EC-Council

C|CT

Certified Cybersecurity Technician

Module - 09

Application Security

Module Objectives

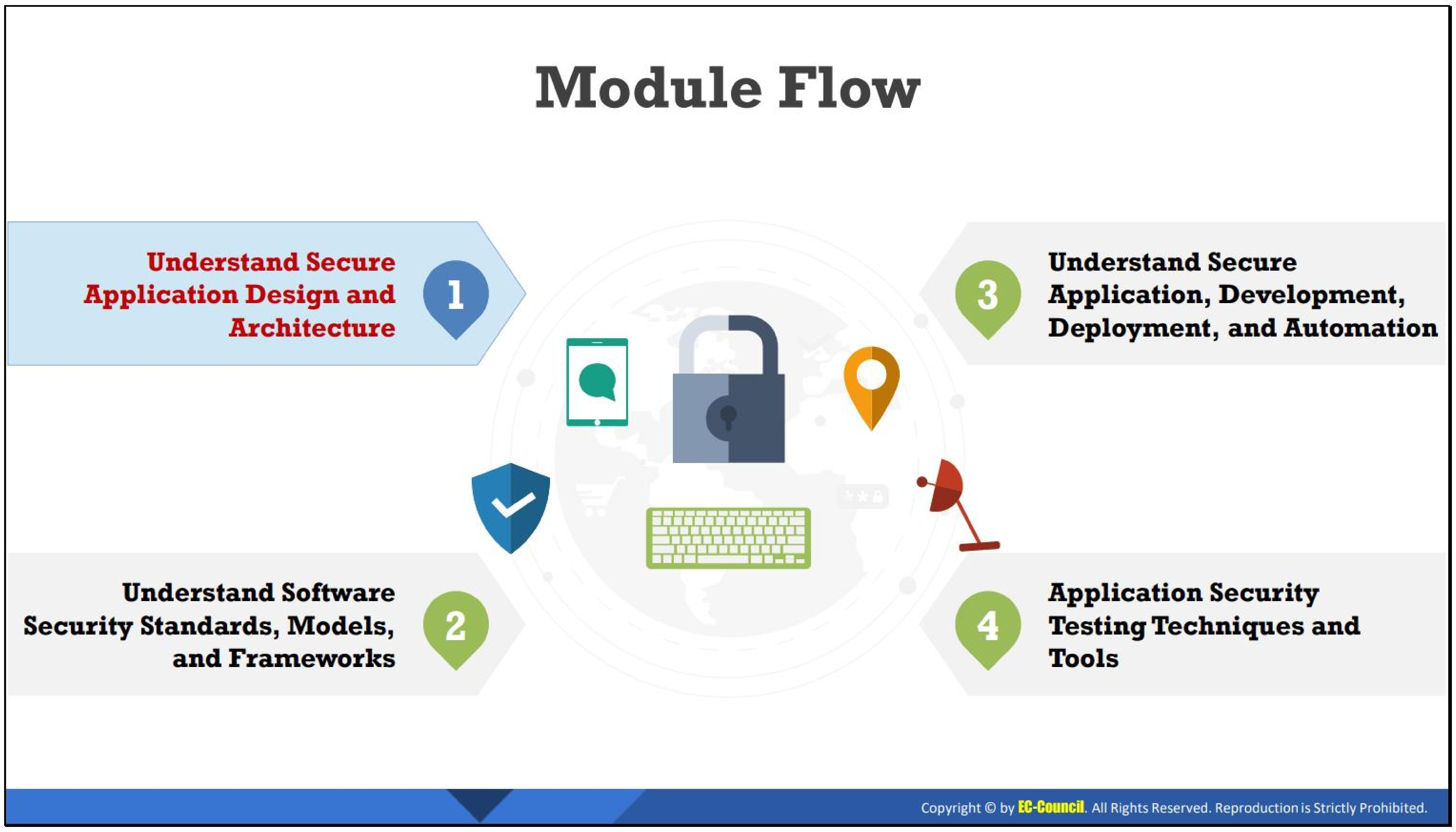
- Understanding Secure Application Design and Architecture
- Understanding Secure Coding Practices
- Overview of Software Security Standards, Models, and Frameworks
- Understanding the Secure Application Development, Deployment, and Automation
- Understanding Various Application Security Testing Techniques and Tools

Module Objectives

The evolution of the Internet and web technologies, combined with rapidly increasing Internet connectivity, has led to the emergence of a new business landscape. Web applications are an integral component of online businesses. Everyone connected via the Internet is using various web applications for different purposes, including online shopping, email, chats, and social networking. Web applications are becoming increasingly vulnerable to sophisticated threats and attack vectors. An outdated or insecure application can pose a serious security threat and, in turn, affect network security. A security professional must manage the security of the deployed applications and constantly monitor, patch, and upgrade the installed applications.

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

- Understand secure application design and architecture
- Explain secure coding practices
- Describe software security standards, models, and frameworks
- Understand the secure application development, deployment, and automation
- Understand various application security testing techniques and tools

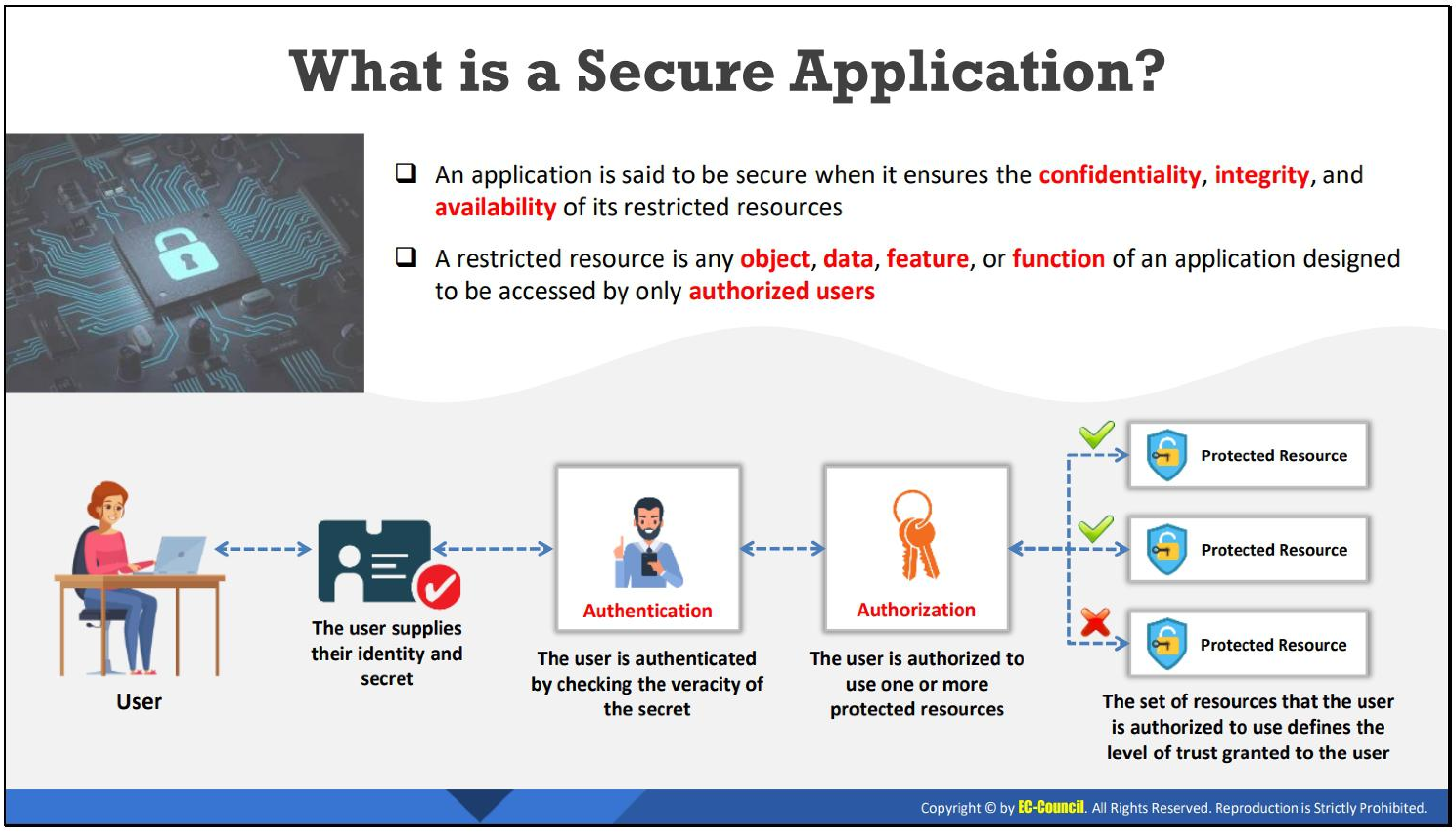


Module Flow

1. Understand Secure Application Design and Architecture
2. Understand Software Security Standards, Models, and Frameworks
3. Understand Secure Application, Development, Deployment, and Automation
4. Application Security Testing Techniques and Tools

Understand Secure Application Design and Architecture

Increasing Internet usage and expanding online businesses have accelerated the development and ubiquity of web applications across the globe. A key factor in the adoption of web applications for business purposes is the multitude of features that they offer. Although web applications enforce certain security policies, they are vulnerable to various attacks. This section discusses the importance of secure application design and architecture.



What is a Secure Application?

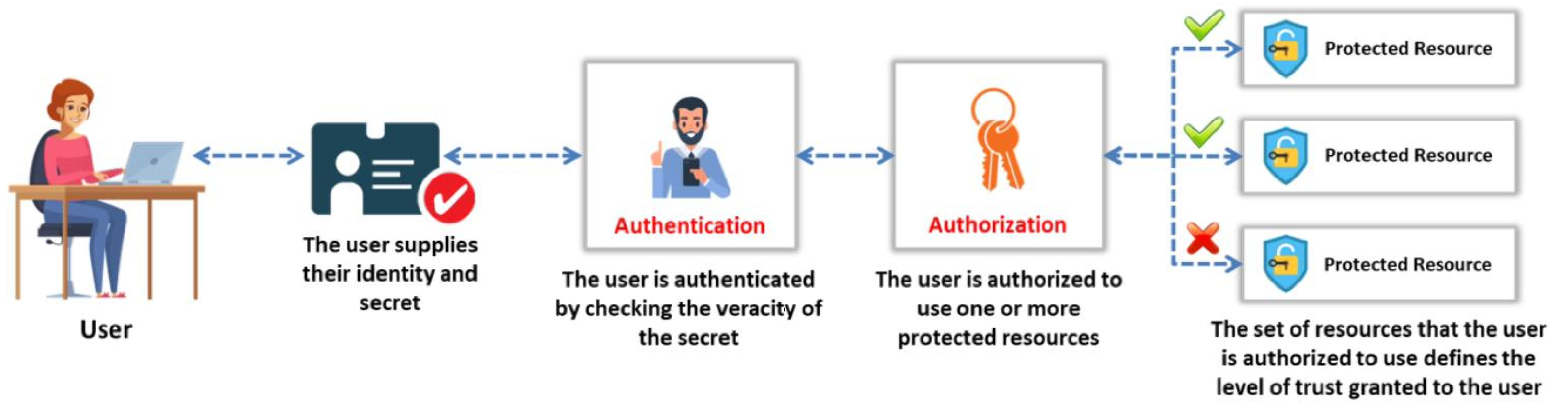
- An application is said to be secure when it ensures the **confidentiality**, **integrity**, and **availability** of its restricted resources
- A restricted resource is any **object**, **data**, **feature**, or **function** of an application designed to be accessed by only **authorized users**

User
The user supplies their identity and secret
Authentication
The user is authenticated by checking the veracity of the secret
Authorization
The user is authorized to use one or more protected resources
Protected Resource
Protected Resource
Protected Resource
The set of resources that the user is authorized to use defines the level of trust granted to the user

What is a Secure Application?

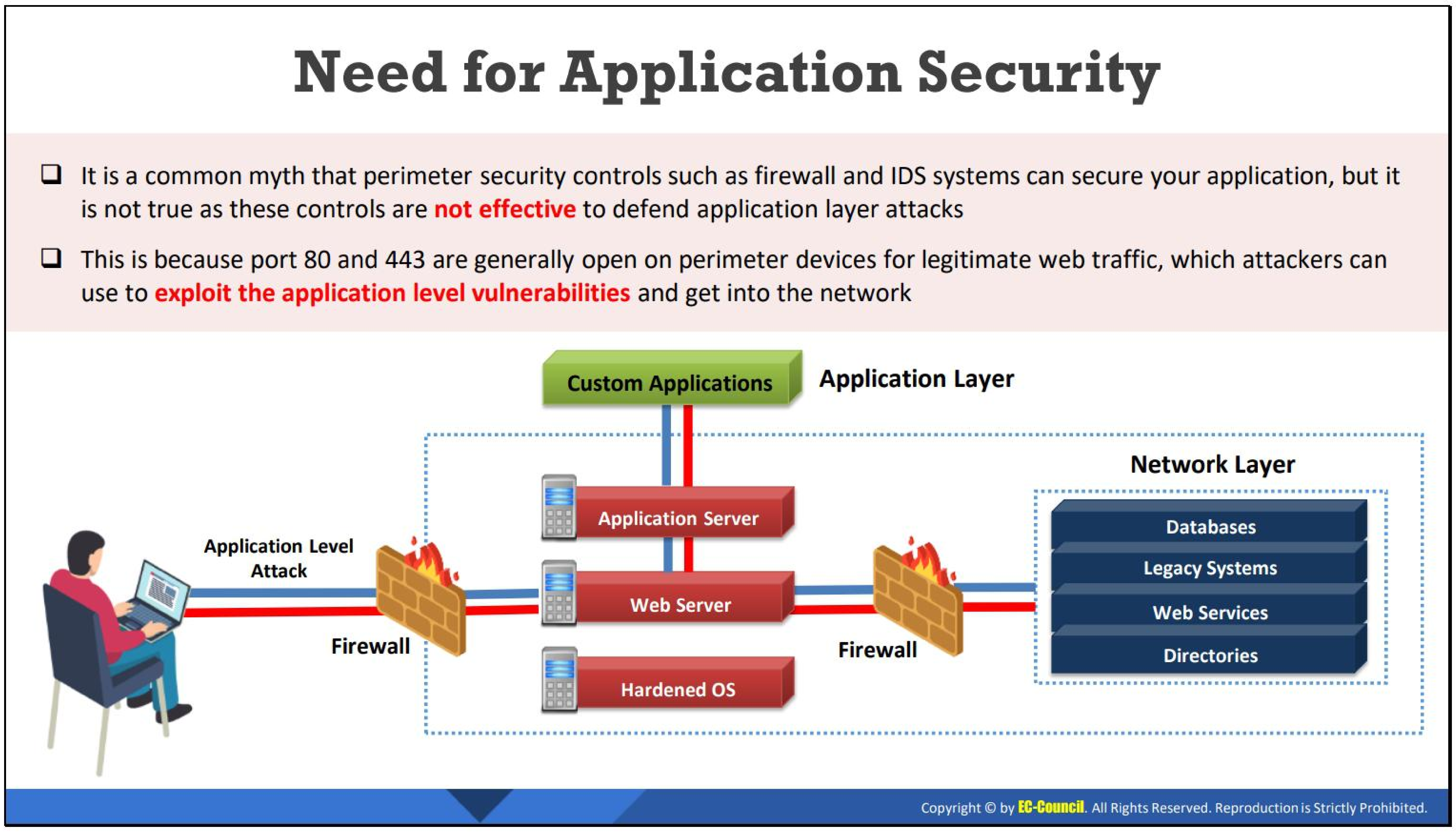
A secure application ensures the confidentiality, integrity, and availability of its restricted resources throughout the application lifecycle. The securing process involves some tools and procedures to protect the application from cyberattacks. Cybercriminals are motivated to target and exploit vulnerabilities in an application to steal confidential data, tamper with code, and compromise the whole application.

The process of securing an application involves deploying, inserting, and testing every component of the application. This procedure identifies all the vulnerabilities in restricted resources such as the objects, data, features, or functions of an application designed to be accessed by only authorized users.



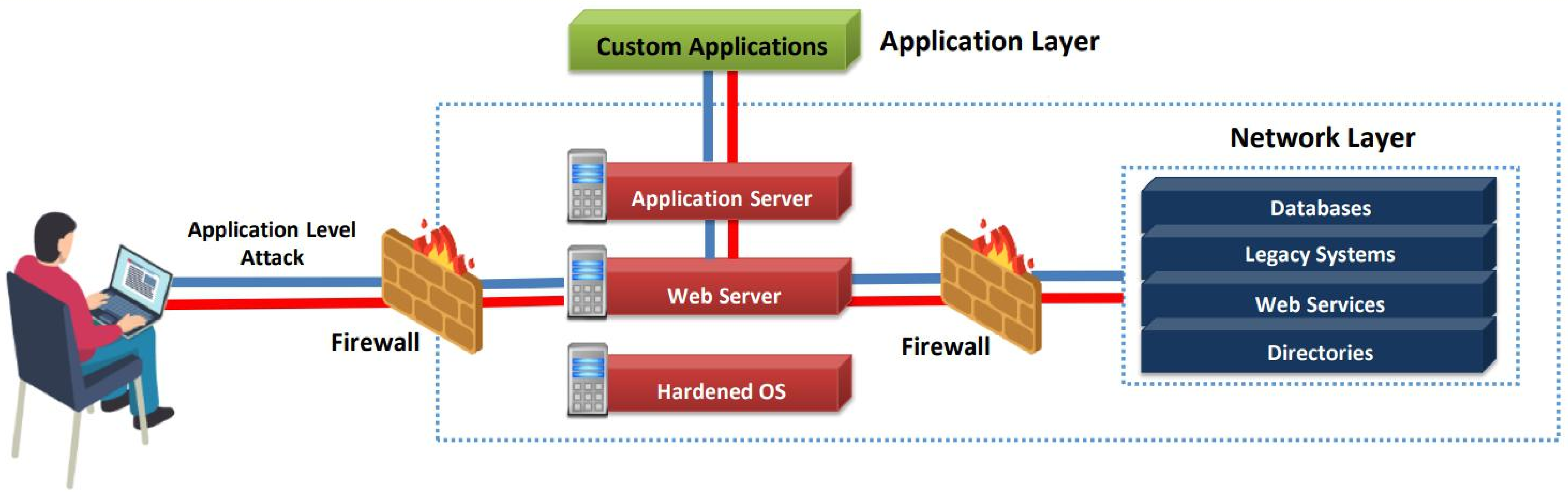
User
The user supplies their identity and secret
Authentication
The user is authenticated by checking the veracity of the secret
Authorization
The user is authorized to use one or more protected resources
Protected Resource
Protected Resource
Protected Resource
The set of resources that the user is authorized to use defines the level of trust granted to the user

Figure 9.1: Illustration of secure web application



Need for Application Security

- It is a common myth that perimeter security controls such as firewall and IDS systems can secure your application, but it is not true as these controls are **not effective** to defend application layer attacks
- This is because port 80 and 443 are generally open on perimeter devices for legitimate web traffic, which attackers can use to **exploit the application level vulnerabilities** and get into the network



Need for Application Security

Organizations are increasingly using web applications to provide high value business functions to their customers such as real-time sales, transactions, inventory management across multiple vendors including both B-B and B-C e-commerce, workflow and supply chain management, etc. Attackers exploit vulnerabilities in the applications to launch various attacks and gain unauthorized access to resources. It is a common myth that perimeter security controls such as firewall and IDS systems can secure your application but it is not true as these controls are not effective to defend application layer attacks. This is because port 80 and 443 are generally open on perimeter devices for legitimate web traffic, which attackers can use to exploit the application level vulnerabilities and get into the network.

A successful application level attack may result into:

- Financial Loss
- Affects Business Continuity
- Closure of Business
- Disclosure of Business Information
- Damages Reputation
- Fraudulent Transactions

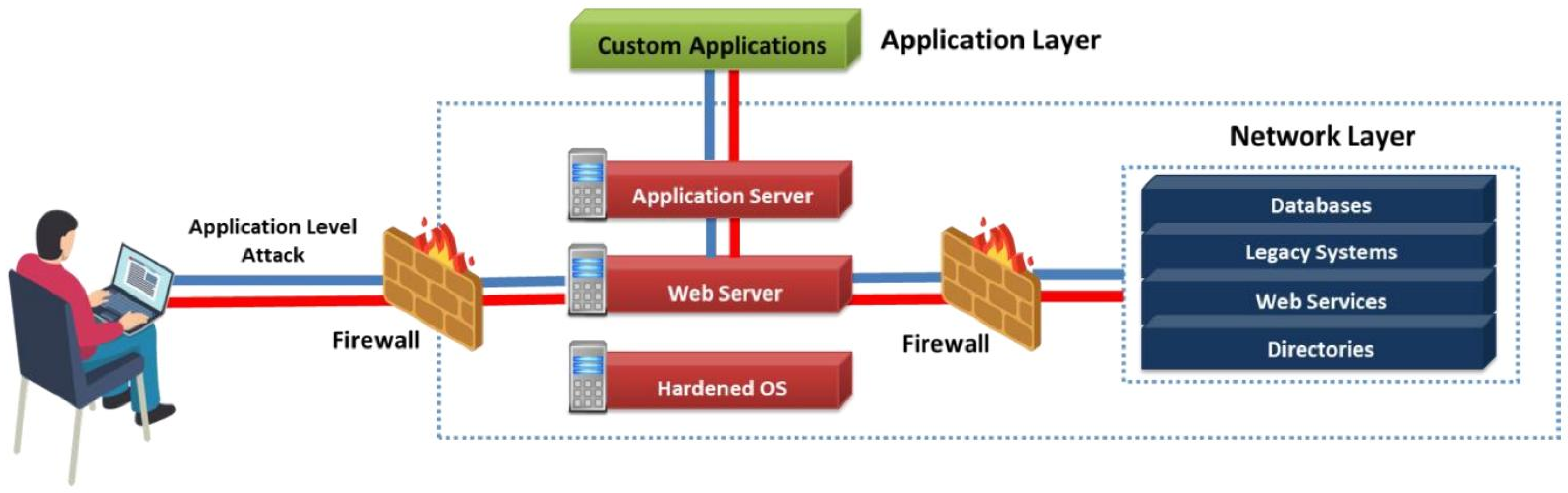
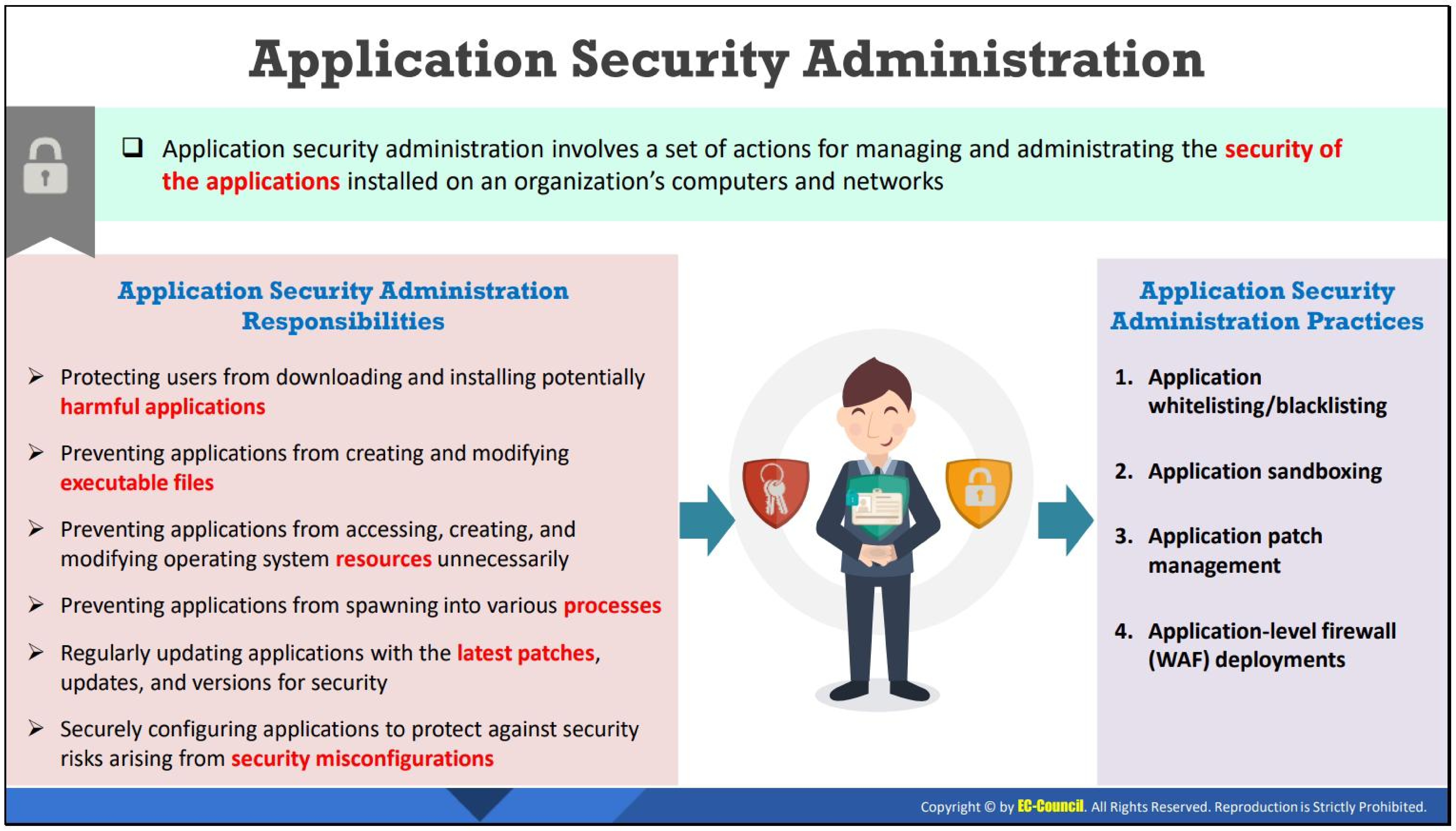


Figure 9.2: Illustration of web application attack



Application Security Administration

- Application security administration involves a set of actions for managing and administrating the **security of the applications** installed on an organization's computers and networks

Application Security Administration Responsibilities

- Protecting users from downloading and installing potentially **harmful applications**
- Preventing applications from creating and modifying **executable files**
- Preventing applications from accessing, creating, and modifying operating system **resources** unnecessarily
- Preventing applications from spawning into various **processes**
- Regularly updating applications with the **latest patches**, updates, and versions for security
- Securely configuring applications to protect against security risks arising from **security misconfigurations**

Application Security Administration Practices

1. **Application whitelisting/blacklisting**
2. **Application sandboxing**
3. **Application patch management**
4. **Application-level firewall (WAF) deployments**

Application Security Administration

Organizations must continuously monitor their applications for vulnerabilities to reduce potential risks and maintain the security of applications. Application security administration involves a set of actions for managing the security of the applications installed on an organization's computers and networks. It is one of the several levels of security that companies consider to secure systems.

The typical responsibilities of application security administration include the following:

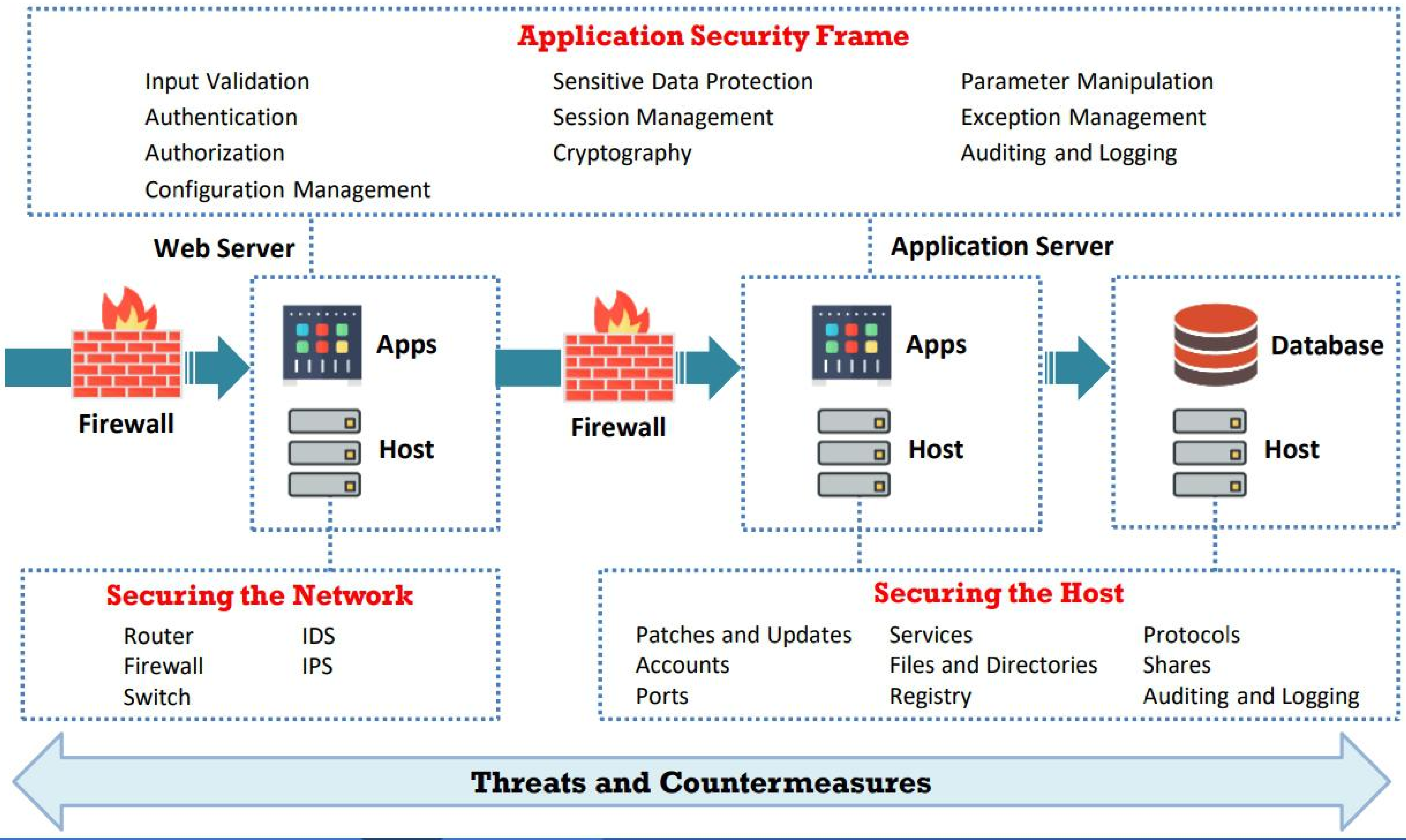
- Users must be protected from downloading and installing potentially harmful applications. Security professionals should not allow the downloading and installation of applications from untrusted sources or third-party sites. Untrusted sources may hide malware inside applications to compromise the system.
- Application must not be allowed to create and modify executable files. Security professionals should ensure the security of an application before it is installed on the system, and applications should be installed using the installation guide provided by the vendor.
- Applications must not be allowed to access, create, or modify OS resources unnecessarily. A security professional should monitor the running applications, and the applications should have only the required permissions to access the system resources to prevent the loss of confidentiality, integrity, and availability.
- Applications must not be allowed to spawn into various processes.

- Applications must be regularly updated with the latest patches and for security. The existence of outdated or insecure applications poses a serious security threat to the organization's network.
- Applications must be configured in a secure manner to protect users from security risks arising from security misconfigurations.

To implement application security in an organization, a security professional performs the following:

- **Application whitelisting/blacklisting**: Control the execution of unwanted or malicious applications.
- **Application sandboxing**: Execute untrusted or untested applications in an isolated environment to protect the system.
- **Application patch management**: Monitor and deploy new or missing patches to ensure the security of applications on hosts.
- **Application-level firewall (WAF) deployment**: Deploy WAF to protect web servers from malicious traffic.

Application Security Frame



Application Security Frame

An application security frame, also referred to as a web-application security schema, incorporates skillful technical operations such as threat modeling to discover and categorize threats, vulnerabilities, and attack surfaces as well as provide appropriate countermeasures. It minimizes risks that can evolve from public platforms while accessing application services. The security frame can establish a regular framework that can merge skills to secure the web server though firewalls, IDSes, routers, and other networking solutions and to secure the host server by releasing on time patches, maintaining individual accounts, logging, etc.

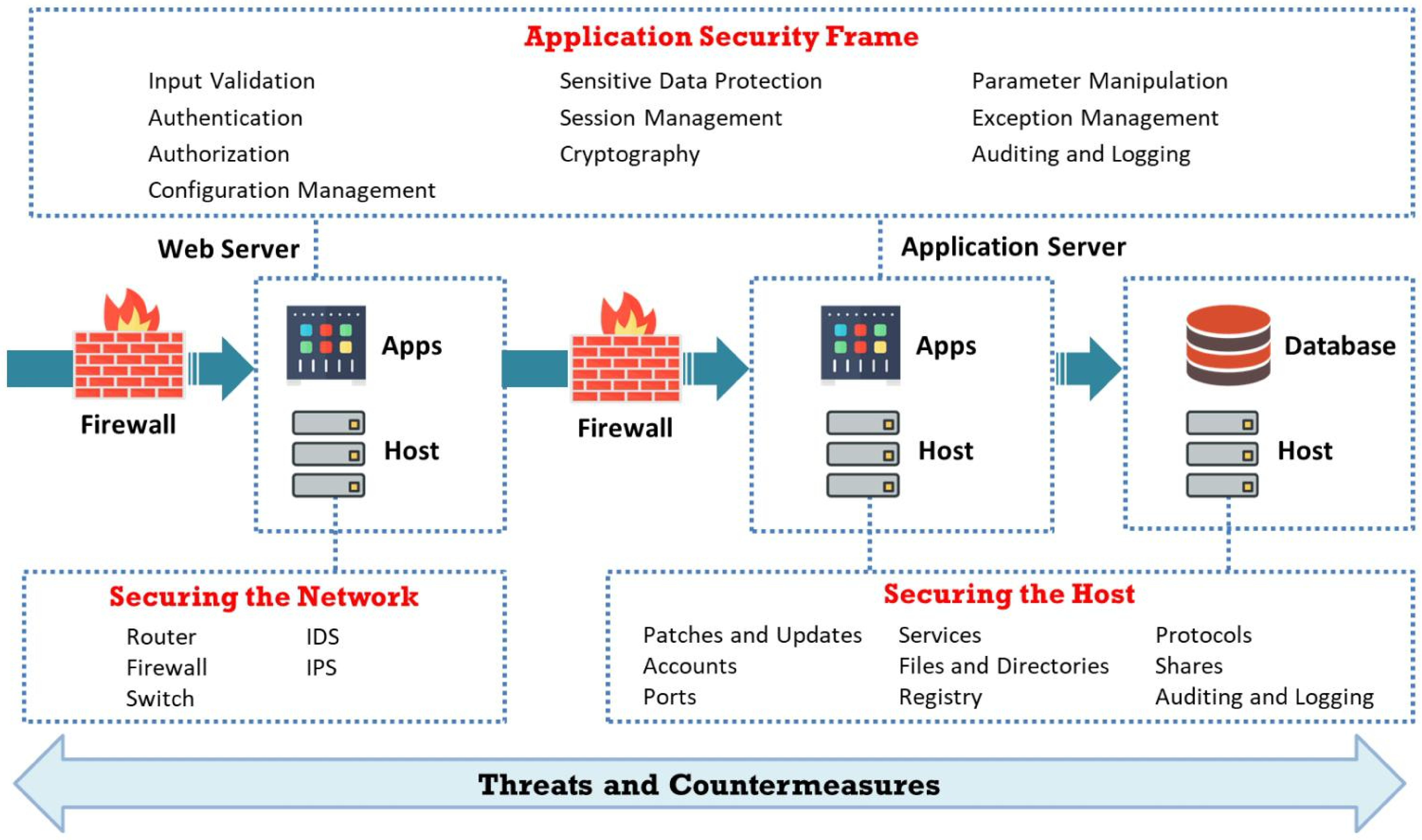
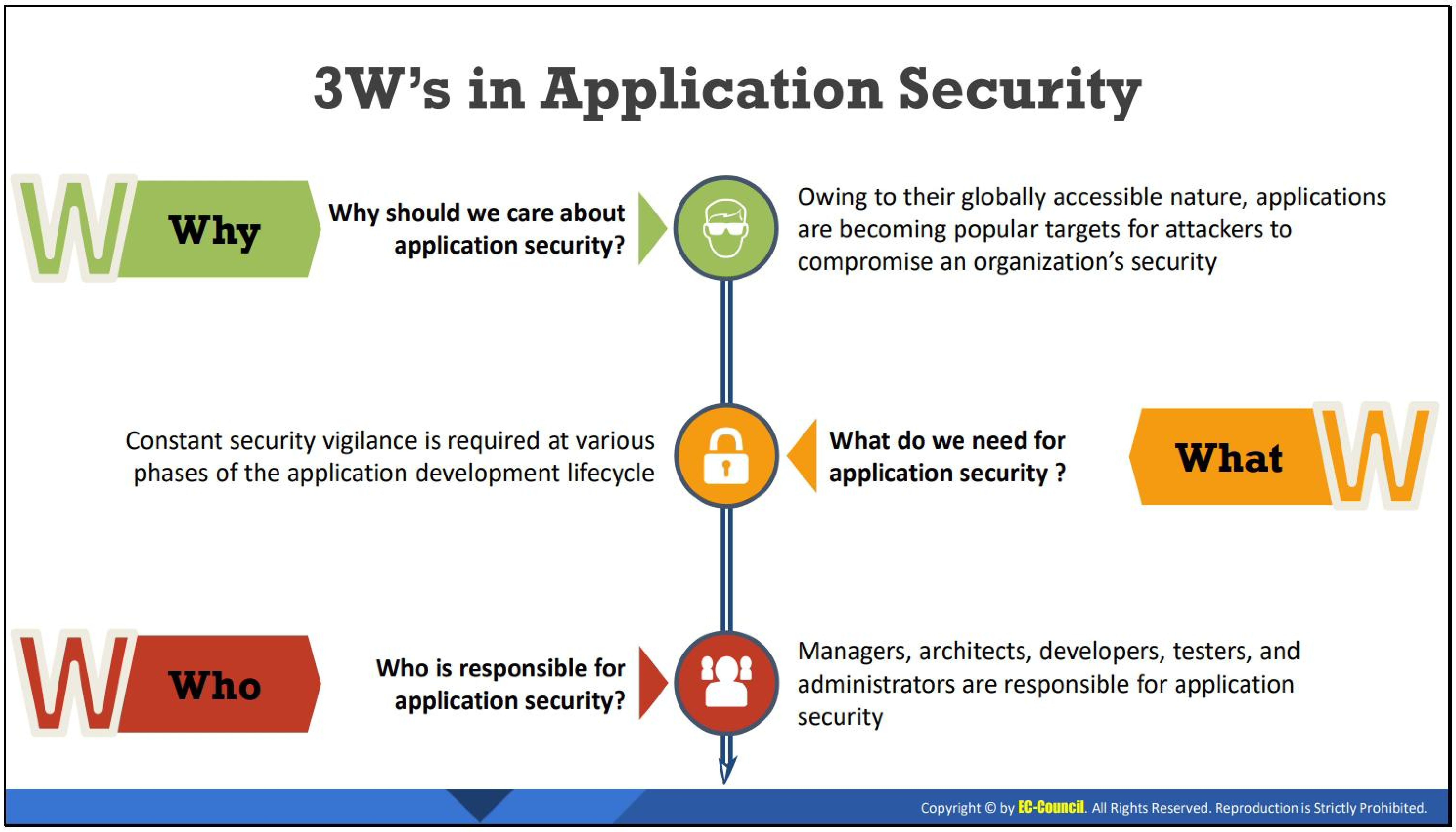


Figure 9.3: Application security frame



3W's in Application Security

W **Why** — Why should we care about application security?

Owing to their globally accessible nature, applications are becoming popular targets for attackers to compromise an organization's security

W **What** — What do we need for application security ?

Constant security vigilance is required at various phases of the application development lifecycle

W **Who** — Who is responsible for application security?

Managers, architects, developers, testers, and administrators are responsible for application security

3W's in Application Security

As a web application passes through complex networks and connects to multiple users, it must be secured with all the necessary security measures, which requires proper planning and expertise. The following are the three Ws involved in providing effective application security.

- **Why: Why should we care about application security?**

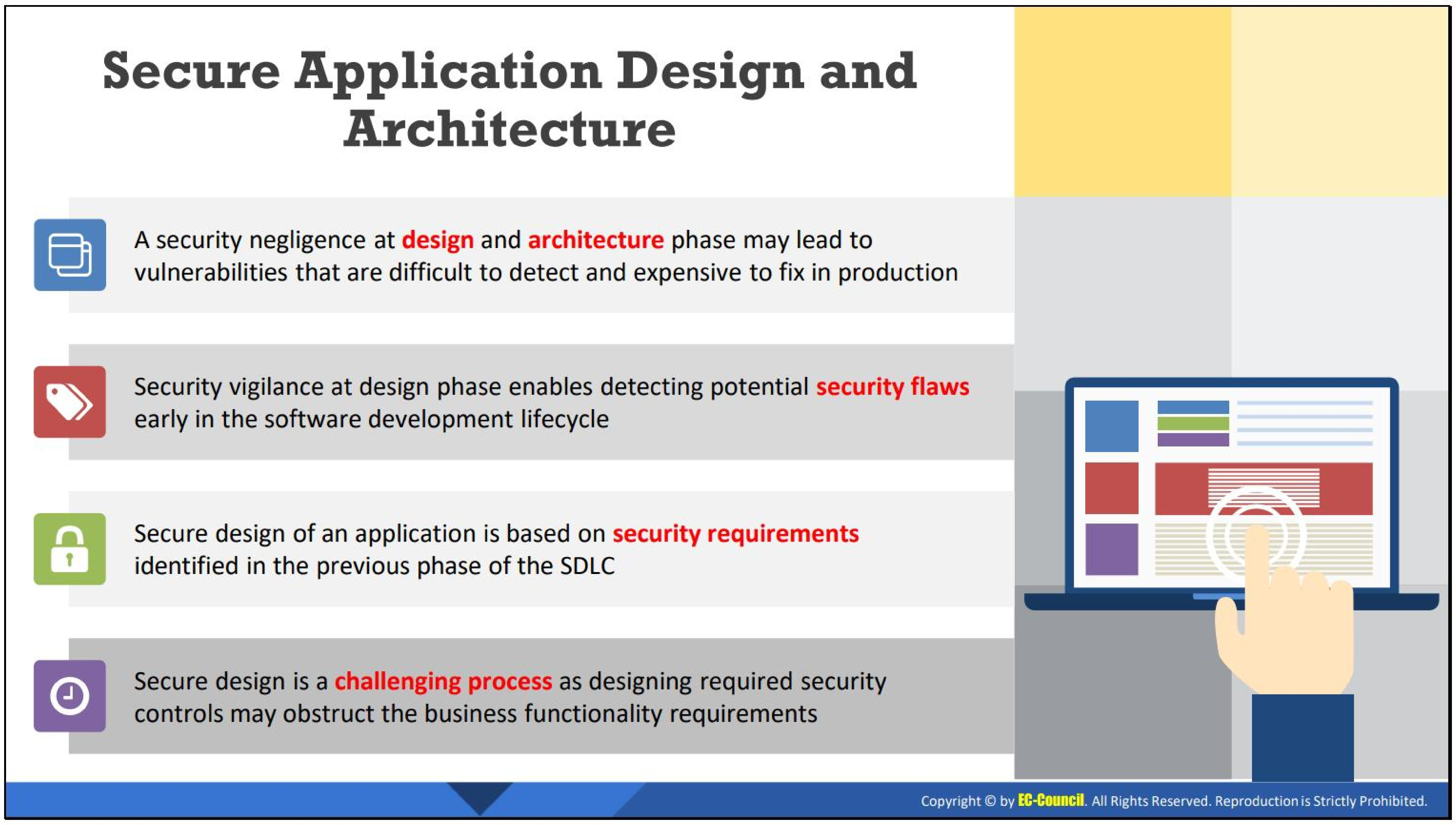
  As applications are globally accessible, they are becoming popular targets for attackers to compromise an organization's security. Therefore, an application must be evaluated while considering all the target portions or attack surfaces. Through appropriate security implementations, the application can maintain confidentiality and integrity of data as well as ensure the uninterrupted availability of services.

- **What: What do we need for application security?**

  To overcome all the security challenges that an application can face in the global network, constant security vigilance is required at various phases of the application development lifecycle. The application also requires security controls or tools to identify, address, and handle threats and to enhance the overall security, thus making it less vulnerable to cyberattacks. Standard policies and guidelines can also play a major role in implementing application security.

- **Who: Who is responsible for application security?**

  Irrespective of where an application is hosted, securing it is a major concern for the organization. Entities such as managers, architects, developers, testers, and administrators take equal responsibility in securing the application. All these parties must collaborate to detect common application security bugs as well as create and deliver patches after thorough inspection.

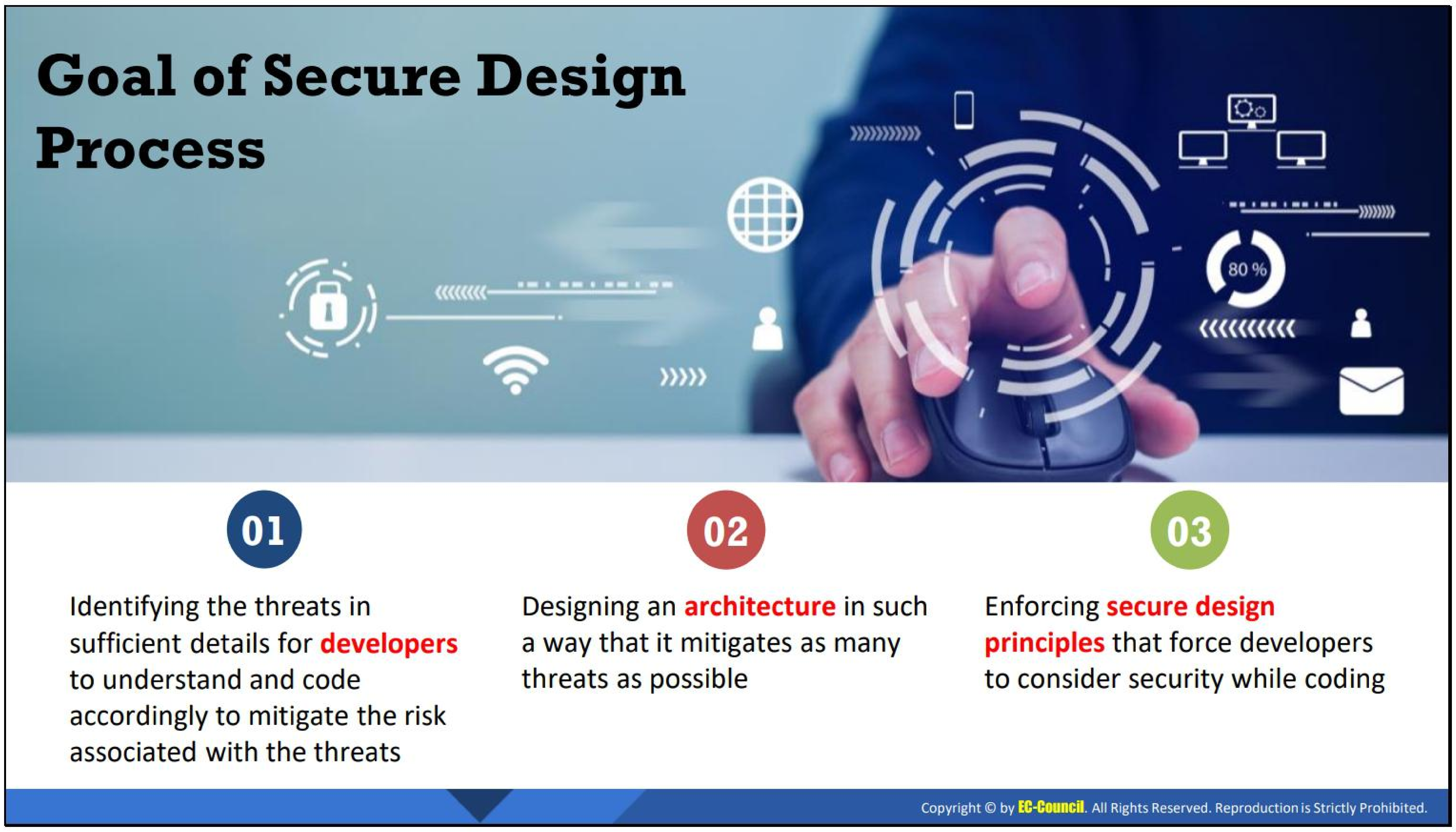


Secure Application Design and Architecture

- A security negligence at **design** and **architecture** phase may lead to vulnerabilities that are difficult to detect and expensive to fix in production
- Security vigilance at design phase enables detecting potential **security flaws** early in the software development lifecycle
- Secure design of an application is based on **security requirements** identified in the previous phase of the SDLC
- Secure design is a **challenging process** as designing required security controls may obstruct the business functionality requirements

Secure Application Design and Architecture

A security negligence at design and architecture phase may lead to vulnerabilities that are difficult to detect and expensive to fix in production. The security vigilance at design phase enables detecting potential security flaws early in the software development lifecycle. Secure design of an application is based on security requirements identified in the previous phase of the SDLC. Secure design is a challenging process as designing required security controls may obstruct the business functionality requirements.



Goal of Secure Design Process

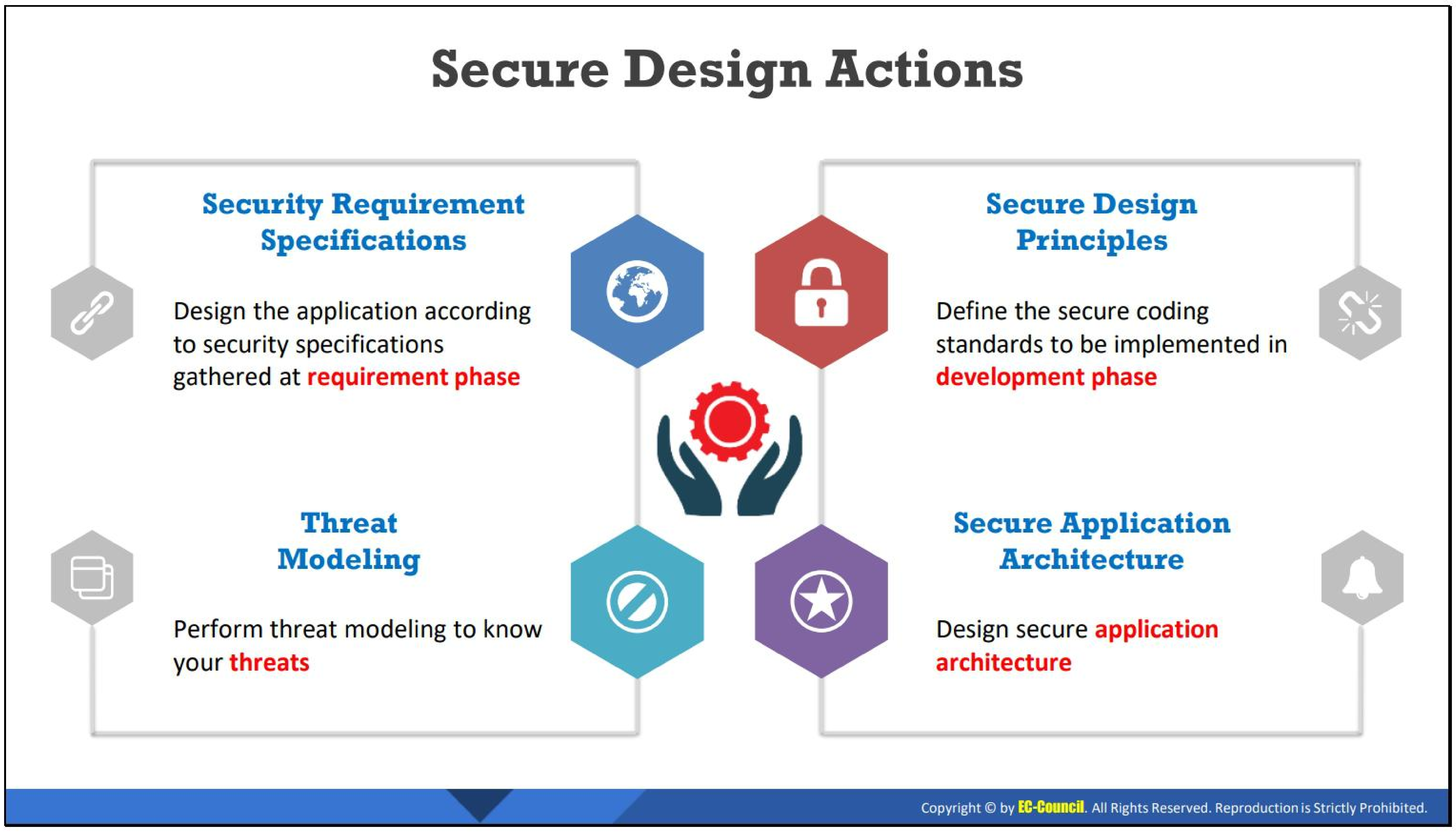
01
Identifying the threats in sufficient details for **developers** to understand and code accordingly to mitigate the risk associated with the threats

02
Designing an **architecture** in such a way that it mitigates as many threats as possible

03
Enforcing **secure design principles** that force developers to consider security while coding

Goal of Secure Design Process

- Identifying the threats in sufficient details for developers to understand and code accordingly to mitigate the risk associated with the threats.
- Designing an architecture in such a way that it mitigates as many threats as possible.
- Enforcing secure design principles that force developers to consider security while coding.
- Ensuring confidentiality, integrity, and availability of data used within the application.



Secure Design Actions

Security Requirement Specifications

Design the application according to security specifications gathered at **requirement phase**

Secure Design Principles

Define the secure coding standards to be implemented in **development phase**

Threat Modeling

Perform threat modeling to know your **threats**

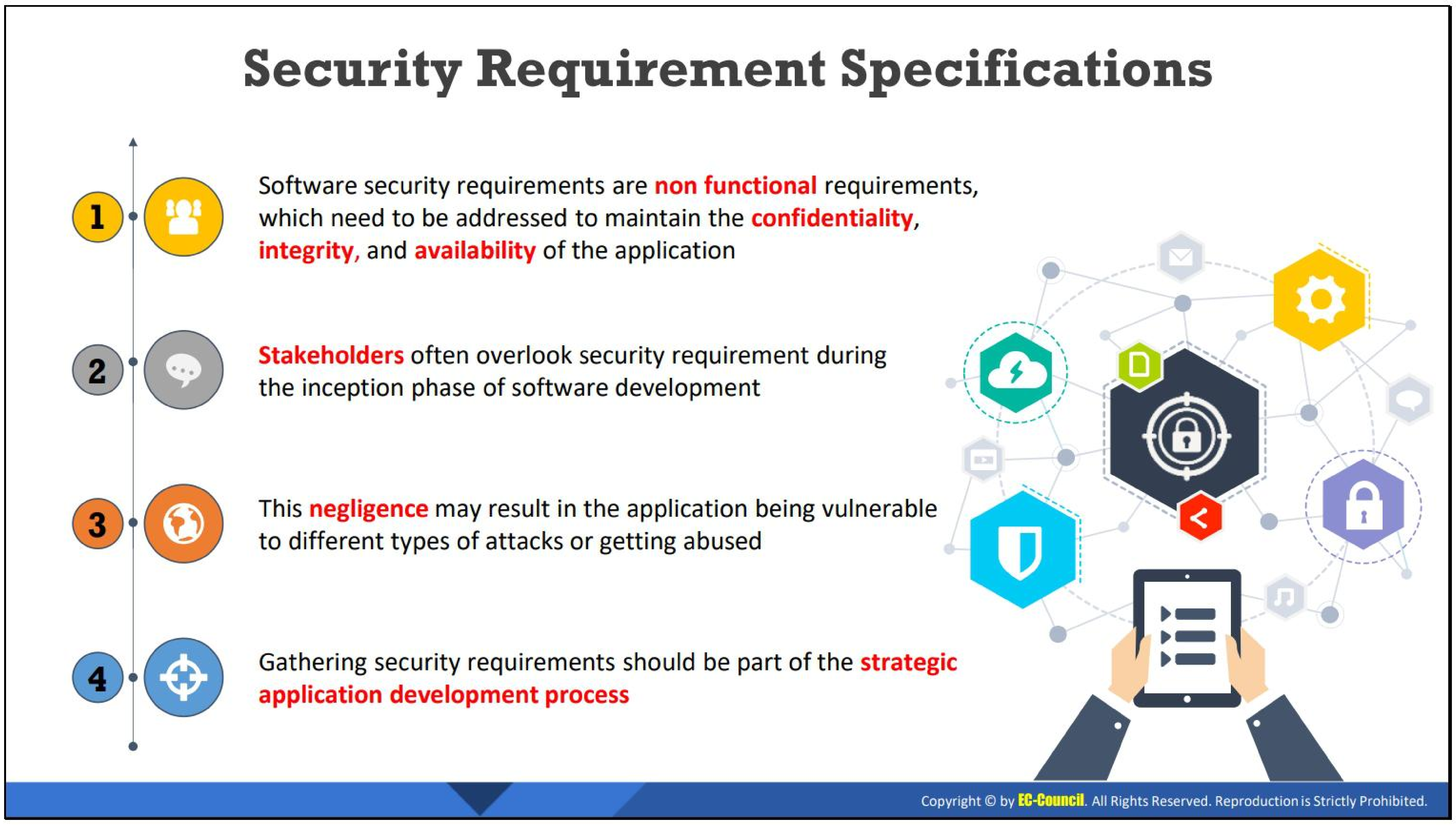
Secure Application Architecture

Design secure **application architecture**

Secure Design Actions

The secure design actions include the following:

- **Security Requirement Specifications:** Design the application according to security specifications gathered at requirement phase.
- **Secure Design Principles:** Define the secure coding standards to be implemented in development phase.
- **Threat Modeling:** Perform threat modeling to know your threats.
- **Secure Application Architecture:** Design secure application architecture.



Security Requirement Specifications

1. Software security requirements are **non functional** requirements, which need to be addressed to maintain the **confidentiality**, **integrity**, and **availability** of the application
2. **Stakeholders** often overlook security requirement during the inception phase of software development
3. This **negligence** may result in the application being vulnerable to different types of attacks or getting abused
4. Gathering security requirements should be part of the **strategic application development process**

Security Requirement Specifications

Software security requirements are non-functional requirements, which need to be addressed to maintain the confidentiality, integrity, and availability of the application. Stakeholders often overlook security requirement during the inception phase of software development. This negligence may result in the application being vulnerable to different types of attacks or getting abused. Gathering security requirements should be part of the strategic application development process. Security requirements give the developer an overview about key security controls required to build secure application. It also specifies the security mechanisms that need to be implemented in order to comply with regulations, standards or requirements for the secure application development and attack protection. Correctly understood security requirements can help in implementing security in design, development, and testing stages.

Define Secure Design Principles

- Secure design principles are the state of practices or guidelines that should be enforced on the developers to follow during development phase
- It helps in deriving secure architectural decisions
- It helps to eliminate design and architecture flaws and mitigate common security vulnerabilities within the application

Define Secure Design Principles (Cont'd)

List of secure design principles to prevent common security vulnerabilities:

- ✓ Security through obscurity
- ✓ Secure the weakest link
- ✓ Use least privilege principle
- ✓ Secure by default
- ✓ Fail securely
- ✓ Apply defense in depth
- ✓ Do not trust user input
- ✓ Reduce attack surface
- ✓ Enable auditing and logging
- ✓ Keep security simple
- ✓ Separation of duties

- ✓ Fix security issues correctly
- ✓ Apply security in design phase
- ✓ Exception handling
- ✓ Secure memory management
- ✓ Protect memory or storage secrets
- ✓ Fault tolerance
- ✓ Fault detection
- ✓ Fault removal
- ✓ Fault avoidance
- ✓ Remove unreachable code
- ✓ Avoid dead code

Define Secure Design Principles

Secure design principles are the state of practices or guidelines that should be enforced on the developers to follow during development phase. It helps in deriving secure architectural decisions. It helps to eliminate design and architecture flaws and mitigate common security vulnerabilities within the application.

List of secure design principles to prevent common security vulnerabilities:

- **Security through obscurity**

  Security Through Obscurity (STO) relies on preventing access to certain users to protect internal data. STO systems may have theoretical or actual security vulnerabilities, but designers believe that flaws are unknown and attackers are unlikely to find them. Its usefulness has declined with the rise of open systems, networking, greater understanding of programming techniques, and increased capabilities of home users.

- **Secure the weakest link**

  Attackers target a system that is easy to penetrate. For example, to gain access to the encrypted data on the network, attackers will not intercept the data and crack encryption; instead they will go after the end points of communication to find a flaw that discloses the data. Identify and strengthen the areas at risk until levels of risk are satisfactory.

- **Use least privilege principle**

  Applications with maximum system privileges are vulnerable to the attacks. For example, many web applications use database admin account though not required to connect to the backend database, enhancing the impact of SQL injection exploits. Using least privilege principle protects application from malicious attacks by:

  - Determining and assigning rights only to those who require privileges to complete the specific task.
  - Avoiding applications that get installed and run by default.
  - Writing applications that can be used by users having non-administrative privileges.

- **Secure by default**

  The software solution or application provided to the users should be security enabled by default. If permitted, it is up to the user to reduce the security. For example, by default the security feature password aging and complexity should be enabled.

- **Fail securely**

  The developer should not give application secrets by default error messages. Application that discloses confidential information on failure assists attackers in creating an attack. When an application fails, determine what may occur and ensure that it does not threaten the application. Always provide logical and useful error messages to the users and store the details in the log file.

- **Apply defense in depth**

  The architects and developers should consider all the levels of the software to impose security while developing software. Implement security mechanisms at different layers that include network layer, kernel layer, physical layer, and the file system layer.

- **Do not trust user input**

  Protect the application from all malicious inputs coming from the user input to the application. Consider all inputs as a malicious input and apply security measures to restrict them.

- **Reduce attack surface**

  Application attack surface area is to be minimized by reducing the number of entry points into the application. Remove or turn off the features, protocols, and functionality which are not in use to minimize number of vulnerabilities and the overall risk. For example, if a vulnerability exists in a way an XML is parsed, denying XML from unknown users minimizes that security vulnerability.

- **Enable auditing and logging**

  Auditing and logging states how the security related events are recorded by an application. Auditing enables identification of attacks or intruders in progress, whereas logging aids in identifying how an attack is performed. Perform auditing and logging to gather information about attacks.

- **Keep security simple**

  If the design is complicated, it is hard to understand and errors are likely to occur in implementation, configuration, and use. On the other hand, if the complexity of security mechanisms increases, the effort required to reach the appropriate level of software assurance also increases. Avoid complex architectures and opt for simpler approaches that are fast and simple.

- **Separation of duties**

  Separation of duties is the key control of fraud. When assigning privileges, system roles are to be considered. In general, system administrators are also the users as some super user privileges are required to make the system run. For example, system administrator can set the password policy, turn off or on the system, etc. but should not be able to log in as a super-privileged user.

- **Fix security issues correctly**

  When a security issue is identified, fix it, considering it as the actual problem, and then go through the security process as you do for the new code, ensuring that the fix does not introduce new errors. For example, user capable of viewing another user's account balance by simply adjusting cookie. In this context once the security issue is fixed it has to be tested on all the applications as cookie handling code is shared among all applications.

- **Apply security in design phase**

  Before starting the application development process, always consider security issues that can help prevent many security vulnerabilities. Considering security issues helps you understand the coding weaknesses and vulnerabilities from the most obvious exploits.

- **Protect sensitive data**

  Do not hard code the sensitive data such as passwords in the program. Use data encryption mechanism to transmit data over the network.

- **Exception handling**

  Events that disrupt the coding process are called exceptions. Exception handling occurs when error conditions interrupt the normal flow of a program's execution. Programmers have difficulty in designing for exception handling, as continuous checking for error conditions is necessary. Proper use of exception handling helps to ensure proper error handling.

- **Secure memory management**

  Check memory bounds on the length of input variables, arrays, and arguments to prevent buffer overflow attacks. Apply coding standards for simplicity, which help in implementing security in the program and keep things simple.

- **Protect memory or storage secrets**

  Encrypt secrets to protect memory storage from ending up in crash dump file. Use a perfect cryptographic method to perform encrypting secrets process. Scrub secrets in memory storage before deletion.

- **Fault tolerance**

  Strategy applied to software design (or system design) to permit system to continue functioning even in the presence of faults by enhancing its robustness.

- **Fault detection**

  Closely linked to fault tolerance, used in detecting faults and producing appropriate responses of system behavior. Examples include system monitors, safety monitors, built-in tests, loop-back tests, etc.

- **Fault removal**

  Removes faults during design process. Examples include error detection, verification through inspection, built-in testing, correction functions, etc.

- **Fault avoidance**

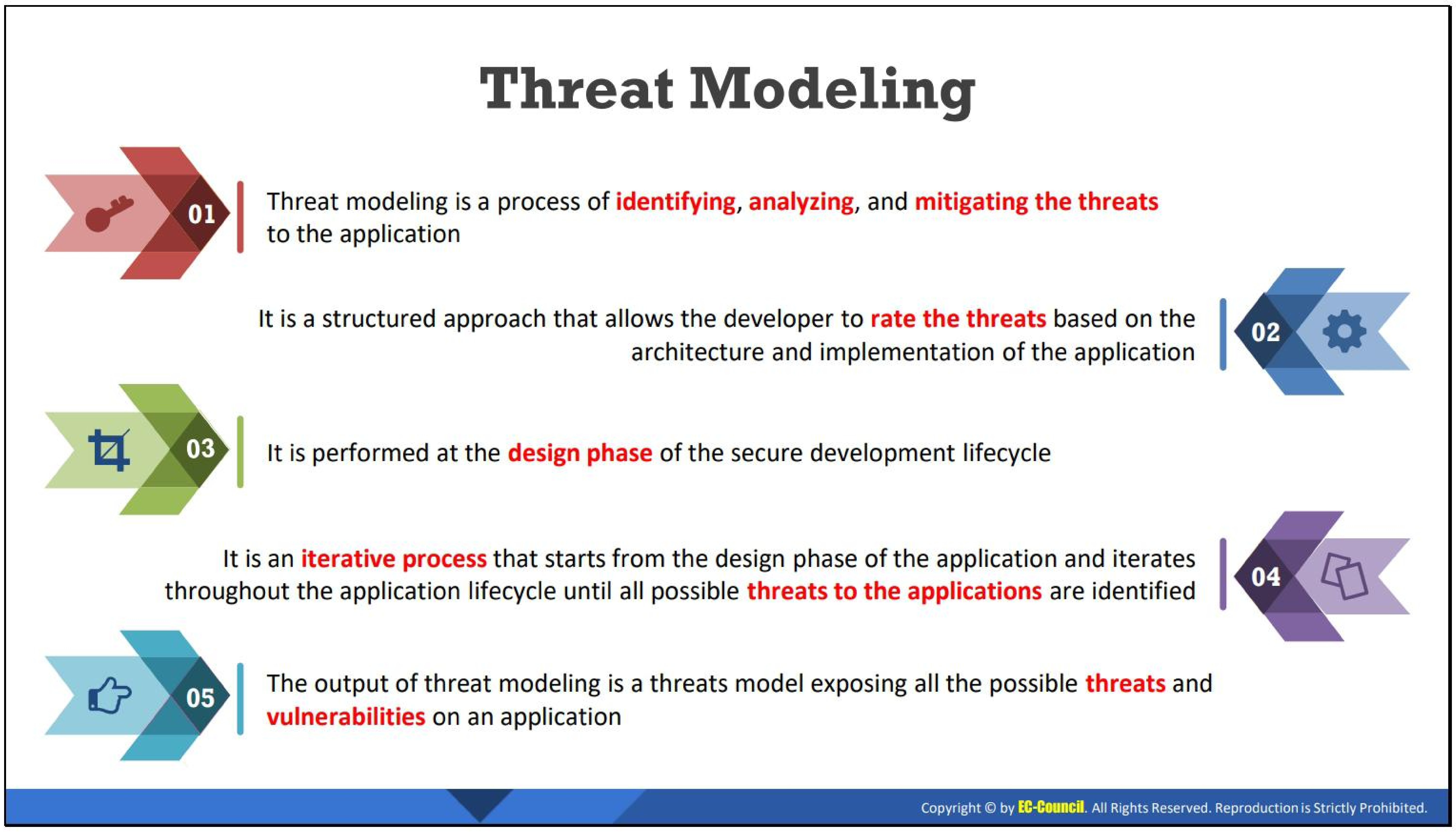
  Avoids errors that contribute to system faults during the development process. Examples include defensive programming, error minimization during design process, minimization of safety critical code, using appropriate SDLC techniques, etc.

- **Remove unreachable code**

  Unreachable code is a portion of the source code used for application development that cannot be executed because of control flow errors in the program. Unreachable code occurs when developers follow poor and insecure coding practices, commit software errors, or neglect to delete redundant code. Unreachable code, if left undetected, can cause unwanted memory overheads and cache cycles, making the application vulnerable to attacks and performance bottlenecks.

- **Avoid dead code**

  Dead code is unwanted code in an application source code that cannot be executed. The execution of dead code does not cause any effect on or change in the behavior of an application. Dead code can make the source code more complex and vaguer, thereby decreasing the performance and security of the application. Dead code should be removed by using any of the latest tools and techniques to prevent any security loopholes or failures that can make the application vulnerable to attacks.



Threat Modeling

01 Threat modeling is a process of **identifying**, **analyzing**, and **mitigating the threats** to the application

02 It is a structured approach that allows the developer to **rate the threats** based on the architecture and implementation of the application

03 It is performed at the **design phase** of the secure development lifecycle

04 It is an **iterative process** that starts from the design phase of the application and iterates throughout the application lifecycle until all possible **threats to the applications** are identified

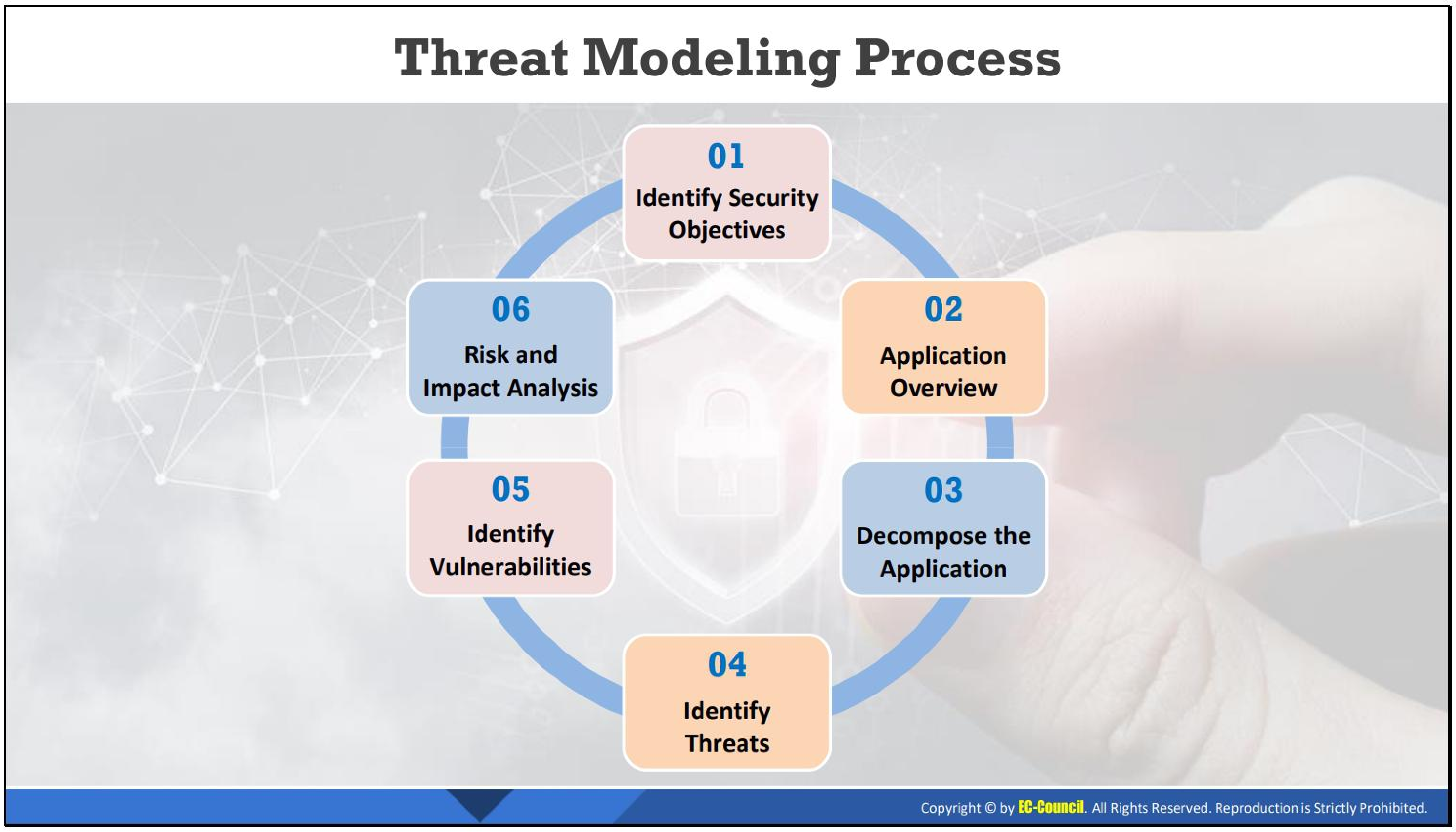
05 The output of threat modeling is a threats model exposing all the possible **threats** and **vulnerabilities** on an application

Threat Modeling

Threat modeling is a process of identifying, analyzing, and mitigating the threats to the application. It is a structured approach that allows the developer to rate the threats based on the architecture and implementation of the application. It is performed at the design phase of the secure development lifecycle. It is an iterative process that starts from the design phase of the application and iterates throughout the application lifecycle until all possible threats to the applications are identified. The output of threat modeling is a threats model exposing all the possible threats and vulnerabilities on an application.

Threat modeling helps to:

- Identify relevant threats to a particular application scenario
- Identify key vulnerabilities in an application's design
- Improve security design


Threat Modeling Process
01 Identify Security Objectives
02 Application Overview
03 Decompose the Application
04 Identify Threats
05 Identify Vulnerabilities
06 Risk and Impact Analysis
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Threat Modeling Process

The threat modeling process involves six steps:

1. **Identify Security Objectives**

   Security objectives are the goals and constraints related to the application's confidentiality, integrity, and availability. Security-specific objectives guide the threat modeling efforts and help to determine how much effort needs to be put toward subsequent steps. To identify security objectives, administrators should ask the following questions:

   - What data should be protected?
   - Are there any compliance requirements?
   - Are there specific quality-of-service requirements?
   - Are there intangible assets to protect?

2. **Application Overview**

   Identify the components, data flows, and trust boundaries. To draw the end-to-end deployment scenario, the administrator should use a whiteboard. First, they should draw a rough diagram that explains the workings and structure of the application, its subsystems, and its deployment characteristics. The deployment diagram should contain the following:

   - End-to-end deployment topology
   - Logical layers

- Key components
- Key services
- Communication ports and protocols
- Identities
- External dependencies

Identify Roles

The administrator should identify people and the roles and actions they can perform within the application. For example, are there higher-privileged groups of users? Who can read data? Who can update data? Who can delete data?

Identify Key Usage Scenarios

The administrator should use the application's use cases to determine its objective. Use cases explain how the application is used and misused.

Identify Technologies

The administrator should list the technologies and key features of the software, as well as the following technologies in use:

- Operating systems
- Web server software
- Database server software
- Technologies for presentation, business, and data access layers
- Development languages

Identifying these technologies helps to focus on technology-specific threats.

Identify Application Security Mechanisms

The administrator should identify some key points regarding the following:

- Input and data validation
- Authorization and authentication
- Sensitive data
- Configuration management
- Session management
- Parameter manipulation
- Cryptography
- Exception management
- Auditing and logging

These efforts aim to identify relevant details and to add details where required, or to identify areas that require more.

3. **Decompose the Application**

   In this step, the administrator breaks down the application to identify the trust boundaries, data flows, entry points, and exit points. Doing so makes it considerably easier to find more relevant and more detailed threats and vulnerabilities.

   Identify Trust Boundaries

   Identifying the application's trust boundaries helps the administrator to focus on the relevant areas of the application. It indicates where trust levels change.

   - Identify outer system boundaries
   - Identify access control points or key places where access requires extra privileges or role membership
   - Identify trust boundaries from a data flow perspective

   Identify Data Flows

   The administrator should list the application's data input from entry to exit. This helps to understand how the application communicates with outside systems and clients and how the internal components interact. They should pay particular attention to the data flow across trust boundaries and the data validation at the trust boundary entry point. A good approach is to start at the highest level and then deconstruct the application by testing the data flow between different subsystems.

   Identify Entry Points

   The application's entry point can also serve as an entry point for attacks. All users interact with the application at these entry points. Other internal entry points uncovered by subcomponents over the layers of the application may be present only to support internal communication with other components. The administrator should identify these entry points to determine the methods used by an intruder to get in through them. They should focus on the entry points that allow access to critical functionalities and provide adequate defense for them.

   Identify Exit Points

   The administrator should also identify the points where the application transfers data to the client or external systems. They should prioritize the exit points at which the application writes data containing client input or data from untrusted sources, such as a shared database.

4. **Identify Threats**

   The administrator should identify threats relevant to the control scenario and context using the information obtained in the application overview and decompose application steps. They should bring members of the development and test teams together to identify potential threats. The team should start with a list of common threats grouped by their application vulnerability category. This step uses a question-driven approach to help identify threats.

5. **Identify Vulnerabilities**

   A vulnerability is a weakness in an application (deployed in an information system) that allows attacker exploitation, thereby leading to security breaches. Security administrators should identify any weaknesses related to the threats found using the vulnerability categories to identifying vulnerabilities and fix them beforehand to keep intruders away.

6. **Risk and Impact Analysis**

   The security administrator should perform risk and impact analysis to determine the amount of damage that a vulnerability in an application can cause when it is exploited as well as to rate the risk or severity level for each threat associated with it. Then, the administrator must prioritize the threats based on the decreasing order of severity level and inform the security management team to identify risk mitigation strategies.

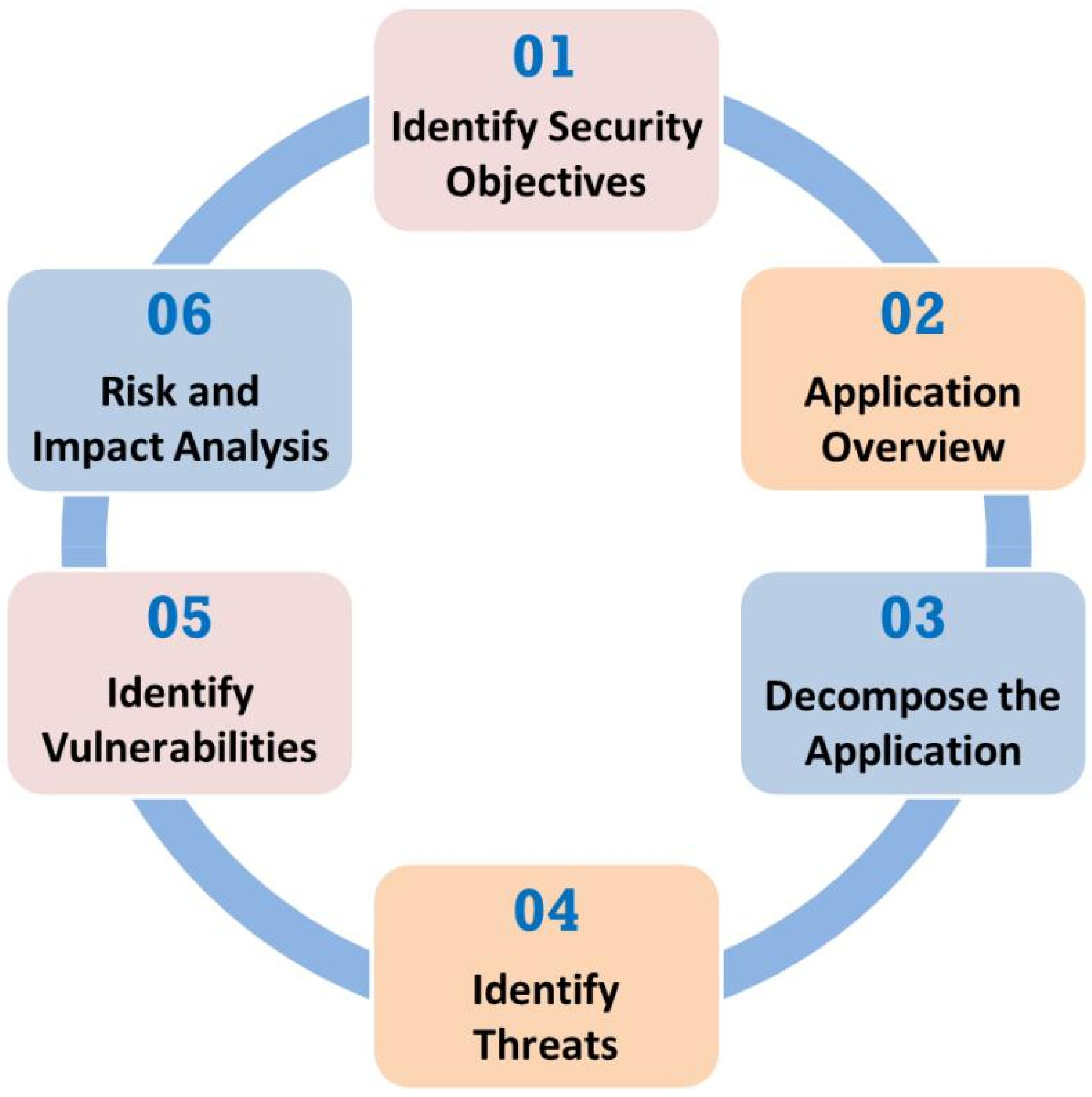
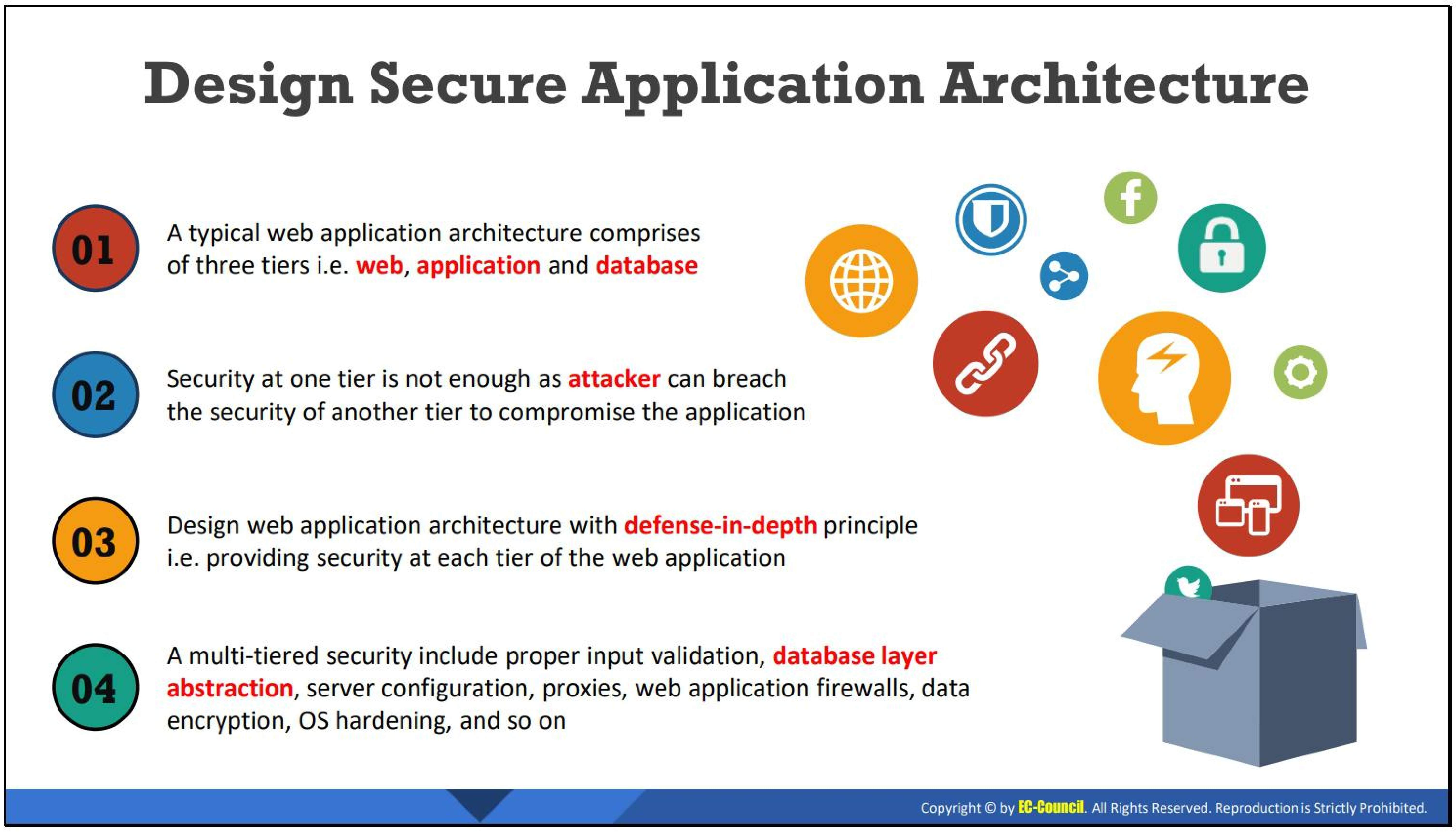


Figure 9.4: Threat modeling process



Design Secure Application Architecture

01 A typical web application architecture comprises of three tiers i.e. **web**, **application** and **database**

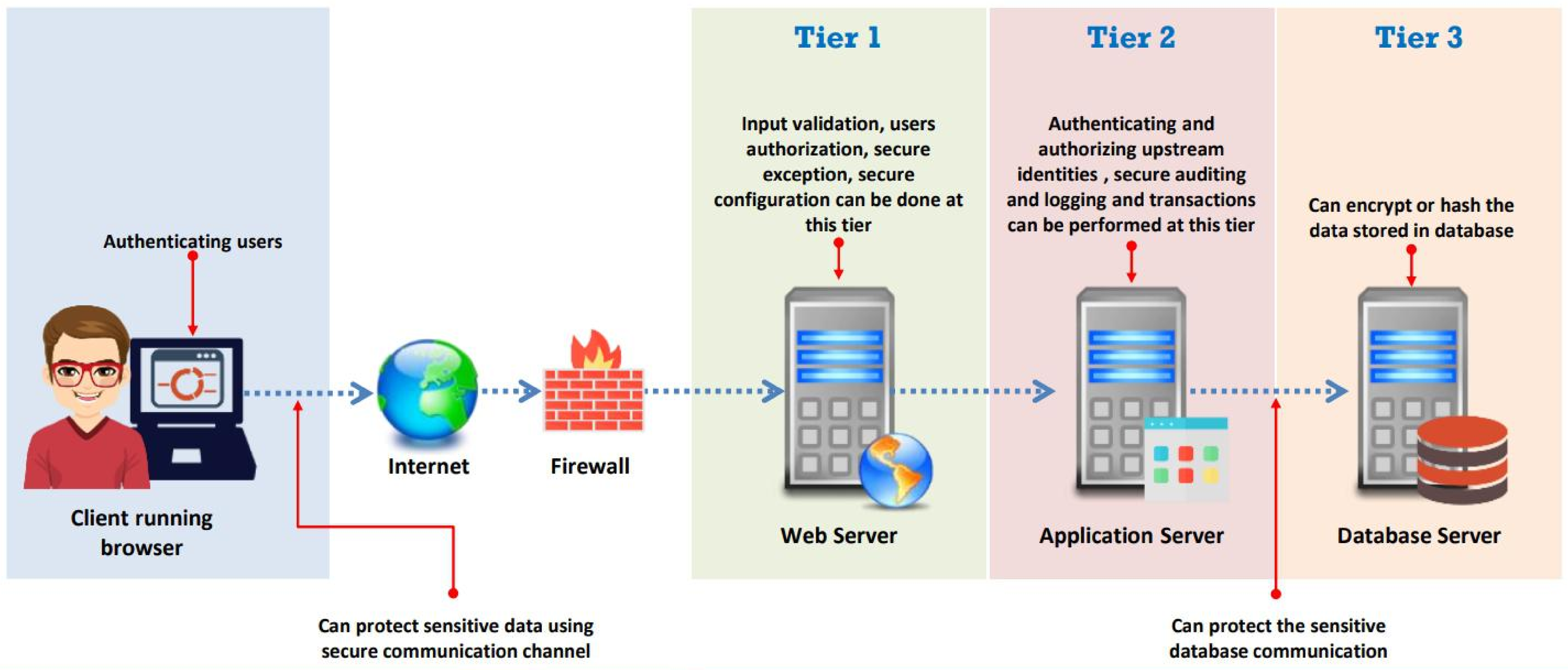
02 Security at one tier is not enough as **attacker** can breach the security of another tier to compromise the application

03 Design web application architecture with **defense-in-depth** principle i.e. providing security at each tier of the web application

04 A multi-tiered security include proper input validation, **database layer abstraction**, server configuration, proxies, web application firewalls, data encryption, OS hardening, and so on

Design Secure Application Architecture (Cont'd)

❑ Applying multiple layer security in application architecture design makes application robust and secure


Tier 1
Tier 2
Tier 3
Input validation, users authorization, secure exception, secure configuration can be done at this tier
Authenticating and authorizing upstream identities , secure auditing and logging and transactions can be performed at this tier
Can encrypt or hash the data stored in database
Authenticating users
Client running browser
Internet
Firewall
Web Server
Application Server
Database Server
Can protect sensitive data using secure communication channel
Can protect the sensitive database communication

Design Secure Application Architecture

A typical web application architecture comprises of three tiers i.e. web, application and database. Security at one tier is not enough as attacker can breach the security of another tier to compromise the application. Design web application architecture with defense-in-depth principle i.e. providing security at each tier of the web application. A multi-tiered security include proper input validation, database layer abstraction, server configuration, proxies, web application firewalls, data encryption, OS hardening, and so on. Applying multiple layer security in application architecture design makes application robust and secure.

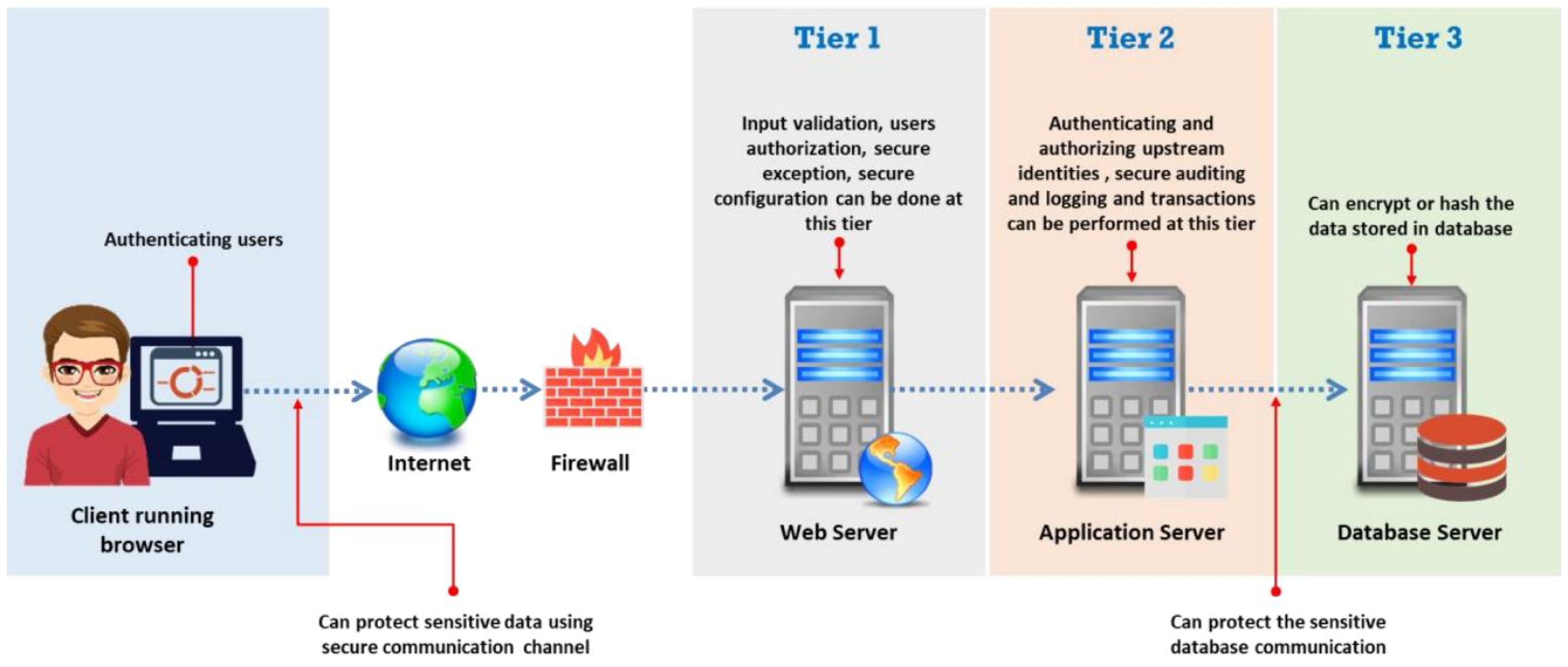
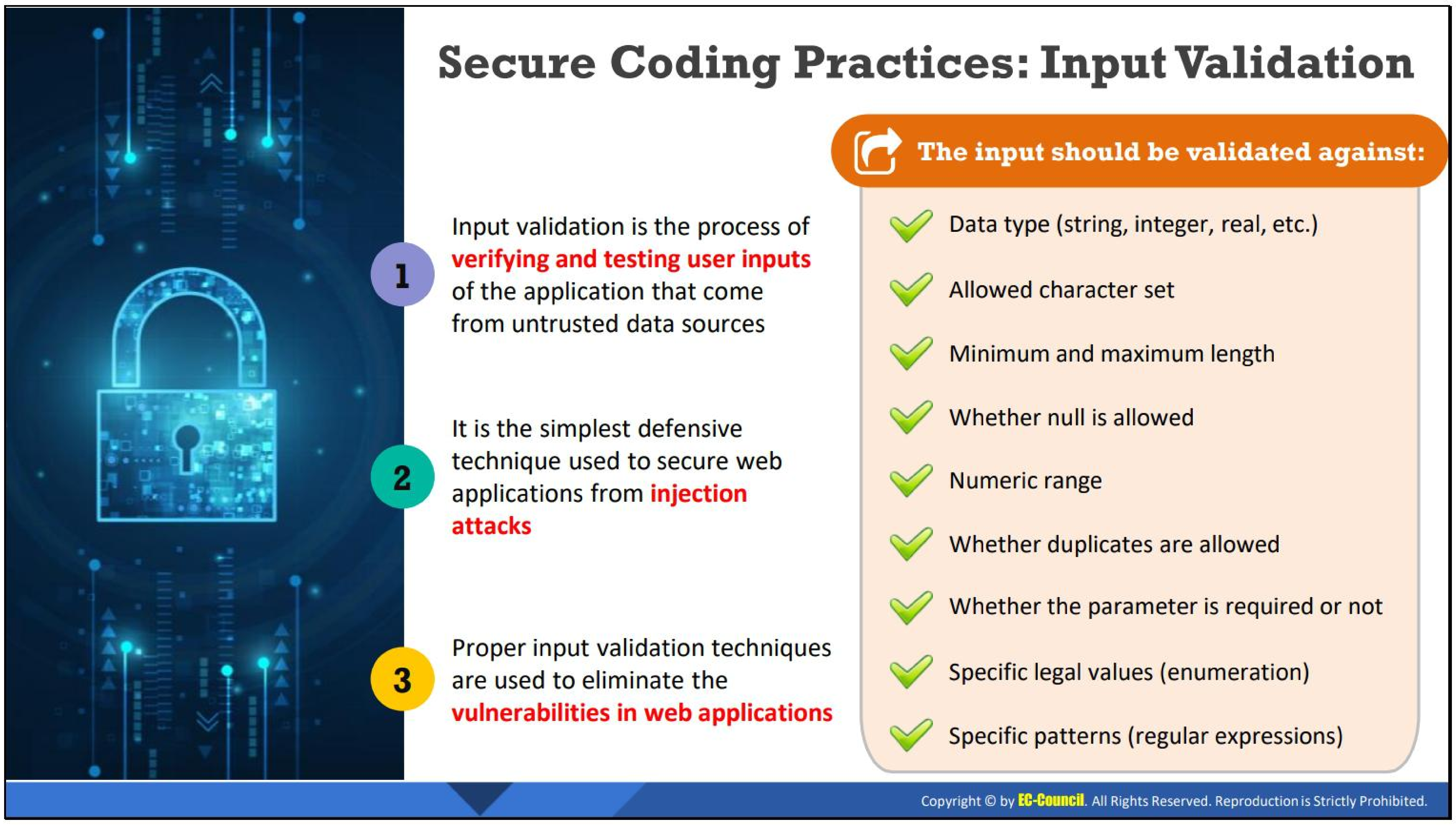

Tier 1
Tier 2
Tier 3
Input validation, users authorization, secure exception, secure configuration can be done at this tier
Authenticating and authorizing upstream identities , secure auditing and logging and transactions can be performed at this tier
Can encrypt or hash the data stored in database
Authenticating users
Internet
Firewall
Client running browser
Web Server
Application Server
Database Server
Can protect sensitive data using secure communication channel
Can protect the sensitive database communication

Figure 9.5: Secure application architecture



Secure Coding Practices: Input Validation

1. Input validation is the process of **verifying and testing user inputs** of the application that come from untrusted data sources
2. It is the simplest defensive technique used to secure web applications from **injection attacks**
3. Proper input validation techniques are used to eliminate the **vulnerabilities in web applications**

The input should be validated against:

- ✓ Data type (string, integer, real, etc.)
- ✓ Allowed character set
- ✓ Minimum and maximum length
- ✓ Whether null is allowed
- ✓ Numeric range
- ✓ Whether duplicates are allowed
- ✓ Whether the parameter is required or not
- ✓ Specific legal values (enumeration)
- ✓ Specific patterns (regular expressions)

Secure Coding Practices

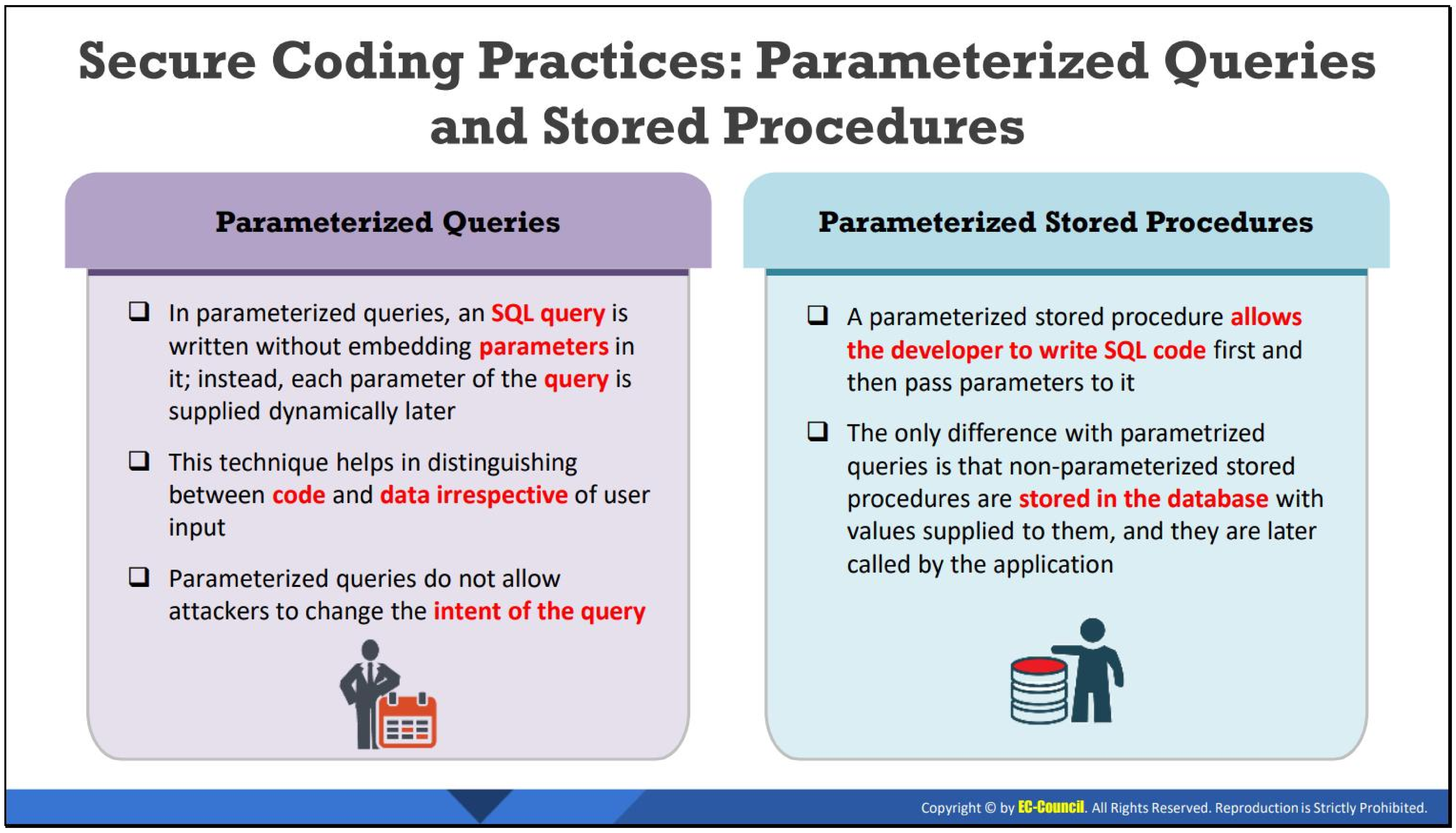
While developing applications, developers must consider many aspects of security in coding such as input validation, secure parameter passing, normalization, and output encoding. Secure coding helps in preventing vulnerabilities and cyberattacks.

Input Validation

Input validation is the process of verifying and testing user inputs of the application that come from untrusted data sources. It is the simplest defensive technique used to secure web applications from injection attacks. Proper input validation techniques are used to eliminate the vulnerabilities in web applications. Improper validation of input may provide the path for the attackers to perform injection attacks such as cross site scripting attacks and SQL injection attacks on the application. Firewalls cannot prevent the attacks caused by malicious or invalid inputs and processing of these inputs without validation can make the application vulnerable to attacks. Attackers can exploit improper input validation vulnerabilities by supplying malicious data to crash the application, manipulate or corrupt databases, etc.

The input should be validated against:

- Data type (string, integer, real, etc.)
- Allowed character set
- Minimum and maximum length
- Whether null is allowed
- Numeric range
- Whether duplicates are allowed
- Whether the parameter is required or not
- Specific legal values (enumeration)
- Specific patterns (regular expressions)



Secure Coding Practices: Parameterized Queries and Stored Procedures

Parameterized Queries

- In parameterized queries, an **SQL query** is written without embedding **parameters** in it; instead, each parameter of the **query** is supplied dynamically later
- This technique helps in distinguishing between **code** and **data irrespective** of user input
- Parameterized queries do not allow attackers to change the **intent of the query**

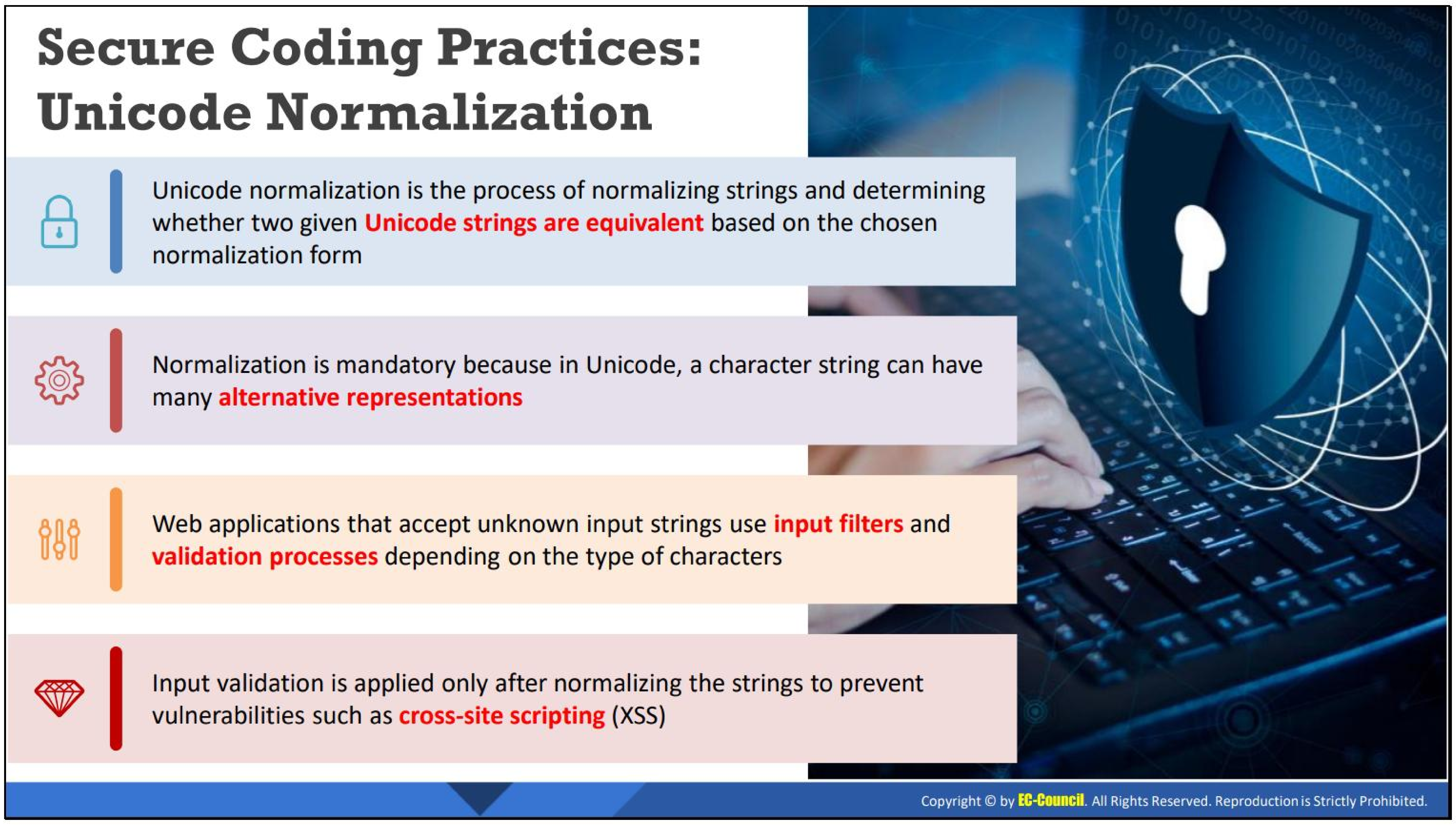
Parameterized Stored Procedures

- A parameterized stored procedure **allows the developer to write SQL code** first and then pass parameters to it
- The only difference with parametrized queries is that non-parameterized stored procedures are **stored in the database** with values supplied to them, and they are later called by the application

Parameterized Queries and Stored Procedures

In parameterized queries, SQL query is written without embedding parameters in it; instead, each parameter of query is supplied dynamically later. This technique helps in distinguishing between code and data irrespective of user input. Parameterized queries do not allow attackers to change the intent of the query.

The parameterized stored procedure also allows the developer to write SQL code first and then pass parameters to it. The only difference is that non-parameterized stored procedures are stored in the database with values supplied to them and then they are called by the application.



Secure Coding Practices: Unicode Normalization

- Unicode normalization is the process of normalizing strings and determining whether two given **Unicode strings are equivalent** based on the chosen normalization form
- Normalization is mandatory because in Unicode, a character string can have many **alternative representations**
- Web applications that accept unknown input strings use **input filters** and **validation processes** depending on the type of characters
- Input validation is applied only after normalizing the strings to prevent vulnerabilities such as **cross-site scripting** (XSS)

Unicode Normalization

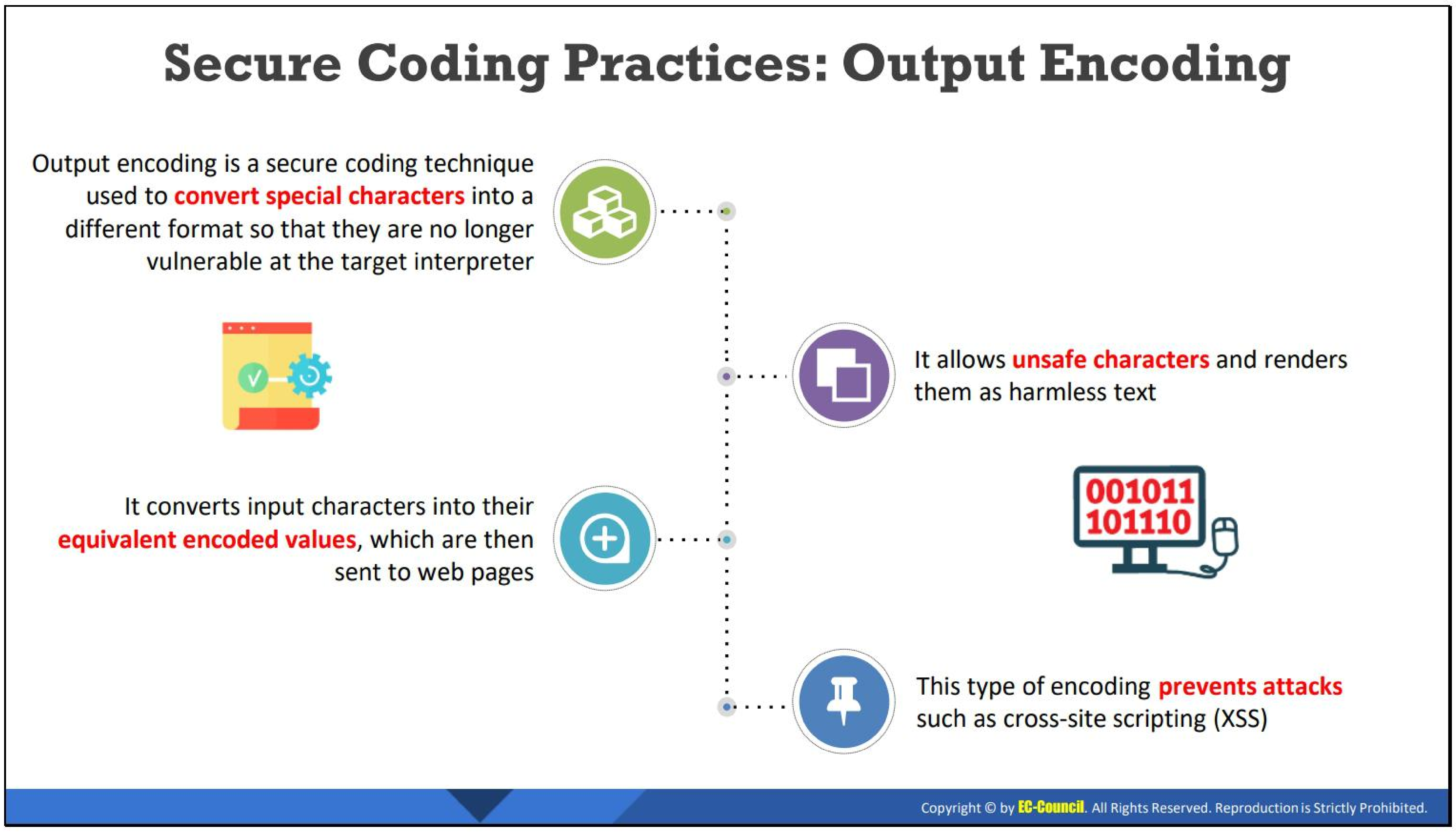
Unicode normalization is the process of normalizing strings and determining whether two given Unicode strings are equivalent based on the chosen normalization form. This mechanism is used while developing secure web applications to normalize input strings. With normalization, two strings with different binary representations acquire the same binary values. Normalization is mandatory because in Unicode, a character string can have many alternative representations. Web applications that accept unknown input strings use input filters and validation processes depending on the type of characters. Input validation is applied only after normalizing the strings to prevent vulnerabilities such as cross-site scripting (XSS). It is performed by correctly identifying and eliminating the <script> tags in the input text. Such processes are necessary in the security strategy, even if they are inadequate for thorough input sanitization and validation. There are two types of equivalence between characters: canonical equivalence and compatibility equivalence.

Canonical equivalence implies that two characters or a sequence of characters has the same meaning and visual appearance. In compatibility equivalence, also called the weaker equivalence of characters, two characters or a sequence of characters has the same abstract character(s) but different meanings and appearances.

Sample Code for Normalization Check

```
>>> str1 = '\u00F1';

>>> str2 = '\u006E\u0303';

>>> str1 == str2

False
```

```
>>>
unicodedata.normlize(“NFKD”,str1)==unicodedata.normlize(“NFKD”,str2)
True
>>>
unicodedata.normlize(“NFKC”,str1)==unicodedata.normlize(“NFKC”,str2)
True
```



Secure Coding Practices: Output Encoding

Output encoding is a secure coding technique used to **convert special characters** into a different format so that they are no longer vulnerable at the target interpreter

It allows **unsafe characters** and renders them as harmless text

It converts input characters into their **equivalent encoded values**, which are then sent to web pages

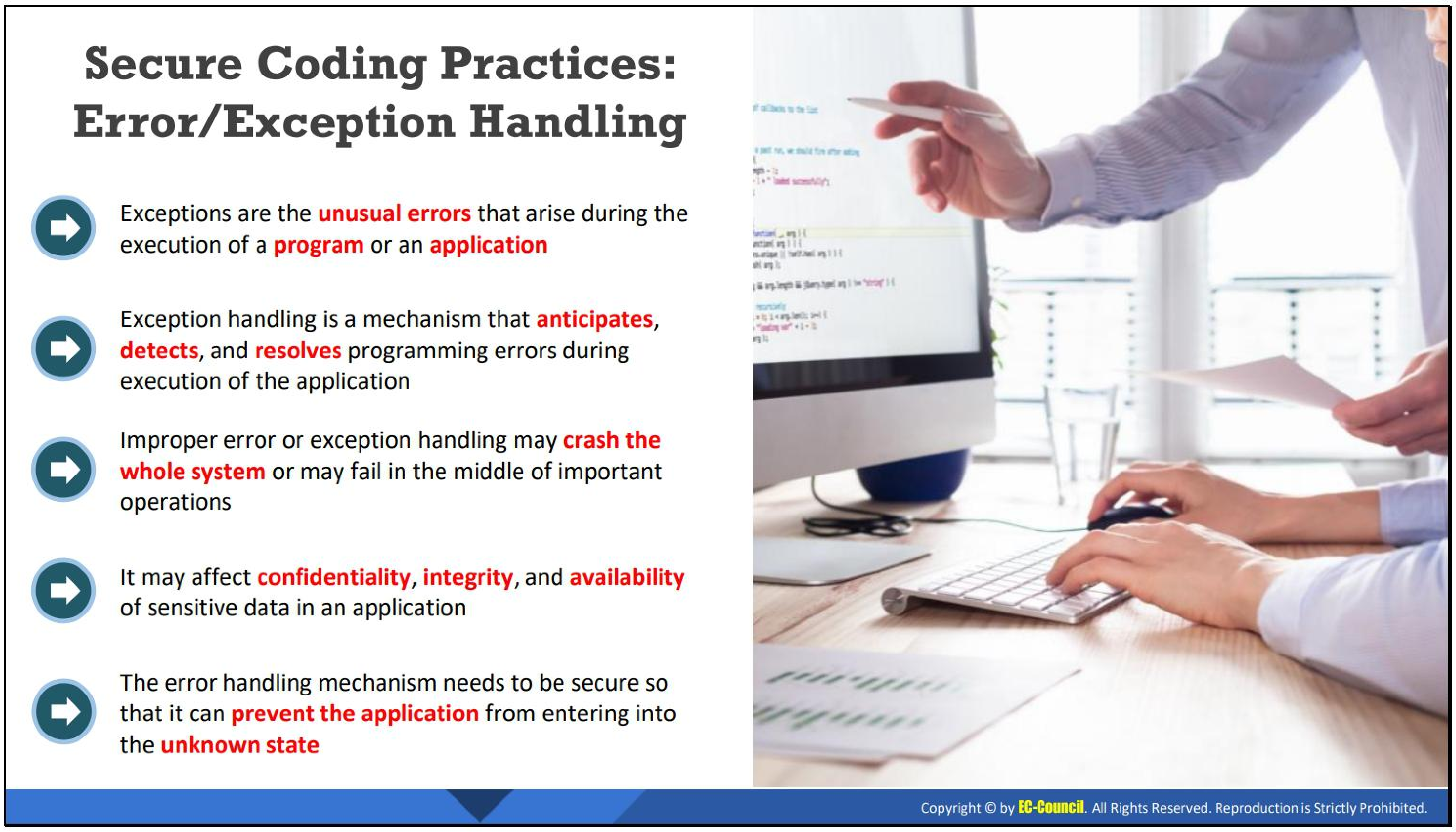
This type of encoding **prevents attacks** such as cross-site scripting (XSS)

Output Encoding

Output encoding is a secure coding technique used to convert special characters into a different format so that they are no longer vulnerable at the target interpreter. It allows unsafe characters and renders them as harmless text. Output encoding converts input characters into their equivalent encoded values, which are then sent to web pages. This type of encoding prevents attacks such as cross-site scripting (XSS).

For example, consider HTML encoding. An HTML encoding scheme is used to represent unusual characters so that they can be safely combined within an HTML document. HTML encoding replaces unusual characters with strings that can be recognized while the various characters define the structure of the document. If you want to use the same characters as those contained in the document, you might encounter problems. These problems can be overcome using HTML encoding. It defines several HTML entities to represent particularly usual characters such as:

- `&amp;   &`
- `&lt;    <`
- `&gt;    >`

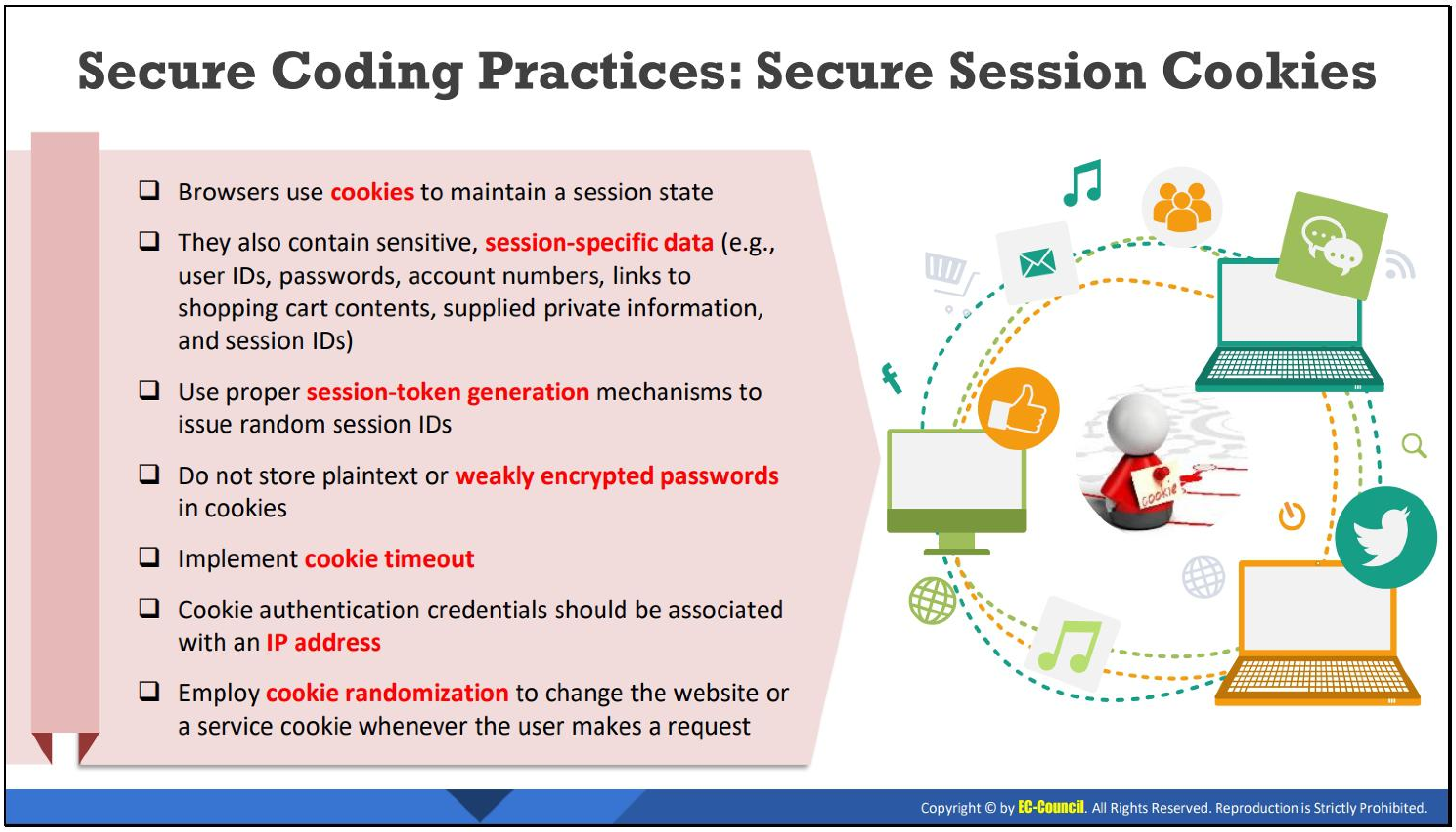


Secure Coding Practices: Error/Exception Handling

- Exceptions are the **unusual errors** that arise during the execution of a **program** or an **application**
- Exception handling is a mechanism that **anticipates**, **detects**, and **resolves** programming errors during execution of the application
- Improper error or exception handling may **crash the whole system** or may fail in the middle of important operations
- It may affect **confidentiality**, **integrity**, and **availability** of sensitive data in an application
- The error handling mechanism needs to be secure so that it can **prevent the application** from entering into the **unknown state**

Error/Exception Handling

Exceptions are the unusual errors that arise during the execution of a program or an application. As these exceptions arise at runtime, they cannot be handled or resolved at compile time. Exception handling is a mechanism that anticipates, detects, and resolves programming errors during execution of the application. Improper error or exception handling may crash the whole system or may fail in the middle of important operations. It may affect confidentiality, integrity, and availability of sensitive data in an application. Insecure error handling may give a path to the attacker to carry out an attack on the application. The detailed error message may give information about the internal structure of the application. The error messages can also help the attacker to constrain the search space to carry out attacks. The error handling mechanism needs to be secure so that it can prevent the application from entering into the unknown state.



Secure Coding Practices: Secure Session Cookies

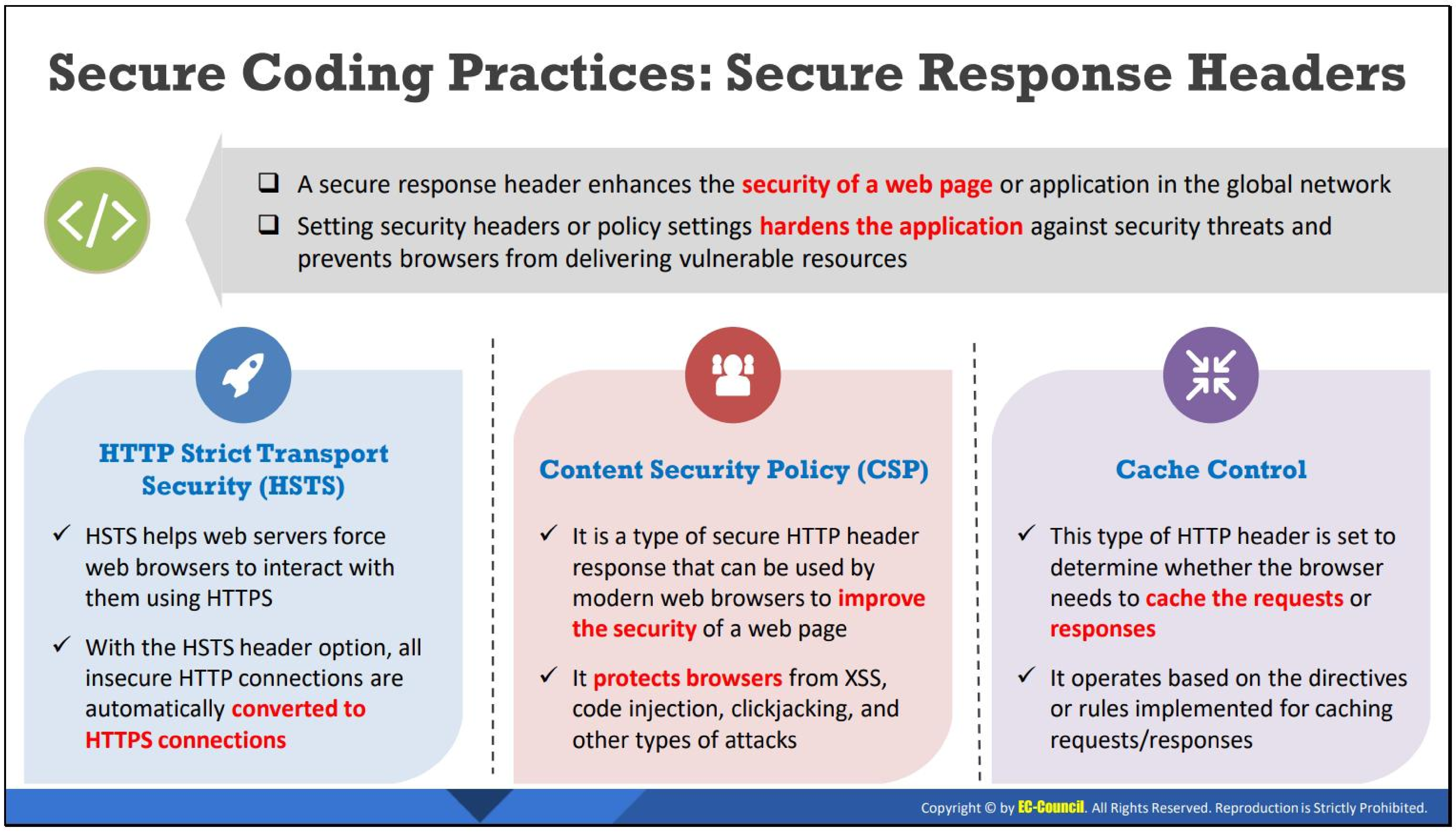
- ❑ Browsers use **cookies** to maintain a session state
- ❑ They also contain sensitive, **session-specific data** (e.g., user IDs, passwords, account numbers, links to shopping cart contents, supplied private information, and session IDs)
- ❑ Use proper **session-token generation** mechanisms to issue random session IDs
- ❑ Do not store plaintext or **weakly encrypted passwords** in cookies
- ❑ Implement **cookie timeout**
- ❑ Cookie authentication credentials should be associated with an **IP address**
- ❑ Employ **cookie randomization** to change the website or a service cookie whenever the user makes a request

Secure Session Cookies

Browsers use cookies to maintain a session state. They also contain sensitive, session-specific data (e.g., user IDs, passwords, account numbers, links to shopping cart contents, supplied private information, and session IDs). Attackers engage in cookie/session poisoning by modifying the data in the cookie to gain escalated access or maliciously affect a user session. Developers must hence follow secure coding practices to secure web applications against such poisoning attacks. They must use proper session-token generation mechanisms to issue random session IDs.

Considerations for secure session cookies:

- Do not store plaintext or weakly encrypted passwords in cookies
- Implement cookie timeout
- Cookie authentication credentials should be associated with an IP address
- Make logout functions available
- Validate all the cookie values to ensure that they are well-formed and correct
- Employ cookie randomization to change the website or a service cookie whenever the user makes a request



Secure Coding Practices: Secure Response Headers

- A secure response header enhances the **security of a web page** or application in the global network
- Setting security headers or policy settings **hardens the application** against security threats and prevents browsers from delivering vulnerable resources

HTTP Strict Transport Security (HSTS)

- ✓ HSTS helps web servers force web browsers to interact with them using HTTPS
- ✓ With the HSTS header option, all insecure HTTP connections are automatically **converted to HTTPS connections**

Content Security Policy (CSP)

- ✓ It is a type of secure HTTP header response that can be used by modern web browsers to **improve the security** of a web page
- ✓ It **protects browsers** from XSS, code injection, clickjacking, and other types of attacks

Cache Control

- ✓ This type of HTTP header is set to determine whether the browser needs to **cache the requests** or **responses**
- ✓ It operates based on the directives or rules implemented for caching requests/responses

Secure Response Headers

Many security implementations can be applied to response headers from a host or server. A secure response header enhances the security of the web page or application in the global network. Security headers or policy settings harden the application against security attacks and prevent browsers from delivering vulnerable resources. Using the application's compatibility and accessibility, programmers can enforce functionality via the headers, making the application secure and flexible. The following are some of the most common security header implementations.

- **HTTP Strict Transport Security (HSTS)**

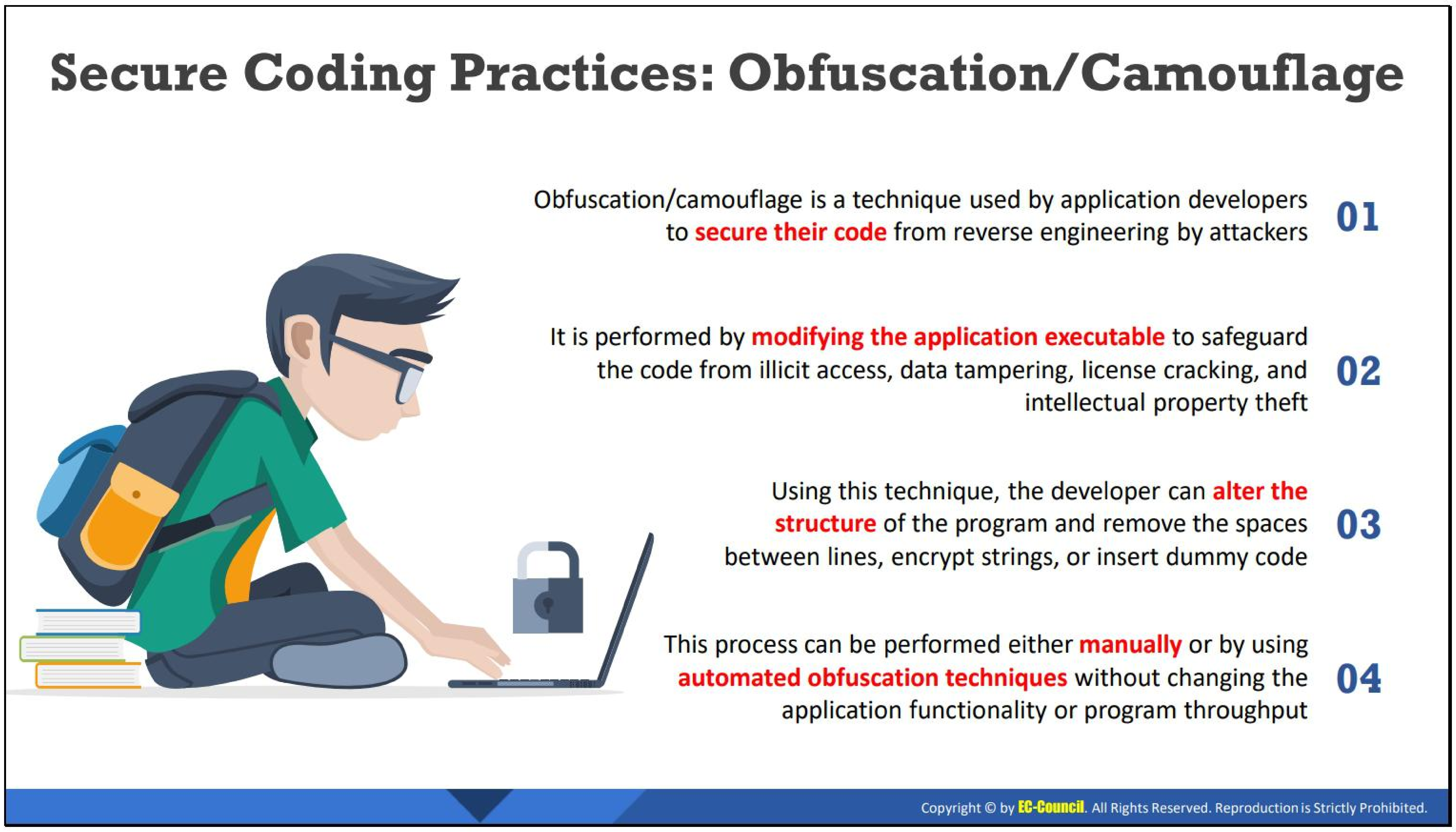
  HSTS enables web servers to force web browsers to interact with them using HTTPS. With the HSTS header option, all insecure HTTP connections are automatically converted into HTTPS connections. This policy ensures that all the communication between a web server and a web browser is encrypted and that all responses that are delivered and received originate from an authenticated server.

- **Content Security Policy (CSP)**

  CSP is another type of secure HTTP header response that can be used by modern web browsers to improve the security of the web page. By implementing CSP, the browser can be safeguarded from XSS, code injection, clickjacking, and other types of attacks. CSP allows an administrator to efficiently control the content resources that are directed to the user agent.

- **Cache Control**

  This type of HTTP header is set to determine whether the browser needs to cache requests or responses because, occasionally, caching private or sensitive information related to users can have unwanted consequences. It operates based on the directives or rules implemented for caching requests/responses. The caching policy can also be used to determine the location at which the resources are to be cached as well as the time-to-live for the cached resources.



Secure Coding Practices: Obfuscation/Camouflage

01 Obfuscation/camouflage is a technique used by application developers to **secure their code** from reverse engineering by attackers

02 It is performed by **modifying the application executable** to safeguard the code from illicit access, data tampering, license cracking, and intellectual property theft

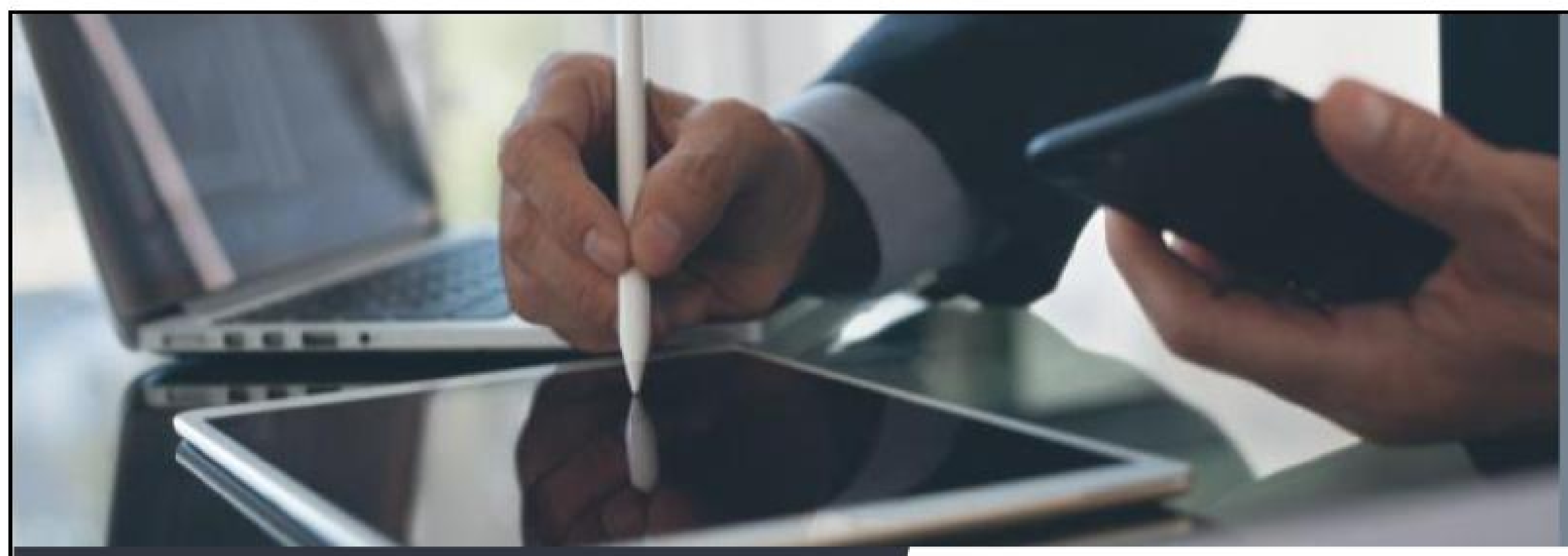
03 Using this technique, the developer can **alter the structure** of the program and remove the spaces between lines, encrypt strings, or insert dummy code

04 This process can be performed either **manually** or by using **automated obfuscation techniques** without changing the application functionality or program throughput

Obfuscation/Camouflage

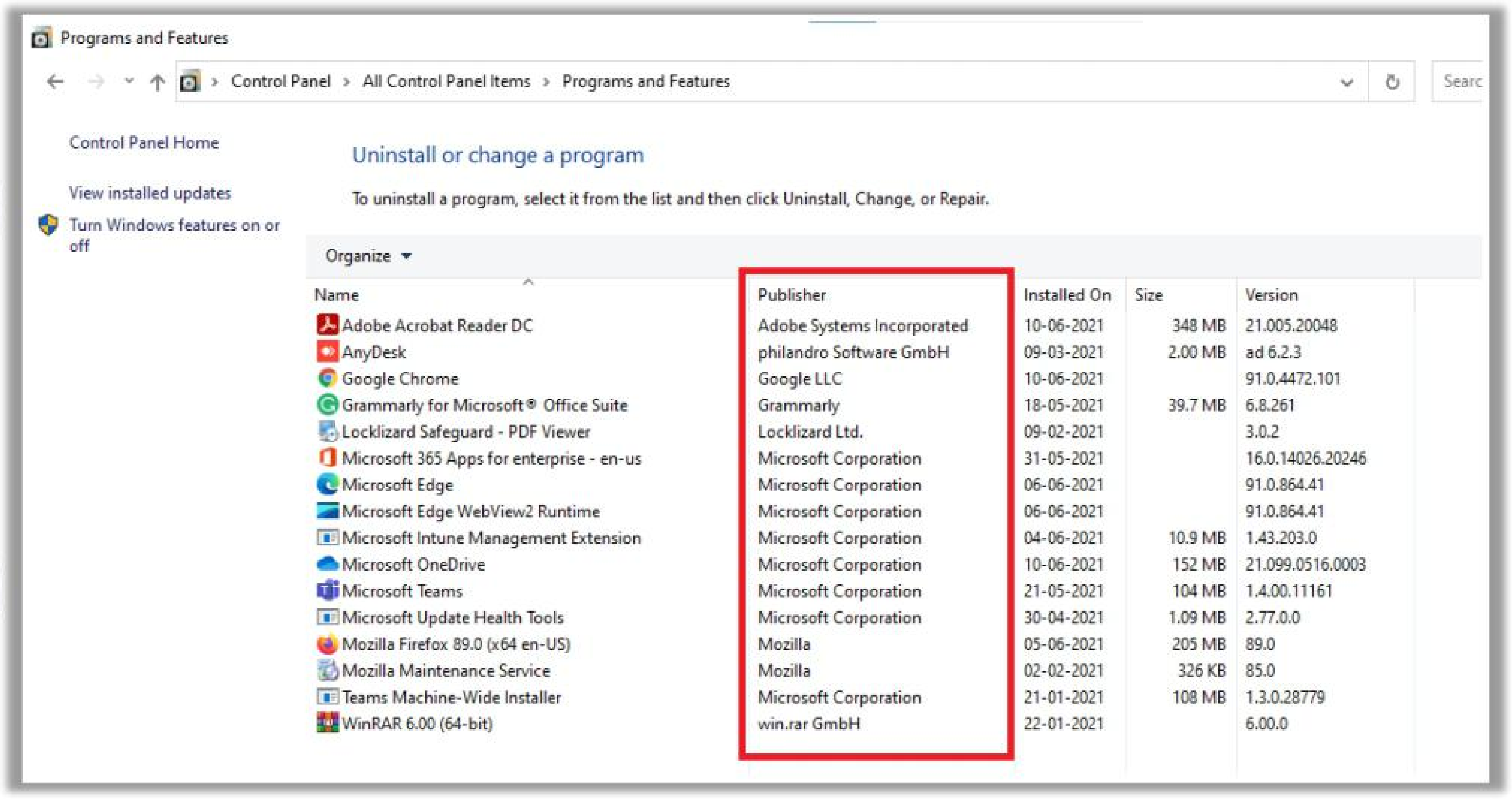
Obfuscation or camouflage is a technique used by application developers to secure their code from being reverse engineered by attackers. It is achieved by modifying the application executable to safeguard the code from illicit access, license cracking, data tampering, and intellectual-property theft. This practice uses many programming techniques to obfuscate the source code, making it impractical for human understanding. Using this technique, the developer can alter the structure of the program by changing variables names, function names, and method names; the developer can also remove spaces between lines, metadata, and unused code; encrypt strings; or insert dummy code.

This process can be performed either manually or by using automated obfuscation techniques. Although it does not change the application functionality or program throughput, in some cases, it might affect the runtime performance. The obfuscation process is followed in different types of applications such as JAVA, Android, iOS, and .NET. Owing to the easy availability of decompilers and deobfuscation tools to reverse engineer and extract source code from executables, application developers should strictly focus on using obfuscation techniques to protect the application. Attackers also use these types of obfuscation techniques to conceal malicious code.



Secure Coding Practices: Code Signing

- Code signing is used by software publishers/developers to **digitally sign software**, product updates, or executables
- This signature ensures the **integrity** and **authenticity** of an application or software upon installation and execution on a system
- Code signing involves the **usage of the PKI mechanism** based on a public and private key to sign and verify the certificate containing the developer and publisher information



Programs and Features

Control Panel > All Control Panel Items > Programs and Features

Control Panel Home
View installed updates
Turn Windows features on or off

Uninstall or change a program

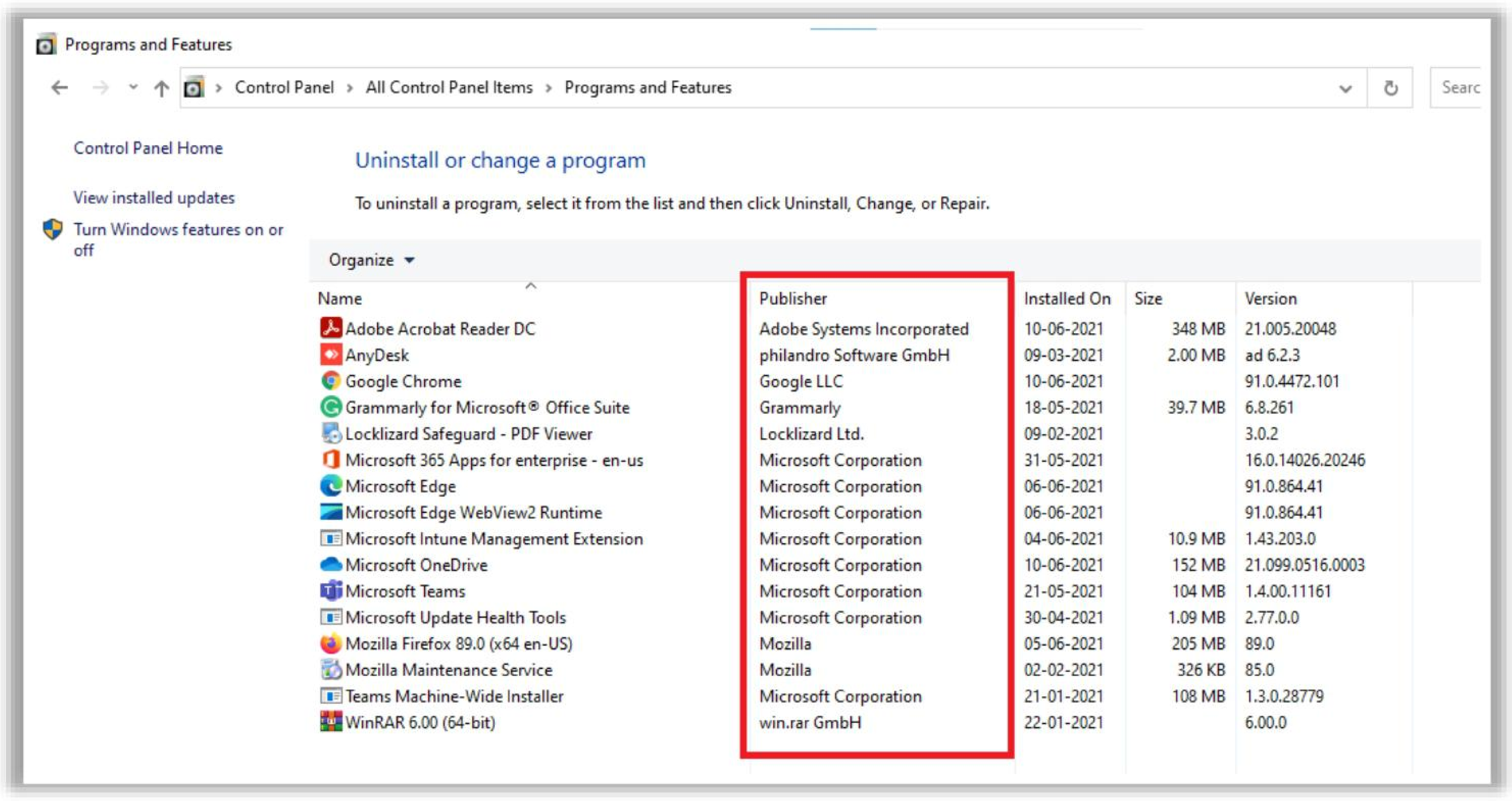
To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Organize

Name	Publisher	Installed On	Size	Version
Adobe Acrobat Reader DC	Adobe Systems Incorporated	10-06-2021	348 MB	21.005.20048
AnyDesk	philandro Software GmbH	09-03-2021	2.00 MB	ad 6.2.3
Google Chrome	Google LLC	10-06-2021		91.0.4472.101
Grammarly for Microsoft® Office Suite	Grammarly	18-05-2021	39.7 MB	6.8.261
Locklizard Safeguard - PDF Viewer	Locklizard Ltd.	09-02-2021		3.0.2
Microsoft 365 Apps for enterprise - en-us	Microsoft Corporation	31-05-2021		16.0.14026.20246
Microsoft Edge	Microsoft Corporation	06-06-2021		91.0.864.41
Microsoft Edge WebView2 Runtime	Microsoft Corporation	06-06-2021		91.0.864.41
Microsoft Intune Management Extension	Microsoft Corporation	04-06-2021	10.9 MB	1.43.203.0
Microsoft OneDrive	Microsoft Corporation	10-06-2021	152 MB	21.099.0516.0003
Microsoft Teams	Microsoft Corporation	21-05-2021	104 MB	1.4.00.11161
Microsoft Update Health Tools	Microsoft Corporation	30-04-2021	1.09 MB	2.77.0.0
Mozilla Firefox 89.0 (x64 en-US)	Mozilla	05-06-2021	205 MB	89.0
Mozilla Maintenance Service	Mozilla	02-02-2021	326 KB	85.0
Teams Machine-Wide Installer	Microsoft Corporation	21-01-2021	108 MB	1.3.0.28779
WinRAR 6.00 (64-bit)	win.rar GmbH	22-01-2021		6.00.0

Code Signing

Code signing is used by software publishers/developers to digitally sign software, product updates, or executables. This signature ensures the integrity and authenticity of the application or software upon its installation and execution on a system. Code signing also determines whether the software has been downloaded from an authorized publisher/developer or from an attacker. It also ensures that the software is not modified or changed after being signed by the developer and is safe to download. Code signing involves the use of a public and private key–based PKI mechanism to sign and verify the certificate containing the developer and publisher information. The programmers or publishers sign the certificate with their private key, and clients utilize the publisher's public key to determine the authenticity.


Programs and Features

Control Panel > All Control Panel Items > Programs and Features

Control Panel Home

View installed updates

Turn Windows features on or off

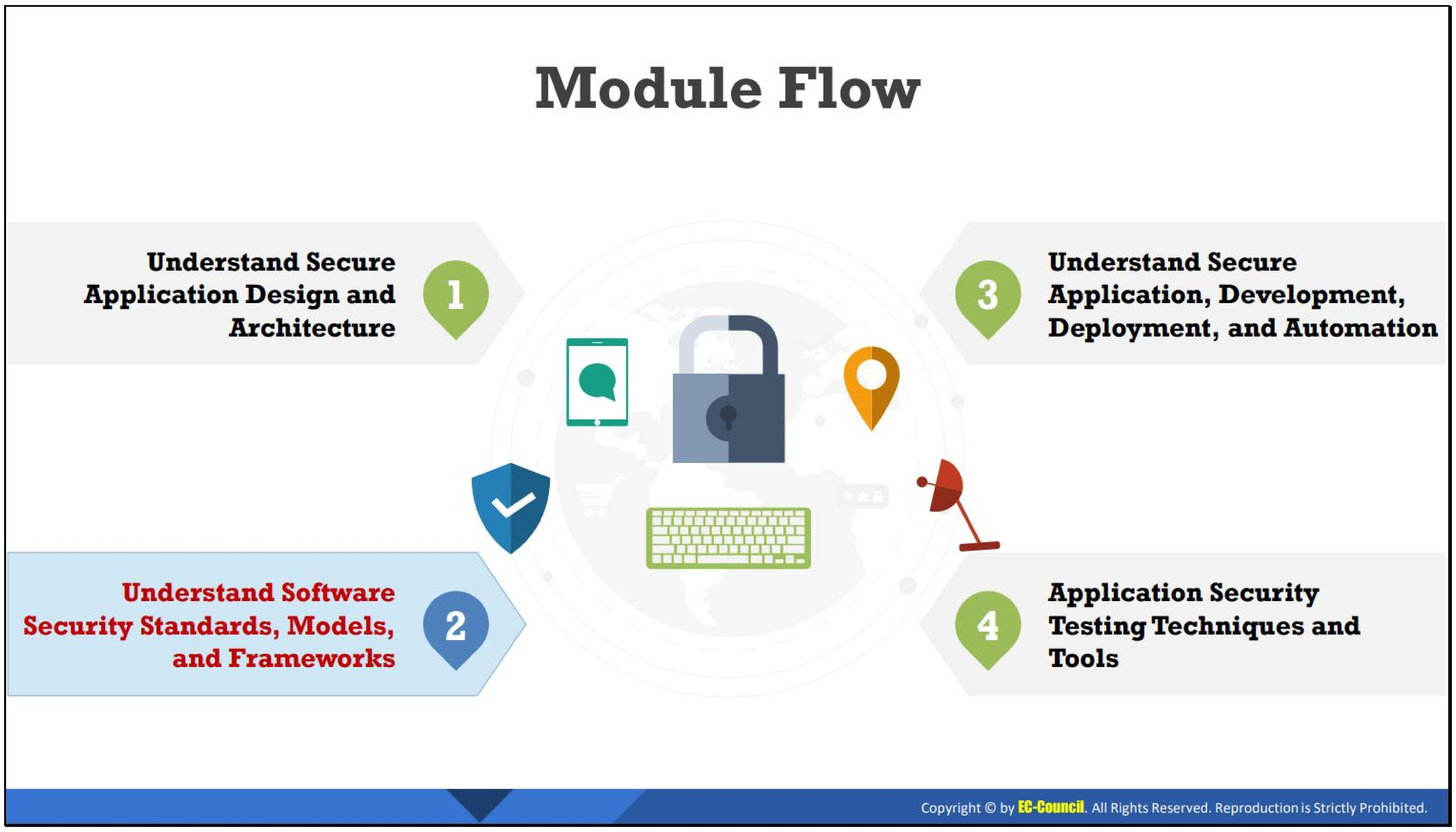
Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Organize

Name	Publisher	Installed On	Size	Version
Adobe Acrobat Reader DC	Adobe Systems Incorporated	10-06-2021	348 MB	21.005.20048
AnyDesk	philandro Software GmbH	09-03-2021	2.00 MB	ad 6.2.3
Google Chrome	Google LLC	10-06-2021		91.0.4472.101
Grammarly for Microsoft® Office Suite	Grammarly	18-05-2021	39.7 MB	6.8.261
Locklizard Safeguard - PDF Viewer	Locklizard Ltd.	09-02-2021		3.0.2
Microsoft 365 Apps for enterprise - en-us	Microsoft Corporation	31-05-2021		16.0.14026.20246
Microsoft Edge	Microsoft Corporation	06-06-2021		91.0.864.41
Microsoft Edge WebView2 Runtime	Microsoft Corporation	06-06-2021		91.0.864.41
Microsoft Intune Management Extension	Microsoft Corporation	04-06-2021	10.9 MB	1.43.203.0
Microsoft OneDrive	Microsoft Corporation	10-06-2021	152 MB	21.099.0516.0003
Microsoft Teams	Microsoft Corporation	21-05-2021	104 MB	1.4.00.11161
Microsoft Update Health Tools	Microsoft Corporation	30-04-2021	1.09 MB	2.77.0.0
Mozilla Firefox 89.0 (x64 en-US)	Mozilla	05-06-2021	205 MB	89.0
Mozilla Maintenance Service	Mozilla	02-02-2021	326 KB	85.0
Teams Machine-Wide Installer	Microsoft Corporation	21-01-2021	108 MB	1.3.0.28779
WinRAR 6.00 (64-bit)	win.rar GmbH	22-01-2021		6.00.0

Figure 9.6: Screenshot showing signed publisher details

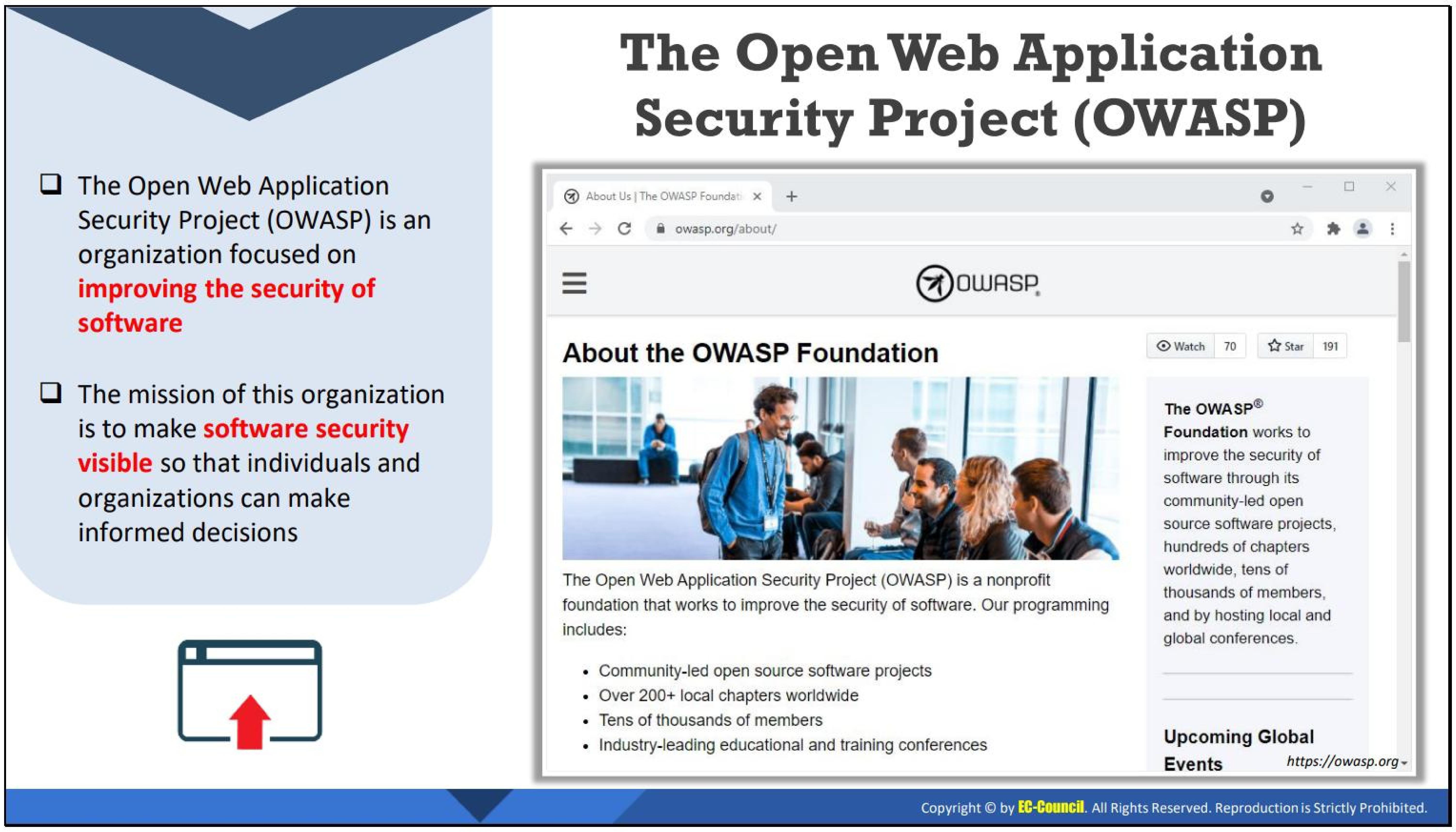


Module Flow

1. Understand Secure Application Design and Architecture
2. Understand Software Security Standards, Models, and Frameworks
3. Understand Secure Application, Development, Deployment, and Automation
4. Application Security Testing Techniques and Tools

Understand Software Security Standards, Models, and Frameworks

Numerous organizations face several impediments in their information security platforms due to tremendous increases in infrastructure. Hence, it is necessary to follow basic software security standards, models, and frameworks, which consist of strategies and techniques for implementing information security controls in an organization. This section provides an overview of various software security standards, models, and frameworks.



The Open Web Application Security Project (OWASP)

- The Open Web Application Security Project (OWASP) is an organization focused on **improving the security of software**
- The mission of this organization is to make **software security visible** so that individuals and organizations can make informed decisions

About the OWASP Foundation

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. Our programming includes:

- Community-led open source software projects
- Over 200+ local chapters worldwide
- Tens of thousands of members
- Industry-leading educational and training conferences

The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

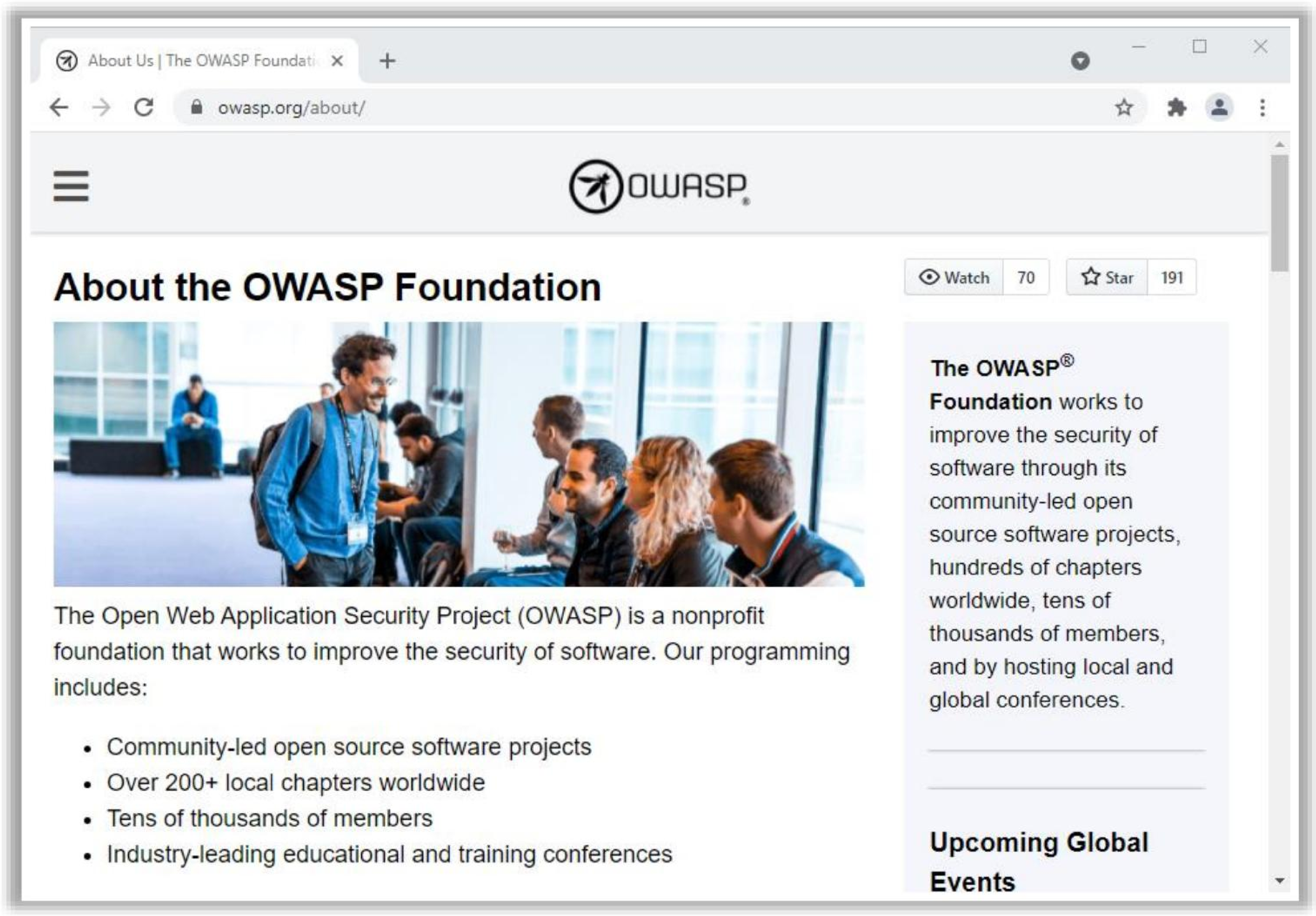
Upcoming Global Events

https://owasp.org

The Open Web Application Security Project (OWASP)

Source: *https://owasp.org*

The Open Web Application Security Project (OWASP) is an organization focused on improving the security of software. The mission of this organization is to make software security visible so that individuals and organizations are able to make informed decisions. OWASP is a community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. Through community-led software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP foundation has become an important source for developers and technologists to secure the web.



About the OWASP Foundation

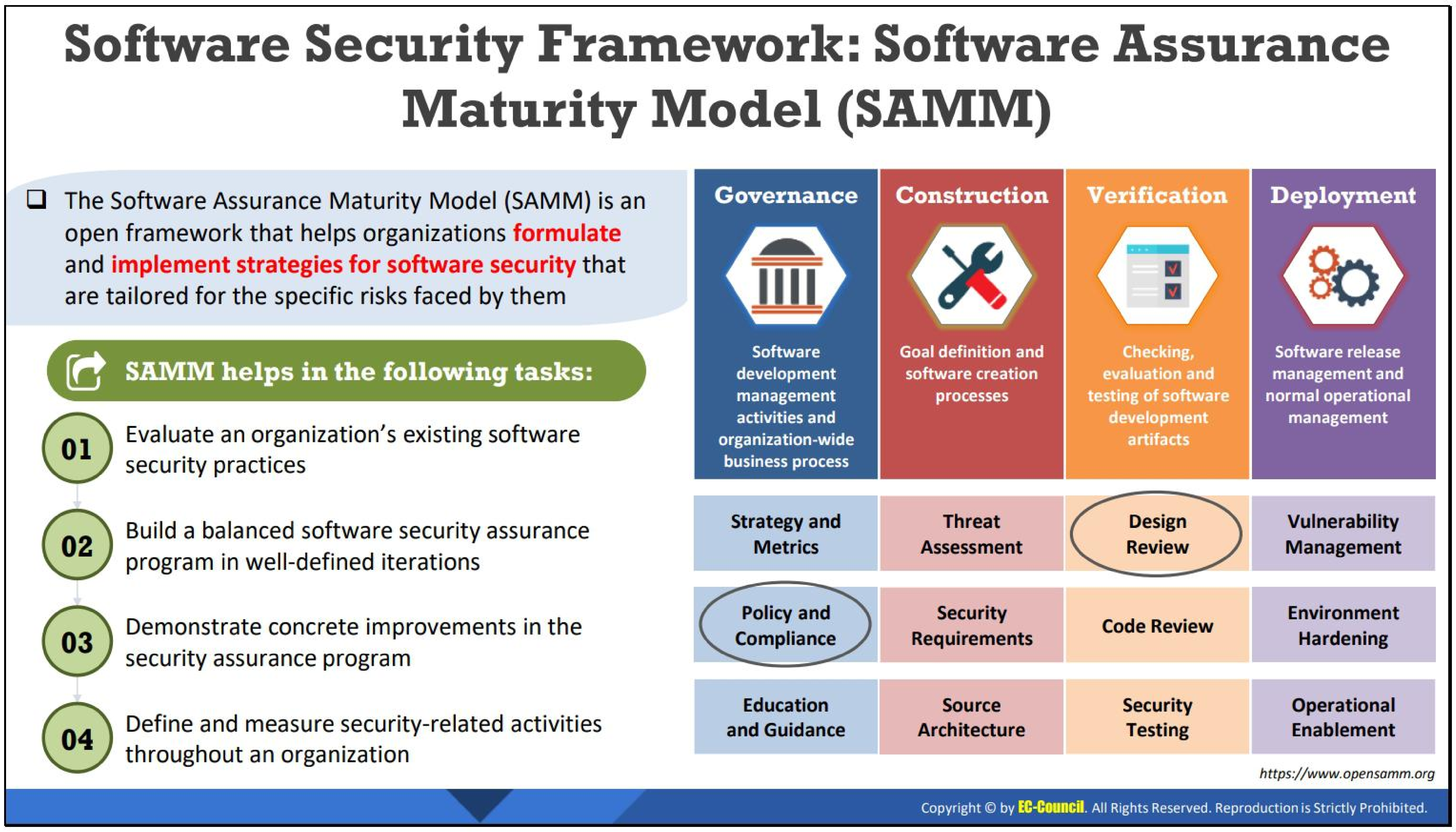
The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. Our programming includes:

- Community-led open source software projects
- Over 200+ local chapters worldwide
- Tens of thousands of members
- Industry-leading educational and training conferences

The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

Upcoming Global Events

Figure 9.7: Screenshot of OWASP



Software Security Framework: Software Assurance Maturity Model (SAMM)

- The Software Assurance Maturity Model (SAMM) is an open framework that helps organizations **formulate** and **implement strategies for software security** that are tailored for the specific risks faced by them

SAMM helps in the following tasks:

01 Evaluate an organization's existing software security practices

02 Build a balanced software security assurance program in well-defined iterations

03 Demonstrate concrete improvements in the security assurance program

04 Define and measure security-related activities throughout an organization

Governance	Construction	Verification	Deployment
Software development management activities and organization-wide business process	Goal definition and software creation processes	Checking, evaluation and testing of software development artifacts	Software release management and normal operational management
Strategy and Metrics	Threat Assessment	Design Review	Vulnerability Management
Policy and Compliance	Security Requirements	Code Review	Environment Hardening
Education and Guidance	Source Architecture	Security Testing	Operational Enablement

https://www.opensamm.org

Software Security Framework: Software Assurance Maturity Model (SAMM)

Source: *https://www.opensamm.org*

The Software Assurance Maturity Model (SAMM) is an open framework that helps organizations formulate and implement strategies for software security that are tailored for the specific risks faced by them. SAMM helps in the following tasks:

- Evaluate an organization's existing software security practices.
- Build a balanced software security assurance program in well-defined iterations.
- Demonstrate concrete improvements in the security assurance program.
- Define and measure security-related activities throughout an organization.

The maturity model consists of four business functions and each function possess three security practices.

- **Governance:** Assess the management of application security in an organization
- **Construction:** Assess the software creation process in an organization
- **Verification:** Assess the software testing of the application
- **Deployment:** Assess the deployment (Software release management) and production of the application

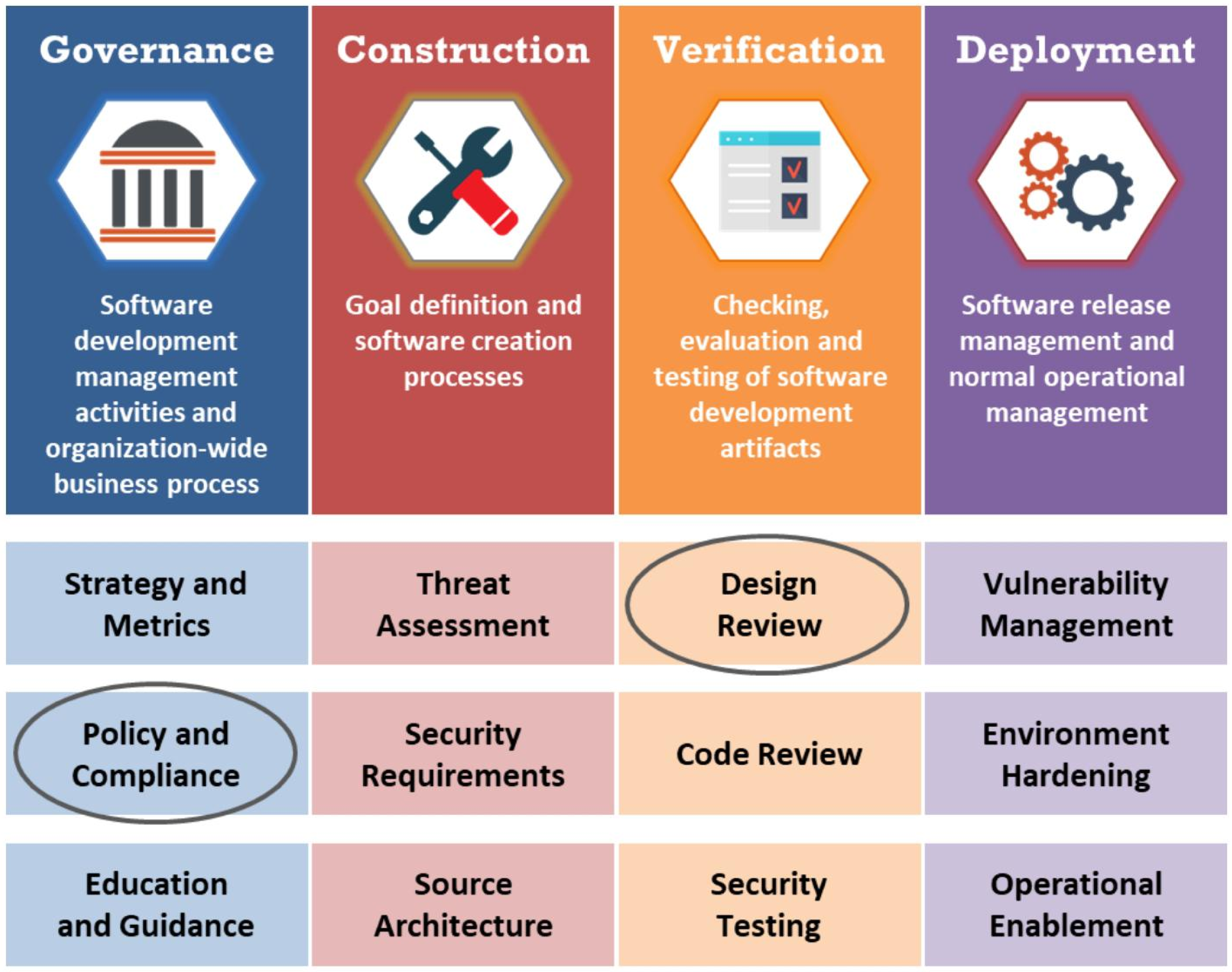
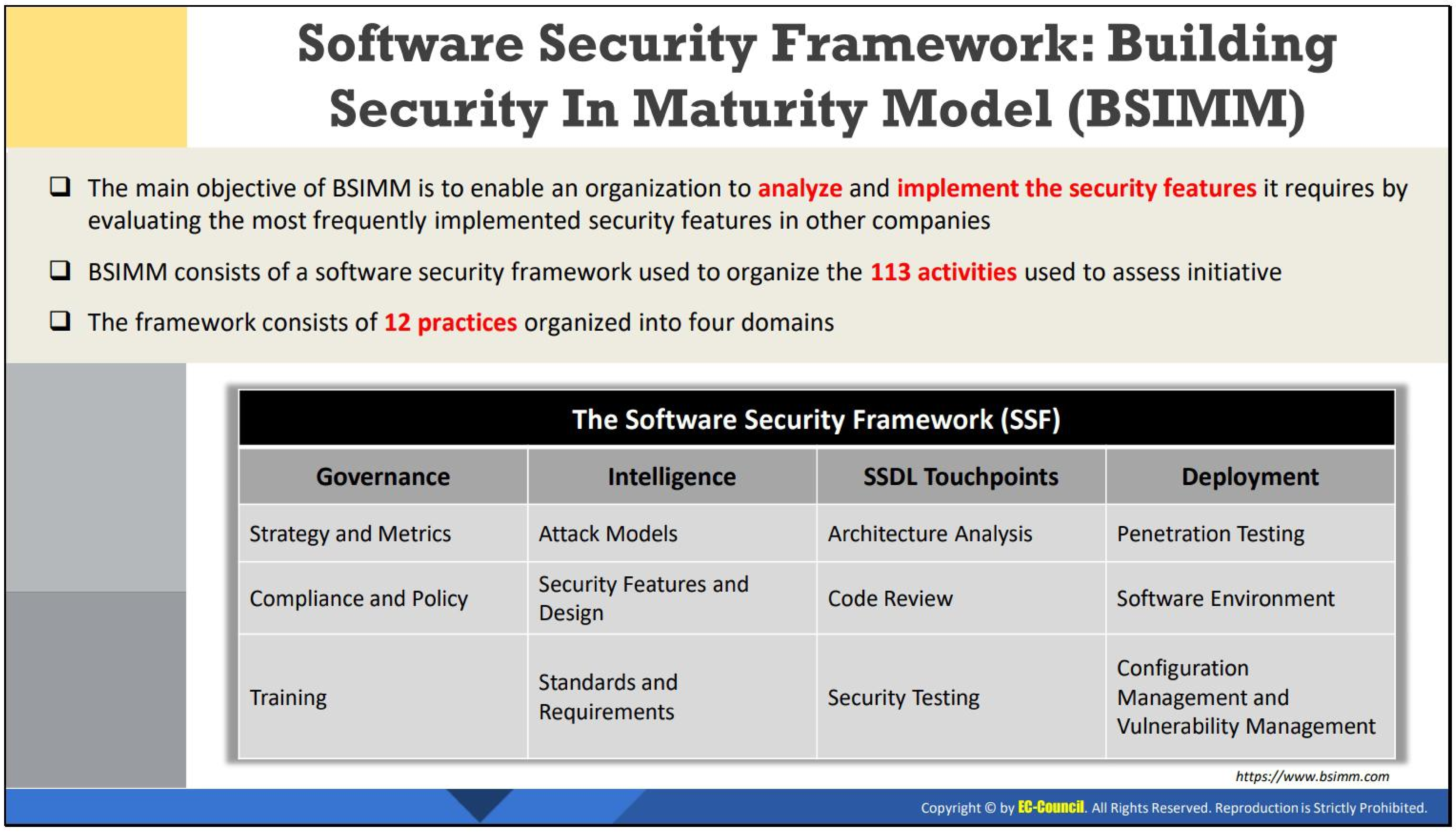


Figure 9.8: Software Assurance Maturity Model (SAMM)



Software Security Framework: Building Security In Maturity Model (BSIMM)

- The main objective of BSIMM is to enable an organization to **analyze** and **implement the security features** it requires by evaluating the most frequently implemented security features in other companies
- BSIMM consists of a software security framework used to organize the **113 activities** used to assess initiative
- The framework consists of **12 practices** organized into four domains

The Software Security Framework (SSF)			
Governance	**Intelligence**	**SSDL Touchpoints**	**Deployment**
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

https://www.bsimm.com

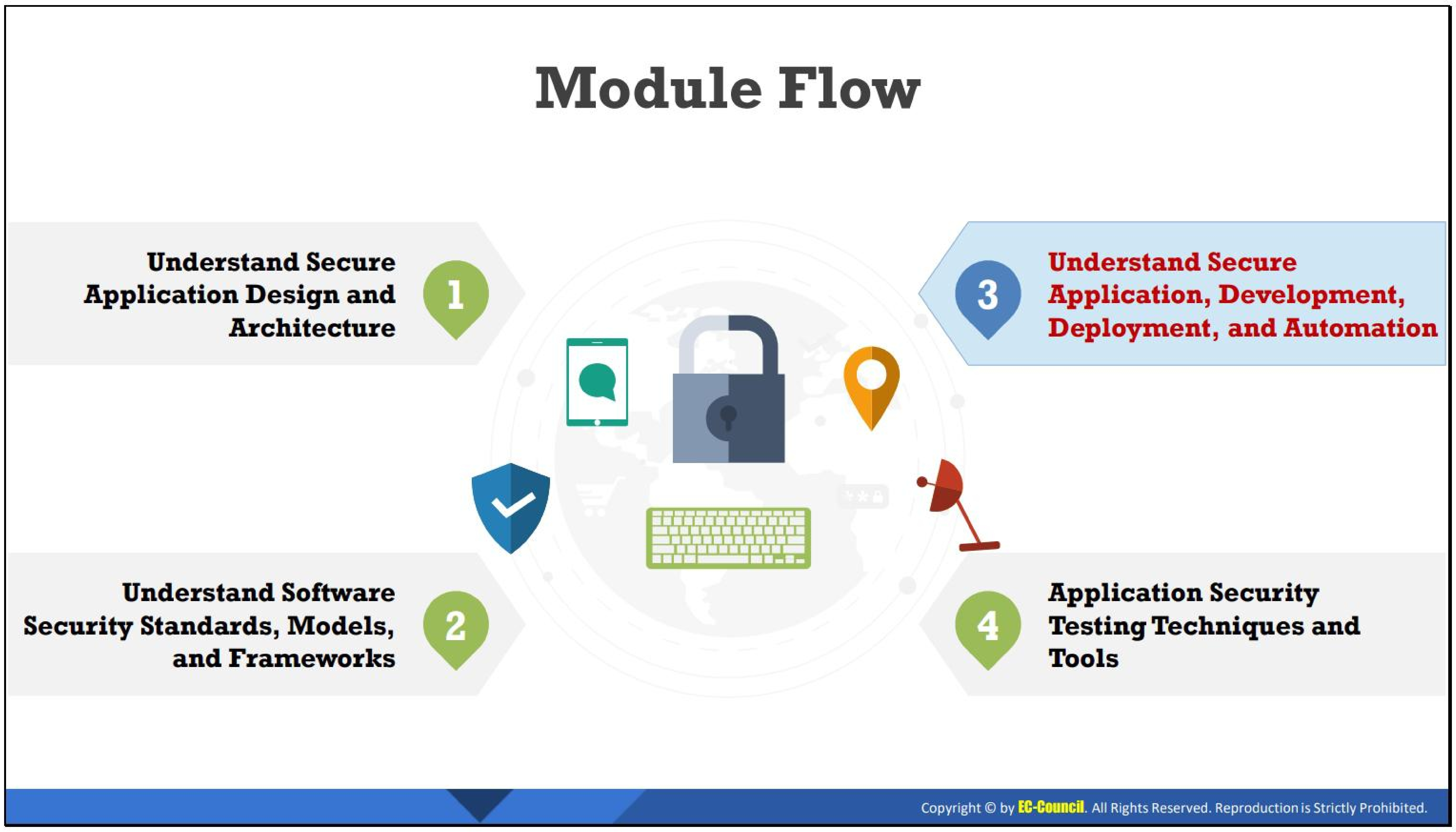
Software Security Framework: Building Security In Maturity Model (BSIMM)

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

The main objective of BSIMM is to enable an organization to analyze and implement the security features it requires by evaluating the most frequently implemented security features in other companies. BSIMM consists of a software security framework used to organize the 113 activities used to assess initiatives. The framework consists of 12 practices organized into four domains. The BSIMM is designed to help the organization understand, measure, and plan a software security initiative. It was created by observing and analyzing real-world data from leading software security initiatives. BSIMM data reflect how many organizations are adapting their approaches to address the new dynamics of modern development and deployment practices, such as shorter release cycles, increased use of automation, and software-defined infrastructure.

The Software Security Framework (SSF)			
Governance	**Intelligence**	**SSDL Touchpoints**	**Deployment**
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

Table 9.1: The Software Security Framework (SSF)

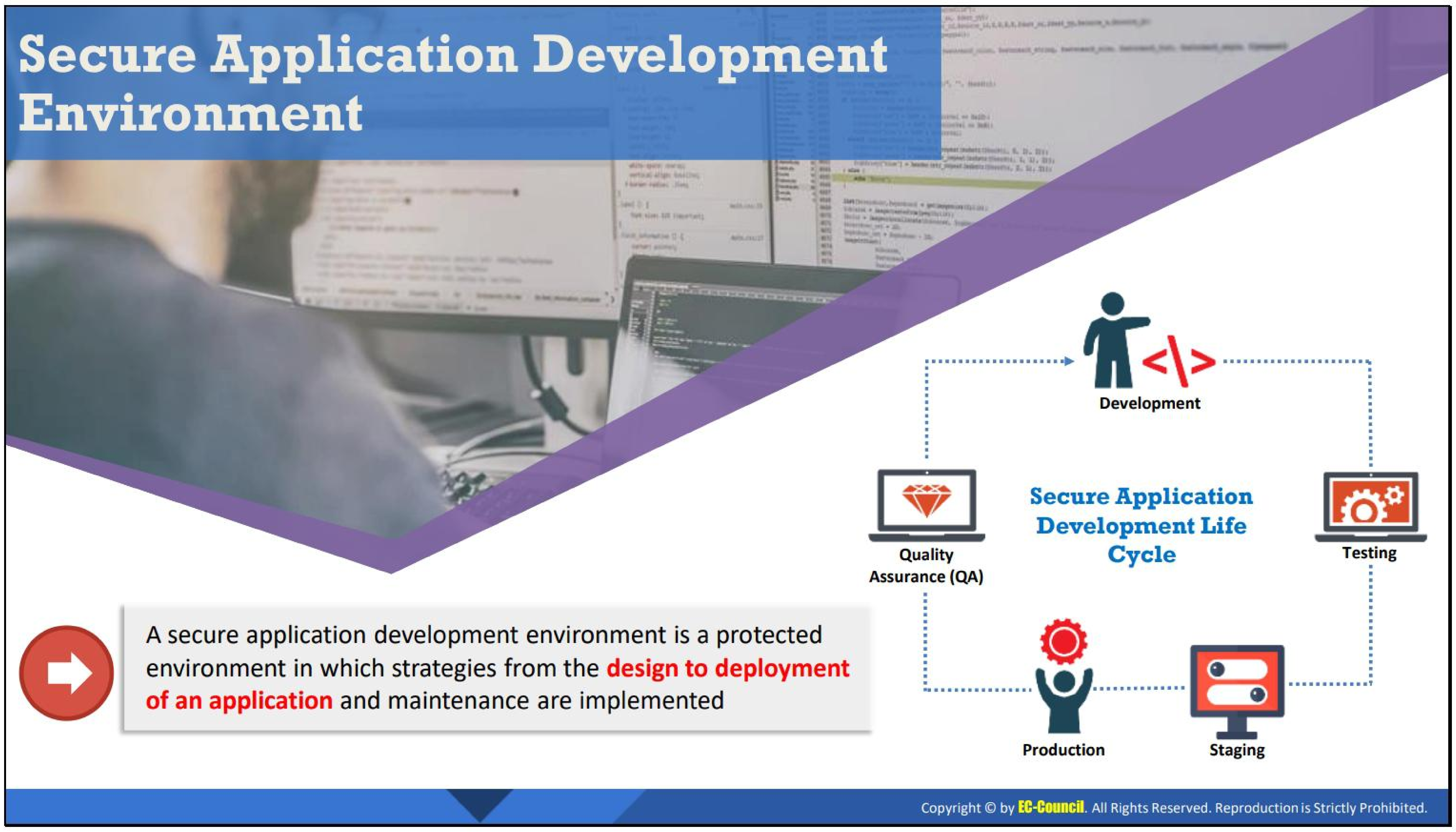


Module Flow

1. Understand Secure Application Design and Architecture
2. Understand Software Security Standards, Models, and Frameworks
3. Understand Secure Application, Development, Deployment, and Automation
4. Application Security Testing Techniques and Tools

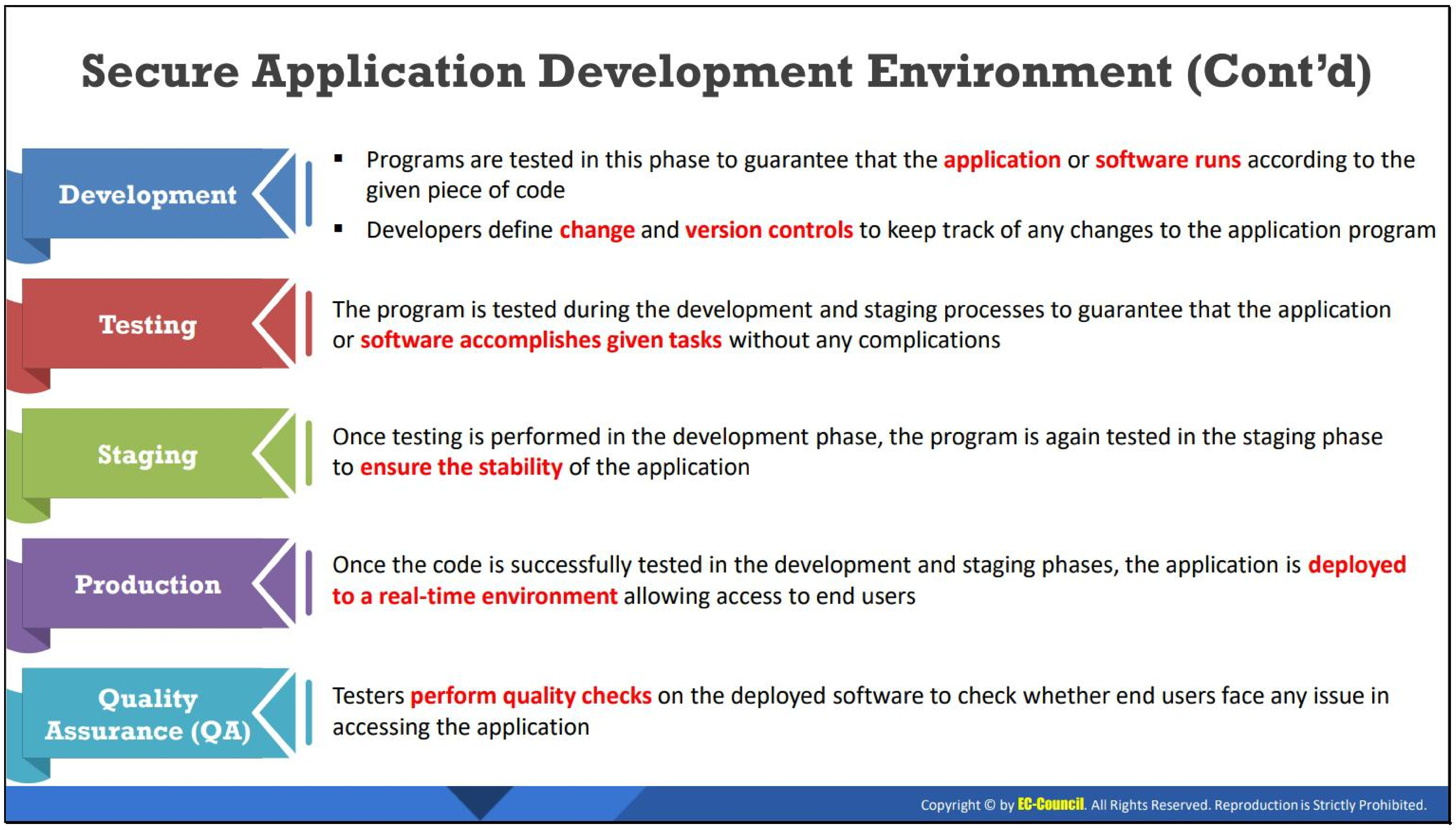
Understand Secure Application, Development, Deployment, and Automation

Most organizations concentrate on automation for the development and deployment of an application to cut superfluous operating costs. Although automation can make the software fast and efficient, it can also degrade the performance if proper security baselining is not followed. A security analyst should follow development lifecycle models and techniques to prevent any unknown vulnerabilities and backdoors in the application. This section discusses secure application development, deployment, and automation.



Secure Application Development Environment

A secure application development environment is a protected environment in which strategies from the **design to deployment of an application** and maintenance are implemented



Secure Application Development Environment (Cont'd)

Development	▪ Programs are tested in this phase to guarantee that the **application** or **software runs** according to the given piece of code ▪ Developers define **change** and **version controls** to keep track of any changes to the application program
Testing	The program is tested during the development and staging processes to guarantee that the application or **software accomplishes given tasks** without any complications
Staging	Once testing is performed in the development phase, the program is again tested in the staging phase to **ensure the stability** of the application
Production	Once the code is successfully tested in the development and staging phases, the application is **deployed to a real-time environment** allowing access to end users
Quality Assurance (QA)	Testers **perform quality checks** on the deployed software to check whether end users face any issue in accessing the application

Secure Application Development Environment

A secure application development environment is a protected environment in which strategies from the design to deployment of an application and maintenance are implemented. The secure application development lifecycle has a five-tier architecture. It begins with development, testing, and staging, following which it moves into the production and quality assurance (QA) levels, thereby ensuring the overall efficiency and integrity of the software or application. The five tiers of the secure application development lifecycle are described below.

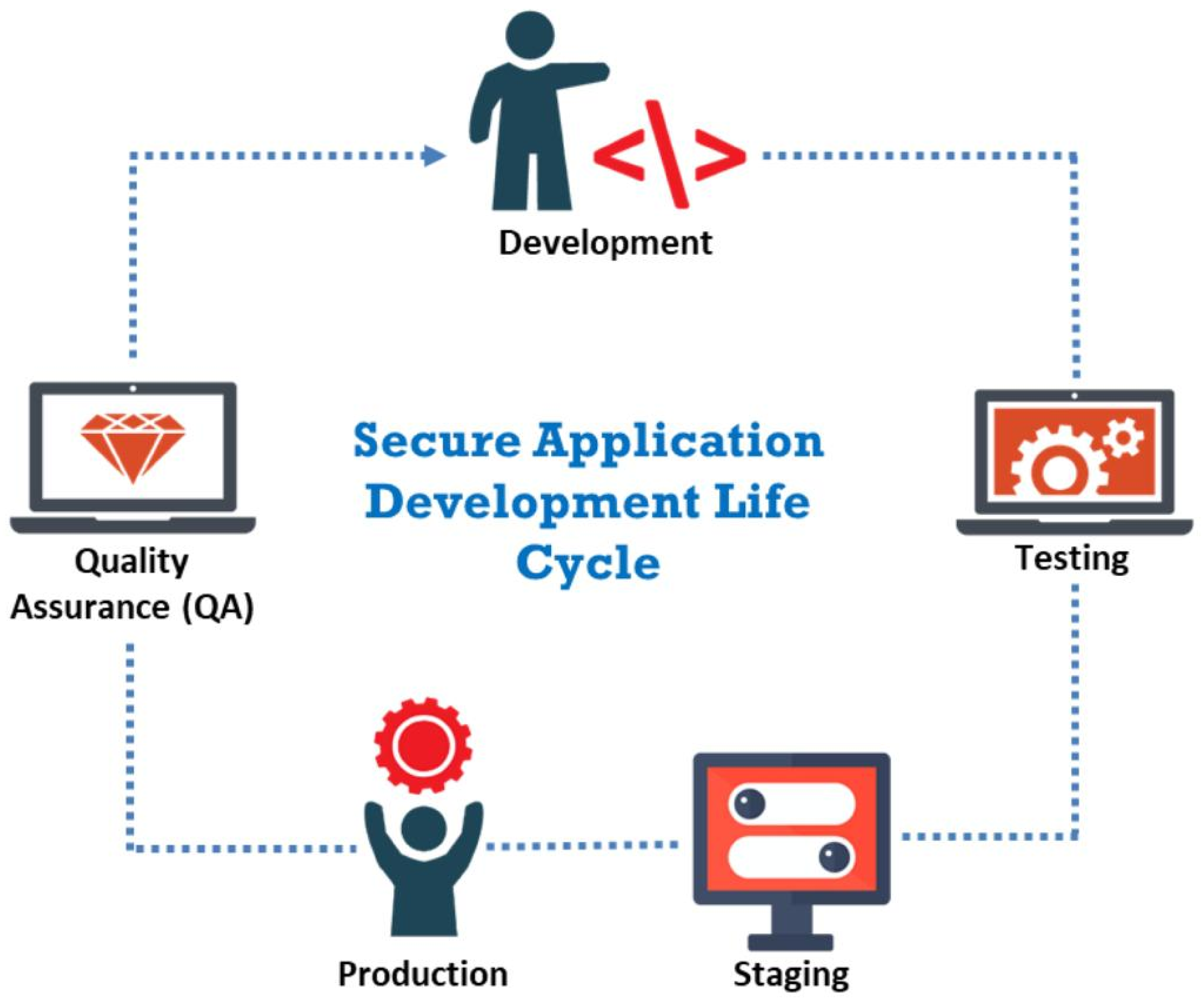


Figure 9.9: Secure application development lifecycle

- **Development**: The program is tested in this phase to guarantee that the application or software runs according to the given piece of code. Subsequently, the program is transferred to the next environment for additional enhancements. In the development phase, developers define change and version controls to keep track of any changes to the application program.
- **Testing**: The program is tested during the development and staging processes to guarantee that the application or software accomplishes given tasks without any complications. In this phase, testers receive the application code from various developers, and they scan the code to identify any bugs or errors.
- **Staging**: After testing is performed in the development phase, the program is again tested in the staging phase to ensure the stability of the application. Once the program is tested in this environment, it is transferred to the production phase. This phase provides a later copy of the testing environment.
- **Production**: Once the code is successfully tested in the development and staging phases, the application is deployed to a real-time environment to allow access to end users. This stage provides everything required to run an application.
- **Quality Assurance (QA)**: In this phase, the application's performance is monitored, and its quality is evaluated in the end users' network. Testers perform quality checks on the deployed software to determine whether end users face any issue in accessing the application.

Secure Baseline

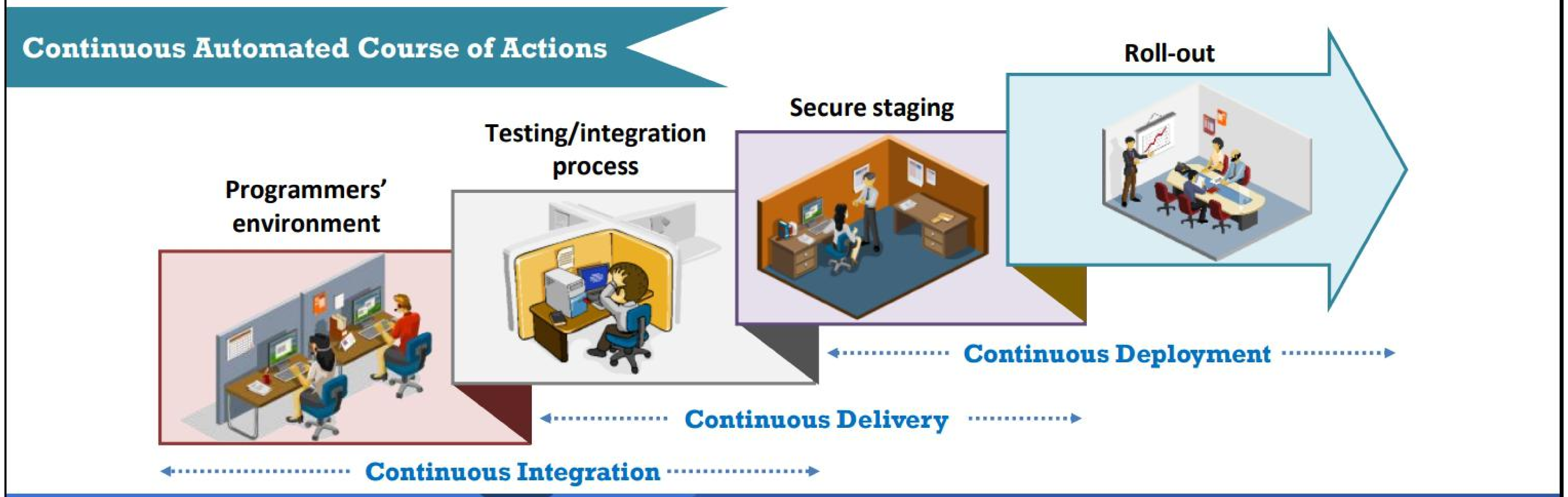
A secure baseline defines how the application should operate in a real-time environment. The baseline should be established with a standard setup. Firewall and operating-system installations, as well as their patches, need to be configured appropriately to secure the application. When patches are released, the baseline should be changed in accordance with the changes or developments to the operating system.

Integrity Measurement

After the successful creation of a secure baseline, testers examine the application against the new baseline provided. They ensure that all the application components maintain this baseline to ensure the integrity of the application.

Resiliency and Automation Strategies

- Systems that can be restored to their usual operating conditions after facing an interruption or outage are known and **resilient systems**
- The resiliency of systems can be improved by **adopting a proper configuration** or using methods such as snapshots, strengthening the ability to return to the original state, and appropriate fault tolerance and redundancy methods
- Automation can also be used to make systems, applications, or networks resilient through **continuous automated operations**



Resiliency and Automation Strategies (Cont'd)

Continuous Integration	It is a practice followed in the software development process in which programmers frequently **integrate**, **develop**, and **inspect their tasks**, typically through automation.
Continuous Delivery	It is a continuous operation in which the integration process ends and the software development teams are allowed to **automate the development** and **testing process**
Continuous Deployment	It simplifies the integration process further through automated testing that **validates changes to the code**
Continuous Monitoring	The continuous monitoring process ensures that all **components are working as intended** and generates alerts if any failure occurs
Continuous Validation	It is used in an application or service environment to **validate application compliance** and check whether the application meets the defined goals

Resiliency and Automation Strategies

Systems that can be restored to their usual operating conditions after an interruption or outage are known as resilient systems. The risks associated with a system are reduced when it is resilient to attacks. The resiliency of systems can be improved by adopting proper configuration or setting methods such as snapshots, strengthening the ability to return to the original state, and also by implementing fault tolerance and redundancy methods. Automation can also be used to make systems, applications, or networks resilient through continuous operations that boost efficiency and ensure accuracy while deploying software and running scripts or commands.

The following are different areas where automation is involved.

- **Continuous integration**: It is a practice followed in software development, wherein programmers frequently integrate, develop, and inspect their tasks, typically through automation. Each integration process is validated by an automated build that runs all the automated tasks, which could discover integration errors or bugs as early as possible.
- **Continuous delivery**: It is a continuous operation performed after the completion of the integration process, and it allows the software development teams to automate the development and testing processes in the software development lifecycle. The secure staging process is performed in this phase. It offers many advantages while provisioning an integrated toolset. The major advantages include the following:
  - Minimization of the deployment time by continuous development and testing
  - Reduction in the costs associated with conventional software development
  - Extension of software development depending on the project size
  - Automatic deployment of code into different phases of SDLC
- **Continuous deployment:** It simplifies the integration process further through automated testing, which validates changes to the code and could result in the faster delivery of updates or new versions of the software.
- **Automated course of actions**: Automation is often implemented by running appropriate scripts in a process known as an automated course of action. It has the following benefits over manual implementation.
  - If they are inspected, predefined scripts can potentially minimize the possibility of user errors.
  - Scripts can be bounded together to automate various commands.
  - They consume less time because they run at the speed of the system and are not dependent on the speed of manual inputs.
  - They can be used to continuously monitor and instantly identify incidents and outages.
- **Continuous monitoring**: It refers to a system that has a default monitoring feature, rather than a monitoring feature implemented externally. The continuous monitoring process ensures that all components are working as expected and generates alerts if any failure occurs. It is tied up with automation because the responses and dashboards are employed in conjunction with monitoring.
- **Continuous validation**: This process can also use automation techniques. In this process, when the system is turned on, the configuration files are automatically validated

according to the specified standards. This process is also used in application or service environments to validate application compliance and check whether the application meets the defined goals. Automation can make the validation mechanism easy to execute and less prone to errors.

The continuous strategy fetches reports from the delivery and deployment stages to monitor and analyze the software to meet the security standards. It also ensures that there is no deviation from the defined baselines.

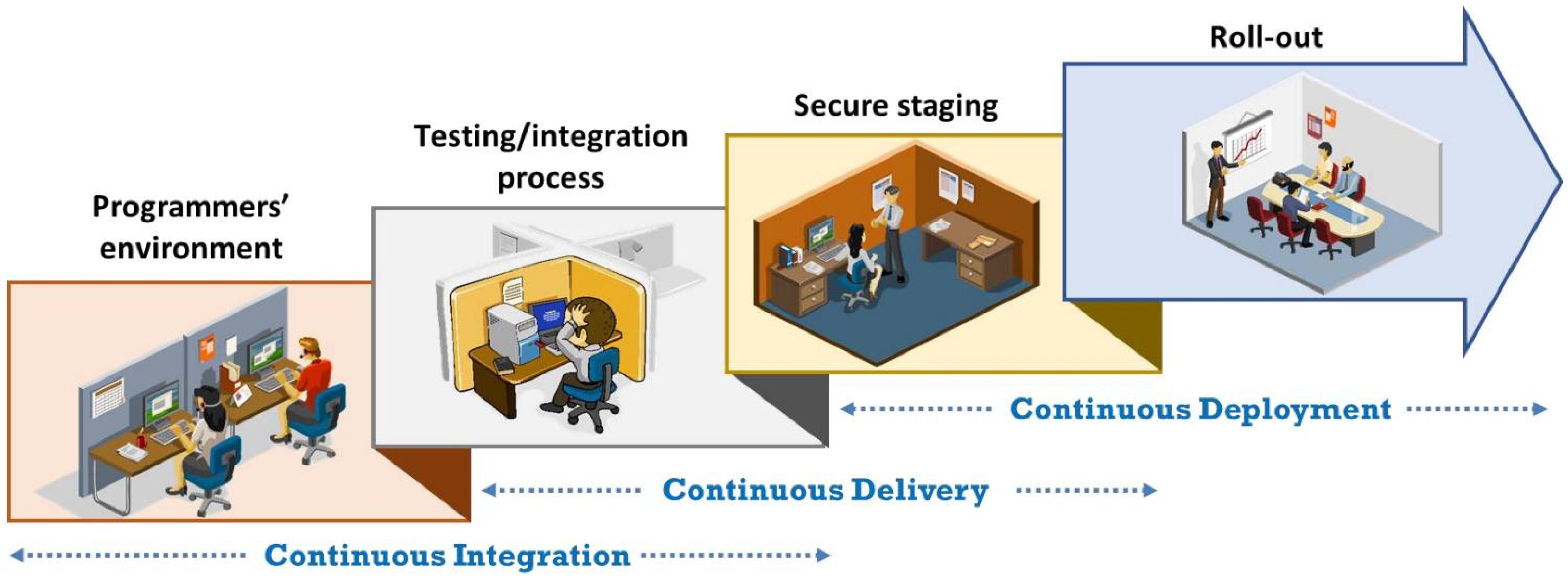
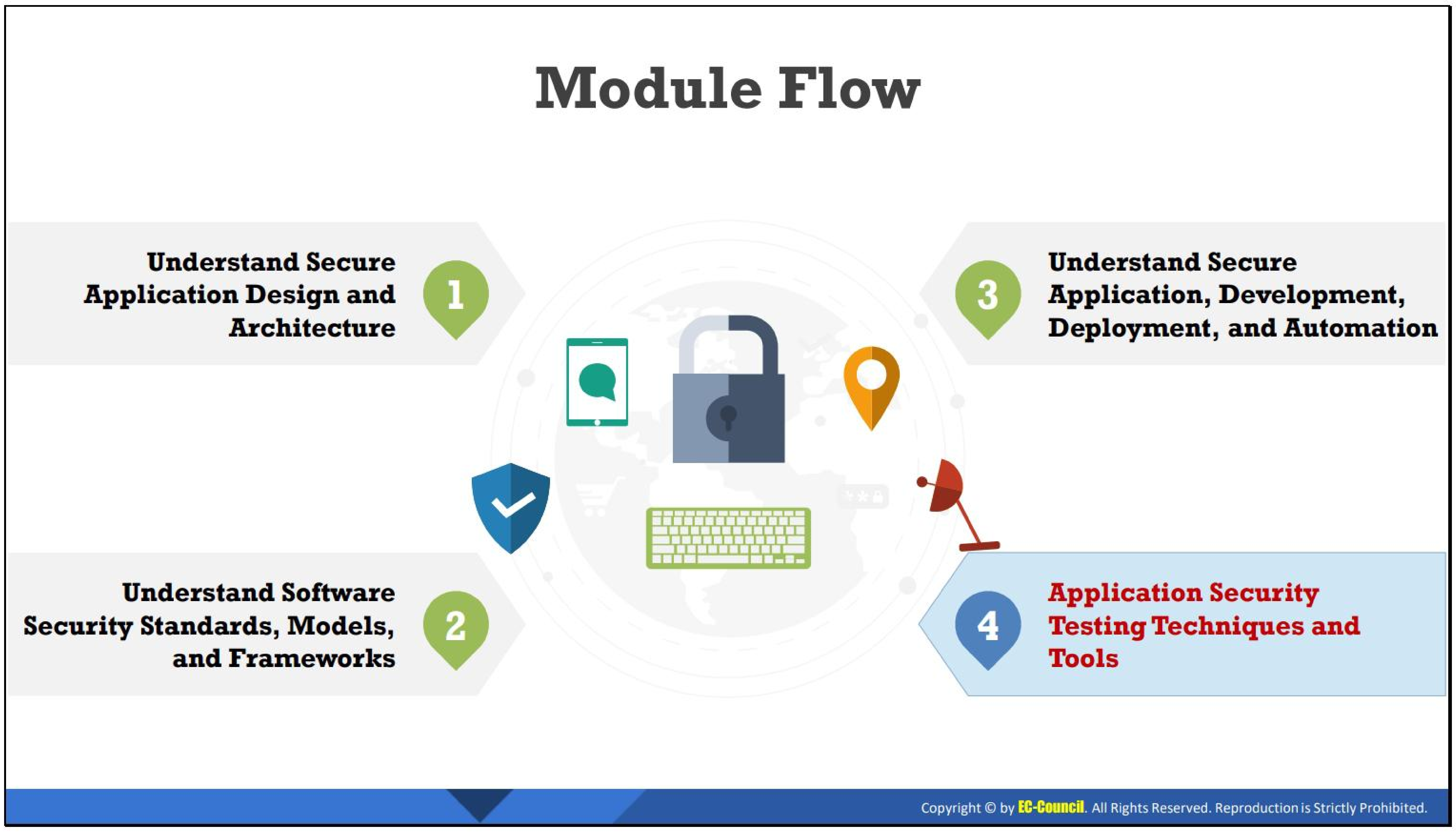


Figure 9.10: Resiliency and automation strategies

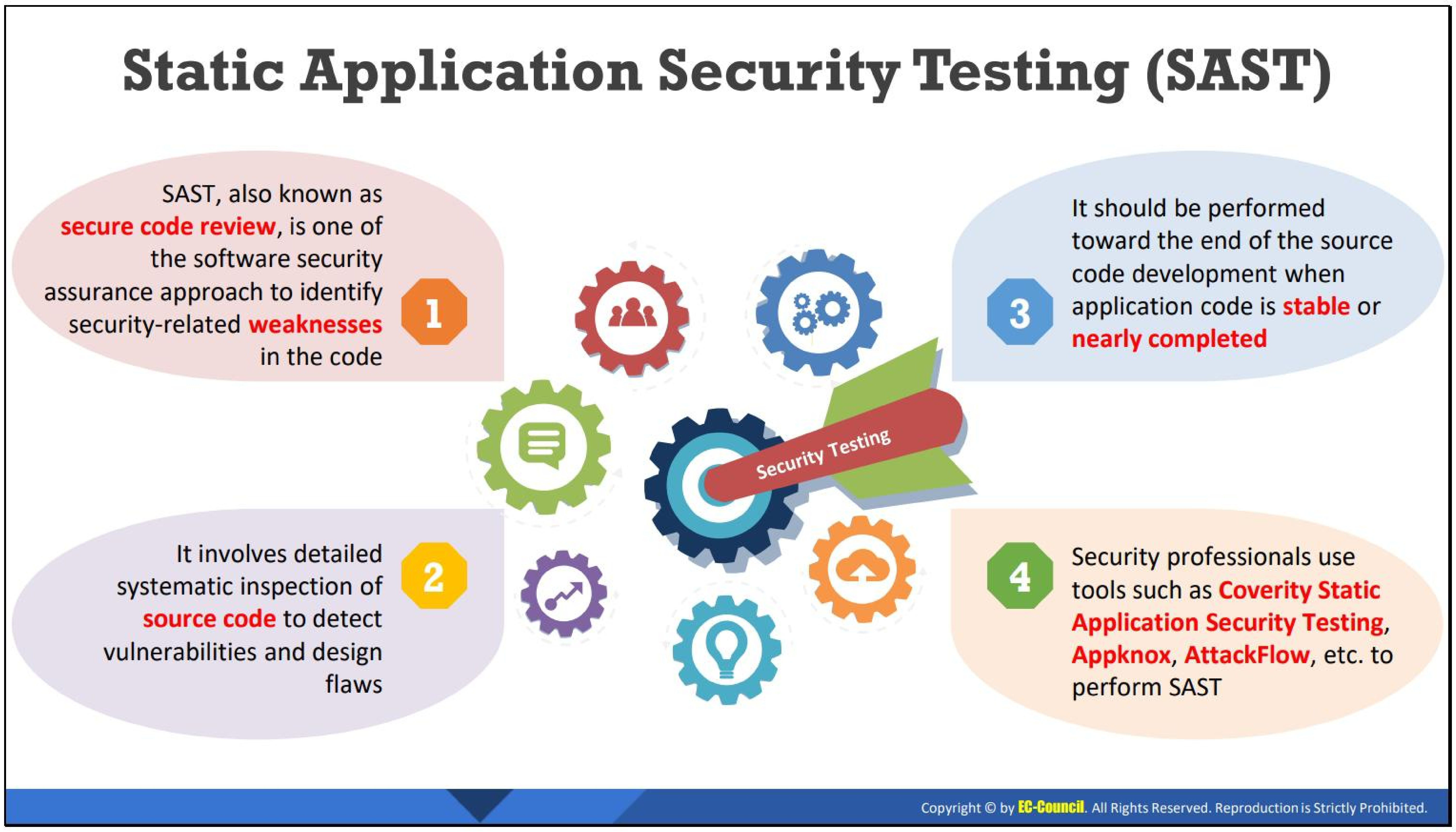


Module Flow

1. Understand Secure Application Design and Architecture
2. Understand Software Security Standards, Models, and Frameworks
3. Understand Secure Application, Development, Deployment, and Automation
4. Application Security Testing Techniques and Tools

Application Security Testing Techniques and Tools

The primary objective of application security testing is to detect flaws or vulnerabilities associated with source code. Application security testing techniques and tools assist developers in detecting all the potential security weaknesses and in making changes to the code to fix the weaknesses. This section discusses various application security testing techniques and tools.



Static Application Security Testing (SAST)

1 SAST, also known as **secure code review**, is one of the software security assurance approach to identify security-related **weaknesses** in the code

2 It involves detailed systematic inspection of **source code** to detect vulnerabilities and design flaws

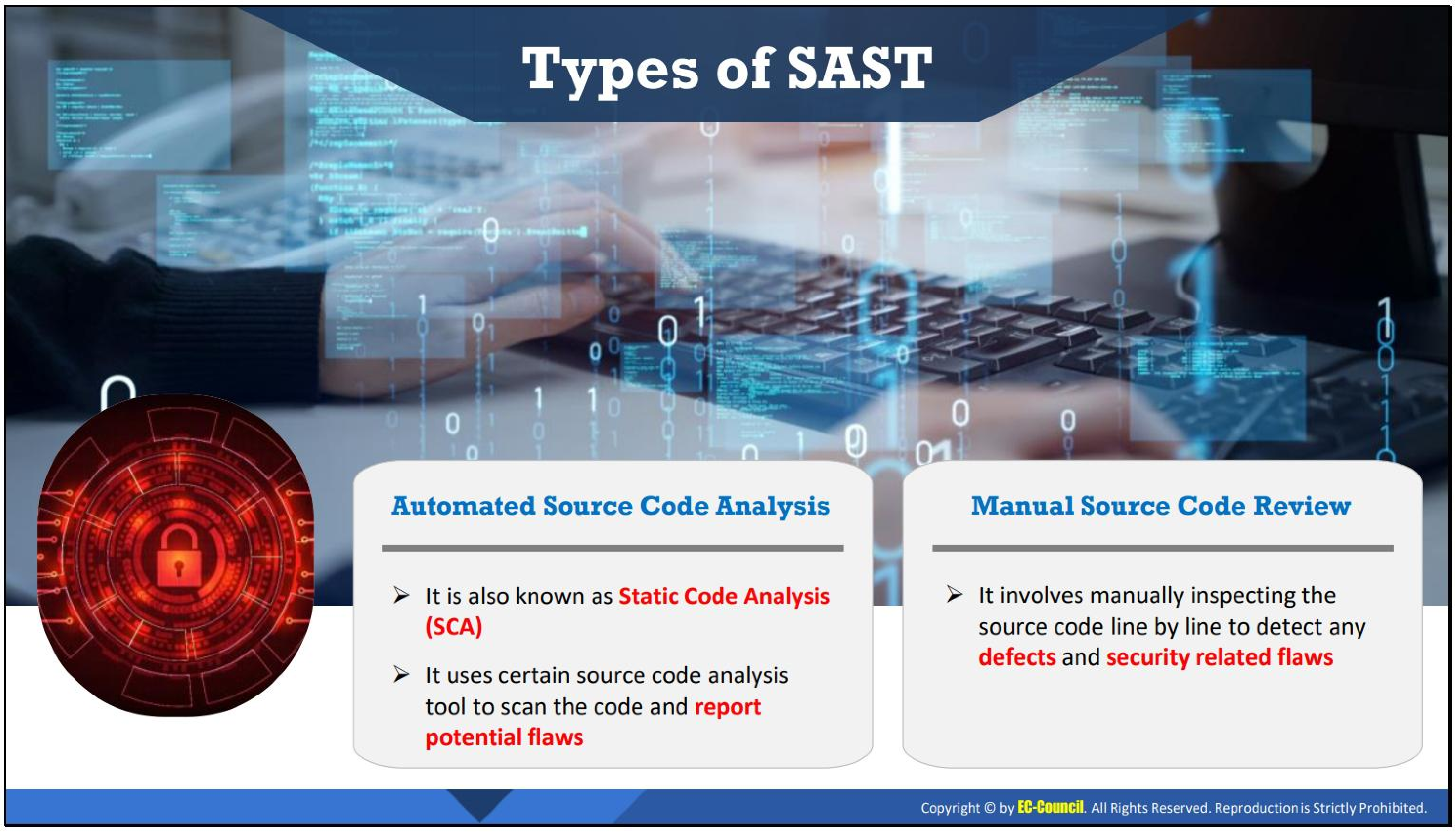
3 It should be performed toward the end of the source code development when application code is **stable** or **nearly completed**

4 Security professionals use tools such as **Coverity Static Application Security Testing**, **Appknox**, **AttackFlow**, etc. to perform SAST

Static Application Security Testing (SAST)

SAST, also known as secure code review, is one of the software security assurance approach to identify security-related weaknesses in the code. It involves detailed systematic inspection of source code to detect vulnerabilities and design flaws. It should be performed toward the end of the source code development when application code is stable or nearly completed. It should always be performed in combination with human effort (Manual) and technology support (Automated). SAST aims to detect application security vulnerabilities and their root causes when code is not running.

SAST tools assist developers in testing the source code to discover and report design flaws associated with the application, which can open doors for various attacks. It also ensures that the source code is compliant with defined rules, standards, and guidelines. Security professionals use tools such as Coverity Static Application Security Testing, Appknox, AttackFlow, bugScout, and PT Application Inspector, to perform SAST.



Types of SAST

Automated Source Code Analysis

- It is also known as **Static Code Analysis (SCA)**
- It uses certain source code analysis tool to scan the code and **report potential flaws**

Manual Source Code Review

- It involves manually inspecting the source code line by line to detect any **defects** and **security related flaws**

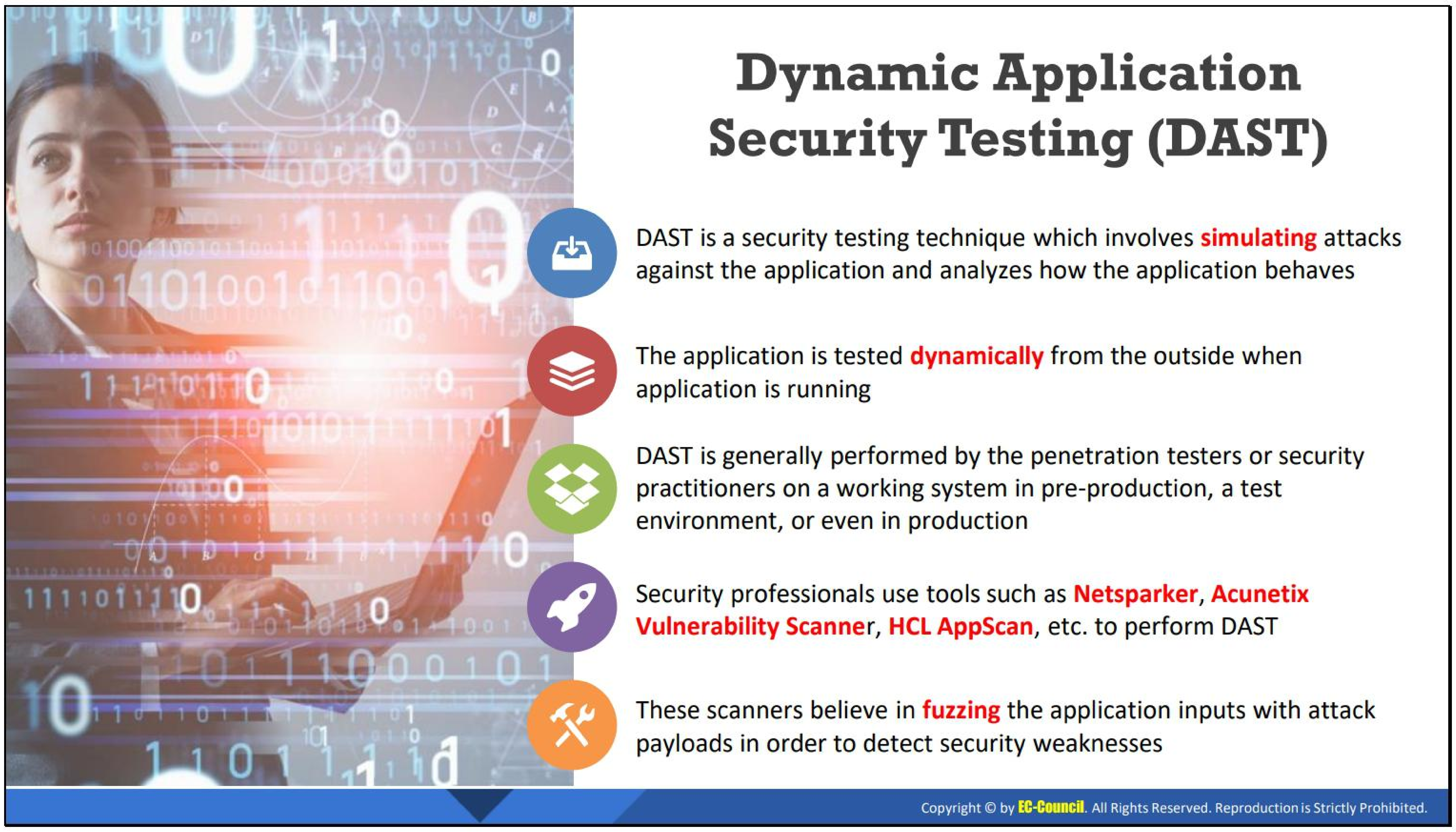
Types of SAST

- **Automated source code analysis**

  It is also known as Static Code Analysis (SCA). It uses certain source code analysis tool to scan the code and report potential flaws.

- **Manual source code review**

  It involves manually inspecting the source code line by line to detect any defects and security related flaws.



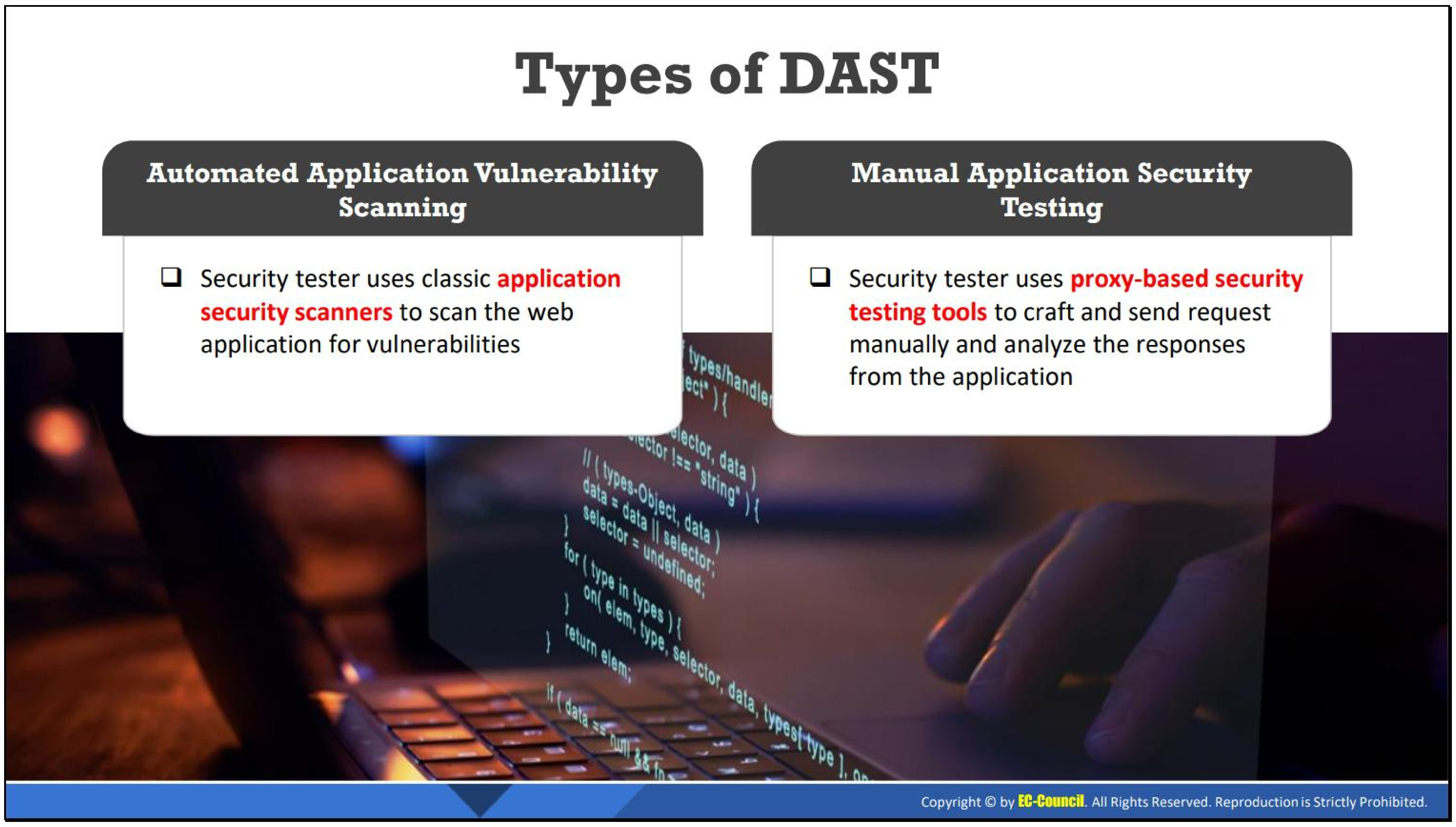
Dynamic Application Security Testing (DAST)

- DAST is a security testing technique which involves **simulating** attacks against the application and analyzes how the application behaves
- The application is tested **dynamically** from the outside when application is running
- DAST is generally performed by the penetration testers or security practitioners on a working system in pre-production, a test environment, or even in production
- Security professionals use tools such as **Netsparker**, **Acunetix Vulnerability Scanne**r, **HCL AppScan**, etc. to perform DAST
- These scanners believe in **fuzzing** the application inputs with attack payloads in order to detect security weaknesses

Dynamic Application Security Testing (DAST)

DAST is a security testing technique which involves simulating attacks against the application and analyzes how the application behaves. The application is tested dynamically from the outside when application is running. DAST is generally performed by the penetration testers or security practitioners on a working system in pre-production, a test environment, or even in production. These individuals typically uses automated web application vulnerability scanners to conduct DAST. These scanners believe in fuzzing the application inputs with attack payloads in order to detect security weaknesses.

DAST tools execute on running code to identify issues related to interfaces, requests/responses, sessions, scripts, authentication processes, code injections, etc. Security professionals use tools such as Netsparker, Acunetix Vulnerability Scanner, HCL AppScan, Micro Focus Fortify on Demand, and Appknox, to perform DAST.



Types of DAST

Automated Application Vulnerability Scanning

- Security tester uses classic **application security scanners** to scan the web application for vulnerabilities

Manual Application Security Testing

- Security tester uses **proxy-based security testing tools** to craft and send request manually and analyze the responses from the application

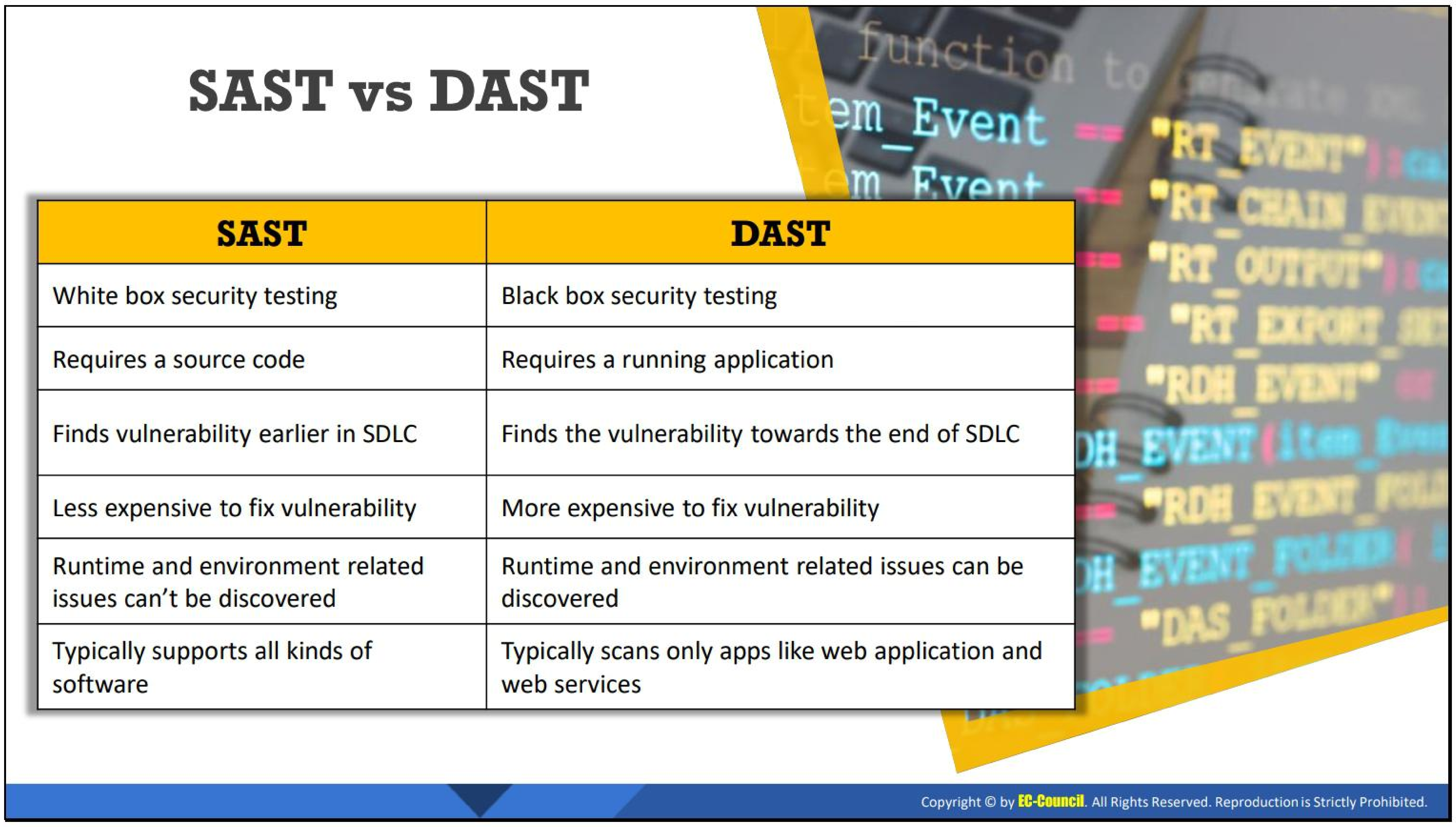
Types of DAST

- **Automated Application Vulnerability Scanning**

  Security tester uses classic application security scanners to scan the web application for vulnerabilities.

- **Manual Application Security Testing**

  Security tester uses proxy-based security testing tools to craft and send request manually and analyze the responses from the application.



SAST vs DAST

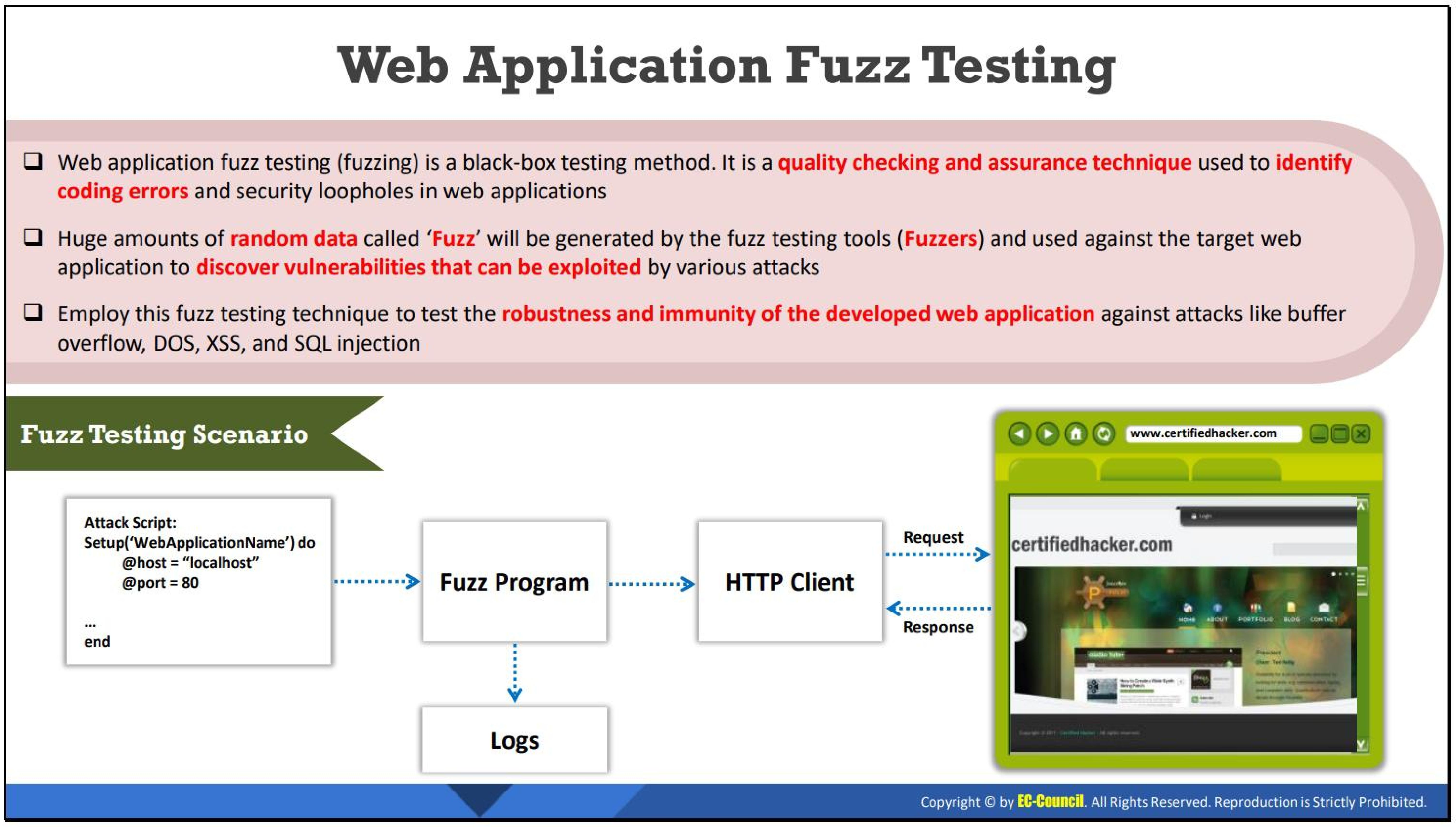
SAST	DAST
White box security testing	Black box security testing
Requires a source code	Requires a running application
Finds vulnerability earlier in SDLC	Finds the vulnerability towards the end of SDLC
Less expensive to fix vulnerability	More expensive to fix vulnerability
Runtime and environment related issues can't be discovered	Runtime and environment related issues can be discovered
Typically supports all kinds of software	Typically scans only apps like web application and web services

SAST vs DAST

The table given below summarizes the differences between SAST and DAST:

SAST	DAST
White box security testing	Black box security testing
Requires a source code	Requires a running application
Finds vulnerability earlier in SDLC	Finds the vulnerability towards the end of SDLC
Less expensive to fix vulnerability	More expensive to fix vulnerability
Runtime and environment related issues can't be discovered	Runtime and environment related issues can be discovered
Typically supports all kinds of software	Typically scans only apps like web application and web services

Table 9.2: SAST vs DAST



Web Application Fuzz Testing

- Web application fuzz testing (fuzzing) is a black-box testing method. It is a **quality checking and assurance technique** used to **identify coding errors** and security loopholes in web applications
- Huge amounts of **random data** called '**Fuzz**' will be generated by the fuzz testing tools (**Fuzzers**) and used against the target web application to **discover vulnerabilities that can be exploited** by various attacks
- Employ this fuzz testing technique to test the **robustness and immunity of the developed web application** against attacks like buffer overflow, DOS, XSS, and SQL injection

Fuzz Testing Scenario

```
Attack Script:
Setup('WebApplicationName') do
        @host = "localhost"
        @port = 80

...
end
```

Web Application Fuzz Testing

Web application fuzz testing (fuzzing) is a black box testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications. Massive amounts of random data called "fuzz" are generated by fuzz testing tools (fuzzers) and used against the target web application to discover vulnerabilities that can be exploited by various attacks. Attackers employ various attack techniques to crash the victim's web applications and cause havoc in the least possible time. Security personnel and web developers employ this fuzz testing technique to test the robustness and immunity of the developed web application against attacks such as buffer overflow, DOS, XSS, and SQL injection.

Steps of Fuzz Testing

Web application fuzz testing involves the following steps:

- Identify the target system
- Identify inputs
- Generate fuzzed data
- Execute the test using fuzz data
- Monitor system behavior
- Log defects

Fuzz Testing Scenario

The diagram below shows an overview of the main components of the fuzzer. An attacker script is fed to the fuzzer, which in turn translates the attacks to the target as http requests. These http requests will get responses from the target and all the requests and their responses are then logged for manual inspection.

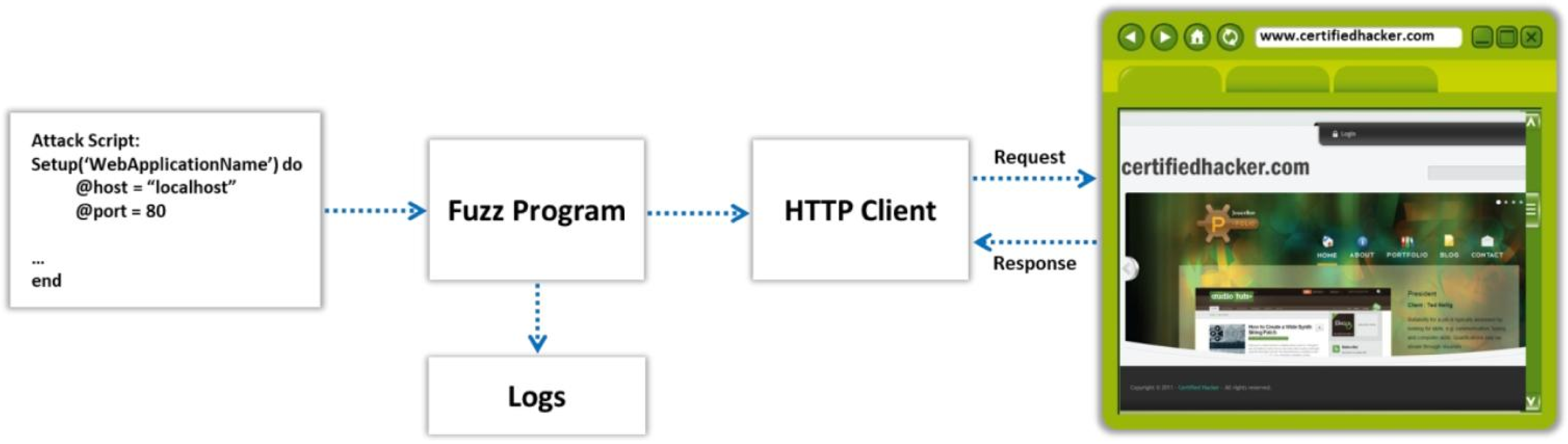
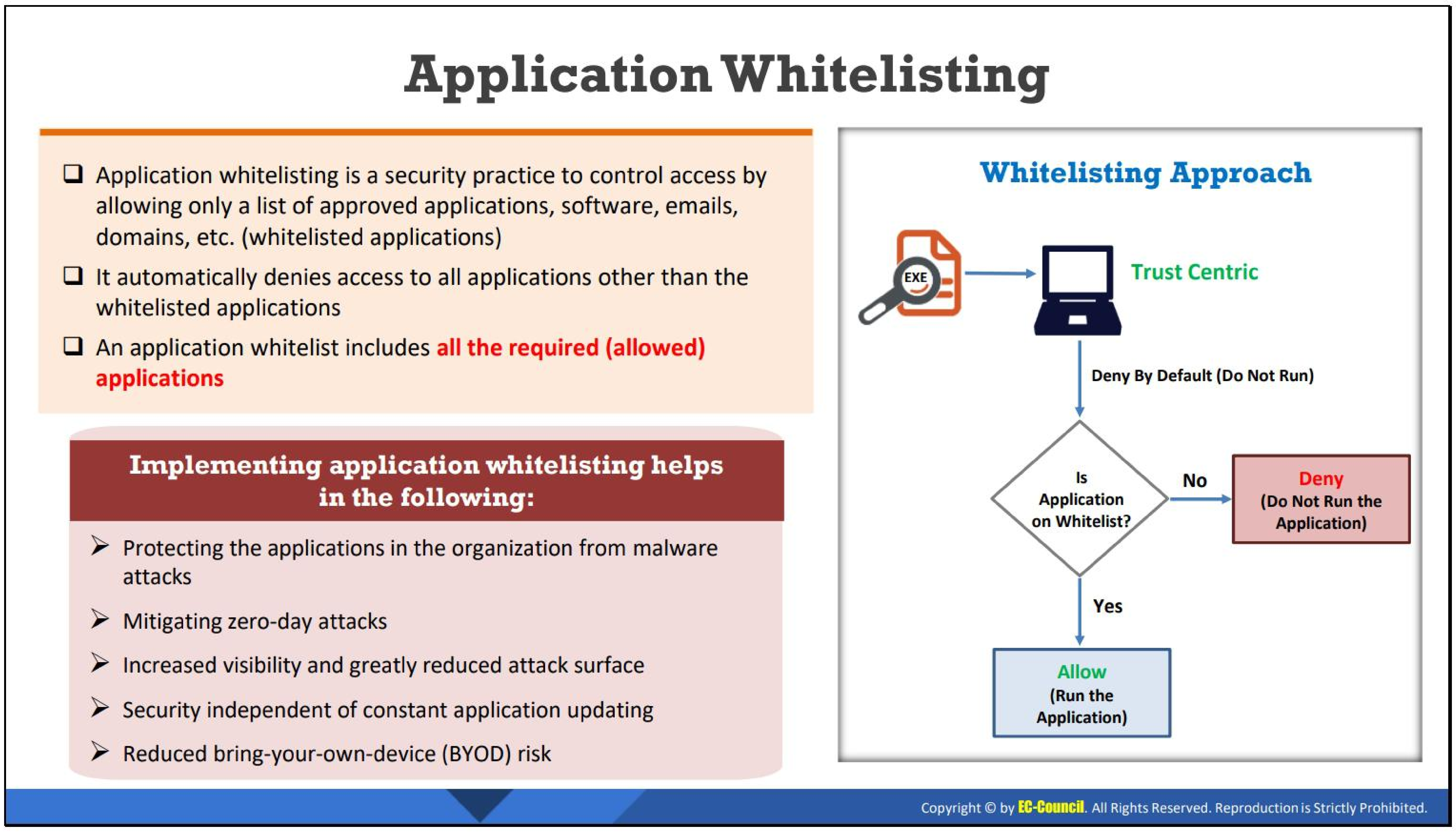


Figure 9.11: Web application fuzz testing scenario



Application Whitelisting

- ❑ Application whitelisting is a security practice to control access by allowing only a list of approved applications, software, emails, domains, etc. (whitelisted applications)
- ❑ It automatically denies access to all applications other than the whitelisted applications
- ❑ An application whitelist includes **all the required (allowed) applications**

Implementing application whitelisting helps in the following:

- ➢ Protecting the applications in the organization from malware attacks
- ➢ Mitigating zero-day attacks
- ➢ Increased visibility and greatly reduced attack surface
- ➢ Security independent of constant application updating
- ➢ Reduced bring-your-own-device (BYOD) risk

Whitelisting Approach

EXE
Trust Centric
Deny By Default (Do Not Run)
Is Application on Whitelist?
No
Deny (Do Not Run the Application)
Yes
Allow (Run the Application)

Application Whitelisting

Application whitelisting is a form of access control that allows only specific programs to run. Unless a program is whitelisted, it is blocked on a host. Application whitelisting technologies are also called application control programs or whitelisting programs.

The approach of application whitelisting is trust centric. By default, applications that are not in the whitelist are prevented from being executed. To allow the execution of any program or application, the security professional must add it in the application whitelist.

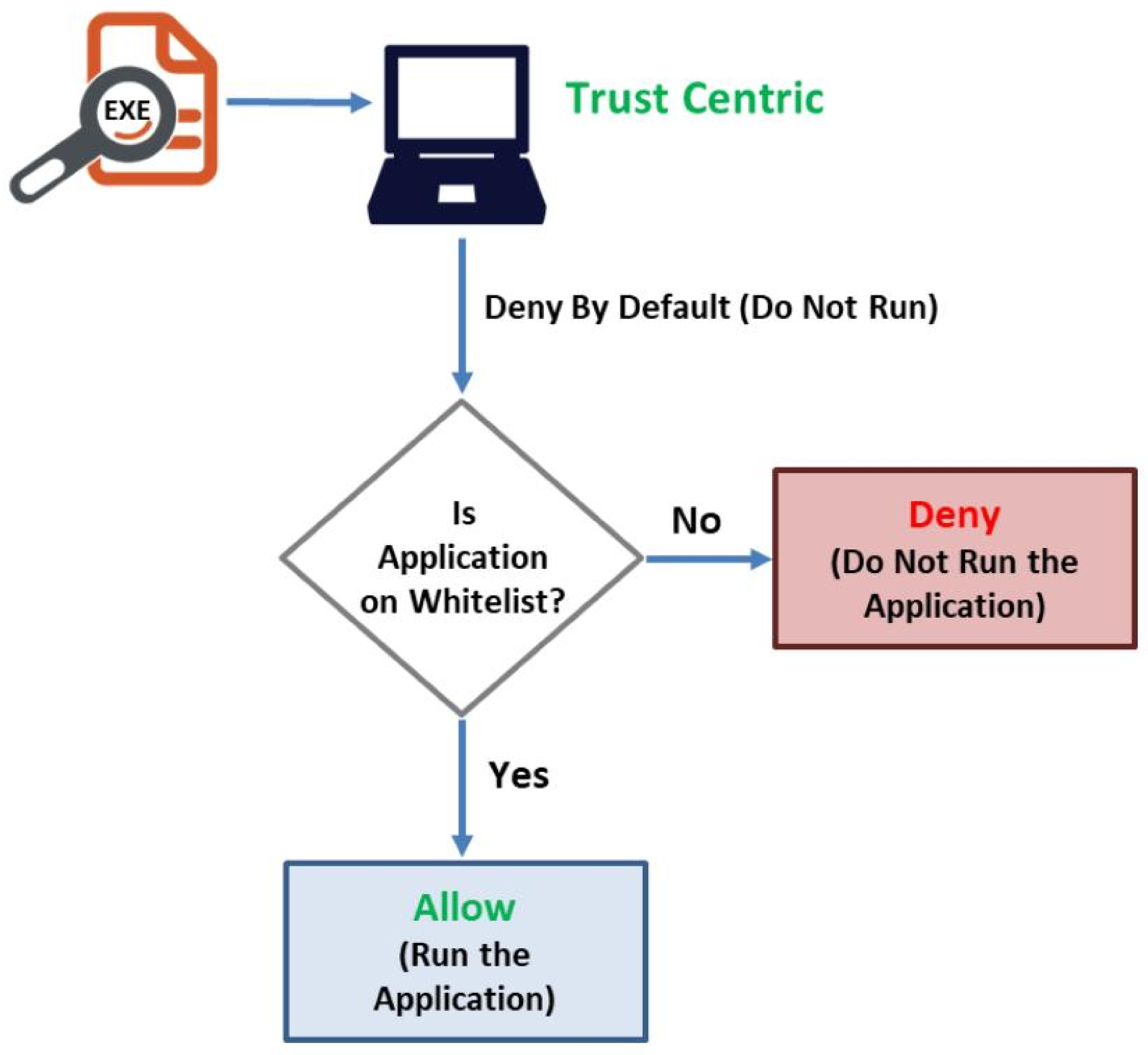

EXE
Trust Centric
Deny By Default (Do Not Run)
Is Application on Whitelist?
No
Deny (Do Not Run the Application)
Yes
Allow (Run the Application)

Figure 9.12: Whitelisting approach

Any runtime process, host, application and application components (plug-ins, configuration files, software libraries, and extensions), email addresses, port numbers, etc. can be used to create a whitelist.

Advantages of Whitelisting

Implementing application whitelisting ensures the confidentiality, integrity, and availability of data. Application whitelisting provides security professionals and organizations the following benefits.

- **Protection against malware attacks**

  The whitelisting of applications in an organization can prevent malware attacks. Any application that is not in the whitelist is blocked.

- **Mitigating zero-day attacks**

  Generally, attackers start exploiting vulnerabilities once a software patch is released. Occasionally, malware for unpatched systems is ready to be deployed in a short time window during which a new patch has not yet been tested or implemented. Antivirus vendors also take time to identify new signatures to produce and distribute. Implementing application whitelisting hinders the execution of such vulnerabilities.

- **Improved efficiency of computers**

  Application whitelisting prevents unauthorized applications from running in organizations, improving the efficiency of computers.

- **Increased visibility and greatly reduced attack surface**

  Application whitelisting removes many basic attacks by protecting against the attack vector of download and execute. Application whitelisting enables organizations to track which applications are running or blocked on company systems. Improving the capability of monitoring and controlling applications greatly reduces the attack surface area, unauthorized changes to applications, and inspection requirements.

- **Reclaiming bandwidth from streaming or sharing applications**

  Application whitelisting avoids the significant use of resources to operate unapproved and unnecessary applications, ensuring the optimal utilization of company resources in organizations. Application whitelisting limits the exposure of social media applications, bans certain websites, eliminates games, and blocks other destructive applications that consume excessive employee time and network bandwidth.

- **Avoiding organizations from facing lawsuits or paying unnecessary license fees**

  Application whitelisting helps organizations avoid troubles such as lawsuits or license fees for unknowingly using unlicensed or illegal applications.

- **Security independent of constant application updating**

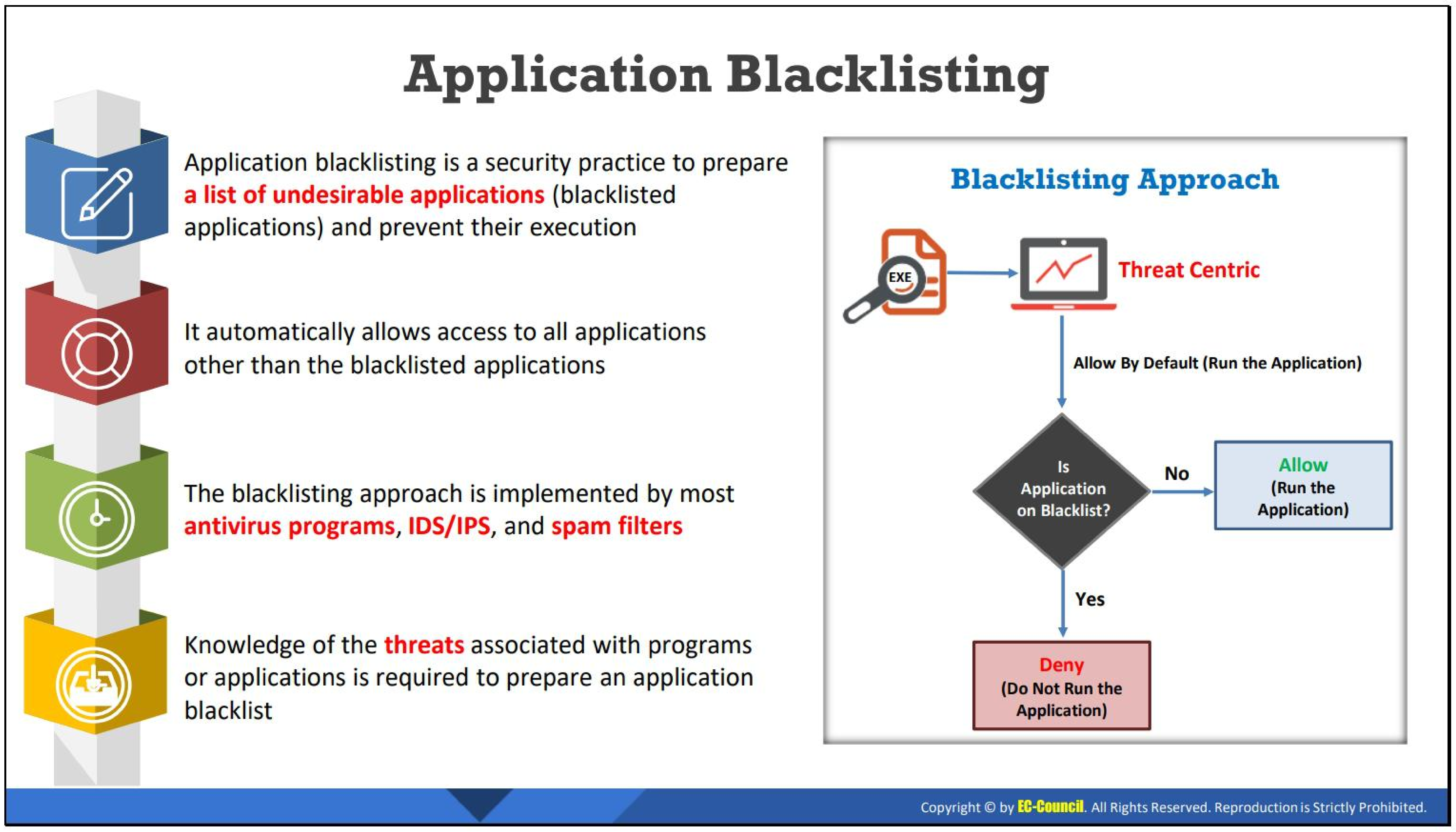
  Unlike antivirus programs, application whitelisting solutions do not need to get updated periodically to be active.

- **Easier attack detection**

  Attack detection becomes easier when many attack activities are blocked, and attacks generate a lot of noise. The noise created by attackers provide valuable information to incident response teams. This helps in measuring how long it takes for an antivirus solution to detect the existence of malware or changes on a system.

- **Reduced bring-your-own-device (BYOD) risk**

  Application whitelisting reduces BYOD risk through the enforcement of mobile-application policies.



Application Blacklisting

- Application blacklisting is a security practice to prepare **a list of undesirable applications** (blacklisted applications) and prevent their execution
- It automatically allows access to all applications other than the blacklisted applications
- The blacklisting approach is implemented by most **antivirus programs**, **IDS/IPS**, and **spam filters**
- Knowledge of the **threats** associated with programs or applications is required to prepare an application blacklist

Blacklisting Approach
EXE
Threat Centric
Allow By Default (Run the Application)
Is Application on Blacklist?
No
Allow (Run the Application)
Yes
Deny (Do Not Run the Application)

Application Blacklisting

Application blacklisting is a security practice of blocking the running and execution of a list of undesirable programs. Application blacklisting is threat centric. By default, it allows all applications that are not in the blacklist to be executed. To block any program or application, the security professional must add it in the application blacklist.

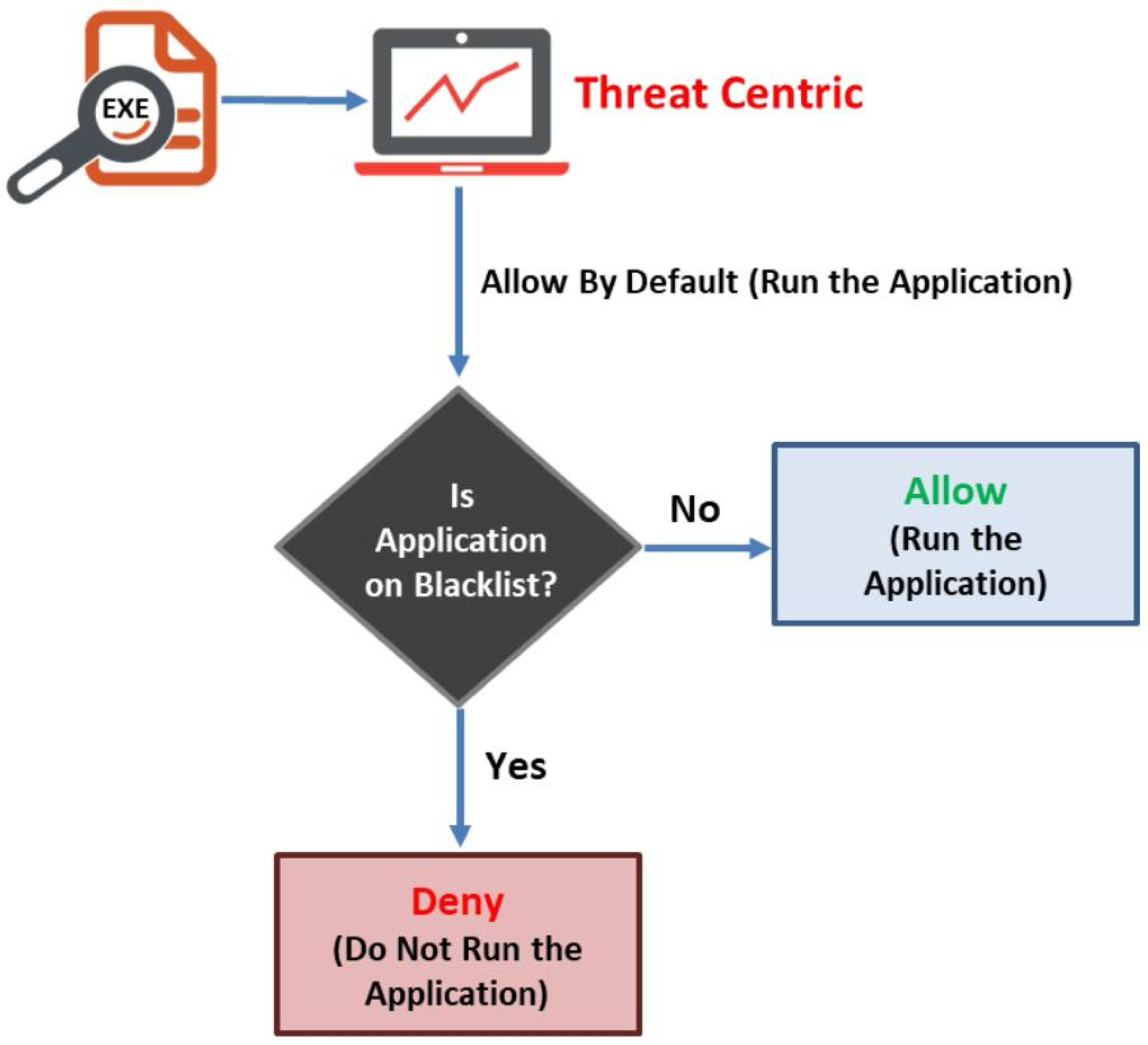

EXE
Threat Centric
Allow By Default (Run the Application)
Is Application on Blacklist?
No
Allow (Run the Application)
Yes
Deny (Do Not Run the Application)

Figure 9.13: Blacklisting approach

Most antivirus programs, spam filters and other intrusion prevention or detection systems use the application blacklisting method. A blacklist often comprises malware, users, IP addresses, applications, email addresses, domains, etc. Knowledge of the threats associated with programs or applications is required to prepare an application blacklist

Advantages of Application Blacklisting

Application blacklisting provides security professionals and organizations the following benefits.

- It is simple to implement. A blacklist simply identifies the blacklisted applications, denies them access, and allows the execution of all other applications not in the blacklist.
- Blacklists need low maintenance since the security software compiles lists and do not ask users for inputs often.

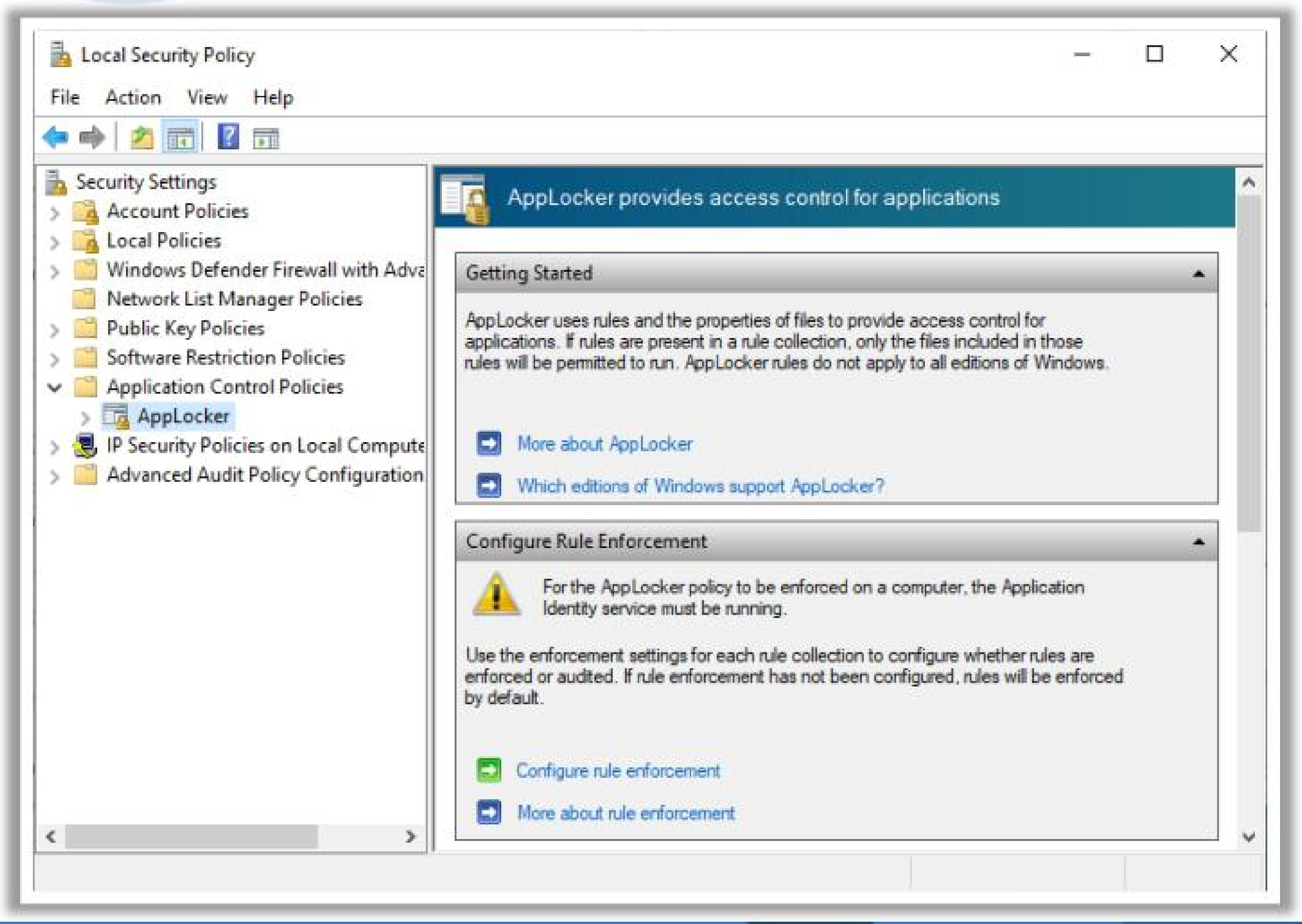
Disadvantages of Application Blacklisting

The following are some of the disadvantages of implementing application blacklisting.

- A blacklist cannot be comprehensive, and the effectiveness of a blacklist is limited as the number of different and complex threats is continuously increasing. Sharing threat information can help make application blacklisting more effective.
- Blacklisting can tackle known attacks well but will not be able to protect against zero-day attacks. If an organization is the first target of new threats, blacklisting cannot stop them.
- Occasionally, hackers create malware to evade detection using blacklisting tools. In these cases, blacklisting fails to recognize the malware and add it to the blacklist.

Using AppLocker for Application Whitelisting

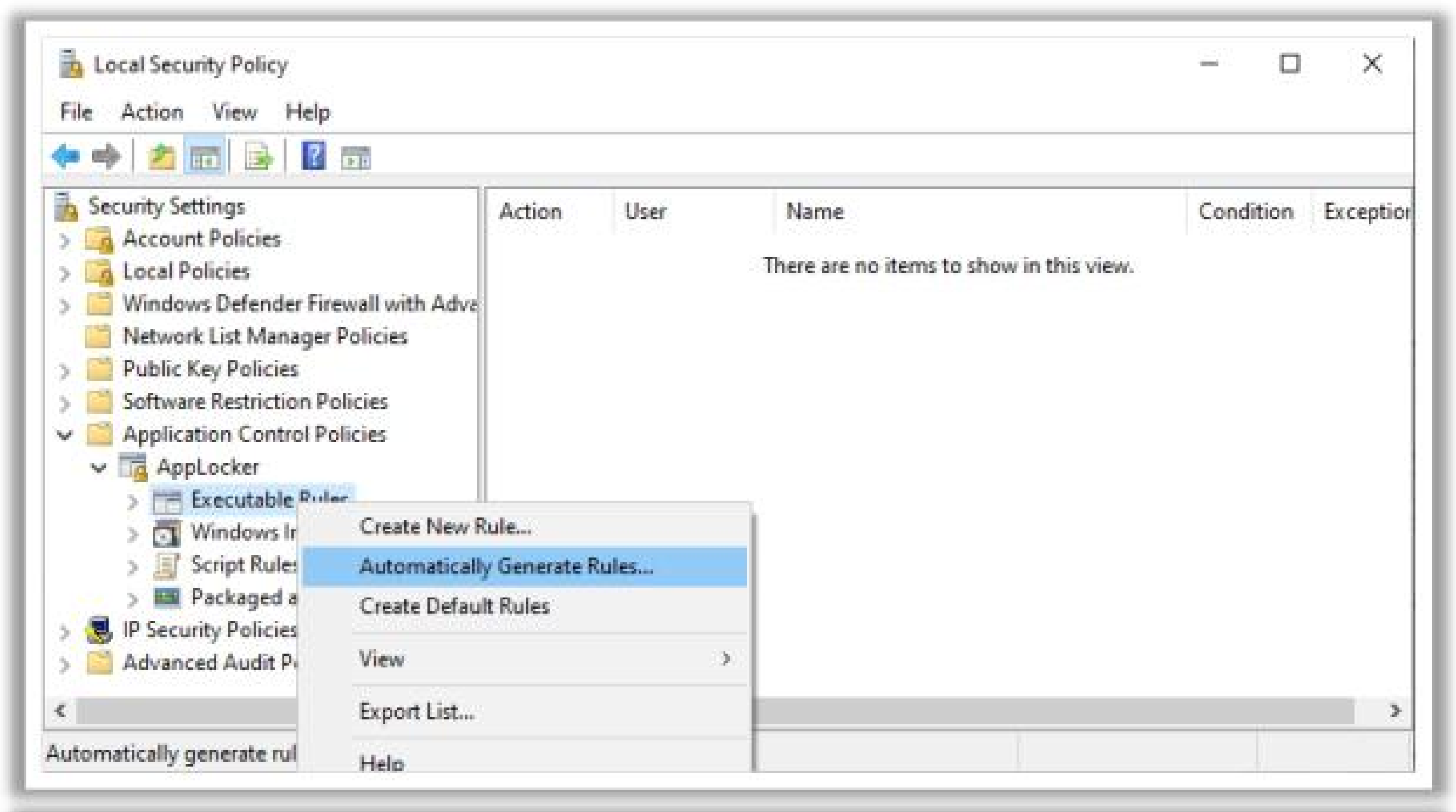
- AppLocker is a **Windows in-built security component** used to control applications (executables, scripts, Windows Installer files, and dynamic-link libraries (DLLs)) users can run
- The default executable rules are based on folder paths, and all files under those paths will be allowed
- Group Policy AppLocker can be used to set rules for applications in a domain


Local Security Policy
File Action View Help
Security Settings
Account Policies
Local Policies
Windows Defender Firewall with Adva
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
AppLocker
IP Security Policies on Local Compute
Advanced Audit Policy Configuration
AppLocker provides access control for applications
Getting Started
More about AppLocker
Which editions of Windows support AppLocker?
Configure Rule Enforcement
Configure rule enforcement
More about rule enforcement

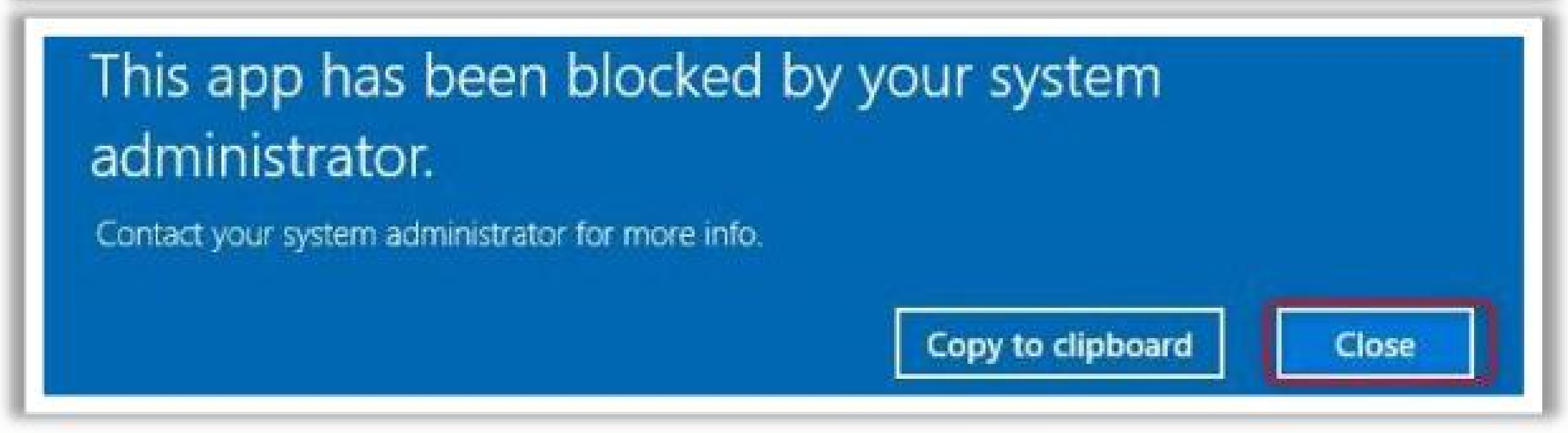
AppLocker uses rules and the properties of files to provide access control for applications. If rules are present in a rule collection, only the files included in those rules will be permitted to run. AppLocker rules do not apply to all editions of Windows.

For the AppLocker policy to be enforced on a computer, the Application Identity service must be running.

Use the enforcement settings for each rule collection to configure whether rules are enforced or audited. If rule enforcement has not been configured, rules will be enforced by default.



There are no items to show in this view.

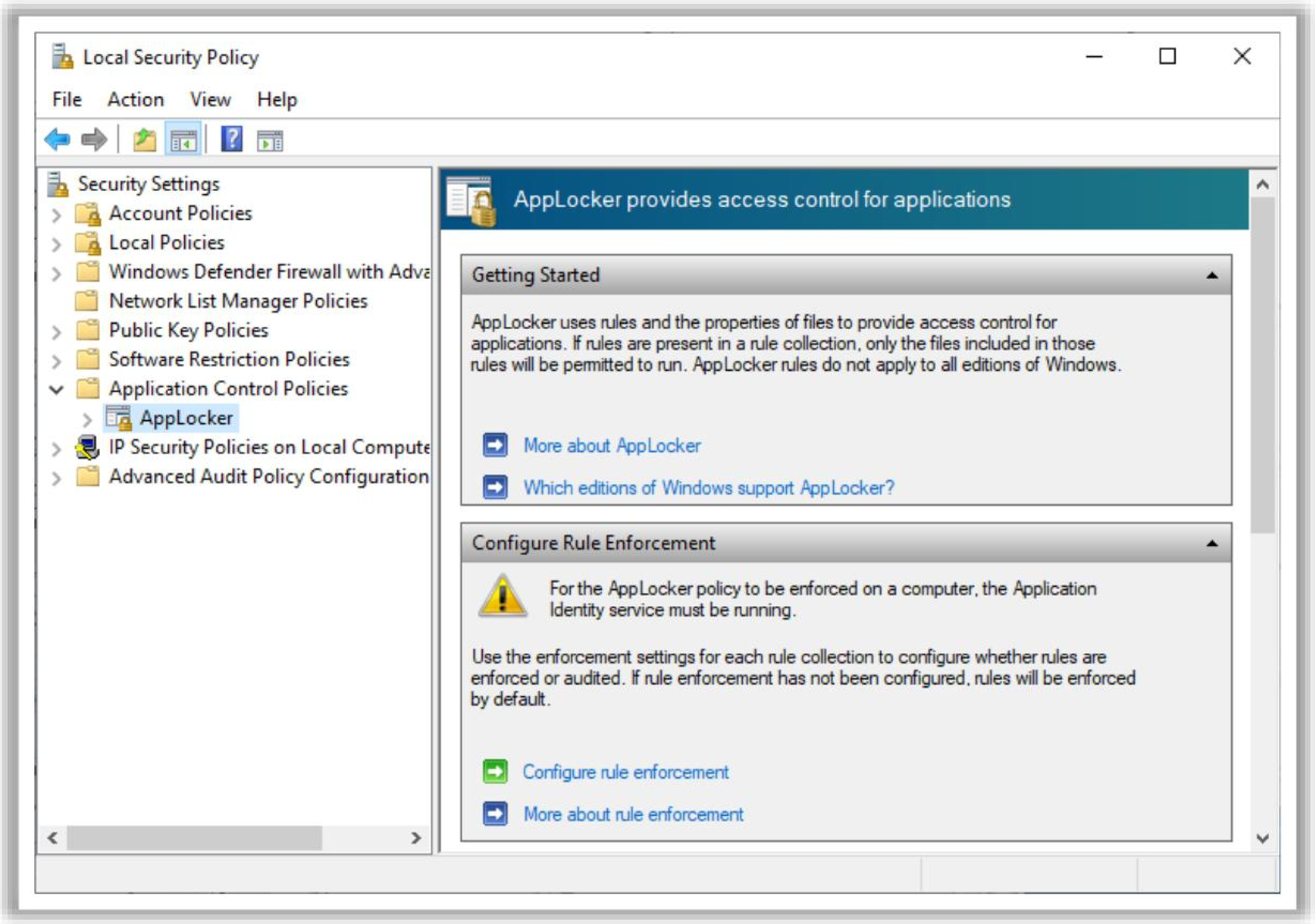


This app has been blocked by your system administrator.

Contact your system administrator for more info.

Using AppLocker for Application Whitelisting

AppLocker is an in-built Windows security component that can be used to control which applications users can run. When AppLocker rules are enforced, apps excluded from the list of allowed apps are prevented from running. The files include executables, Windows Installer files, and dynamic-link library (DLL) files. The default executable rules are based on paths, and all files under those paths are allowed. Group Policy AppLocker is used to set rules for applications in a domain.

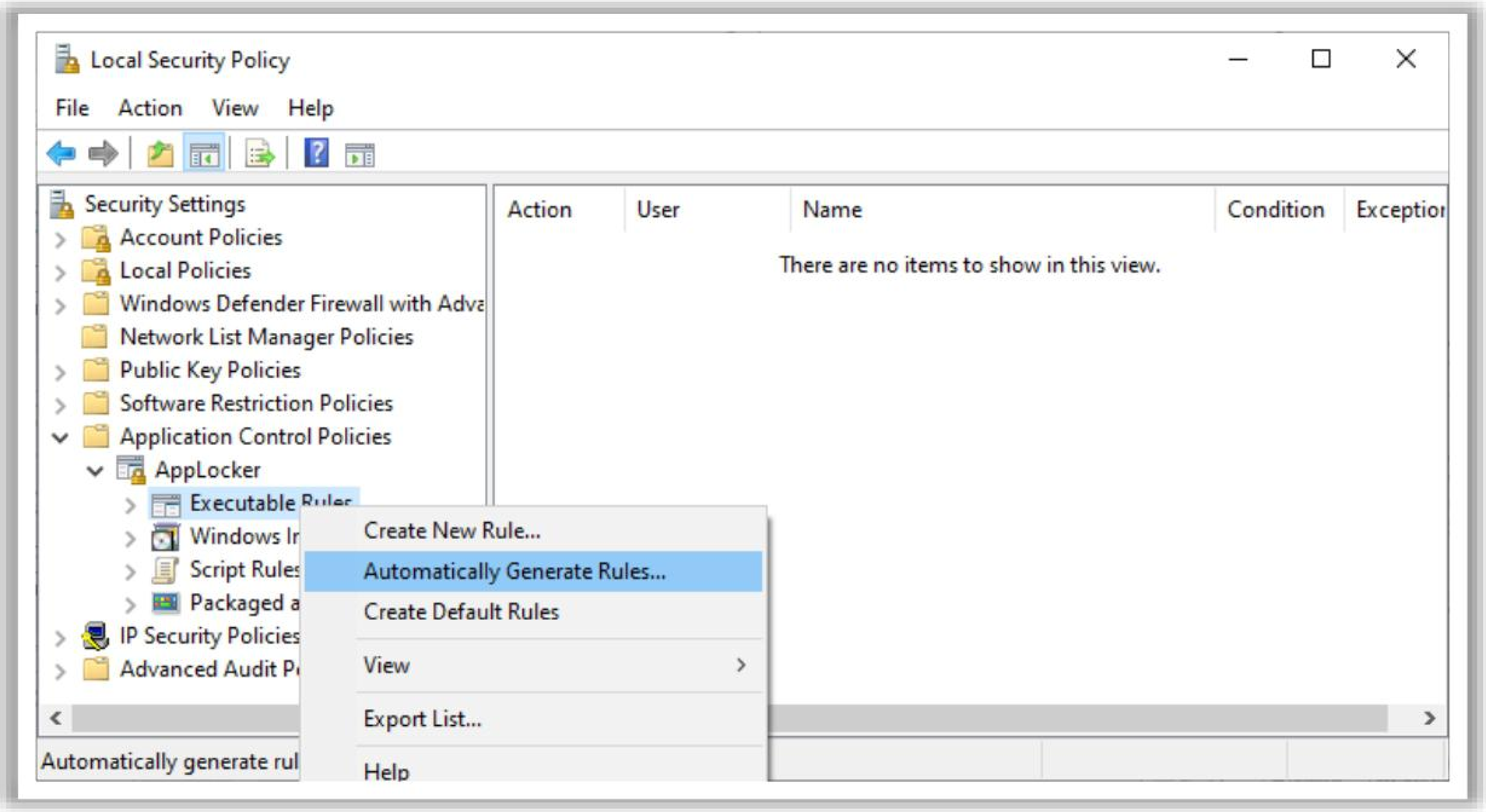

Local Security Policy
File Action View Help
Security Settings
Account Policies
Local Policies
Windows Defender Firewall with Adva
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
AppLocker
IP Security Policies on Local Compute
Advanced Audit Policy Configuration
AppLocker provides access control for applications
Getting Started
More about AppLocker
Which editions of Windows support AppLocker?
Configure Rule Enforcement
Configure rule enforcement
More about rule enforcement

AppLocker uses rules and the properties of files to provide access control for applications. If rules are present in a rule collection, only the files included in those rules will be permitted to run. AppLocker rules do not apply to all editions of Windows.

For the AppLocker policy to be enforced on a computer, the Application Identity service must be running.

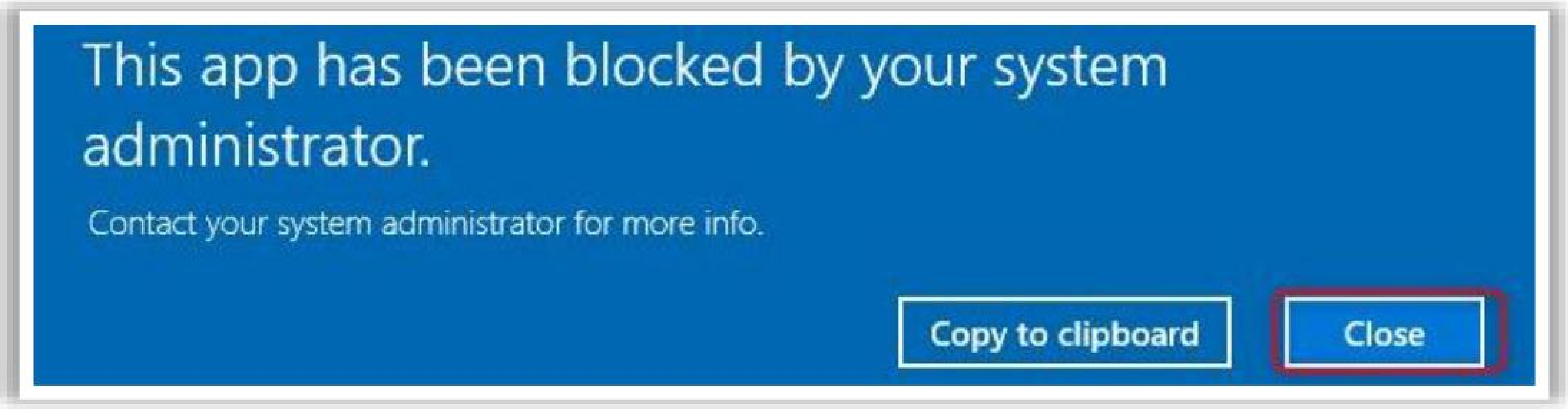
Use the enforcement settings for each rule collection to configure whether rules are enforced or audited. If rule enforcement has not been configured, rules will be enforced by default.

Figure 9.14: AppLocker for application whitelisting



There are no items to show in this view.

Figure 9.15: Generate an executable rule automatically



This app has been blocked by your system administrator.

Contact your system administrator for more info.

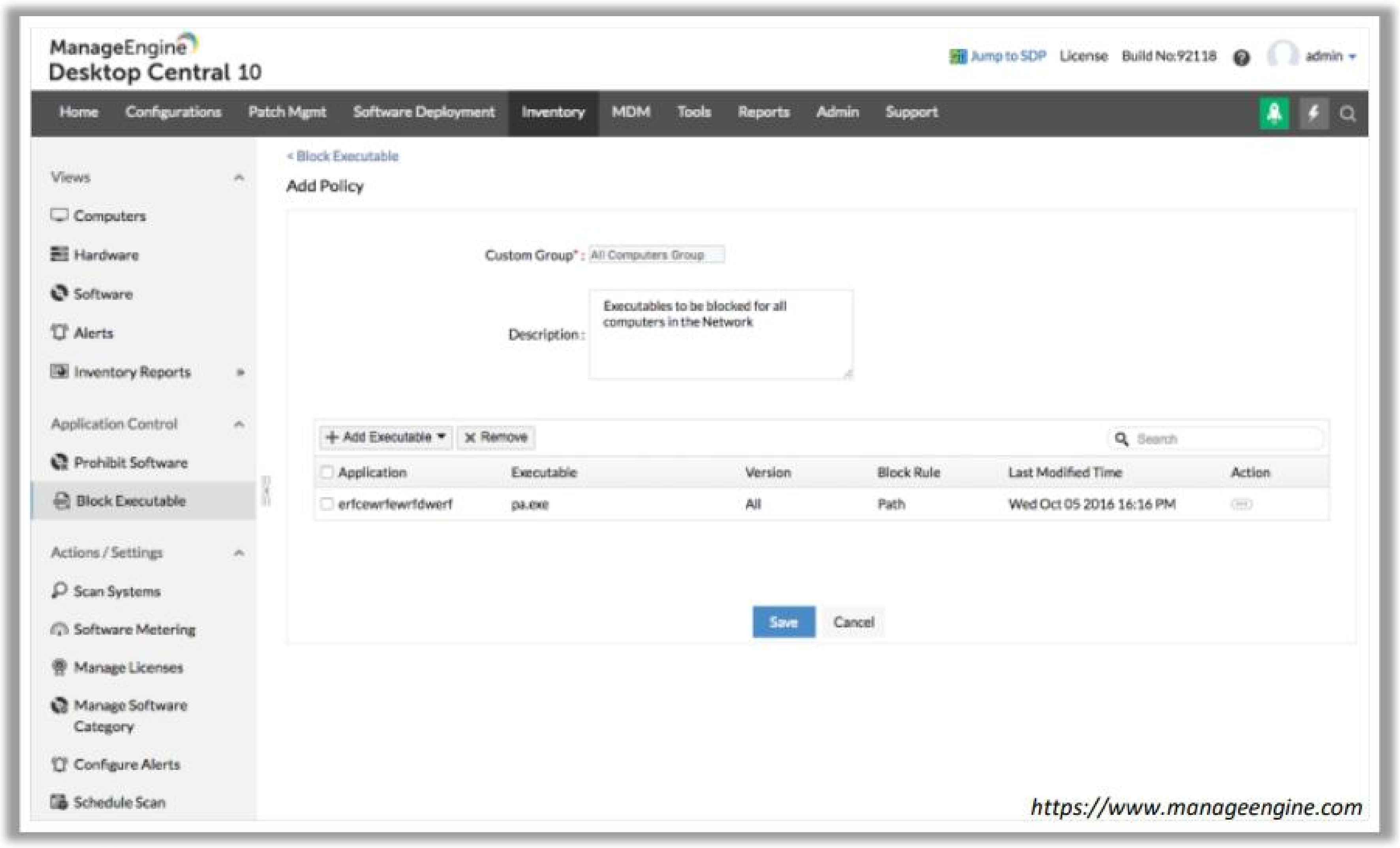
Figure 9.16: App gets blocked

Using ManageEngine Desktop Central for Application Blacklisting

- Desktop Central helps in restricting the usage of **blacklisted applications** as well as **portable executables**, which can be accessed without installation

Block Executable Features

- Enables security professionals to **block** the required applications/executables
- Block applications using the following:
  - Path rules
  - Hash values



Add Policy

Custom Group* : All Computers Group

Description : Executables to be blocked for all computers in the Network

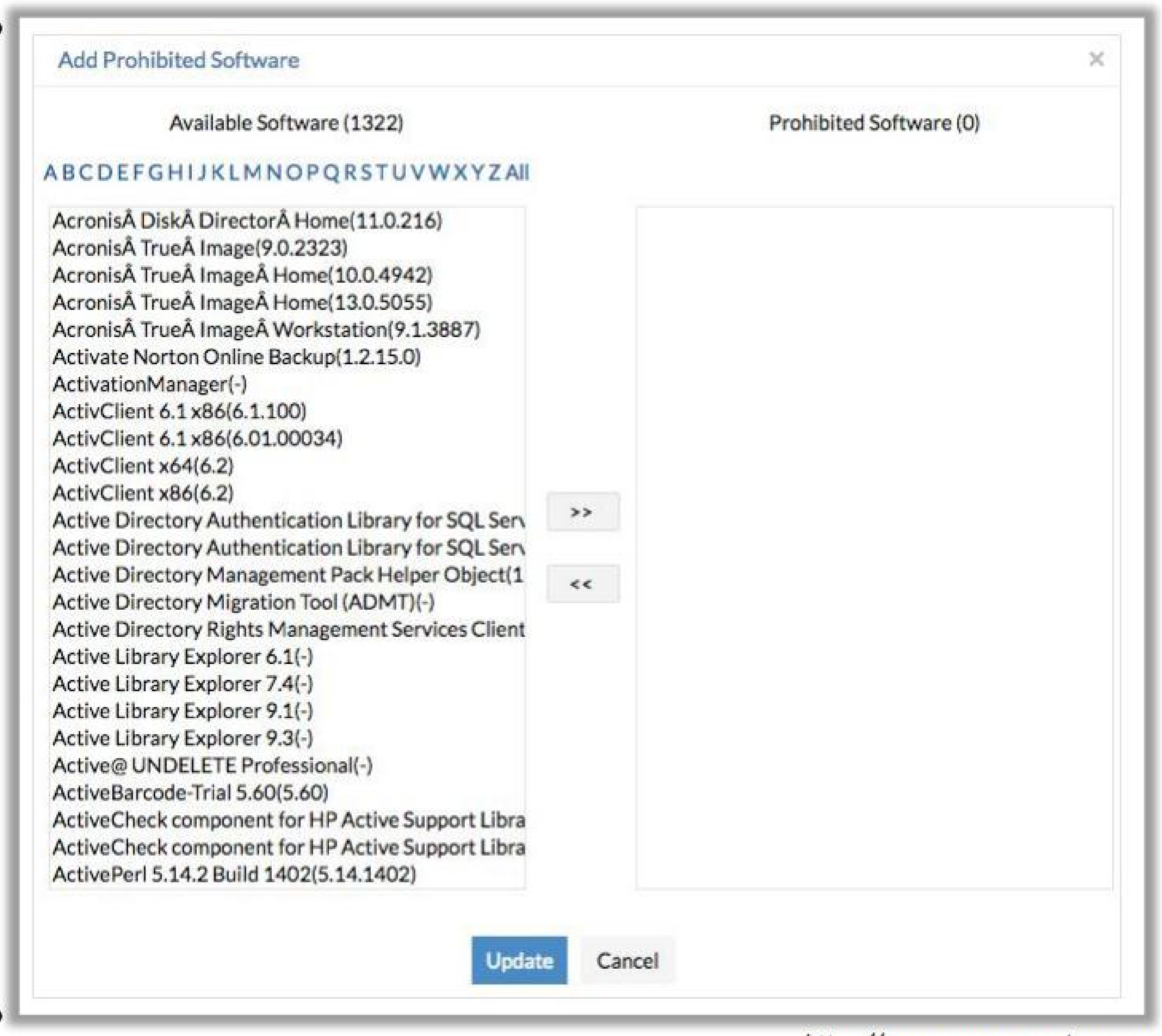
Application	Executable	Version	Block Rule	Last Modified Time	Action
erfcewrfewrfdwerf	pa.exe	All	Path	Wed Oct 05 2016 16:16 PM	

https://www.manageengine.com

Using ManageEngine Desktop Central for Application Blacklisting (Cont'd)

Prohibited Software Feature

- Enables automatic **detection** and removal of **blacklisted** applications (prohibited applications)
- Security professionals can perform the following:
  - Blacklist applications and block blacklisted applications
  - Identify blacklisted application in the network
  - Auto-uninstall the blacklisted applications
  - Exempt computers from the auto-uninstallation routine
  - Generate a report on prohibited software



Available Software (1322)

AcronisÂ DiskÂ DirectorÂ Home(11.0.216)
AcronisÂ TrueÂ Image(9.0.2323)
AcronisÂ TrueÂ ImageÂ Home(10.0.4942)
AcronisÂ TrueÂ ImageÂ Home(13.0.5055)
AcronisÂ TrueÂ ImageÂ Workstation(9.1.3887)
Activate Norton Online Backup(1.2.15.0)
ActivationManager(-)
ActivClient 6.1 x86(6.1.100)
ActivClient 6.1 x86(6.01.00034)
ActivClient x64(6.2)
ActivClient x86(6.2)
Active Directory Authentication Library for SQL Serv
Active Directory Authentication Library for SQL Serv
Active Directory Management Pack Helper Object(1
Active Directory Migration Tool (ADMT)(-)
Active Directory Rights Management Services Client
Active Library Explorer 6.1(-)
Active Library Explorer 7.4(-)
Active Library Explorer 9.1(-)
Active Library Explorer 9.3(-)
Active@ UNDELETE Professional(-)
ActiveBarcode-Trial 5.60(5.60)
ActiveCheck component for HP Active Support Libra
ActiveCheck component for HP Active Support Libra
ActivePerl 5.14.2 Build 1402(5.14.1402)

Prohibited Software (0)

https://www.manageengine.com

Using ManageEngine Desktop Central for Application Blacklisting

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

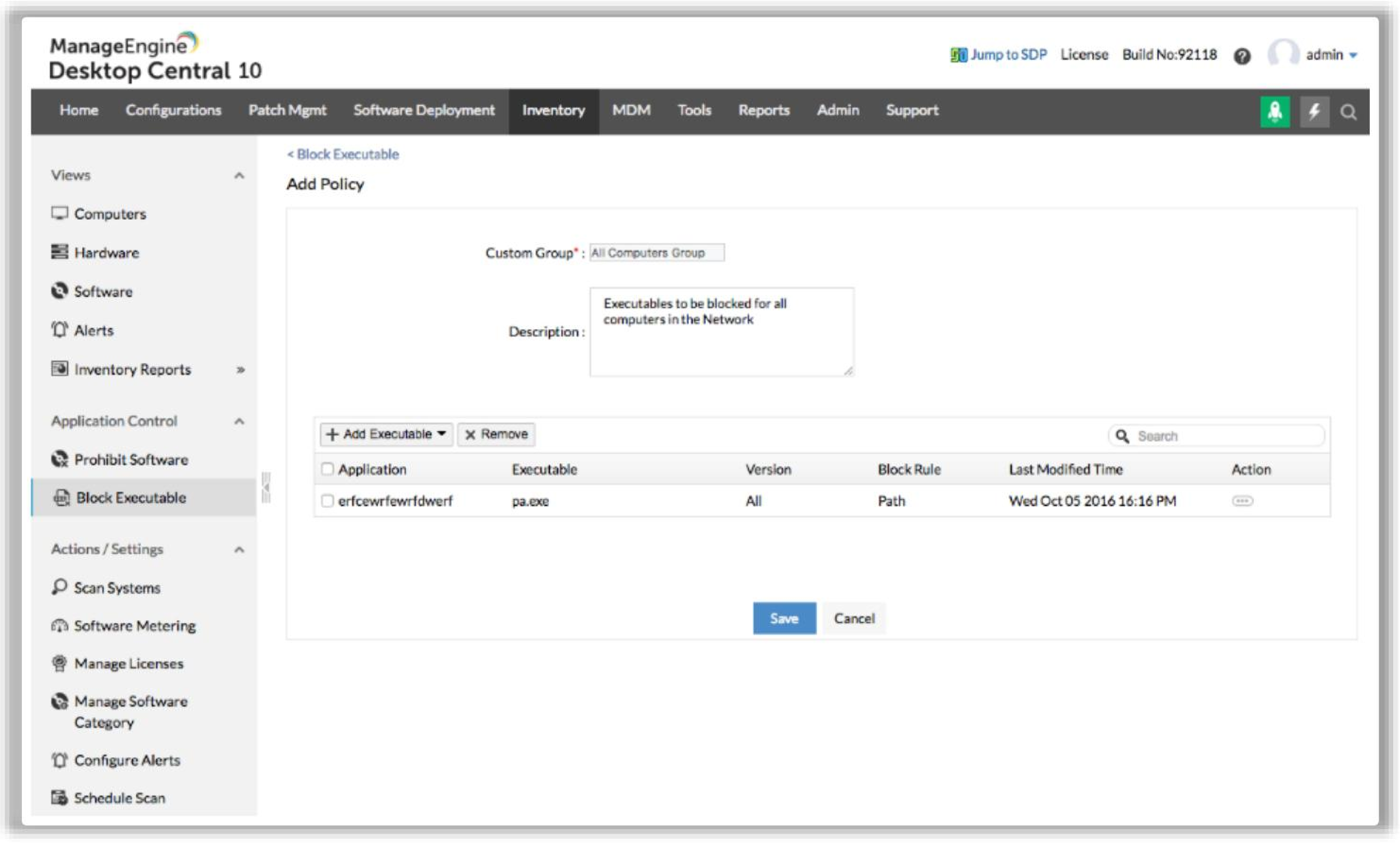
ManageEngine Desktop Central prevents blacklisted applications based on the organization's policies. It helps in restricting the usage of blacklisted applications as well as portable executables, which can be accessed without installation. The Block Executable and Prohibit Software features of ManageEngine Desktop Central can be used for Application Blacklisting.

Block Executable Feature

The Block Executable feature enables security professionals to block applications/executables. It is possible to block executables in all computers or block them for specific users/computers.

There are two methods to block an executable/application.

- **A path rule** can be used to block all versions of specific applications based on the name of the executable and its file extension.
- **A hash value** can be used to block executables even if they are renamed.



Custom Group* : All Computers Group

Description : Executables to be blocked for all computers in the Network

Application	Executable	Version	Block Rule	Last Modified Time	Action
erfcewrfewrfdwerf	pa.exe	All	Path	Wed Oct 05 2016 16:16 PM	

Figure 9.17: Screenshot of ManageEngine Desktop Central

Steps to Block an Executable/Application Using the Block Executable Feature

- **Create a policy**

  Create a policy to block an executable for a specific target. Creating a policy involves selecting the target system, selecting and adding the executable to the list, and applying the block rule as a path or hash. It is possible to create two different policies for a single executable, where one uses a path rule and the other uses a hash rule. The system must be restarted to let the changes take effect.

- **Block executables for all users/computers**

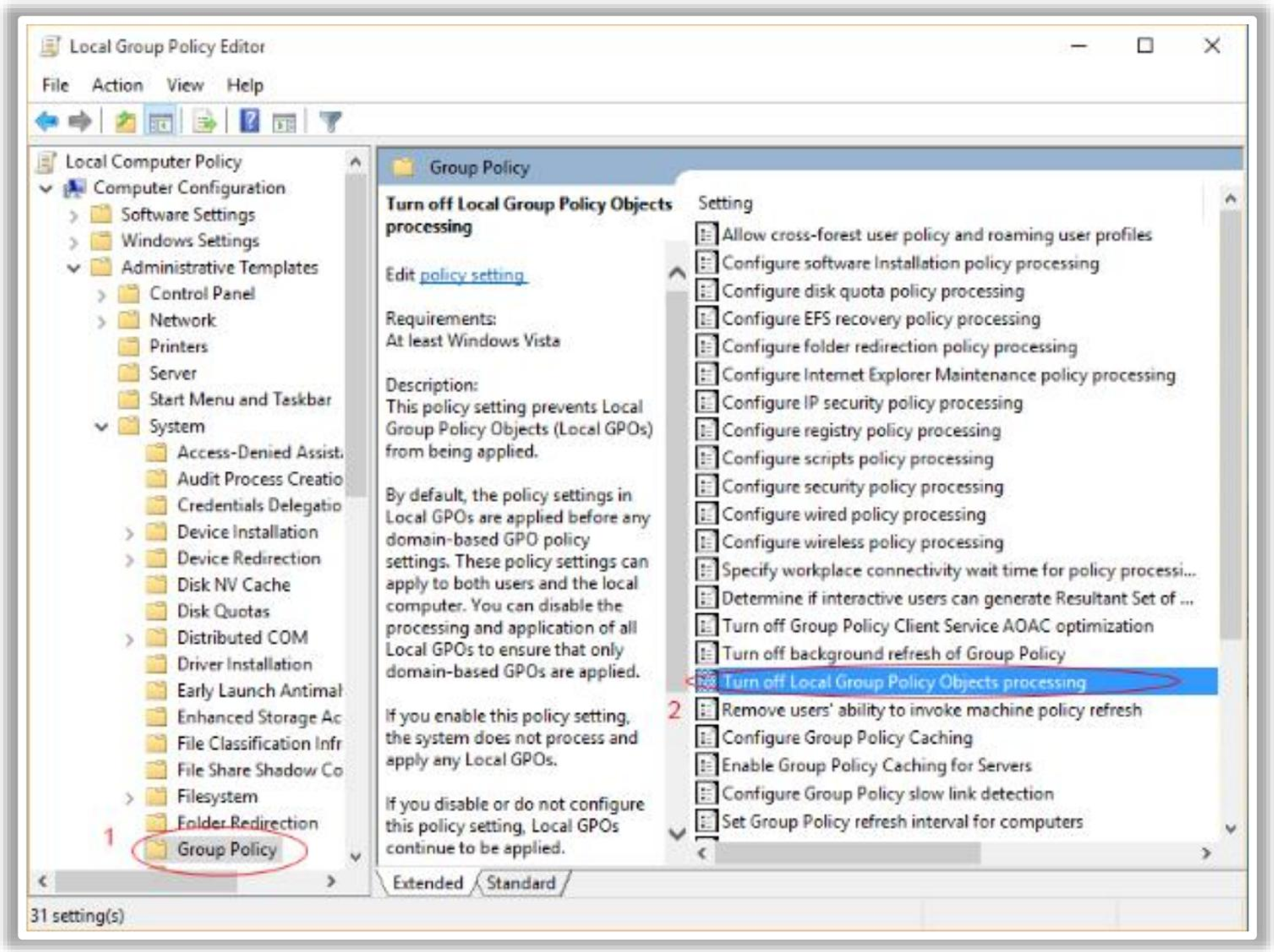
  By default, the Desktop Central features a custom group that comprises all the managed systems. Choose the **All Managed Computers** group and select the executable to be blocked if the blocking is to be applied to all the managed systems. Create a policy by specifying the target and the executable to be blocked.

- **Block executable for specific users/computers**

  Create a new custom group or use existing custom groups to block an executable only for specific targets (users or computers). Create a policy by specifying the target and the executable to be blocked.

Prerequisites for Blocking Executables/Applications

- **Enable Local Group Policy on the target machine**
  - Go to **Run**.
  - Enter **gpedit.msc**.
  - Click **Group Policy**.
  - Click **Turn Off Local Group Policy Objects Processing** in the right pane.



Turn off Local Group Policy Objects processing

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting prevents Local Group Policy Objects (Local GPOs) from being applied.

By default, the policy settings in Local GPOs are applied before any domain-based GPO policy settings. These policy settings can apply to both users and the local computer. You can disable the processing and application of all Local GPOs to ensure that only domain-based GPOs are applied.

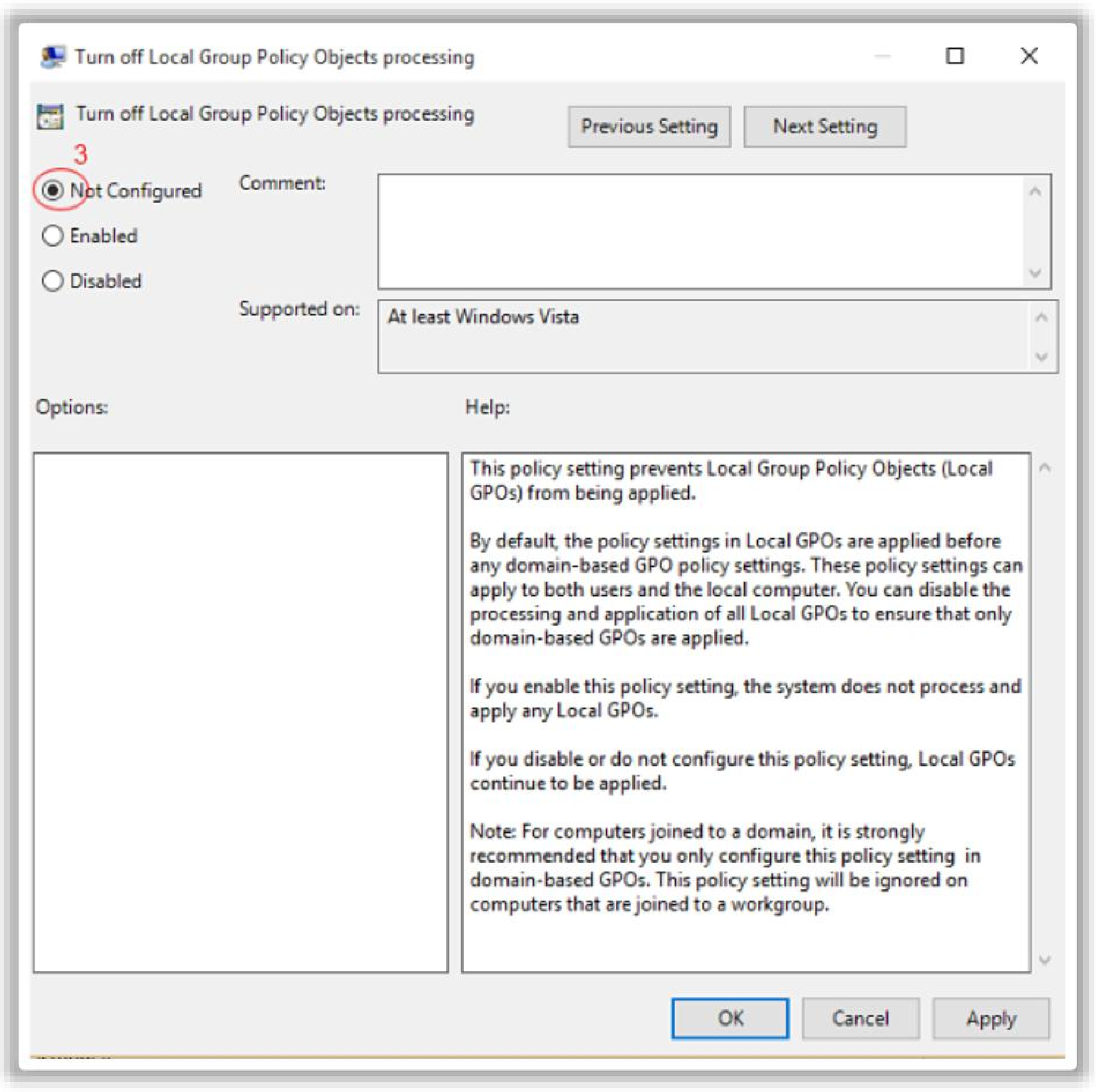
If you enable this policy setting, the system does not process and apply any Local GPOs.

If you disable or do not configure this policy setting, Local GPOs continue to be applied.

Setting
- Allow cross-forest user policy and roaming user profiles
- Configure software Installation policy processing
- Configure disk quota policy processing
- Configure EFS recovery policy processing
- Configure folder redirection policy processing
- Configure Internet Explorer Maintenance policy processing
- Configure IP security policy processing
- Configure registry policy processing
- Configure scripts policy processing
- Configure security policy processing
- Configure wired policy processing
- Configure wireless policy processing
- Specify workplace connectivity wait time for policy processi...
- Determine if interactive users can generate Resultant Set of ...
- Turn off Group Policy Client Service AOAC optimization
- Turn off background refresh of Group Policy
- Turn off Local Group Policy Objects processing
- Remove users' ability to invoke machine policy refresh
- Configure Group Policy Caching
- Enable Group Policy Caching for Servers
- Configure Group Policy slow link detection
- Set Group Policy refresh interval for computers

Figure 9.18: Select "Turn Off Local Group Policy Objects Processing" Policy Setting

- Choose **Not Configured** and click **OK**.



This policy setting prevents Local Group Policy Objects (Local GPOs) from being applied.

By default, the policy settings in Local GPOs are applied before any domain-based GPO policy settings. These policy settings can apply to both users and the local computer. You can disable the processing and application of all Local GPOs to ensure that only domain-based GPOs are applied.

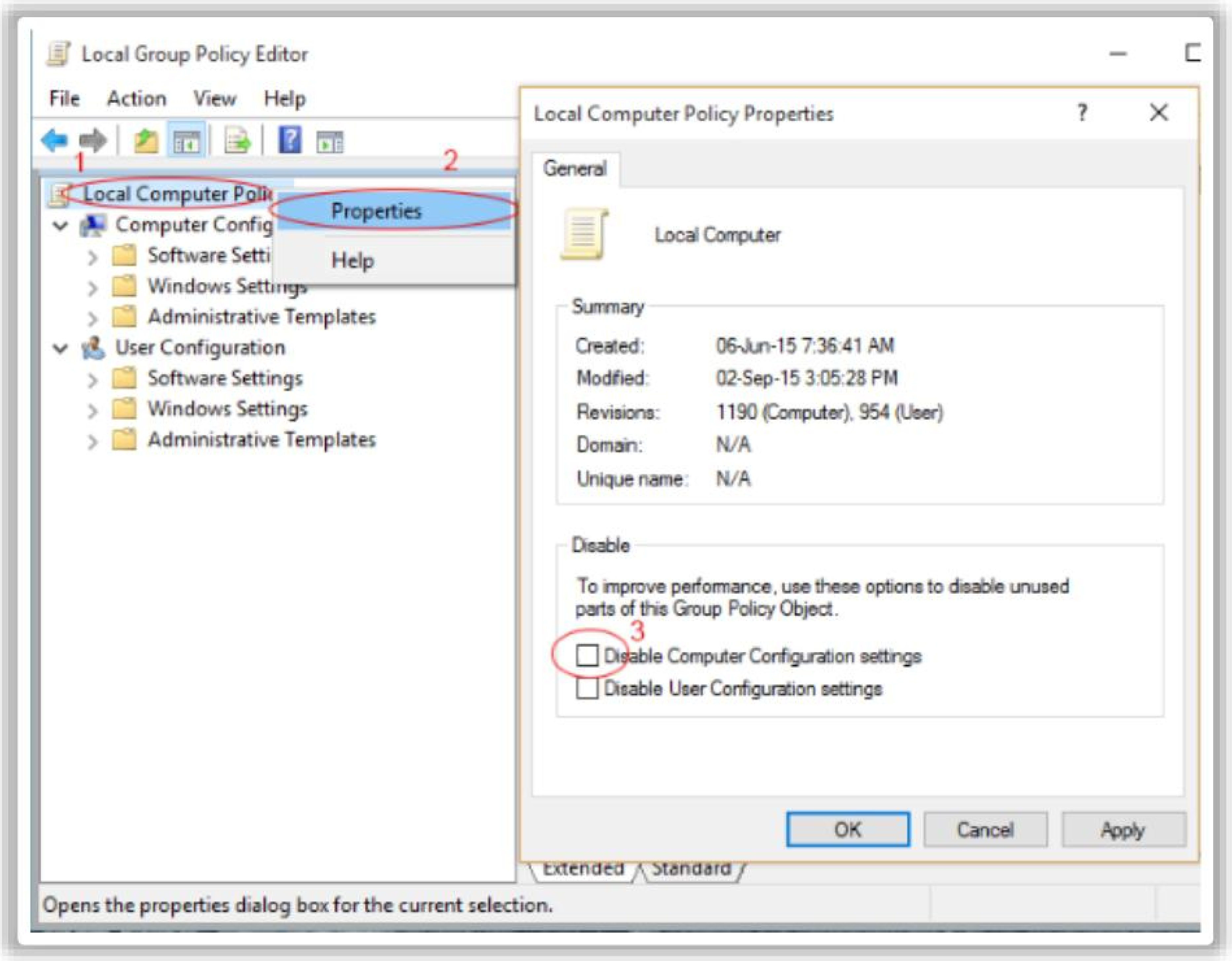
If you enable this policy setting, the system does not process and apply any Local GPOs.

If you disable or do not configure this policy setting, Local GPOs continue to be applied.

Note: For computers joined to a domain, it is strongly recommended that you only configure this policy setting in domain-based GPOs. This policy setting will be ignored on computers that are joined to a workgroup.

Figure 9.19: Selecting Not Configured option

- **Enable Local Group Policy on the target system**
  - Right-click **Local Computer Policy** in the **Local Group Policy Editor**, select **Properties**, and check **Disable Computer Configuration Settings**.

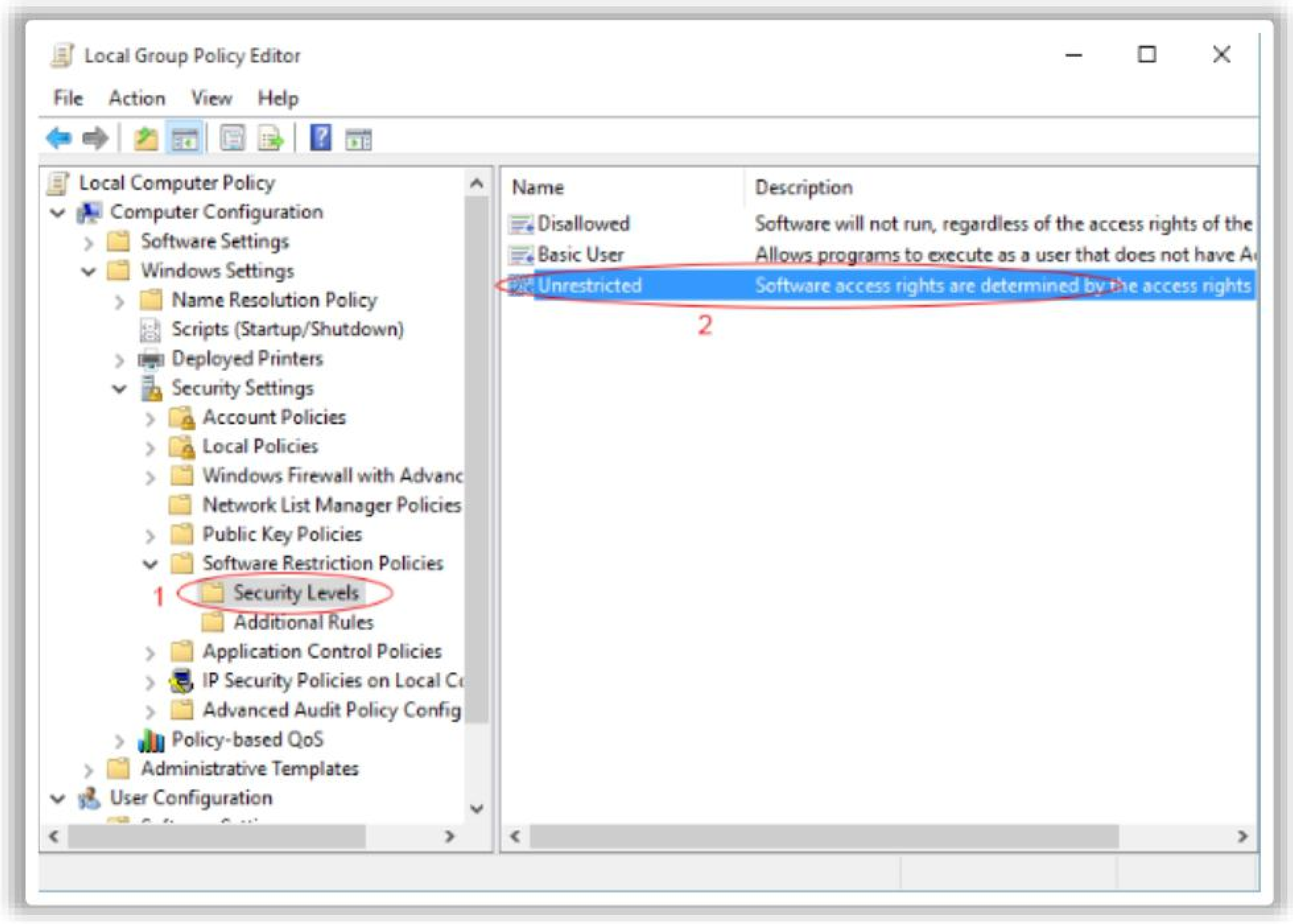


Created: 06-Jun-15 7:36:41 AM
Modified: 02-Sep-15 3:05:28 PM
Revisions: 1190 (Computer), 954 (User)
Domain: N/A
Unique name: N/A

To improve performance, use these options to disable unused parts of this Group Policy Object.

Figure 9.20: Disabling Computer Configuration Settings

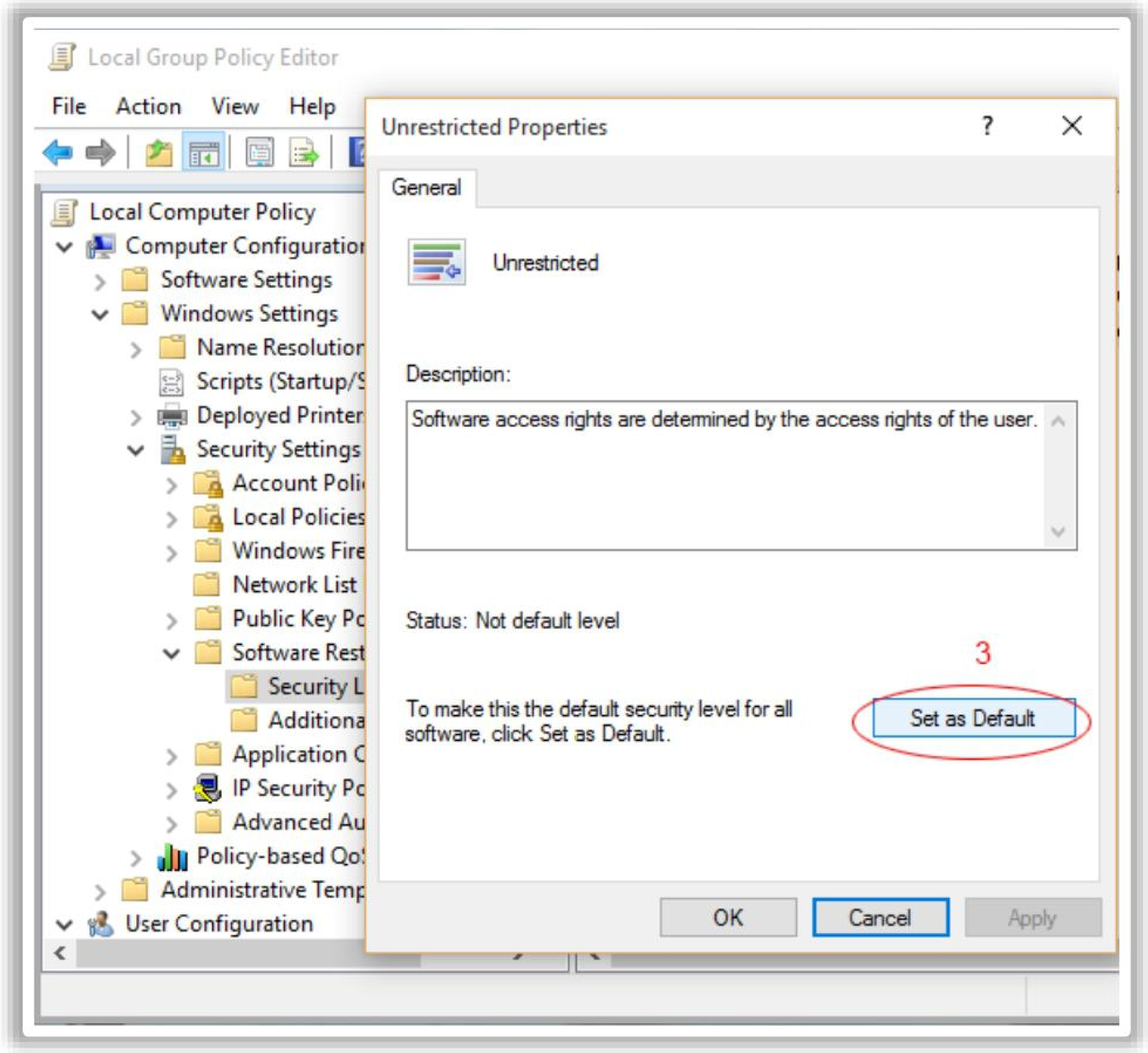
- **Set the default security policy as "Unrestricted"**
  - Go to **Local Computer Policy → Windows Settings → Security Settings → Software Restriction Policies.**
  - Click **Security levels** and double-click **Unrestricted** in the right-side pane.



Name	Description
Disallowed	Software will not run, regardless of the access rights of the
Basic User	Allows programs to execute as a user that does not have A
Unrestricted	Software access rights are determined by the access rights

Figure 9.21: Setting Security Levels

  - Click **Set as Default** in **Unrestricted Properties** window and click **OK**.



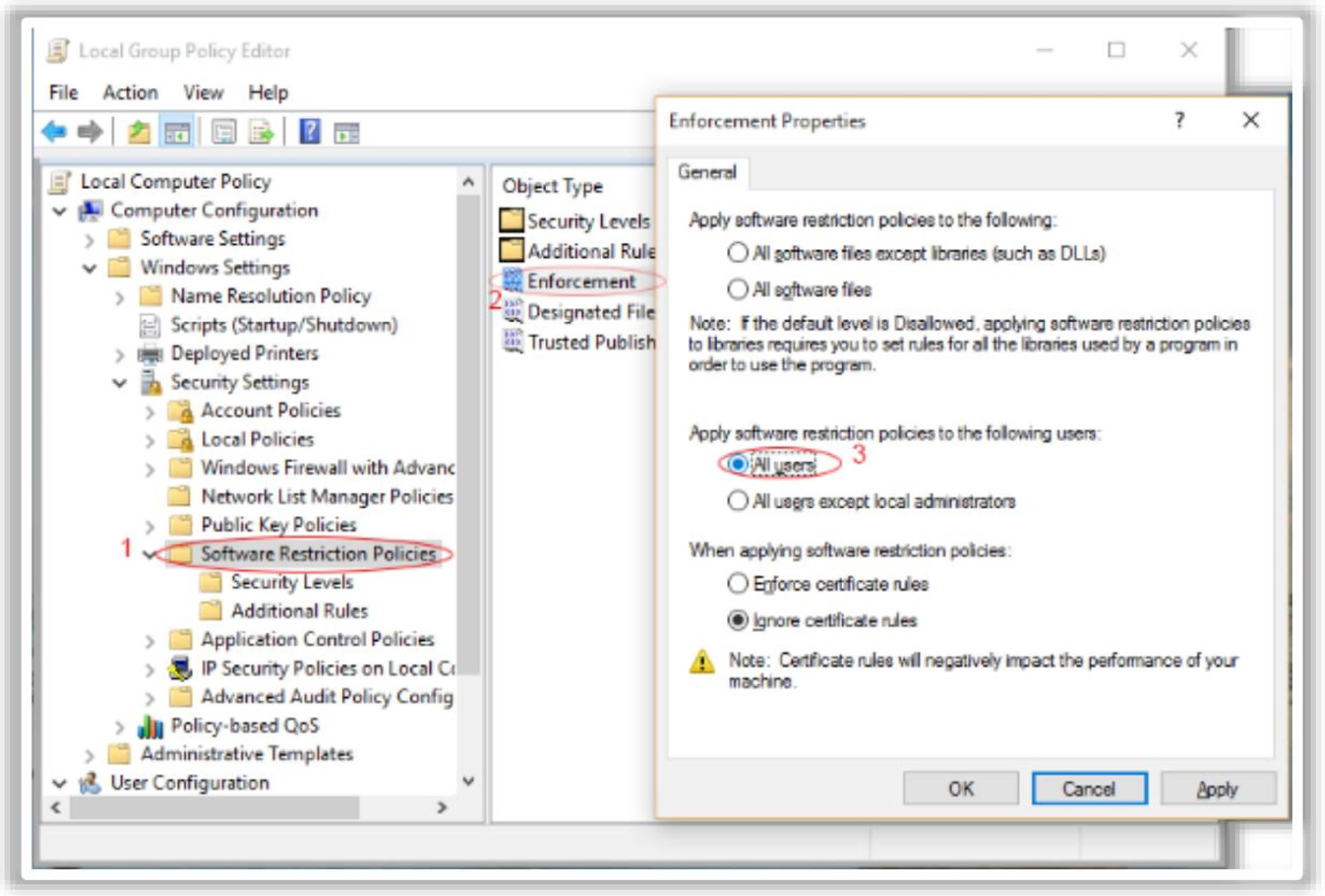
Description: Software access rights are determined by the access rights of the user.

Status: Not default level

To make this the default security level for all software, click Set as Default.

Figure 9.22: Setting the Properties of "Unrestricted" Security Level

- **Enable Local Group Policy for the administrator**
  - Go to **Local Computer Policy → Computer Configuration → Windows Settings → Security Settings → Software Restriction Policy.**
  - Double-click **Unrestricted** in the right-side pane.
  - Click **Set as Default** in **Unrestricted Properties** window and click **OK**.



Enforcement Properties

General

Apply software restriction policies to the following:

- ( ) All software files except libraries (such as DLLs)
- ( ) All software files

Note: If the default level is Disallowed, applying software restriction policies to libraries requires you to set rules for all the libraries used by a program in order to use the program.

Apply software restriction policies to the following users:

- [✓ All users]
- ( ) All users except local administrators

When applying software restriction policies:

- ( ) Enforce certificate rules
- [✓ Ignore certificate rules]

Note: Certificate rules will negatively impact the performance of your machine.

Figure 9.23: Applying Software Restriction Policies to All Users

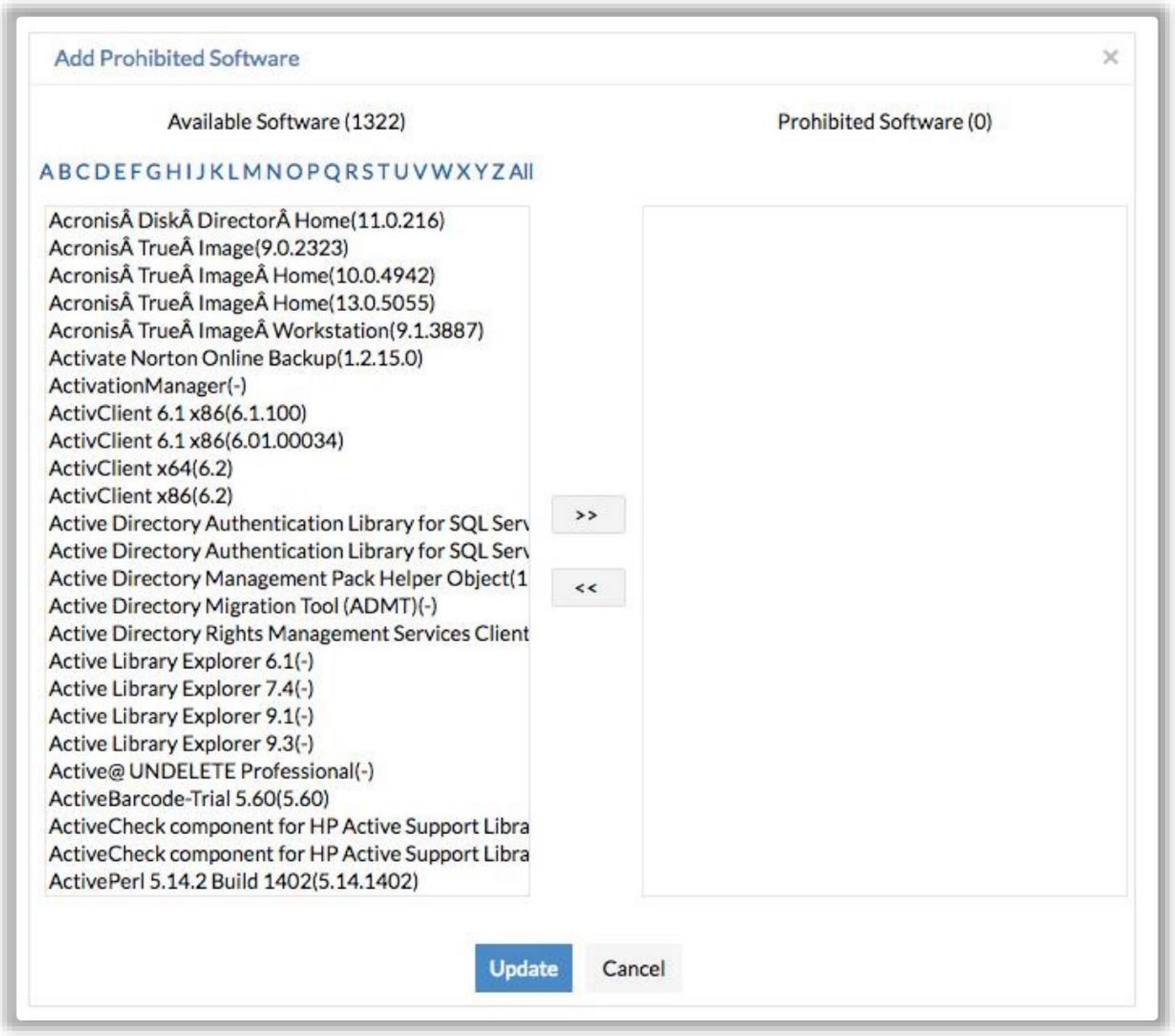
  - Double-click **Enforcement** and select **All Users.**
  - Click **OK**.

Prohibit Software Features

ManageEngine Desktop Central's Prohibited Software feature or module fully automates the detection and removal of prohibited applications.

Steps to Prohibit Applications Using Prohibit Software Feature

- **Add prohibited software to a list**
  - Navigate to **Prohibit Software** from the **Inventory** tab to view the details of all the software that have already been prohibited.
  - Click **Add Prohibited Software**. The dialog **Add Prohibited Software** lists all the software detected in the managed systems. Scan the OS at least once to know the details of the software here.


Add Prohibited Software
×
Available Software (1322)
Prohibited Software (0)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All
>>
<<
Update
Cancel

AcronisÂ DiskÂ DirectorÂ Home(11.0.216)
AcronisÂ TrueÂ Image(9.0.2323)
AcronisÂ TrueÂ ImageÂ Home(10.0.4942)
AcronisÂ TrueÂ ImageÂ Home(13.0.5055)
AcronisÂ TrueÂ ImageÂ Workstation(9.1.3887)
Activate Norton Online Backup(1.2.15.0)
ActivationManager(-)
ActivClient 6.1 x86(6.1.100)
ActivClient 6.1 x86(6.01.00034)
ActivClient x64(6.2)
ActivClient x86(6.2)
Active Directory Authentication Library for SQL Serv
Active Directory Authentication Library for SQL Serv
Active Directory Management Pack Helper Object(1
Active Directory Migration Tool (ADMT)(-)
Active Directory Rights Management Services Client
Active Library Explorer 6.1(-)
Active Library Explorer 7.4(-)
Active Library Explorer 9.1(-)
Active Library Explorer 9.3(-)
Active@ UNDELETE Professional(-)
ActiveBarcode-Trial 5.60(5.60)
ActiveCheck component for HP Active Support Libra
ActiveCheck component for HP Active Support Libra
ActivePerl 5.14.2 Build 1402(5.14.1402)

Figure 9.24: Add Available Software to Prohibited Software

- Select the software and move it to the **Prohibited List** to be blacklisted. Adding a **Software Group** under the **Prohibited Software List** blacklists all the software in that group.
- Click **Update** to confirm the addition of the software to the prohibited list.

- **Auto-uninstall the identified prohibited application**

  The steps below should be followed to configure the auto-uninstall policy to automatically uninstall prohibited software detected on the system.

  - Select the **Auto-Uninstall Policy** tab and check **Enable Automatic Uninstallation**.
  - Specify the maximum number of software that can be uninstalled from a system during the subsequent refresh cycle.

    Note: Increasing the number of software will cause high CPU usage during uninstallation. If the prohibited software count is detected to exceed the allowed maximum number of software to be uninstalled, the remaining software will be uninstalled during the subsequent startup.
  - Check **Notify User before Uninstalling** and specify a custom message to prompt the user before the software uninstallation.

Note: The user is given an alert message during login and whenever the agent identifies prohibited software. This functionality is applicable only if the **Notify User Settings** is configured.

- Specify a number for the wait window for software uninstallation if the software are to be removed a few days after detection.
- Click **Save**.

By default, the auto-uninstallation option is available for **.msi** and **.exe** applications and requires silent switches.

- **Steps to auto-uninstall .exe-based software**
  - Select the **Prohibited SW** tab and click **Not Configured link** under **Uninstall command** against the **.exe** application.
  - The **Add/Edit Uninstall Command** window pops up.
  - Choose any one of the following options:
    - **Pre-fill Uninstall Command**—This command fetches the uninstall command of the Add/Remove Programs application and displays it. Specify only the silent switch.
    - **I Will Specify Myself**—Enter the uninstall command and silent switch manually. Test the uninstallation command manually to verify its correctness.
  - Click **Save.**
  - Verify the status in the **Auto Uninstallation Status** tab.

    Note: The uninstallation occurs based on the configured auto-uninstall policy.
  - Select **Detailed View** under **Auto Uninstallation Status** to view the status and remarks.

    Note: Uninstalling a software by configuring the auto-uninstall policy does not prevent users from installing a software. Once a software is installed, it will get uninstalled automatically.

- **Exempt computers from auto-uninstallation routine**

  The following are the steps to exempt computers from the auto-uninstallation routine to allow the usage of prohibited software for certain users:

  - Navigate to **Prohibit Software** from the **Inventory** tab to view the details of all the software that are already prohibited.
  - Select the checkbox corresponding to the specified software and click the link under the **Exclusions** column to open the **Add Exclusions** dialog.
  - Select whether to exclude **custom groups** or **computers**, select **the groups/computers**, and move them to the **Excluded** list.

- Click **Save**.

- **Approve requests to use prohibited software**
  - Select the specific prohibited software from the list of prohibited software from the agent tray icon.
  - Handle requests from **Desktop Central web console → Inventory → Prohibit Software → User Requests**.

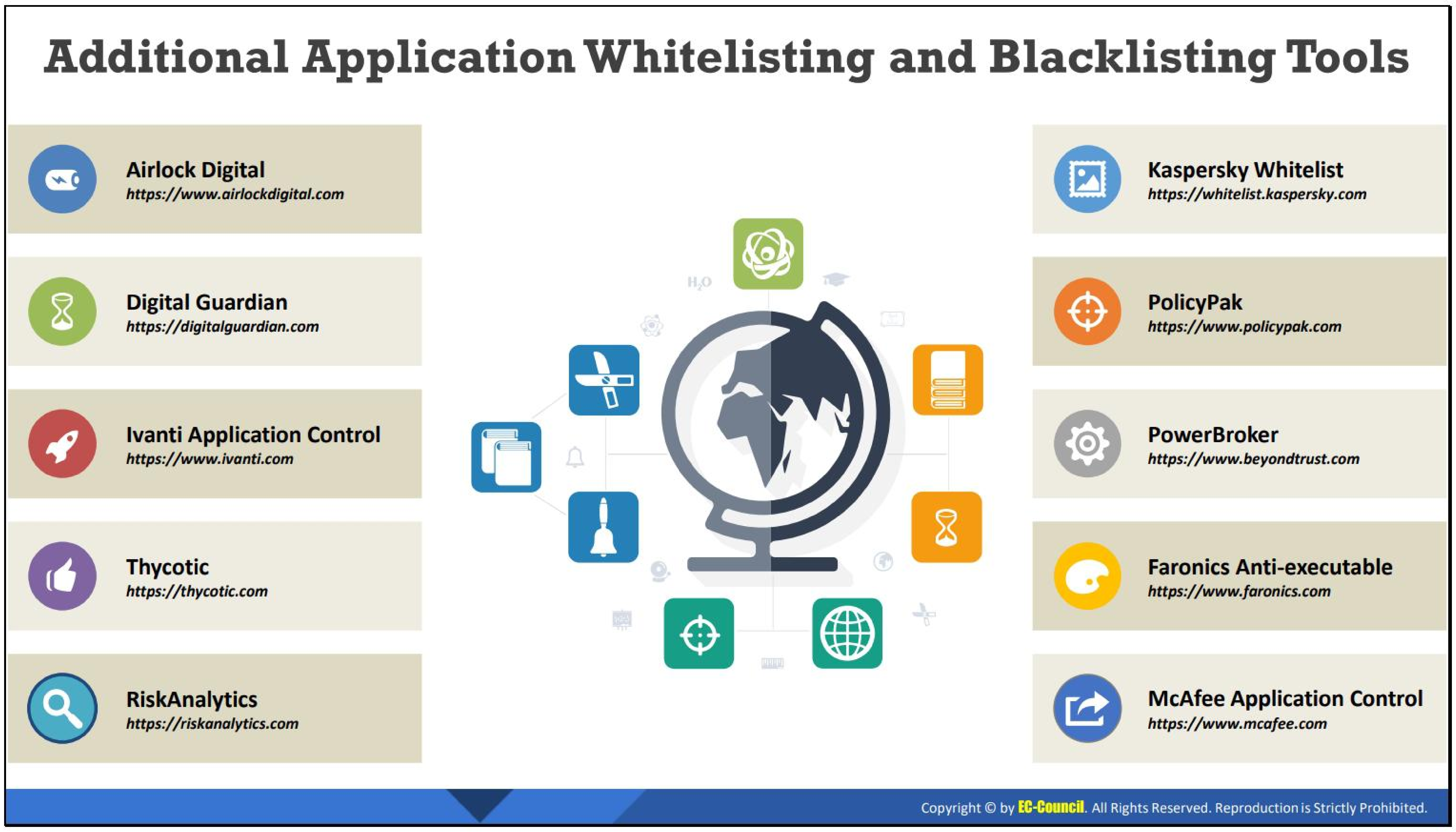
  Users are allowed to install and use the prohibited software they request once the request is approved.

- **Notify admin and end users when prohibited software is detected**

  The following are the steps to notify the admin and end users when prohibited software is detected:
  - Navigate to the **Inventory** tab.
  - Click **Configure E-mail Alerts** in the left pane under Actions/Settings.
  - Under **Notifications**, specify when the notifications should be sent, and configure alerts based on requirements.
  - Specify the email address or addresses to which the notifications must be sent.
  - Click **Save.**

- **Generate a report on prohibited software**

  The following are the steps to generate a report on prohibited software to find the computers in the network using applications at any point of time:
  - Select the **Inventory** tab.
  - Choose the **Prohibited Software** link under the **Software Reports** category by moving the mouse over **Inventory Reports**.



Additional Application Whitelisting and Blacklisting Tools

- **Airlock Digital** *https://www.airlockdigital.com*
- **Digital Guardian** *https://digitalguardian.com*
- **Ivanti Application Control** *https://www.ivanti.com*
- **Thycotic** *https://thycotic.com*
- **RiskAnalytics** *https://riskanalytics.com*
- **Kaspersky Whitelist** *https://whitelist.kaspersky.com*
- **PolicyPak** *https://www.policypak.com*
- **PowerBroker** *https://www.beyondtrust.com*
- **Faronics Anti-executable** *https://www.faronics.com*
- **McAfee Application Control** *https://www.mcafee.com*

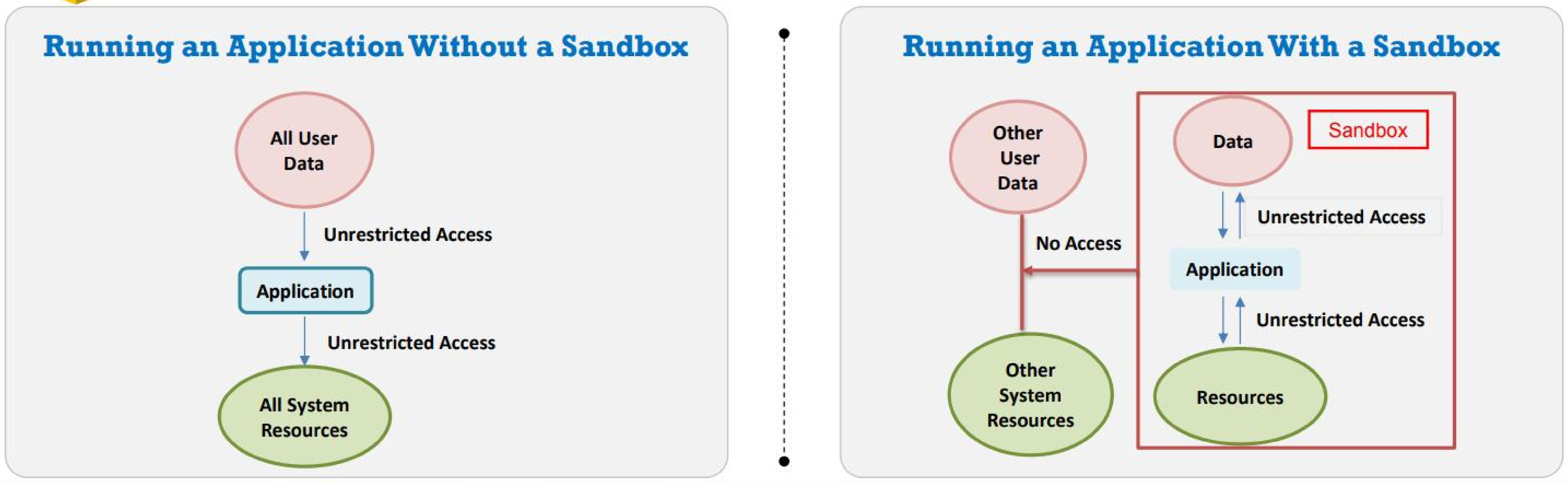
Additional Application Whitelisting and Blacklisting Tools

Some additional application whitelisting and blacklisting tools are listed below:

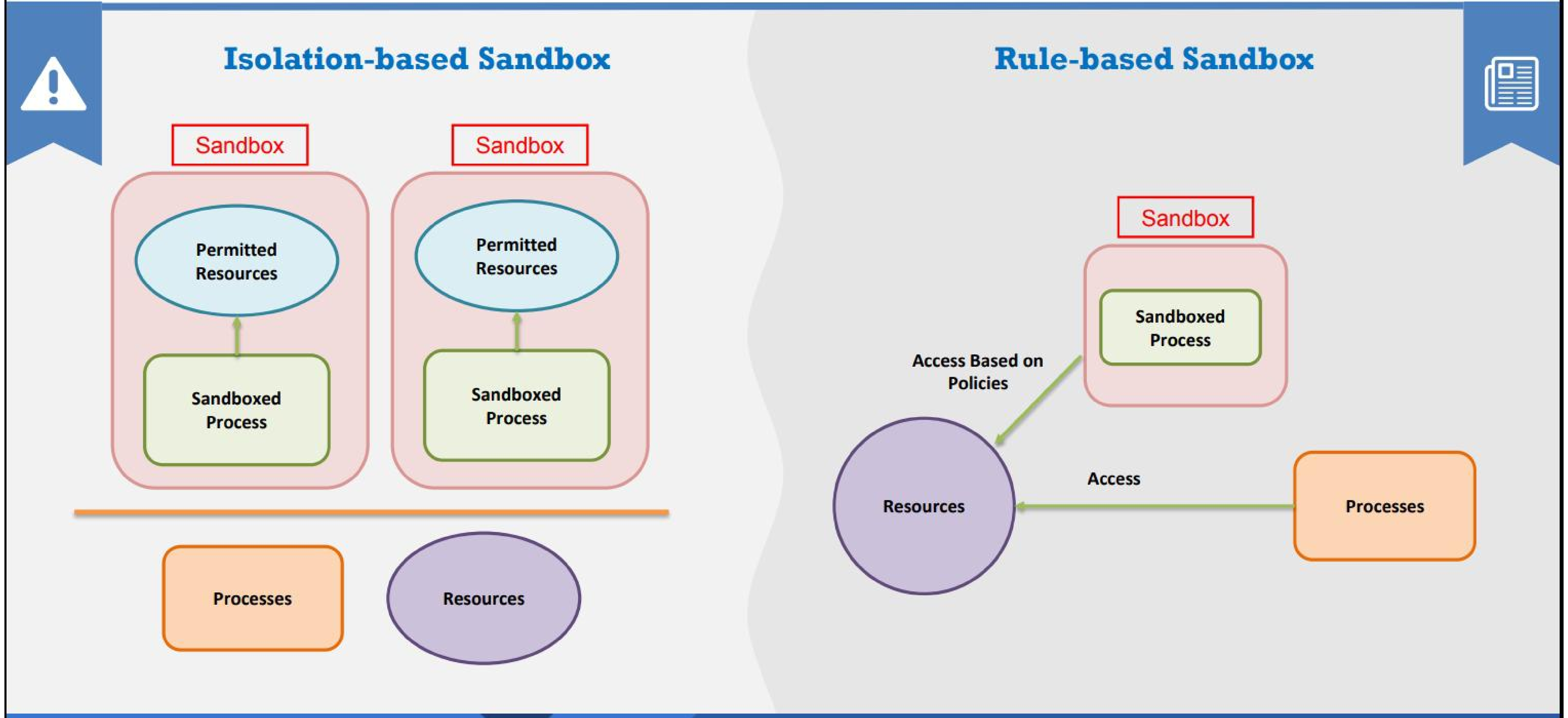
- Airlock Digital (*https://www.airlockdigital.com*)
- Digital Guardian (*https://digitalguardian.co*m)
- Ivanti Application Control (*https://www.ivanti.com*)
- Thycotic (*https://thycotic.com*)
- RiskAnalytics (*https://riskanalytics.com*)
- Kaspersky Whitelist (*https://whitelist.kaspersky.com*)
- PolicyPak (*https://www.policypak.com*)
- PowerBroker (*https://www.beyondtrust.com*)
- Faronics Anti-executable (*https://www.faronics.com*)
- McAfee Application Control (*https://www.mcafee.com*)

Application Sandboxing

- Sandboxing runs applications in a sealed container (sandbox) such that they cannot access critical system resources or other programs
- Sandboxing is used to execute untrusted or untested programs from third parties
- It provides an extra layer of security and protects apps and the system from malicious apps
- Security professionals can test their tasks in a sandbox without affecting the system



Application Sandboxing (Cont'd)



Application Sandboxing

Application sandboxing is the process of running applications in a sealed container (sandbox) so that the applications cannot access critical system resources and other programs. It provides an extra layer of security and protects apps and the system from malicious apps. It is often used to execute untrusted or untested programs or code from untrusted or unverified third parties without risking the host system or OS. The protection provided by the sandbox is not sufficiently robust against advanced malware that target the OS kernel.

When an application is executed without a sandbox, it has unrestricted access to system resources and all user data. In contrast, an application executed within a sandbox has restricted access to the system resources and data outside the sandbox.

Installing a sandboxed app in a system creates a specific directory (sandboxed directory). By default, the app has unlimited read and write access to the directory. However, apps within the directory are not allowed to read or write the files outside the directory or access other system resources, unless authorized.

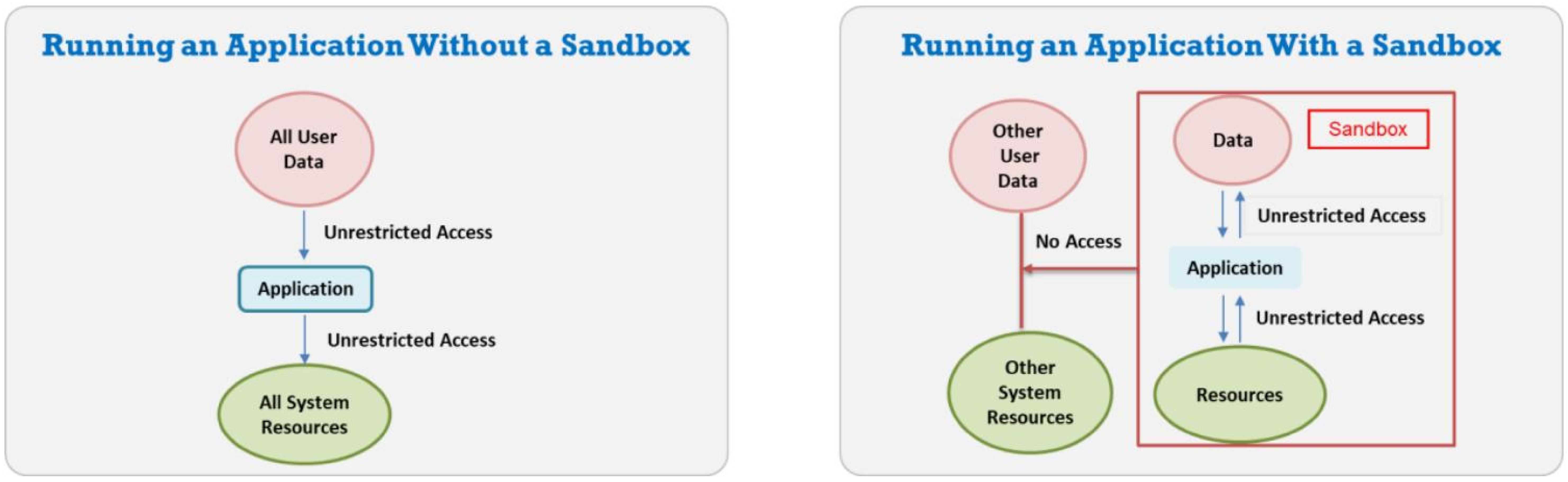


Figure 9.25: Execution of an application with and without a sandbox

The following approaches can be used to implement an application sandbox.

- **Isolation-based approach:** In this approach, a program running in the sandbox is isolated from the system resources and programs running outside the sandbox.

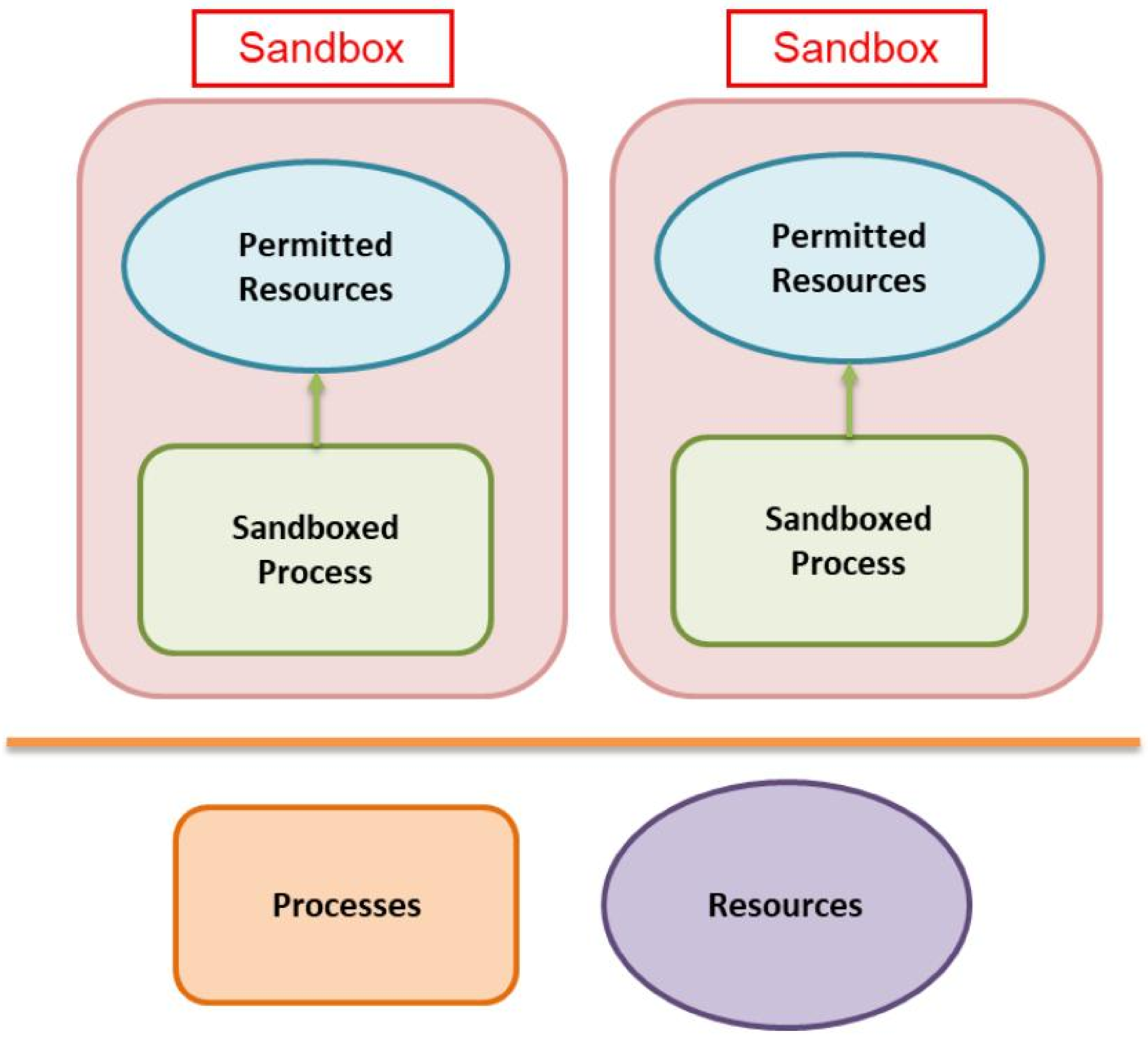


Figure 9.26: Isolation-based sandbox

- **Rule-based approach:** In this approach, the sandbox controls what each application can do and permits applications to share resources based on the set rules.

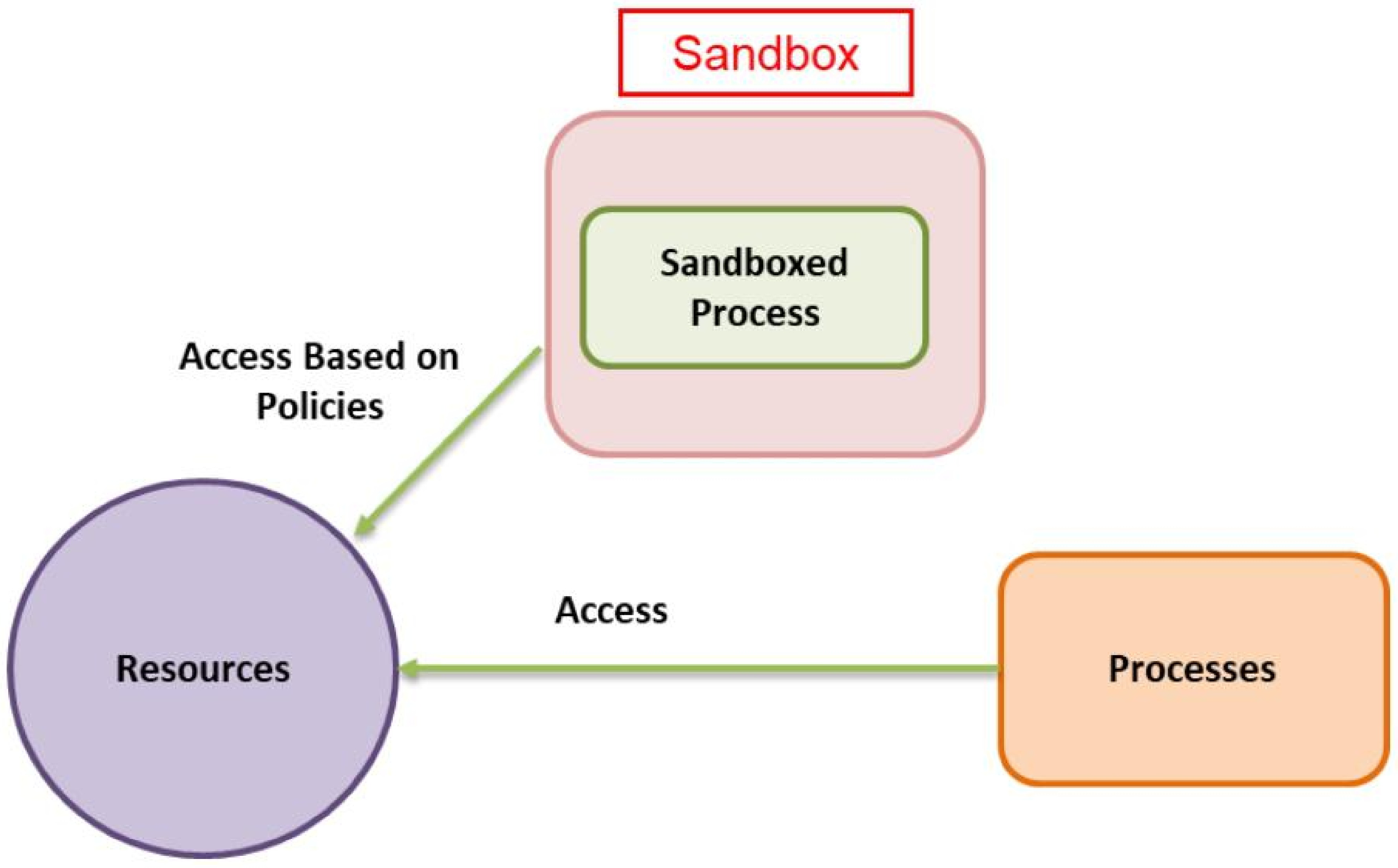
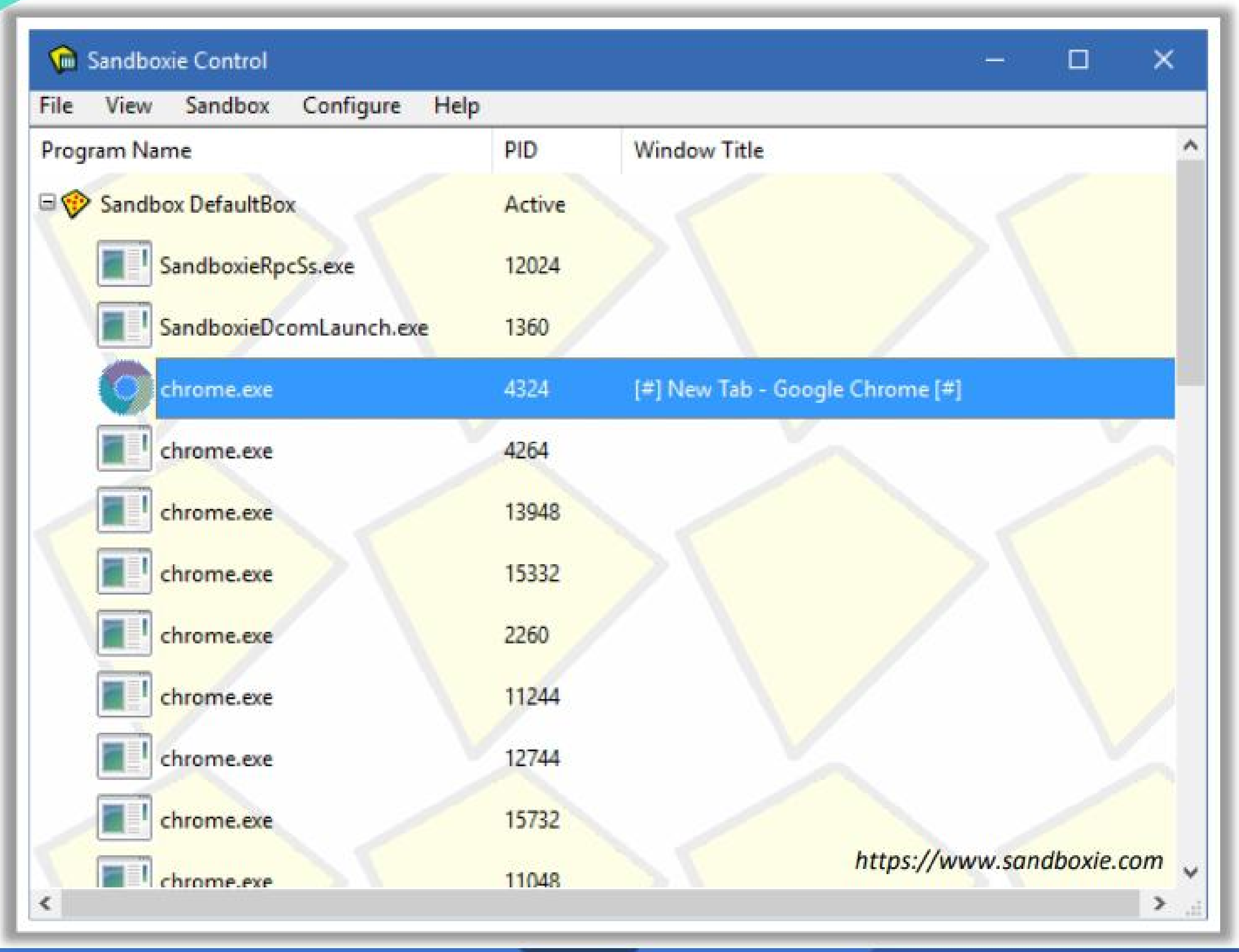


Figure 9.27: Rule-based sandbox

Application Sandboxing Tools

Sandboxie

Sandboxie keeps the browser isolated and blocks malicious software, viruses, ransomware, and zero-day threats



Sandboxie Control

File View Sandbox Configure Help

Program Name	PID	Window Title
Sandbox DefaultBox	Active	
SandboxieRpcSs.exe	12024	
SandboxieDcomLaunch.exe	1360	
chrome.exe	4324	[#] New Tab - Google Chrome [#]
chrome.exe	4264	
chrome.exe	13948	
chrome.exe	15332	
chrome.exe	2260	
chrome.exe	11244	
chrome.exe	12744	
chrome.exe	15732	
chrome.exe	11048	

https://www.sandboxie.com

BUFFERZONE
https://bufferzonesecurity.com

SHADE Sandbox
https://www.shadesandbox.com

Shadow Defender
http://www.shadowdefender.com

Browser in the Box TS
https://www.rohde-schwarz.com

Toolwiz Time Freeze
http://www.toolwiz.com

Application Sandboxing Tools

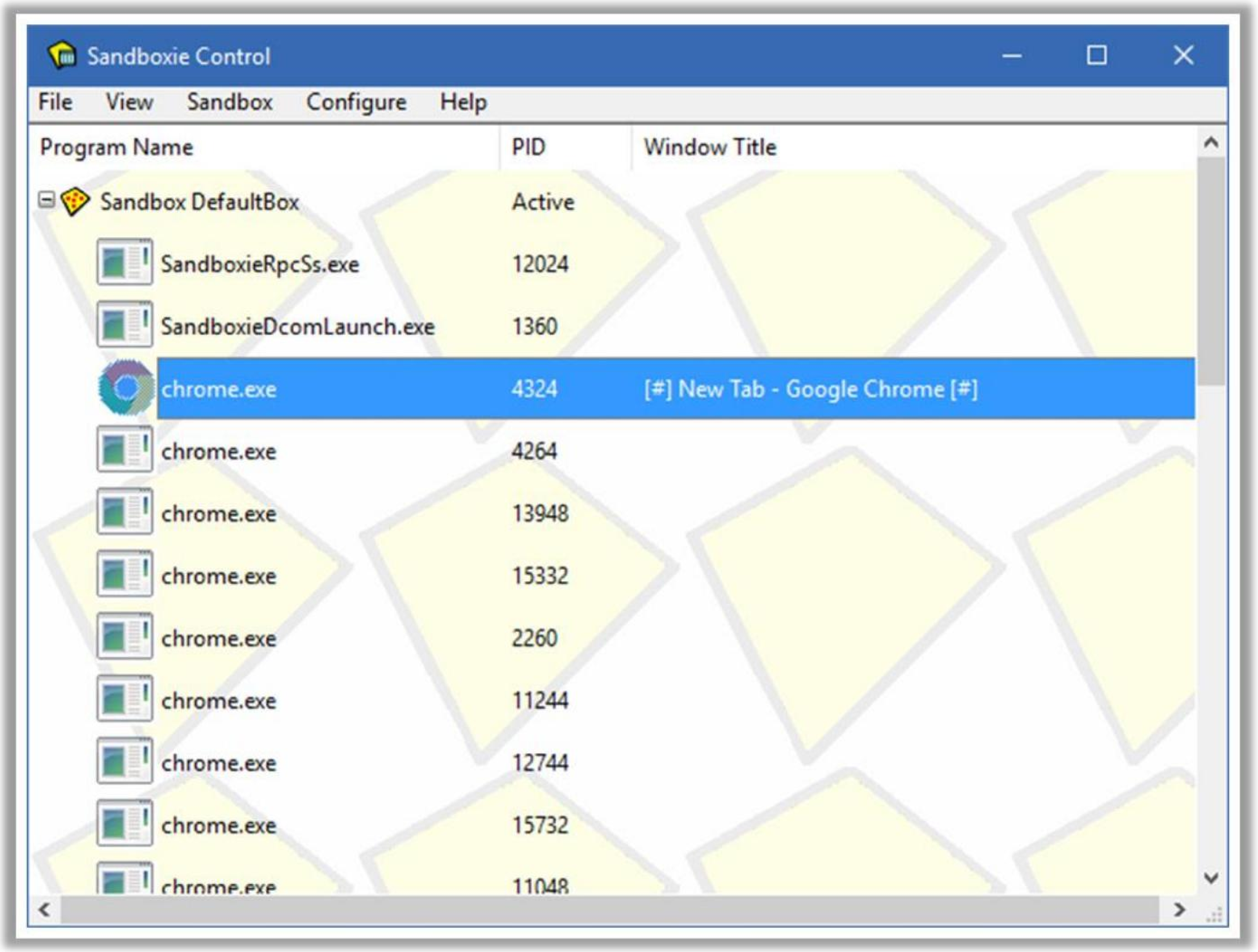
- **Sandboxie**

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

  Sandboxie is a sandboxing tool developed by Sophos. It keeps the browser isolated and blocks malicious software, viruses, ransomware, and zero-day threats. It prevents websites from modifying files and folders on the system.

  The following are the steps to allow already installed programs (e.g., a browser) in Sandboxie:

  - Select **Sandbox → Default Box → Run Sandboxed → Run Web browser**.
  - Select **Run Any Program** to allow any other application.

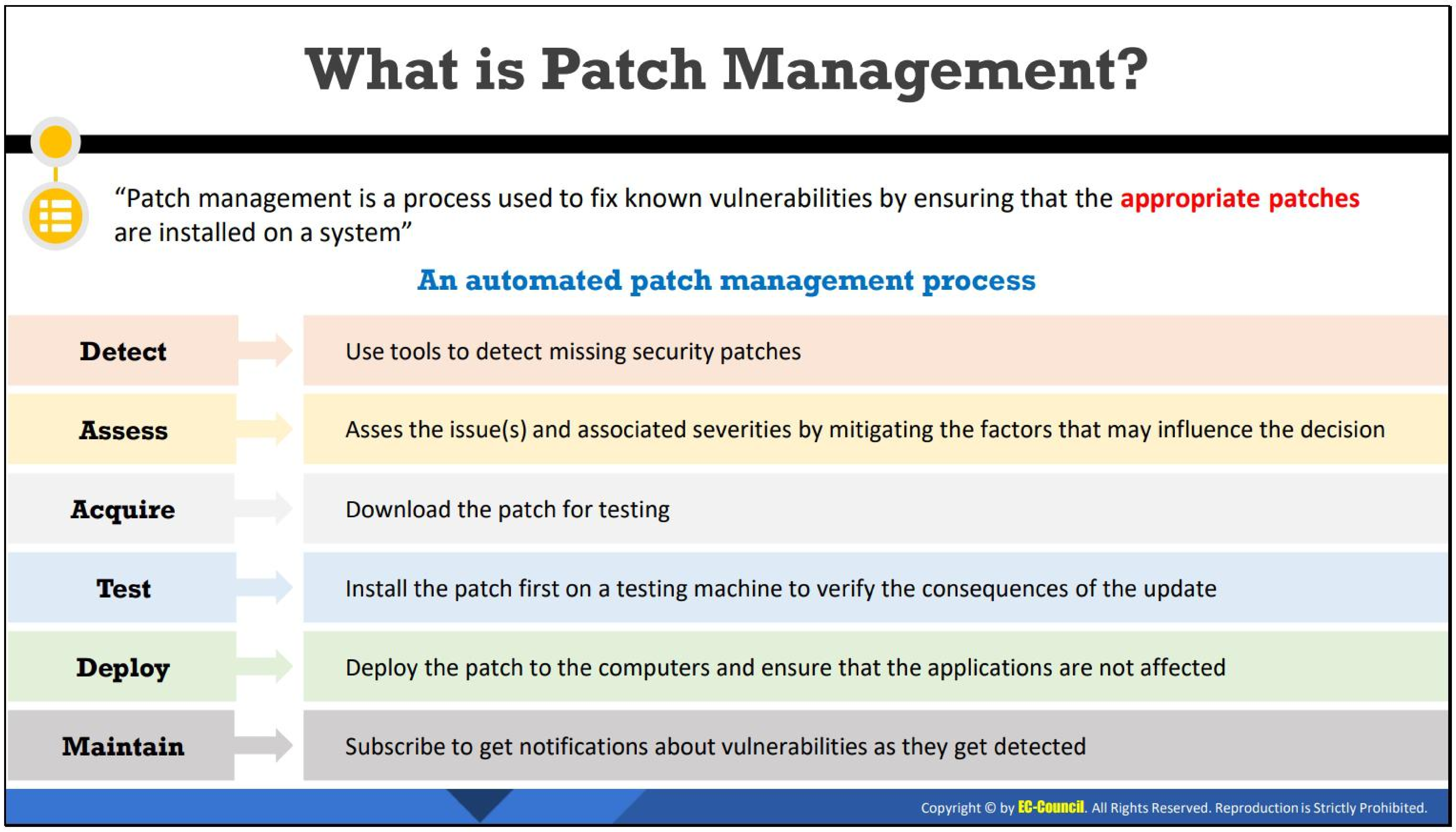


```
Program Name                  PID      Window Title
Sandbox DefaultBox            Active
  SandboxieRpcSs.exe          12024
  SandboxieDcomLaunch.exe     1360
  chrome.exe                  4324     [#] New Tab - Google Chrome [#]
  chrome.exe                  4264
  chrome.exe                  13948
  chrome.exe                  15332
  chrome.exe                  2260
  chrome.exe                  11244
  chrome.exe                  12744
  chrome.exe                  15732
  chrome.exe                  11048
```

Figure 9.28: Working of Sandboxie Control

Some additional application sandboxing tools are listed below:

- BUFFERZONE (*https://bufferzonesecurity.com*)
- SHADE Sandbox (*https://www.shadesandbox.com*)
- Shadow Defender (*http://www.shadowdefender.com*)
- Browser in the Box TS (*https://www.rohde-schwarz.com*)
- Toolwiz Time Freeze (*http://www.toolwiz.com*)



What is Patch Management?

"Patch management is a process used to fix known vulnerabilities by ensuring that the **appropriate patches** are installed on a system"

An automated patch management process

Detect	Use tools to detect missing security patches
Assess	Asses the issue(s) and associated severities by mitigating the factors that may influence the decision
Acquire	Download the patch for testing
Test	Install the patch first on a testing machine to verify the consequences of the update
Deploy	Deploy the patch to the computers and ensure that the applications are not affected
Maintain	Subscribe to get notifications about vulnerabilities as they get detected

What is Patch Management?

According to *https://searchenterprisedesktop.techtarget.com*, patch management is an area of systems management that involves acquiring, testing, and installing multiple patches (code changes) in an administered computer system. Patch management is a method of defense against vulnerabilities that cause security weaknesses or corrupt data. It is a process of scanning for network vulnerabilities, detecting missed security patches and hotfixes, and then deploying the relevant patches as soon as they are available to secure the network. It involves the following tasks:

- Choosing, verifying, testing, and applying patches
- Updating previously applied patches with current patches
- Listing patches applied previously to the current software
- Recording repositories or depots of patches for easy selection
- Assigning and deploying the applied patches

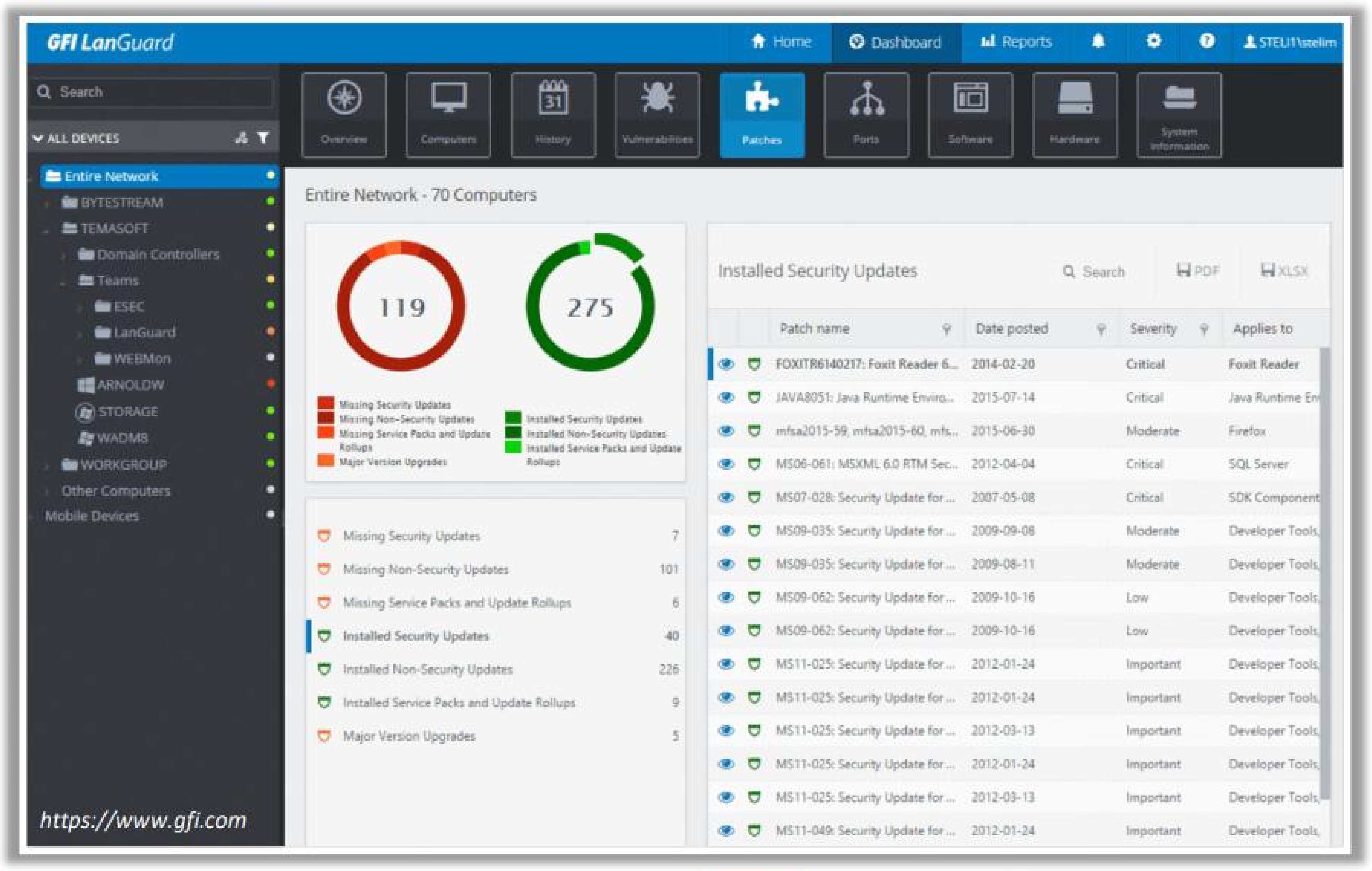
An automated patch management process includes the following steps.

- **Detect**: Use tools to detect missing security patches.
- **Assess**: Asses the issue(s) and its associated severity by mitigating the factors that may influence the decision.
- **Acquire**: Download the patch for testing.
- **Test**: Install the patch first on a test machine to verify the consequences of the update.
- **Deploy**: Deploy the patch to computers and ensure that applications are not affected.
- **Maintain**: Subscribe to receive notifications about vulnerabilities when they are reported.

Patch Management Tools

GFI LanGuard

GFI LanGuard's patch management automatically scans your network and **installs and manages security and non-security patches**



Entire Network - 70 Computers

Missing Security Updates	7
Missing Non-Security Updates	101
Missing Service Packs and Update Rollups	6
Installed Security Updates	40
Installed Non-Security Updates	226
Installed Service Packs and Update Rollups	9
Major Version Upgrades	5

Installed Security Updates

Patch name	Date posted	Severity	Applies to
FOXITR6140217: Foxit Reader 6...	2014-02-20	Critical	Foxit Reader
JAVA8051: Java Runtime Enviro...	2015-07-14	Critical	Java Runtime En
mfsa2015-59, mfsa2015-60, mfs...	2015-06-30	Moderate	Firefox
MS06-061: MSXML 6.0 RTM Sec...	2012-04-04	Critical	SQL Server
MS07-028: Security Update for ...	2007-05-08	Critical	SDK Component
MS09-035: Security Update for ...	2009-09-08	Moderate	Developer Tools,
MS09-035: Security Update for ...	2009-08-11	Moderate	Developer Tools,
MS09-062: Security Update for ...	2009-10-16	Low	Developer Tools,
MS09-062: Security Update for ...	2009-10-16	Low	Developer Tools,
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools,
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools,
MS11-025: Security Update for ...	2012-03-13	Important	Developer Tools,
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools,
MS11-025: Security Update for ...	2012-03-13	Important	Developer Tools,
MS11-049: Security Update for ...	2012-01-24	Important	Developer Tools,

https://www.gfi.com

Symantec Client Management Suite
https://www.broadcom.com

Solarwinds Patch Manager
https://www.solarwinds.com

Kaseya Patch Management
https://www.kaseya.com

Software Vulnerability Manager
https://www.flexera.com

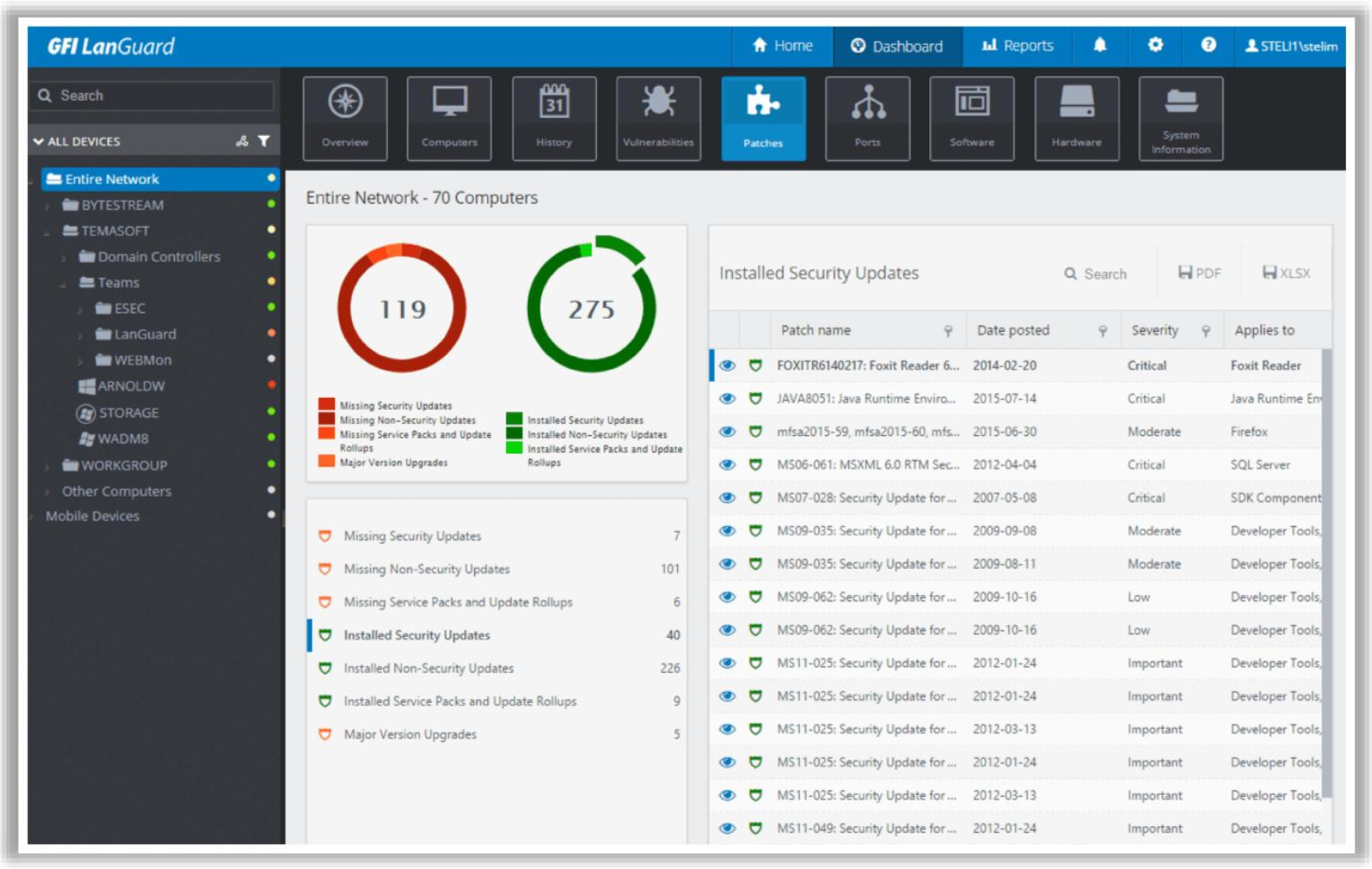
Ivanti Patch for Endpoint Manager
https://www.ivanti.com

Patch Management Tools

- **GFI LanGuard**

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

  The GFI LanGuard patch management software scans the user's network automatically as well as installs and manages security and non-security patches. It supports machines across Microsoft®, MAC OS X®, and Linux® operating systems, as well as many third-party applications. It allows auto-downloads of missing patches as well as patch rollback, resulting in a consistently configured environment that is protected from threats and vulnerabilities.



Entire Network - 70 Computers

Missing Security Updates	7
Missing Non-Security Updates	101
Missing Service Packs and Update Rollups	6
Installed Security Updates	40
Installed Non-Security Updates	226
Installed Service Packs and Update Rollups	9
Major Version Upgrades	5

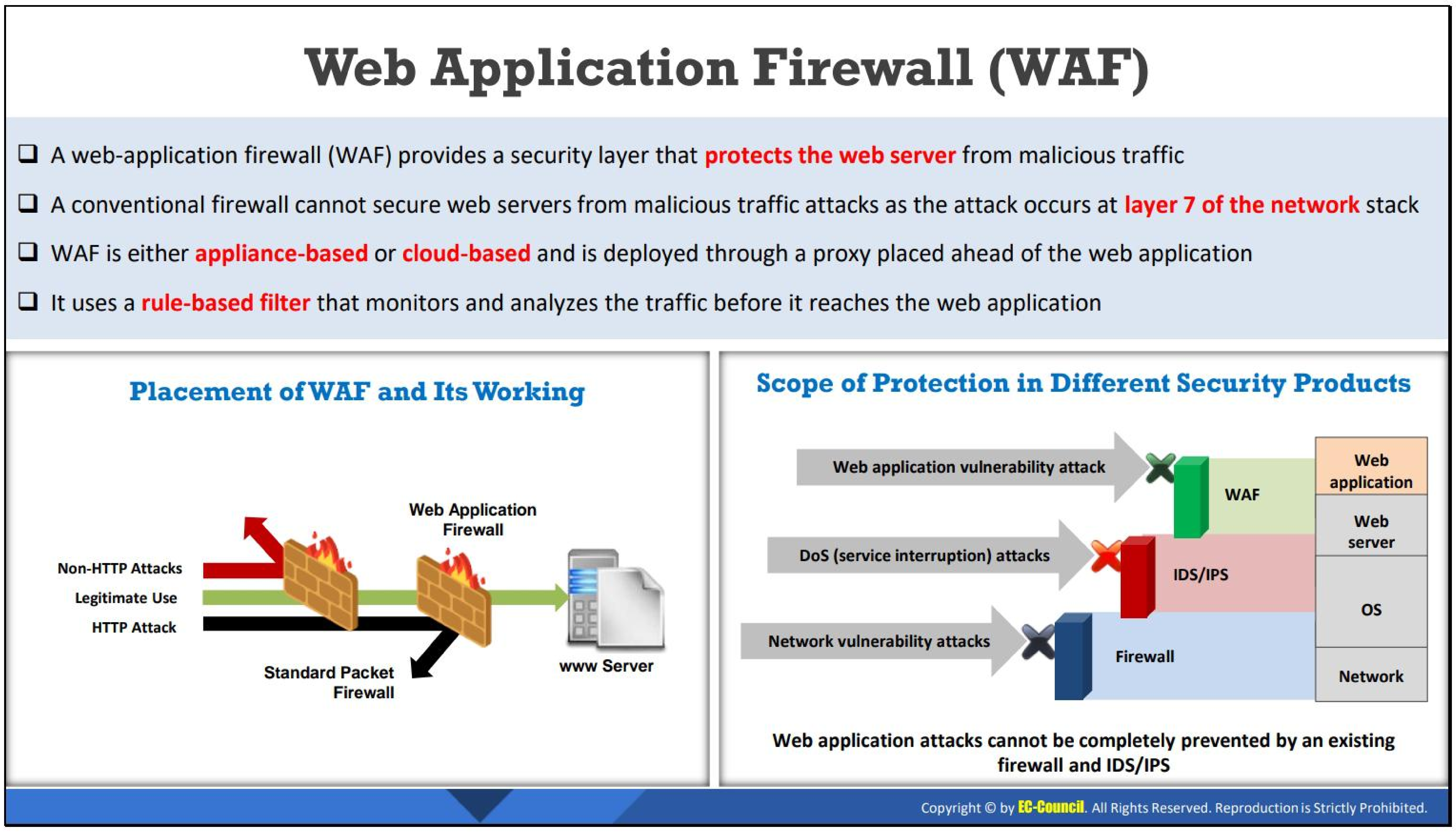
Installed Security Updates

Patch name	Date posted	Severity	Applies to
FOXITR6140217: Foxit Reader 6...	2014-02-20	Critical	Foxit Reader
JAVA8051: Java Runtime Enviro...	2015-07-14	Critical	Java Runtime En
mfsa2015-59, mfsa2015-60, mfs...	2015-06-30	Moderate	Firefox
MS06-061: MSXML 6.0 RTM Sec...	2012-04-04	Critical	SQL Server
MS07-028: Security Update for ...	2007-05-08	Critical	SDK Component
MS09-035: Security Update for ...	2009-09-08	Moderate	Developer Tools,
MS09-035: Security Update for ...	2009-08-11	Moderate	Developer Tools,
MS09-062: Security Update for ...	2009-10-16	Low	Developer Tools,
MS09-062: Security Update for ...	2009-10-16	Low	Developer Tools,
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools,
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools,
MS11-025: Security Update for ...	2012-03-13	Important	Developer Tools,
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools,
MS11-025: Security Update for ...	2012-03-13	Important	Developer Tools,
MS11-049: Security Update for ...	2012-01-24	Important	Developer Tools,

Figure 9.29: Screenshot of GFI LanGuard patch management software

The following are some additional patch management tools:

- Symantec Client Management Suite (*https://www.broadcom.com*)
- Solarwinds Patch Manager (*https://www.solarwinds.com*)
- Kaseya Patch Management (*https://www.kaseya.com*)
- Software Vulnerability Manager (*https://www.flexera.com*)
- Ivanti Patch for Endpoint Manager (*https://www.ivanti.com*)



Web Application Firewall (WAF)

- A web-application firewall (WAF) provides a security layer that **protects the web server** from malicious traffic
- A conventional firewall cannot secure web servers from malicious traffic attacks as the attack occurs at **layer 7 of the network** stack
- WAF is either **appliance-based** or **cloud-based** and is deployed through a proxy placed ahead of the web application
- It uses a **rule-based filter** that monitors and analyzes the traffic before it reaches the web application

Placement of WAF and Its Working

Scope of Protection in Different Security Products

Web application attacks cannot be completely prevented by an existing firewall and IDS/IPS

Web Application Firewall (WAF)

WAF provides a security layer that protects a web server from malicious traffic. A conventional firewall cannot secure web servers from a malicious traffic attack as the attack occurs at layer 7 of the network stack.

WAF is either appliance-based or cloud-based and is deployed through a proxy placed ahead of the web application. It uses a rule-based filter that monitors and analyzes the traffic before it reaches the web application.

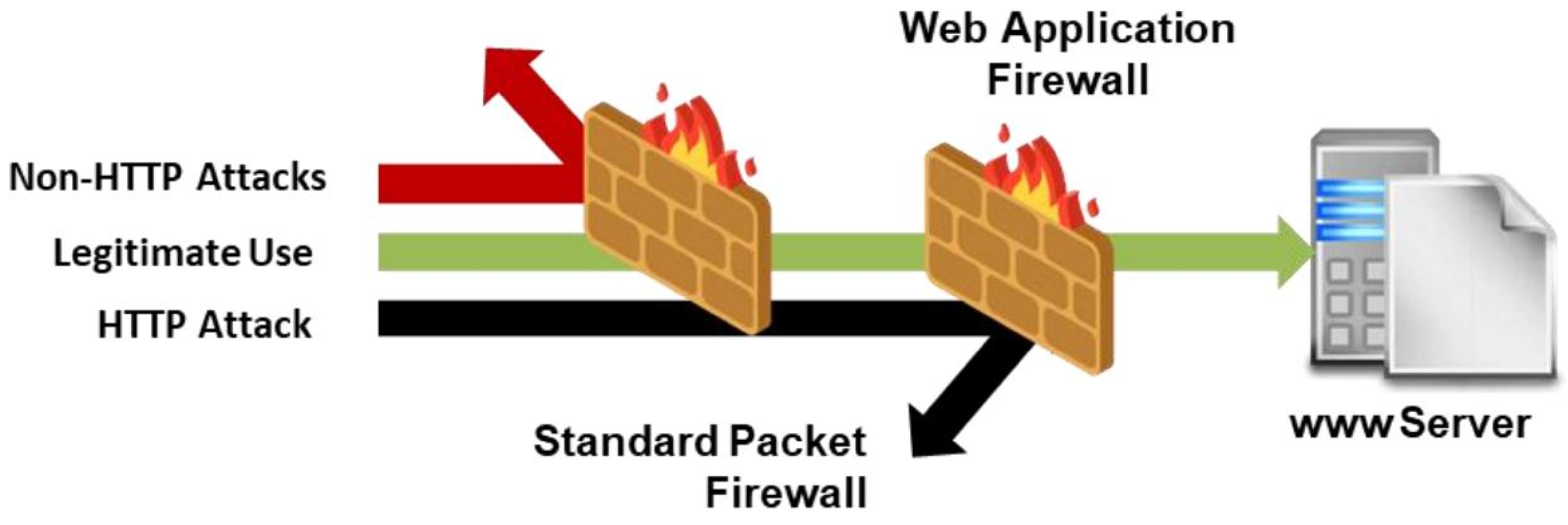
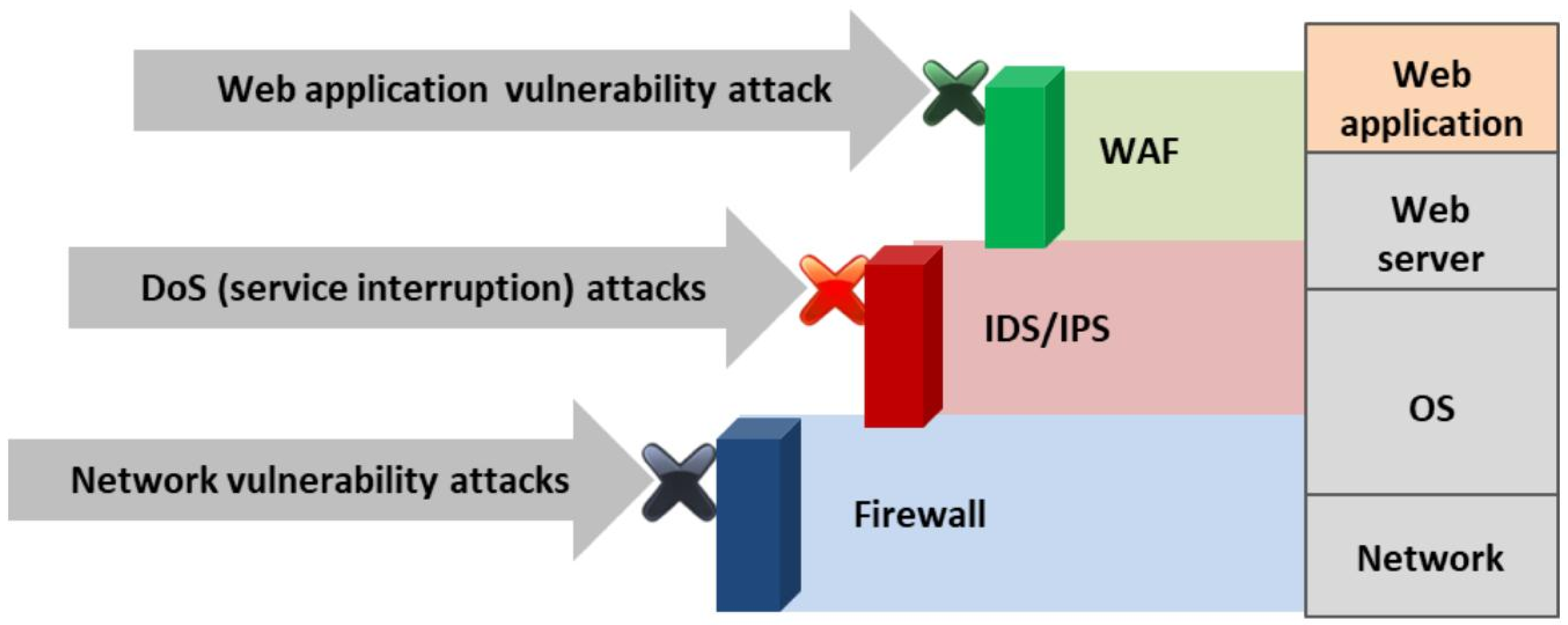


Figure 9.30: Placement of WAF and its working


Web application vulnerability attack
DoS (service interruption) attacks
Network vulnerability attacks
WAF
IDS/IPS
Firewall
Web application
Web server
OS
Network

Web application attacks cannot be completely prevented by an existing firewall and IDS/IPS

Figure 9.31: Working and features of WAF

Benefits of WAF

The benefits of WAF that can help an organization strengthen its web application security from evolving threats include the following:

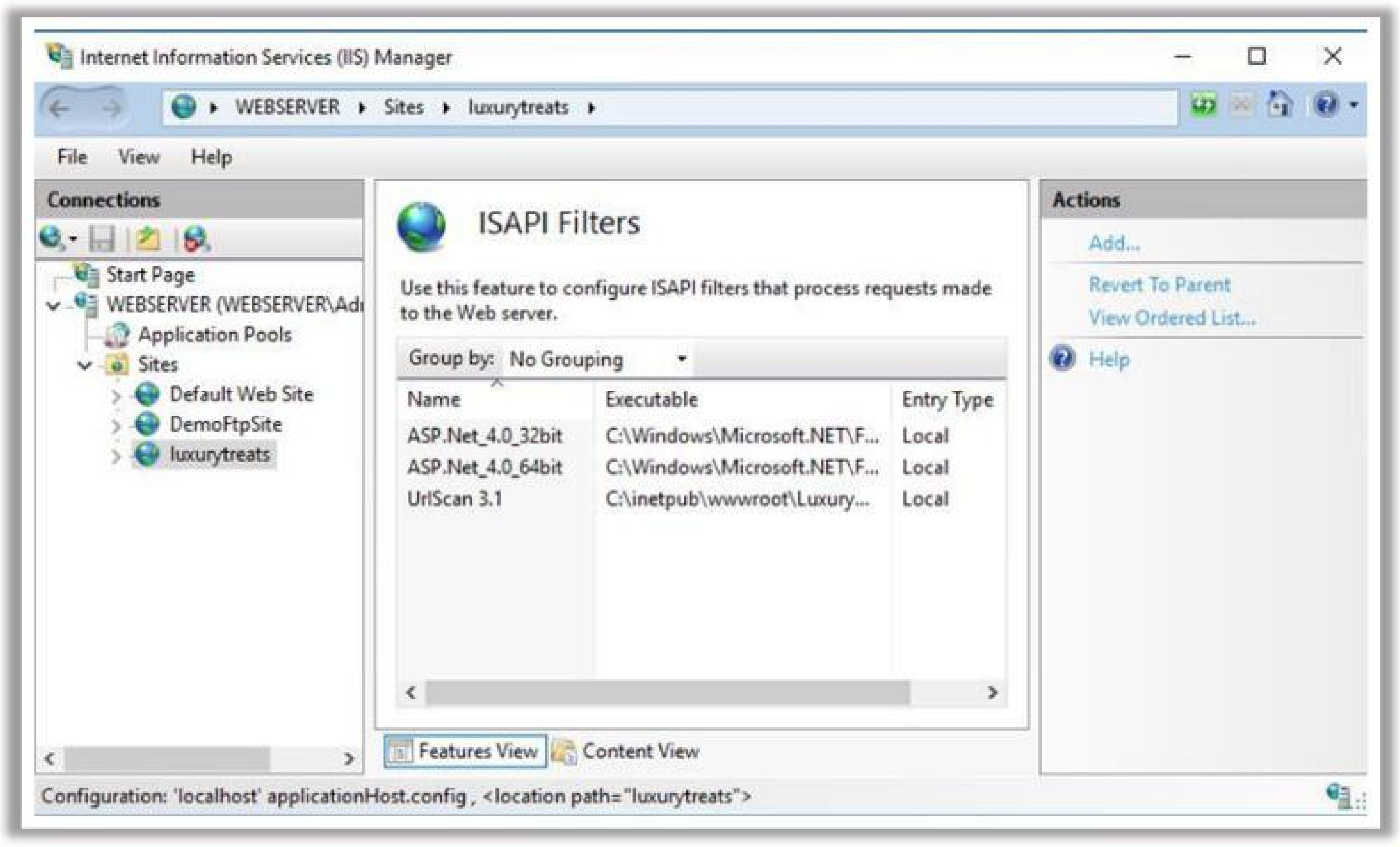
- WAF implementation secures existing and productive web applications.
- Many WAFs have functionalities that can be used in the design process to minimize the workload.
- It provides cookies protection with encryption and signature methodology.
- It secures applications from cross-site request forgery and negates parameter tampering by URL encryption.
- A WAF can detect data-validation issues through the in-depth testing of characters, character length, the range of a value, etc.
- It allows network defender to illustrate compliance with regulatory standards such as Payment Card Industry (PCI), Health Insurance Portability and Accountability Act (HIPAA), and General Data Protection Regulation (GDPR).

Configuring URLScan to Setup as WAF For IIS Server

Microsoft URLScan is a WAF tool that analyzes and **filters all HTTP requests** received by IIS and protects web applications against SQL injection or cross-site scripting XSS attacks

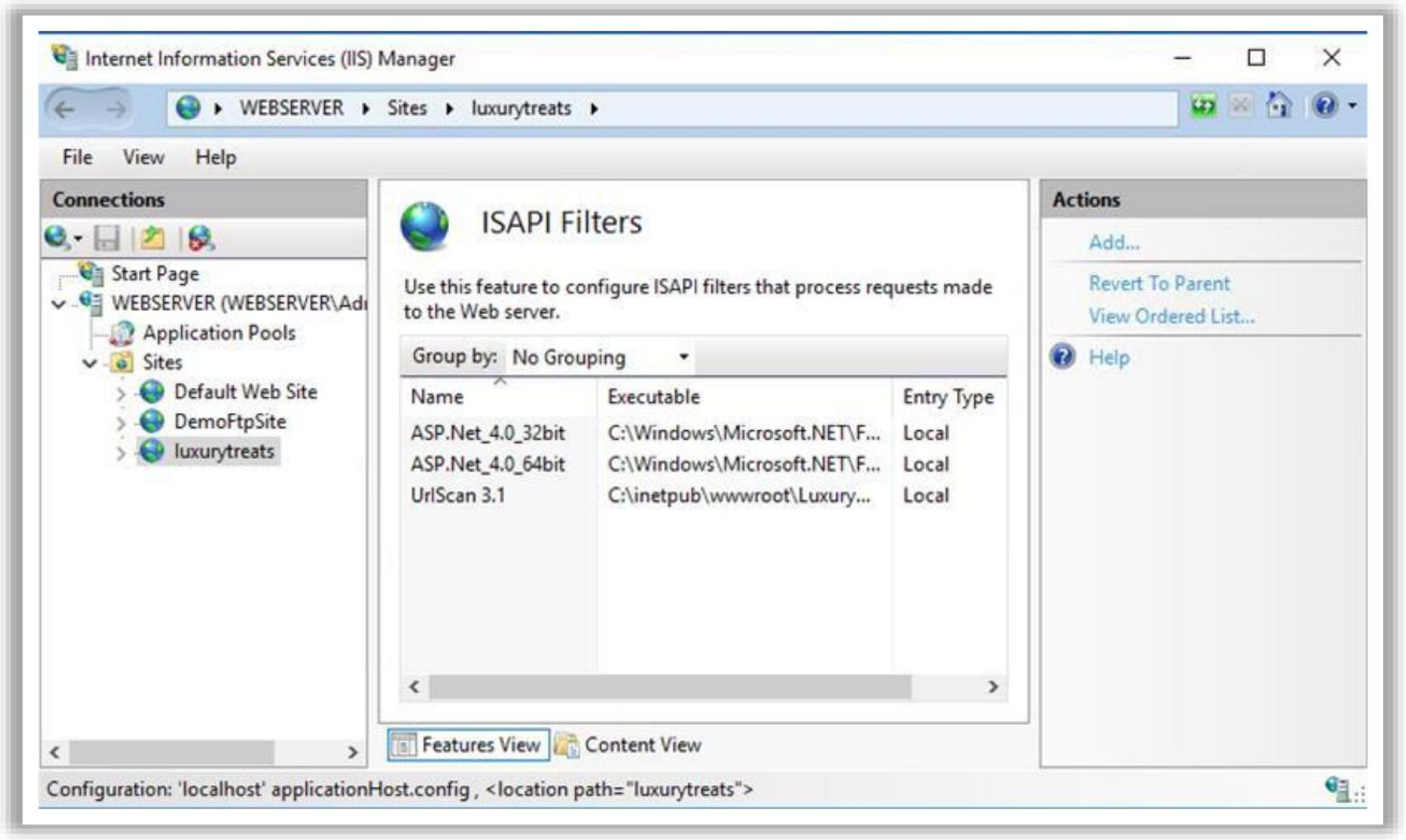
The administrator can **configure the URLScan filter rules** to reject HTTP requests based on following criteria:

- HTTP request method or verb
- File extension of the requested resource
- Suspicious URL encoding
- Presence of non-ASCII characters in the URL
- Presence of specified character sequences in the URL
- Presence of specified headers in the request



Configuring URLScan to Setup as WAF For IIS Server

Microsoft URLScan is a WAF tool that analyzes and filters all Hypertext Transfer Protocol (HTTP) requests received by the Internet Information Service (IIS) web service and protects web applications against Structured Query Language (SQL) injection or cross-site scripting (XSS) attacks. It can log requests to allow the diagnosis of attempts to upset a server. If a request is identified as a risk, the script immediately returns an HTTP 404 message to the client. This mechanism protects the script, website, and server.



Use this feature to configure ISAPI filters that process requests made to the Web server.

Group by: No Grouping

Name	Executable	Entry Type
ASP.Net_4.0_32bit	C:\Windows\Microsoft.NET\F...	Local
ASP.Net_4.0_64bit	C:\Windows\Microsoft.NET\F...	Local
UrlScan 3.1	C:\inetpub\wwwroot\Luxury...	Local

Configuration: 'localhost' applicationHost.config , <location path="luxurytreats">

Figure 9.32: Screenshot of Internet Information Service (IIS) Manager

Its key features include denying rules independently, a global DenyQueryString section for adding deny rules, a global AlwaysAllowedUrls section to specify safe query strings, the use of escape sequences, installation of multiple URLScan instances, propagating configuration change notifications to IIS worker processes, and enhanced W3C formatted logging.

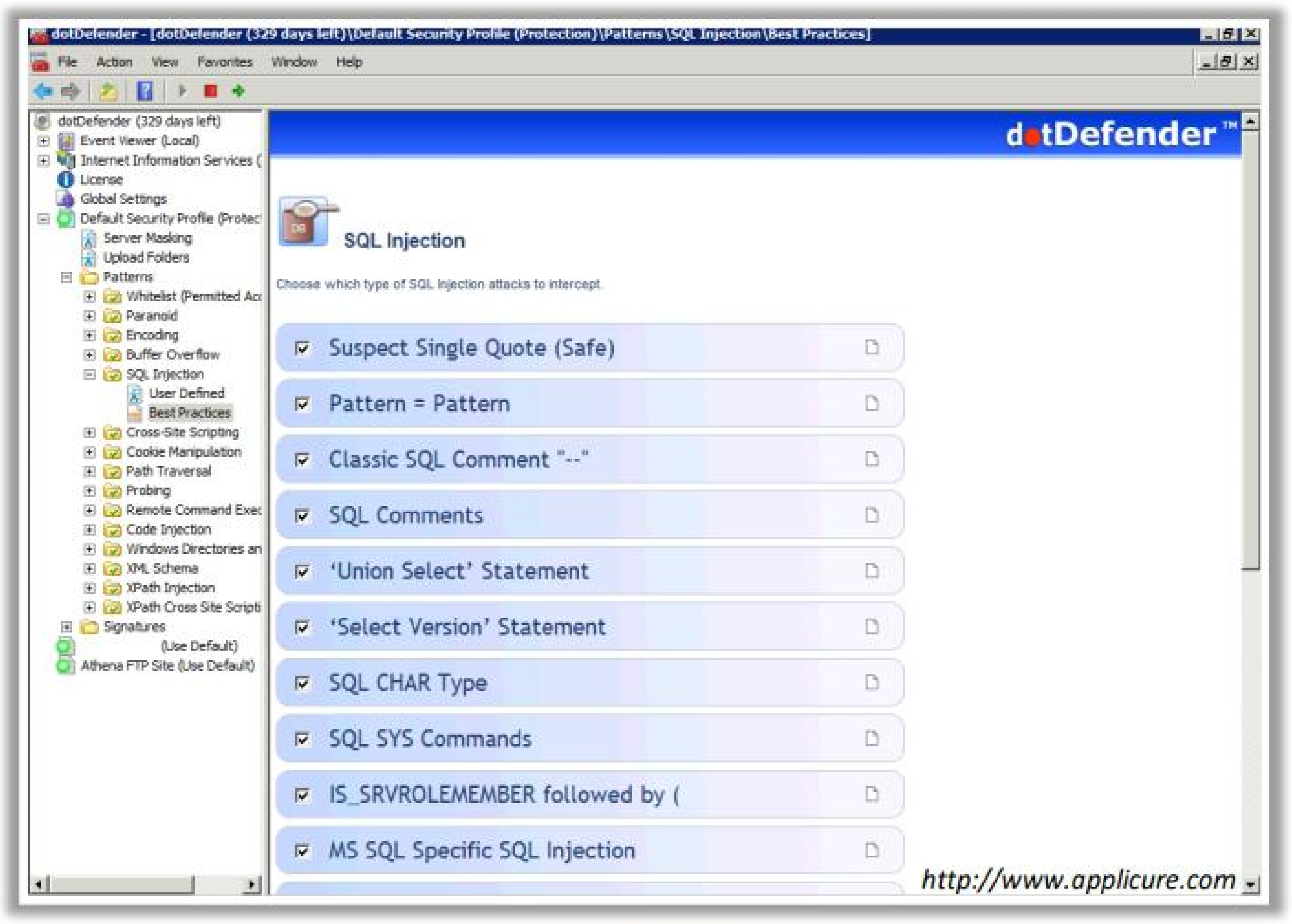
An administrator can configure URLScan filter rules to reject HTTP requests based on the following criteria:

- HTTP request method or verb
- File extension of the requested resource
- Suspicious URL encoding
- Presence of non-ASCII characters in the URL
- Presence of specified character sequences in the URL
- Presence of specified headers in the request

Additional WAF Solutions

dotDefender

- dotDefender **protects websites** from malicious attacks such as SQL injection, path traversal, cross-site scripting, and others that result in **website defacement**
- It inspects **HTTP/HTTPS traffic** for suspicious behavior



SQL Injection

Choose which type of SQL injection attacks to intercept

- ☑ Suspect Single Quote (Safe)
- ☑ Pattern = Pattern
- ☑ Classic SQL Comment "--"
- ☑ SQL Comments
- ☑ 'Union Select' Statement
- ☑ 'Select Version' Statement
- ☑ SQL CHAR Type
- ☑ SQL SYS Commands
- ☑ IS_SRVROLEMEMBER followed by (
- ☑ MS SQL Specific SQL Injection

http://www.applicure.com

- **ServerDefender VP**
  https://www.iis.net
- **ModSecurity**
  https://github.com
- **Radware's AppWall**
  https://www.radware.com
- **Qualys WAF**
  https://www.qualys.com
- **Barracuda Web Application Firewall**
  https://www.barracuda.com

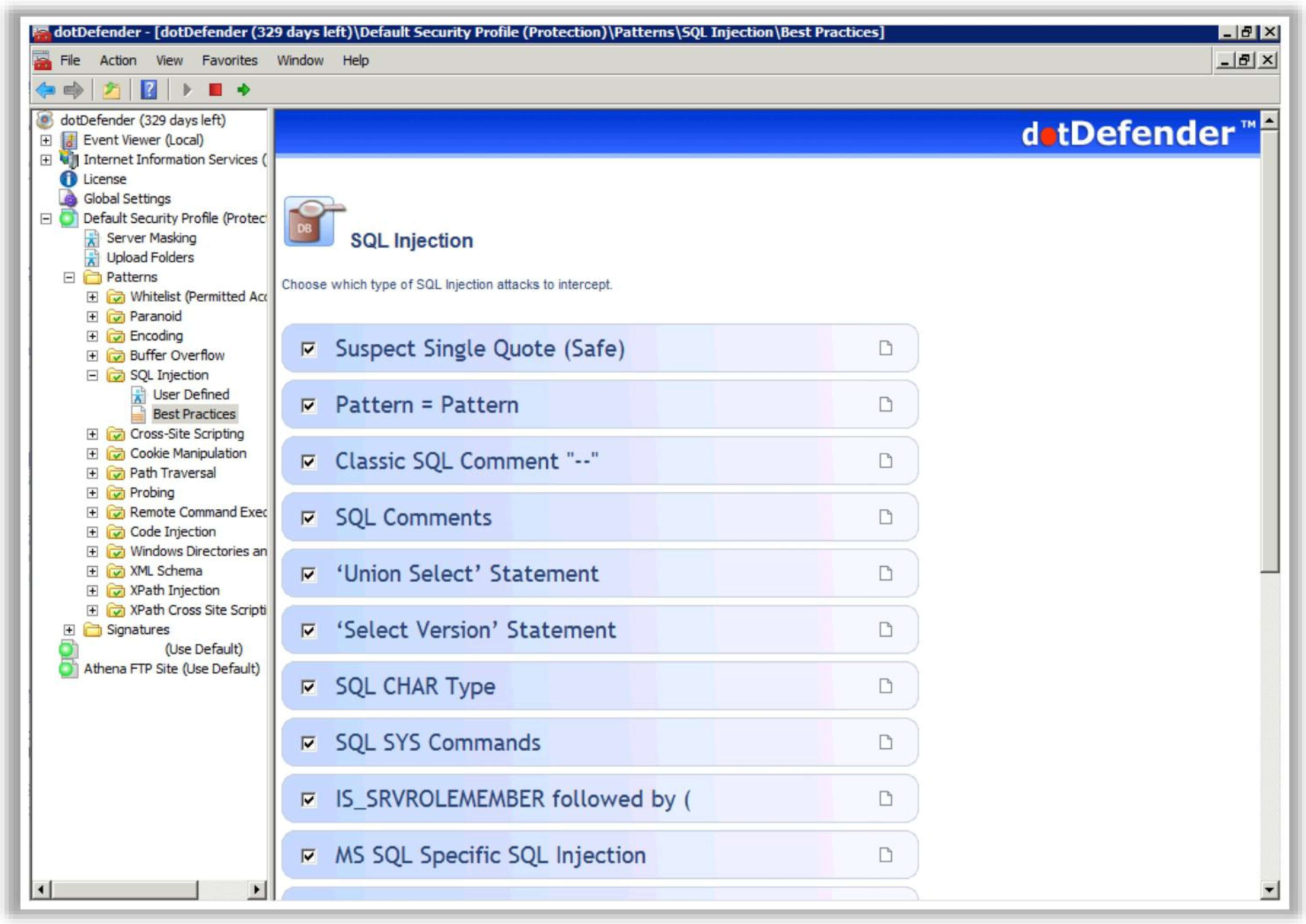
Additional WAF Solutions

Web application firewalls (WAFs) secure websites, web applications, and web services against known and unknown attacks. They prevent data theft and manipulation of sensitive corporate and customer information. Some of the most commonly used WAFs are as follows:

- **dotDefender**

  Source: *http://www.applicure.com*

  dotDefender™ is a software-based WAF that protects your website from malicious attacks such as SQL injection, path traversal, cross-site scripting, and others that result in website defacement. It complements the network firewall, IPS, and other network-based Internet security products. It inspects HTTP/HTTPS traffic for suspicious behavior.



SQL Injection

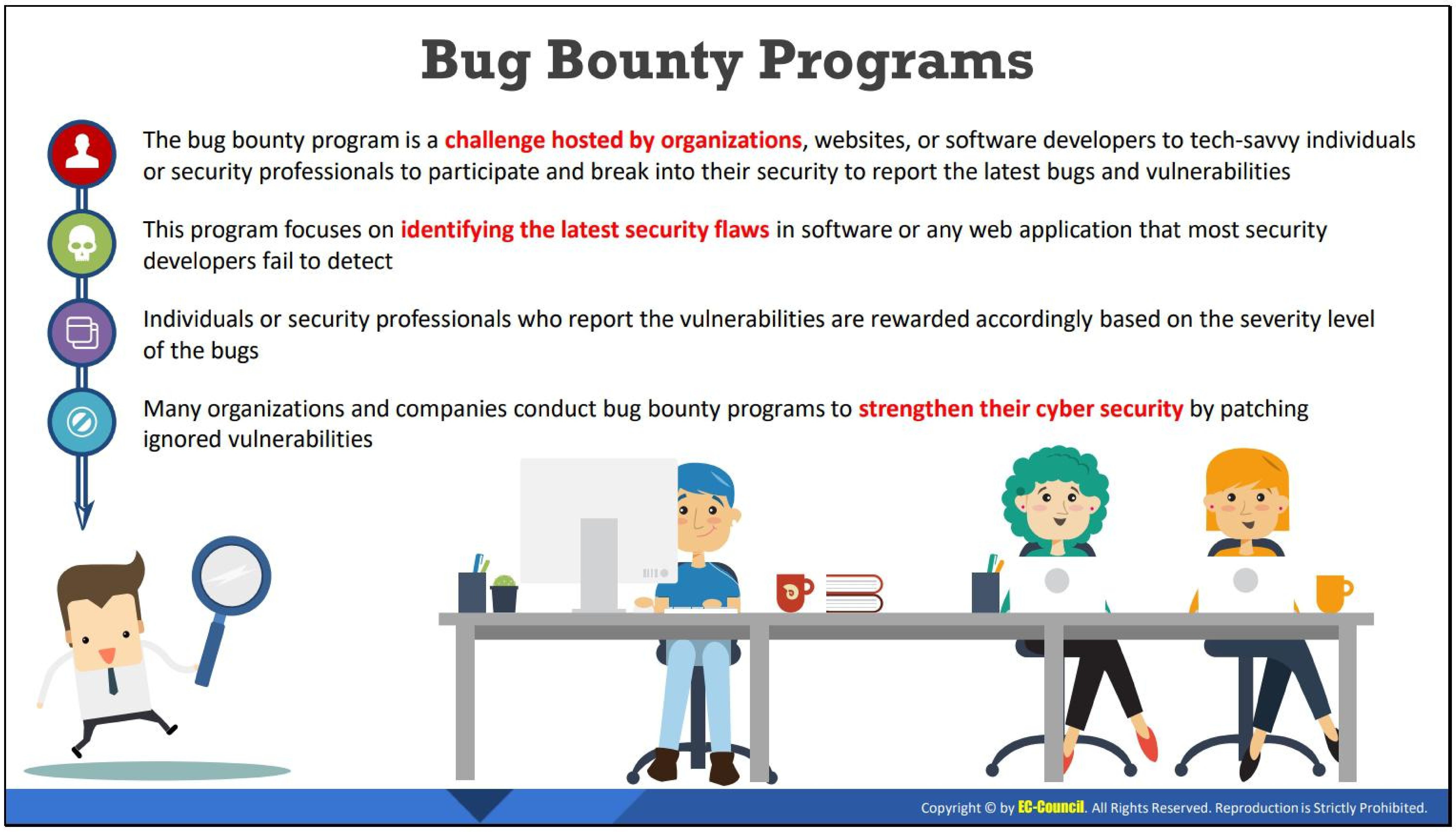
Choose which type of SQL Injection attacks to intercept.

- ☑ Suspect Single Quote (Safe)
- ☑ Pattern = Pattern
- ☑ Classic SQL Comment "--"
- ☑ SQL Comments
- ☑ 'Union Select' Statement
- ☑ 'Select Version' Statement
- ☑ SQL CHAR Type
- ☑ SQL SYS Commands
- ☑ IS_SRVROLEMEMBER followed by (
- ☑ MS SQL Specific SQL Injection

Figure 9.33: Screenshot of dotDefender web application firewall

Some additional web application firewalls are as follows:

- ServerDefender VP (*https://www.iis.net*)
- ModSecurity (*https://github.com*)
- Radware's AppWall (*https://www.radware.com*)
- Qualys WAF (*https://www.qualys.com*)
- Barracuda Web Application Firewall (*https://www.barracuda.com*)



Bug Bounty Programs

- The bug bounty program is a **challenge hosted by organizations**, websites, or software developers to tech-savvy individuals or security professionals to participate and break into their security to report the latest bugs and vulnerabilities
- This program focuses on **identifying the latest security flaws** in software or any web application that most security developers fail to detect
- Individuals or security professionals who report the vulnerabilities are rewarded accordingly based on the severity level of the bugs
- Many organizations and companies conduct bug bounty programs to **strengthen their cyber security** by patching ignored vulnerabilities

Bug Bounty Programs

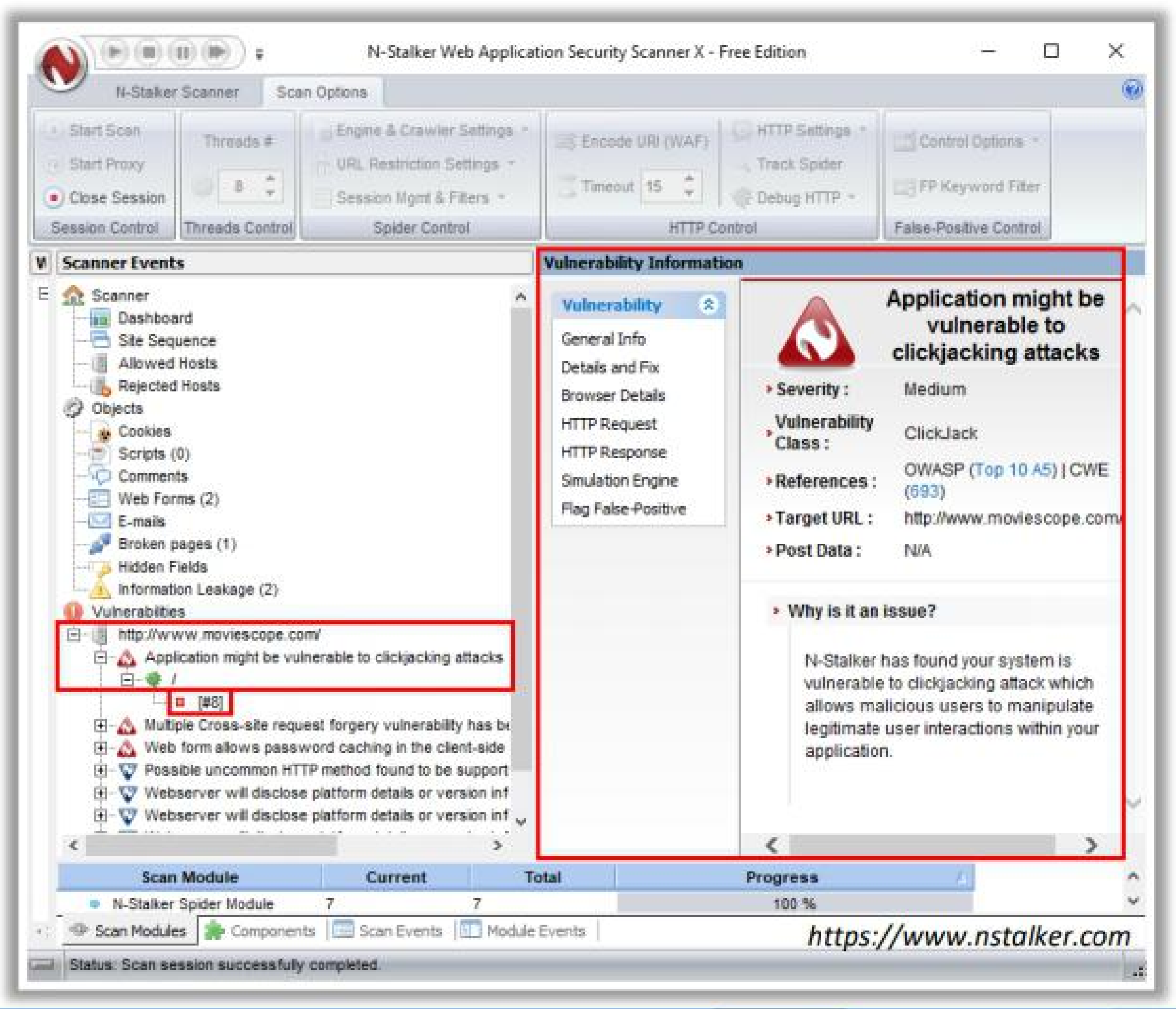
A bug bounty program is a challenge or agreement hosted by organizations, websites, or software developers for tech-savvy individuals or security professionals to participate and break into their security to report the latest bugs and vulnerabilities. This program focuses on identifying the latest security flaws in the software or any web application that most security developers fail to detect and which may hence pose a great threat. Therefore, individuals or security professionals who report the vulnerabilities are rewarded accordingly based on the severity of the bugs. Thus, any threat or flaw that evades the developer can be mitigated before it paves the way to sophisticated cyber-attacks. Many white-hat hackers contribute to this program as part of a comprehensive vulnerability disclosure framework and get rewarded for their work.

Many organizations benefit from such programs, as they need to maintain a keen watch on their system security and identify ignored vulnerabilities. Most of the latest bugs that are not detected by legacy security testing techniques and software tools can be exploited, resulting in major data loss. Such programs can also help organizations to avoid loss of money and reputation in the case of a data breach, as offering rewards through the bug bounty program is more economical. Therefore, most of the large companies use this program for strengthening their security, which in turn enhances websites and programs.

Web Application Security Scanners

N-Stalker Web App Security Scanner

N-Stalker web app security scanner checks for vulnerabilities such as **SQL injection**, **XSS**, and other known attacks



http://www.moviescope.com/
Application might be vulnerable to clickjacking attacks
/
[#8]
Multiple Cross-site request forgery vulnerability has be
Web form allows password caching in the client-side
Possible uncommon HTTP method found to be support
Webserver will disclose platform details or version inf
Webserver will disclose platform details or version inf

Application might be vulnerable to clickjacking attacks

Severity :	Medium
Vulnerability Class :	ClickJack
References :	OWASP (Top 10 A5) | CWE (693)
Target URL :	http://www.moviescope.com/
Post Data :	N/A

Why is it an issue?

N-Stalker has found your system is vulnerable to clickjacking attack which allows malicious users to manipulate legitimate user interactions within your application.

Scan Module	Current	Total	Progress
N-Stalker Spider Module	7	7	100 %

Status: Scan session successfully completed.

https://www.nstalker.com

Acunetix WVS
https://www.acunetix.com

Browser Exploitation Framework (BeEF)
http://beefproject.com

Metasploit
https://www.metasploit.com

PowerSploit
https://github.com

Watcher
https://www.casaba.com

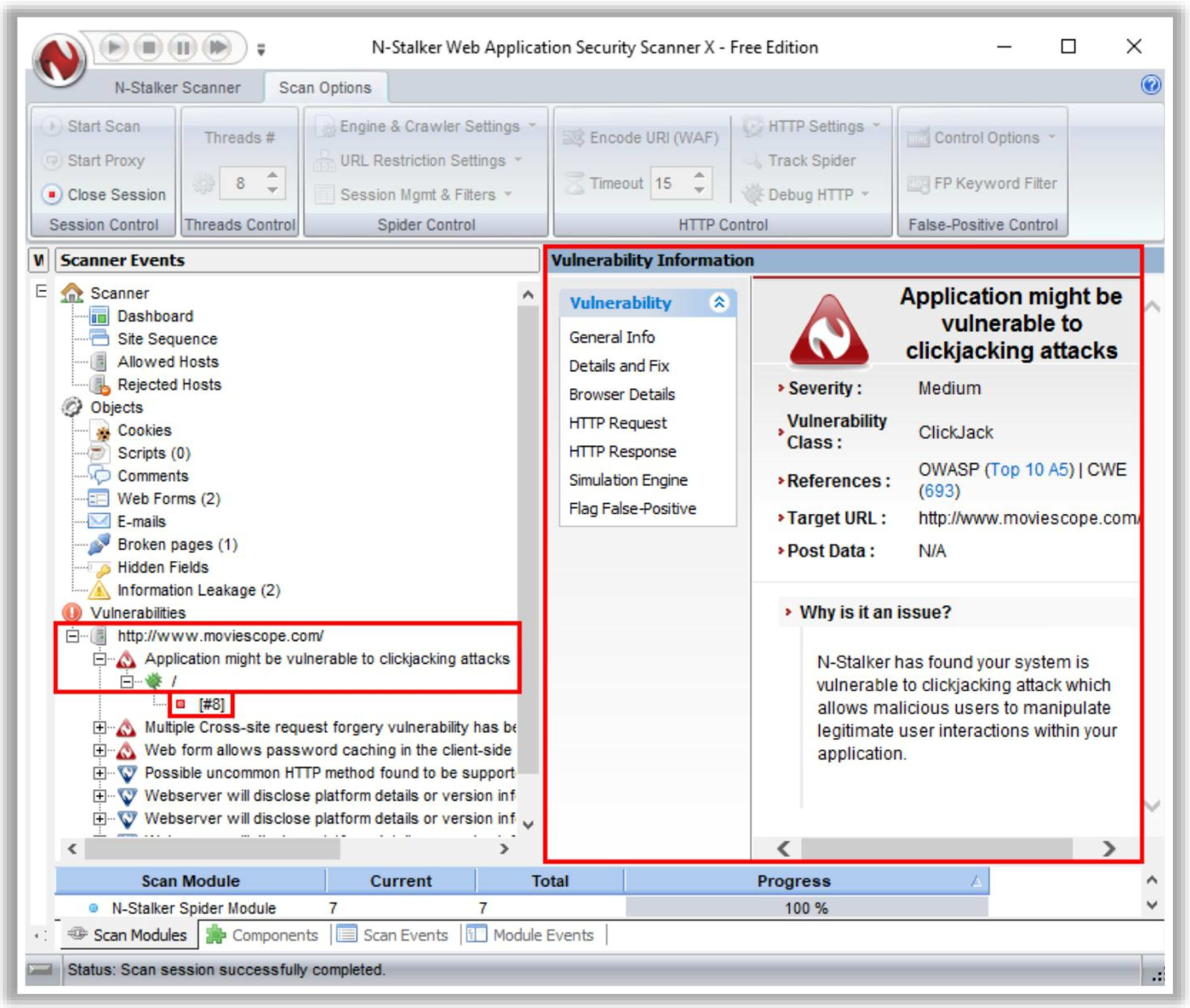
Web Application Security Scanners

There are various web application security assessment tools available for scanning, detecting, and assessing the vulnerabilities/security of web applications. These tools reveal their security posture; you can use them to find ways to harden security and create robust web applications. Furthermore, these tools automate the process of accurate web application security assessment.

- **N-Stalker Web App Security Scanner**

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

  N-Stalker Web App Security Scanner checks for vulnerabilities such as SQL injection, XSS, and other known attacks. It is a useful security tool for developers, system/security administrators, IT auditors, and staff, as it incorporates the well-known "N-Stealth HTTP Security Scanner" and its database of 39,000 web attack signatures along with a component-oriented web application security assessment technology.



Severity : Medium
Vulnerability Class : ClickJack
References : OWASP (Top 10 A5) | CWE (693)
Target URL : http://www.moviescope.com/
Post Data : N/A

Why is it an issue?

N-Stalker has found your system is vulnerable to clickjacking attack which allows malicious users to manipulate legitimate user interactions within your application.

Status: Scan session successfully completed.

Figure 9.34: Screenshot of N-Stalker Web Application Security Scanner

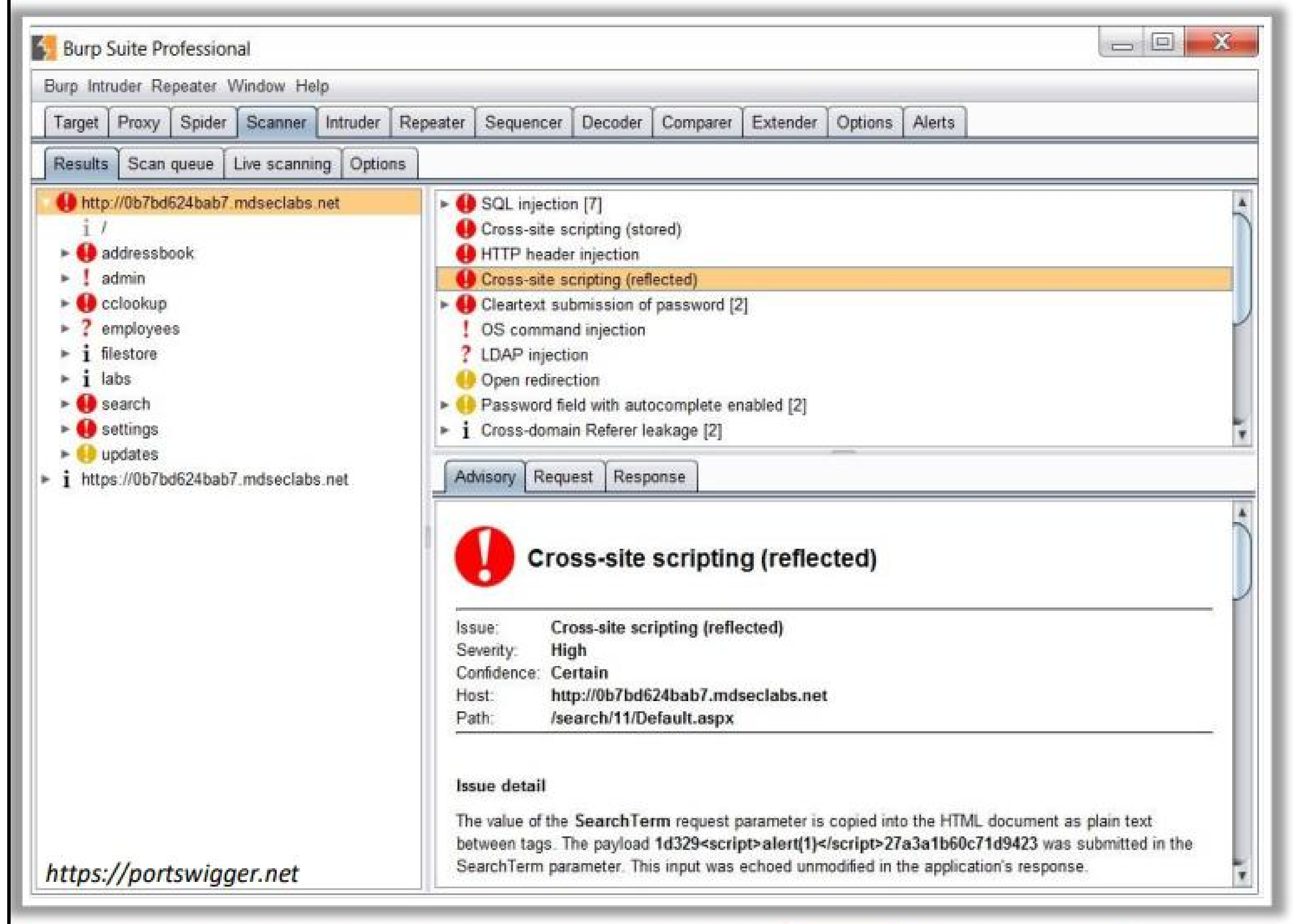
Some additional web application security testing tools are as follows:

- Acunetix WVS (*https://www.acunetix.com*)
- Browser Exploitation Framework (BeEF) (*http://beefproject.com*)
- Metasploit (*https://www.metasploit.com*)
- PowerSploit (*https://github.com*)
- Watcher (*https://www.casaba.com*)

Proxy-based Security Testing Tools

Burp Suite

It provides various tools that work together to support the entire testing process, from initial mapping and analysis of an **application's attack surface**, through to finding and exploiting security vulnerabilities


Burp Suite Professional
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Results Scan queue Live scanning Options
http://0b7bd624bab7.mdseclabs.net
/
addressbook
admin
cclookup
employees
filestore
labs
search
settings
updates
https://0b7bd624bab7.mdseclabs.net
SQL injection [7]
Cross-site scripting (stored)
HTTP header injection
Cross-site scripting (reflected)
Cleartext submission of password [2]
OS command injection
LDAP injection
Open redirection
Password field with autocomplete enabled [2]
Cross-domain Referer leakage [2]
Advisory Request Response

Cross-site scripting (reflected)

Issue:	Cross-site scripting (reflected)
Severity:	High
Confidence:	Certain
Host:	http://0b7bd624bab7.mdseclabs.net
Path:	/search/11/Default.aspx

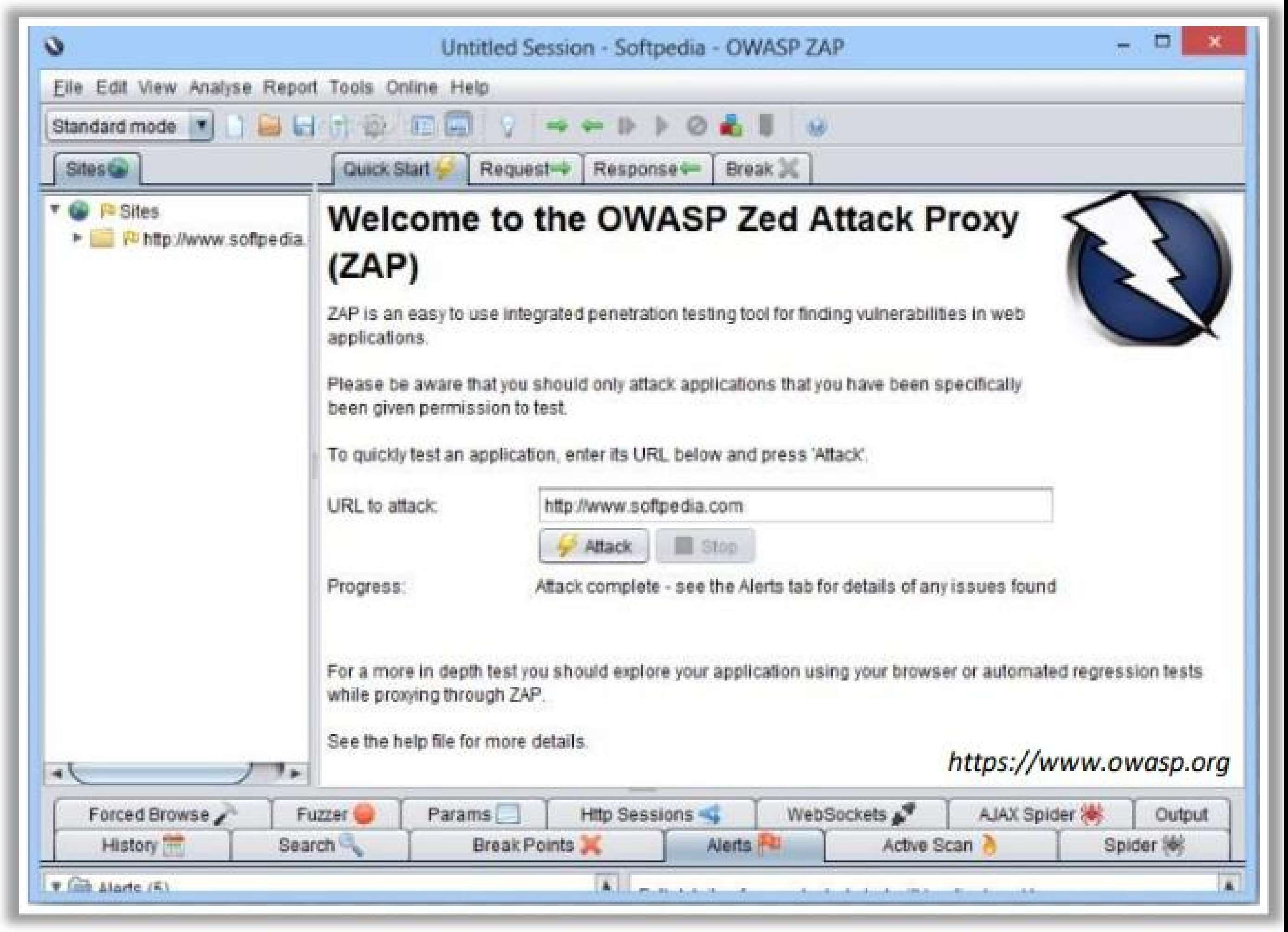
Issue detail

The value of the **SearchTerm** request parameter is copied into the HTML document as plain text between tags. The payload **1d329<script>alert(1)</script>27a3a1b60c71d9423** was submitted in the SearchTerm parameter. This input was echoed unmodified in the application's response.

https://portswigger.net

OWASP Zed Attack Proxy (ZAP)

OWASP ZAP is an open source, easy to use, integrated penetration testing tool for **finding vulnerabilities** in web applications


Untitled Session - Softpedia - OWASP ZAP
File Edit View Analyse Report Tools Online Help
Standard mode
Sites
Quick Start Request Response Break
Sites
http://www.softpedia.
Attack Stop
Forced Browse Fuzzer Params Http Sessions WebSockets AJAX Spider Output
History Search Break Points Alerts Active Scan Spider
Alerts (5)

Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://www.softpedia.com

Progress: Attack complete - see the Alerts tab for details of any issues found

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

See the help file for more details.

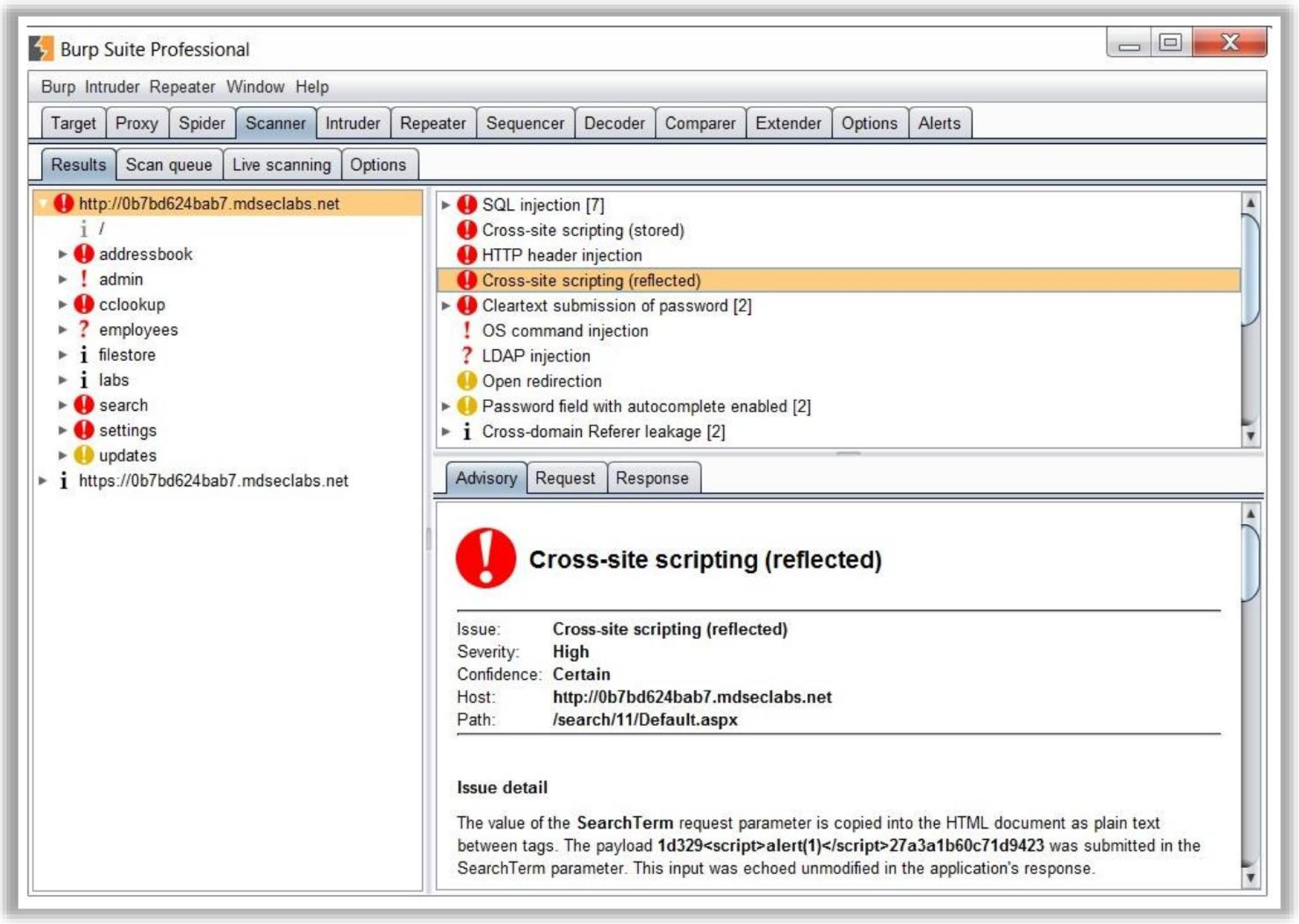
https://www.owasp.org

Proxy-based Security Testing Tools

- **Burp Suite**

  Source: *https://portswigger.net*

  Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities. Install and configure Burp Suite in the browser of your interest.


Burp Suite Professional
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Results Scan queue Live scanning Options
http://0b7bd624bab7.mdseclabs.net
/
addressbook
admin
cclookup
employees
filestore
labs
search
settings
updates
https://0b7bd624bab7.mdseclabs.net
SQL injection [7]
Cross-site scripting (stored)
HTTP header injection
Cross-site scripting (reflected)
Cleartext submission of password [2]
OS command injection
LDAP injection
Open redirection
Password field with autocomplete enabled [2]
Cross-domain Referer leakage [2]
Advisory Request Response

Cross-site scripting (reflected)

Issue:	**Cross-site scripting (reflected)**
Severity:	**High**
Confidence:	**Certain**
Host:	**http://0b7bd624bab7.mdseclabs.net**
Path:	**/search/11/Default.aspx**

Issue detail

The value of the **SearchTerm** request parameter is copied into the HTML document as plain text between tags. The payload **1d329<script>alert(1)</script>27a3a1b60c71d9423** was submitted in the SearchTerm parameter. This input was echoed unmodified in the application's response.

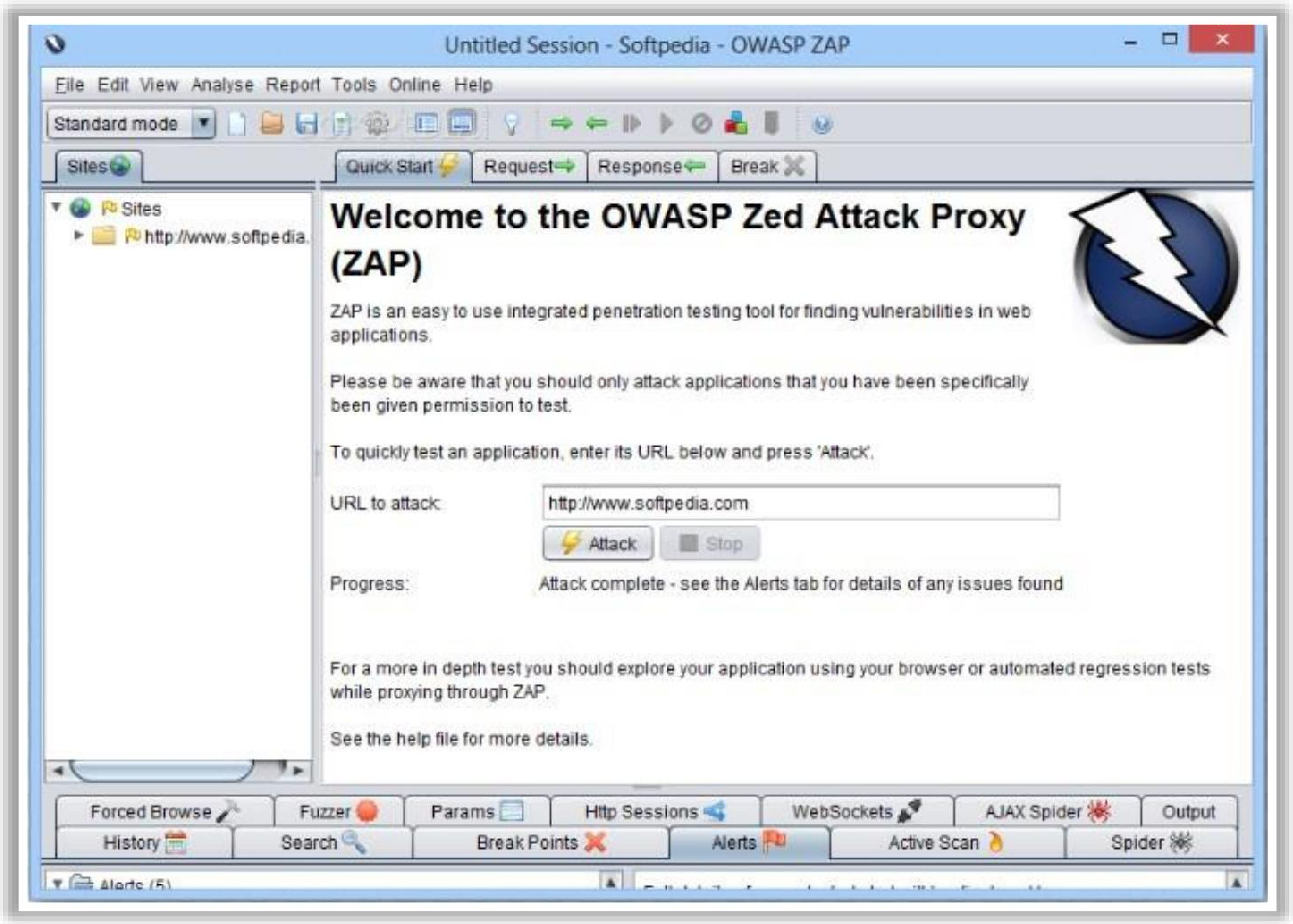
Figure 9.35: Screenshot of Burp Suite

- **OWASP Zed Attack Proxy (ZAP)**

  Source: *https://www.owasp.org*

  OWASP ZAP is an open source, easy to use, integrated penetration testing tool for finding vulnerabilities in web applications. Perform DAST using OWASP ZAP:

  - Install and configure ZAP in the browser of your interest
  - Insert the URL of web application that you want to test and start the attack



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

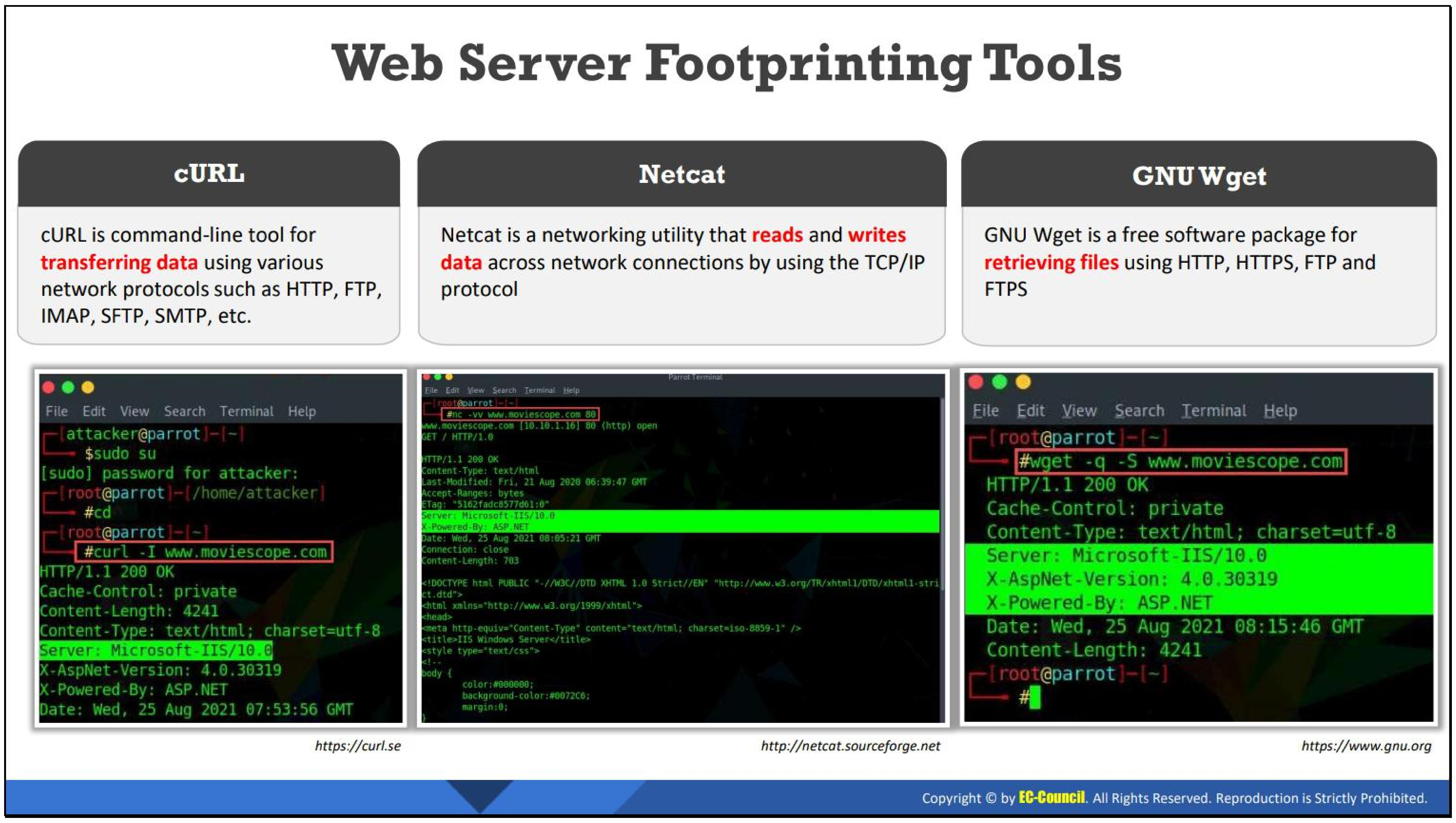
URL to attack: http://www.softpedia.com

Progress: Attack complete - see the Alerts tab for details of any issues found

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

See the help file for more details.

Figure 9.36: Screenshot of OWASP Zed Attack Proxy (ZAP)



Web Server Footprinting Tools

cURL

cURL is command-line tool for **transferring data** using various network protocols such as HTTP, FTP, IMAP, SFTP, SMTP, etc.

Netcat

Netcat is a networking utility that **reads** and **writes data** across network connections by using the TCP/IP protocol

GNU Wget

GNU Wget is a free software package for **retrieving files** using HTTP, HTTPS, FTP and FTPS

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #curl -I www.moviescope.com
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 4241
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 25 Aug 2021 07:53:56 GMT
```

https://curl.se

```
┌─[root@parrot]─[~]
└──╼ #nc -vv www.moviescope.com 80
www.moviescope.com [10.10.1.16] 80 (http) open
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Fri, 21 Aug 2020 06:39:47 GMT
Accept-Ranges: bytes
ETag: "5162fadc8577d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 25 Aug 2021 08:05:21 GMT
Connection: close
Content-Length: 703

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-stri
ct.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS Windows Server</title>
<style type="text/css">
<!--
body {
        color:#000000;
        background-color:#0072C6;
        margin:0;
}
```

http://netcat.sourceforge.net

```
┌─[root@parrot]─[~]
└──╼ #wget -q -S www.moviescope.com
  HTTP/1.1 200 OK
  Cache-Control: private
  Content-Type: text/html; charset=utf-8
  Server: Microsoft-IIS/10.0
  X-AspNet-Version: 4.0.30319
  X-Powered-By: ASP.NET
  Date: Wed, 25 Aug 2021 08:15:46 GMT
  Content-Length: 4241
┌─[root@parrot]─[~]
└──╼ #
```

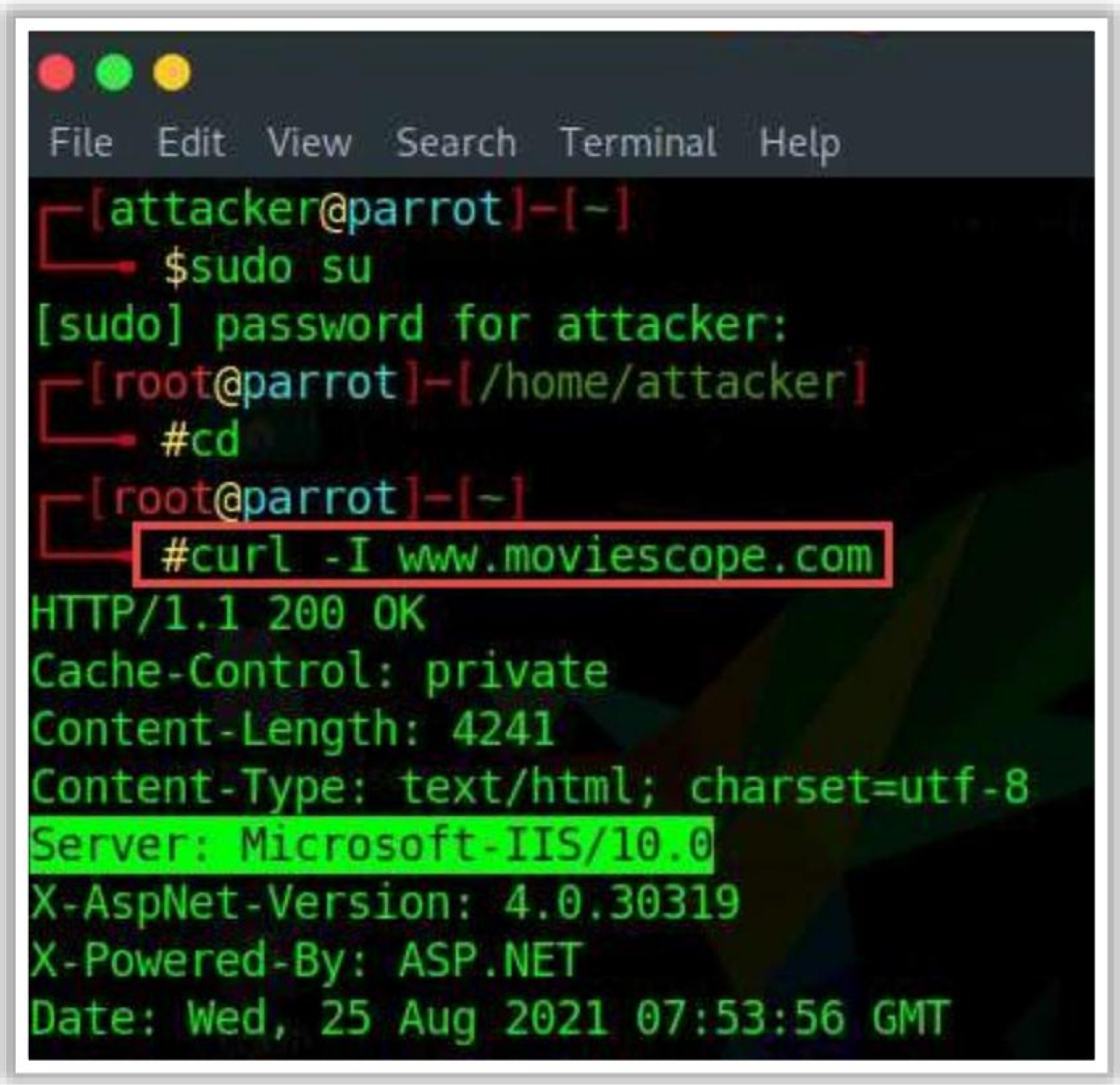
https://www.gnu.org

Web Server Footprinting Tools

- **cURL**

  Source: *https://curl.se*

  cURL is command-line tool for transferring data using various network protocols such as HTTP, FTP, IMAP, SFTP, SMTP, etc.



```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #curl -I www.moviescope.com
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 4241
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 25 Aug 2021 07:53:56 GMT
```

Figure 9.37: Screenshot of cURL

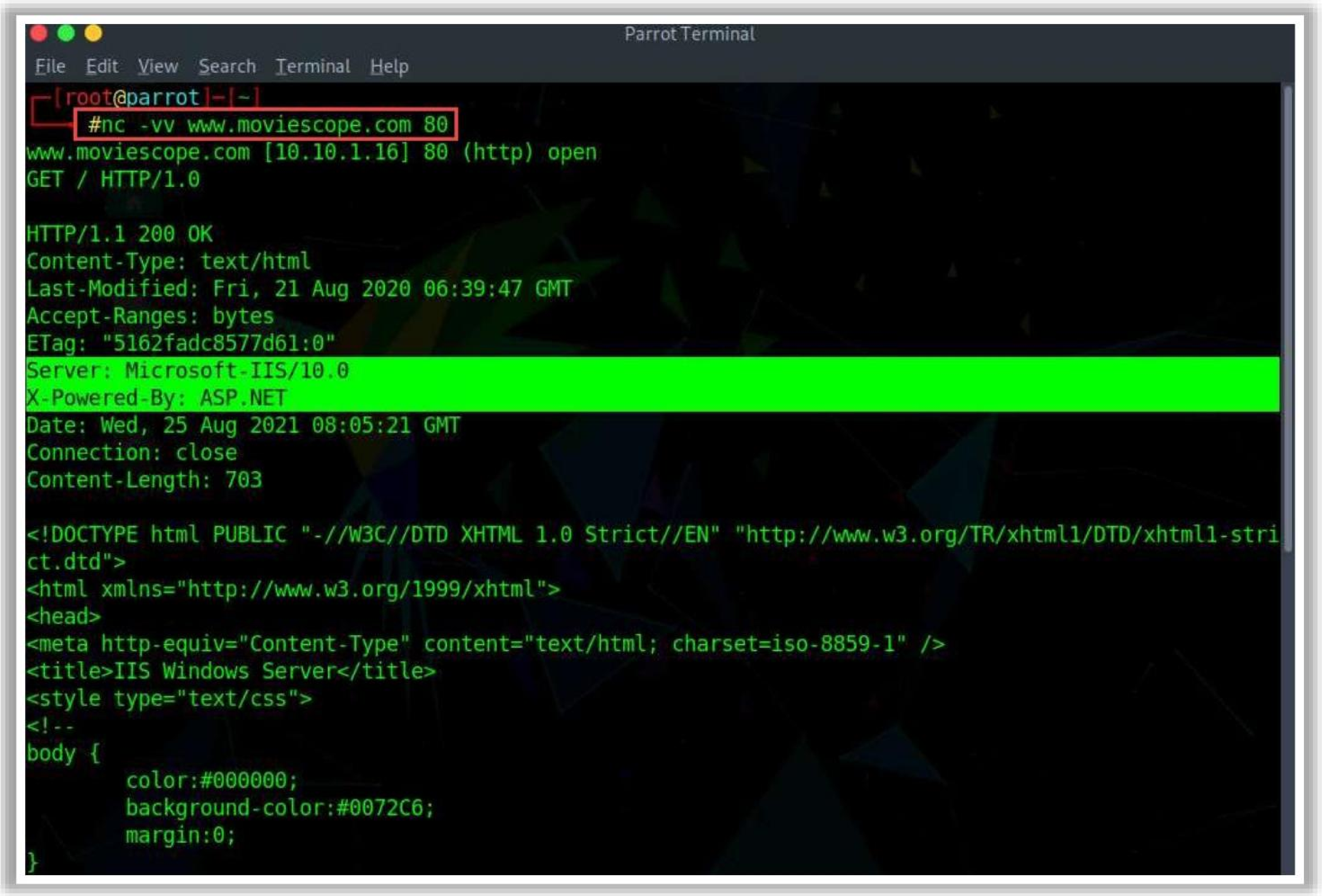
- **Netcat**

  Source: *http://netcat.sourceforge.net*

  Netcat is a networking utility that reads and writes data across network connections by using the TCP/IP protocol. It is a reliable "back-end" tool used directly or driven by other programs and scripts. It is also a network debugging and exploration tool.

  The following are the commands used to perform banner grabbing for www.moviescope.com as an example to gather information such as server type and version.

  - **# nc -vv www. moviescope.com 80** – press [**Enter**]
  - **GET / HTTP/1.0** - press [**Enter**] twice



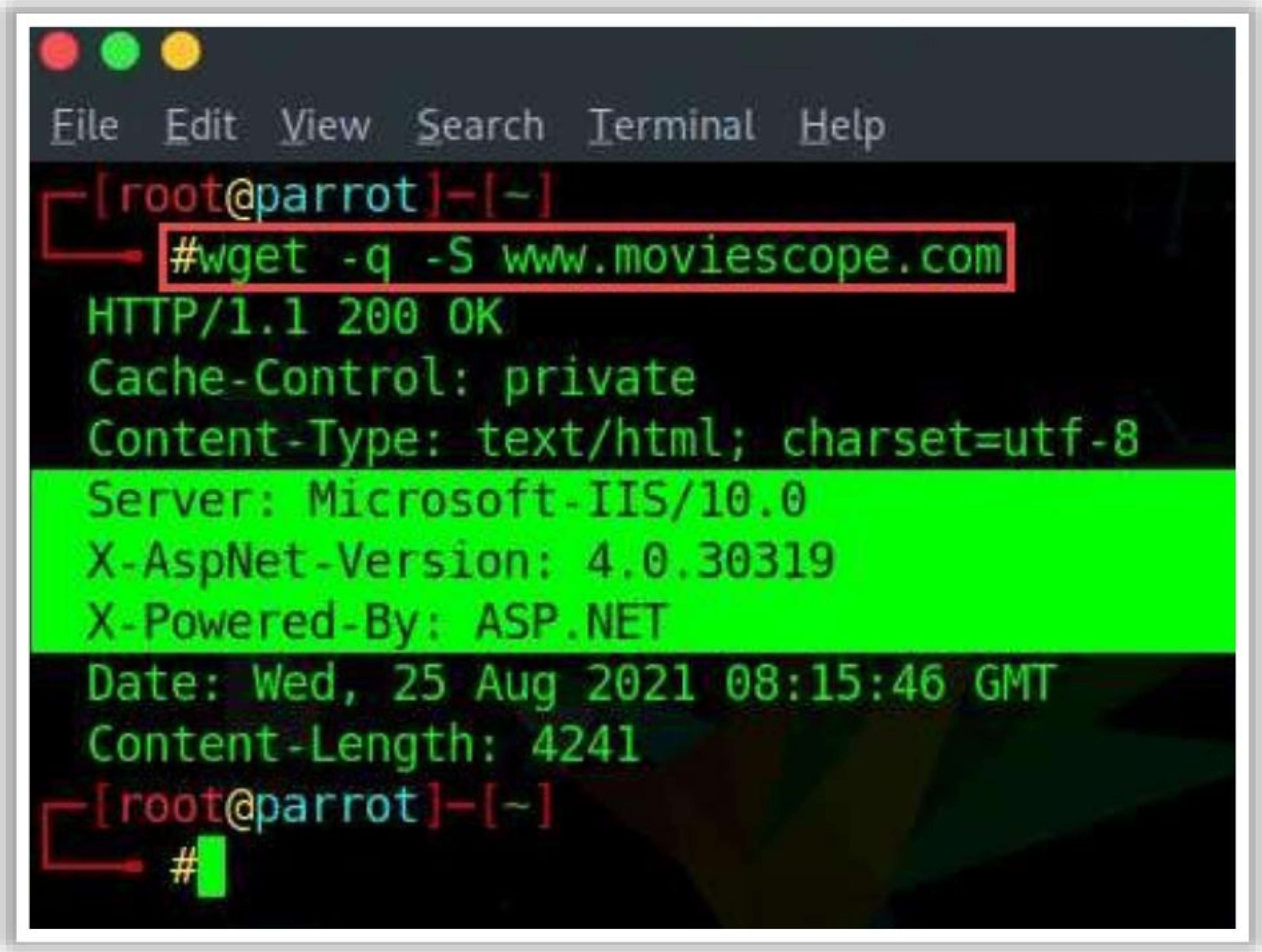
```
┌─[root@parrot]─[~]
└──╼ #nc -vv www.moviescope.com 80
www.moviescope.com [10.10.1.16] 80 (http) open
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Fri, 21 Aug 2020 06:39:47 GMT
Accept-Ranges: bytes
ETag: "5162fadc8577d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 25 Aug 2021 08:05:21 GMT
Connection: close
Content-Length: 703

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-stri
ct.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS Windows Server</title>
<style type="text/css">
<!--
body {
        color:#000000;
        background-color:#0072C6;
        margin:0;
}
```

Figure 9.38: Screenshot of Netcat

- **GNU Wget**

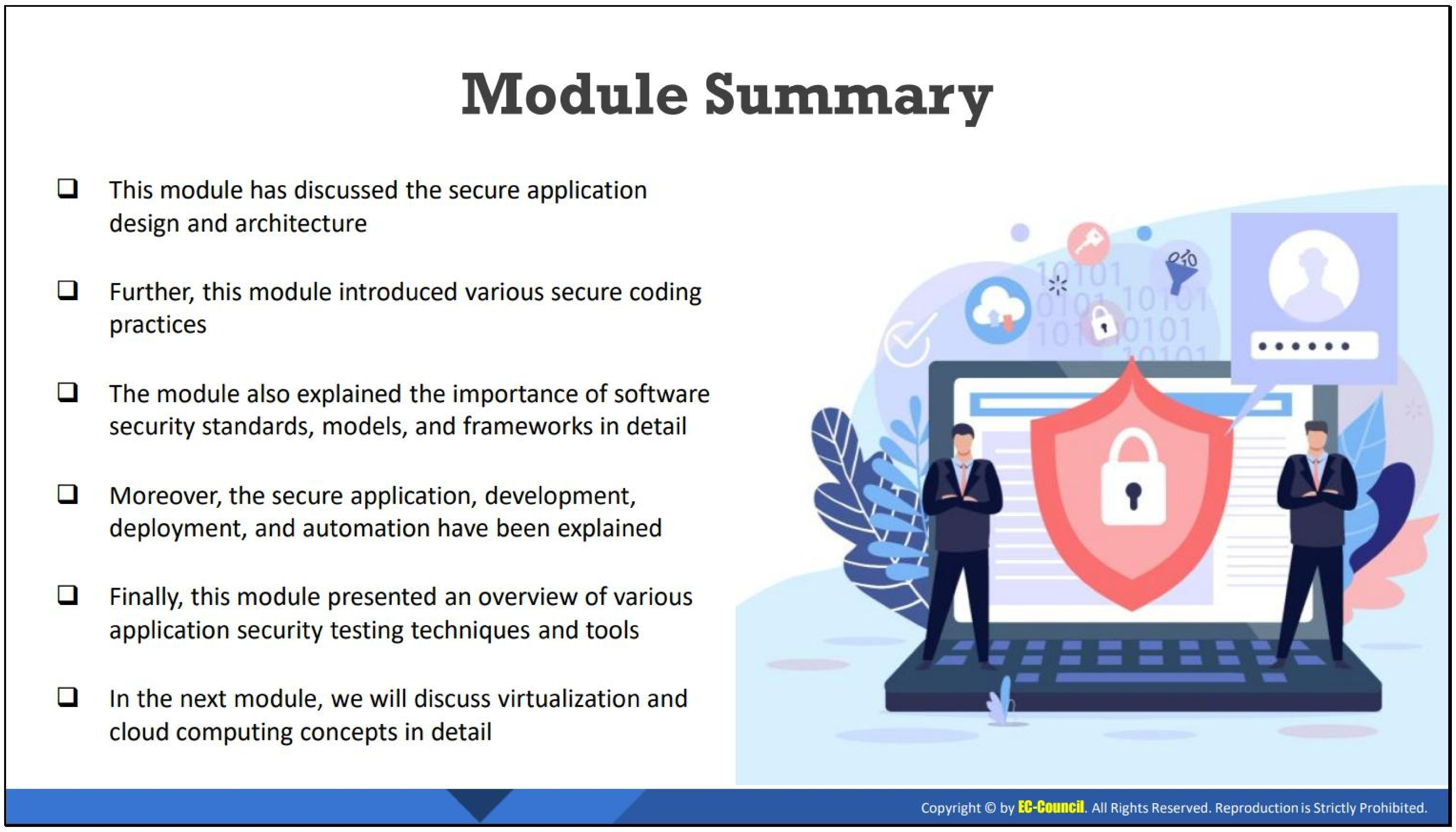
  Source: *https://www.gnu.org*

  GNU Wget is a free software package for retrieving files using HTTP, HTTPS, FTP and FTPS, the most widely used Internet protocols. It is a non-interactive command line tool, called from scripts, cron jobs, terminals without X-Windows support, etc.



```
┌─[root@parrot]─[~]
└──╼ #wget -q -S www.moviescope.com
  HTTP/1.1 200 OK
  Cache-Control: private
  Content-Type: text/html; charset=utf-8
  Server: Microsoft-IIS/10.0
  X-AspNet-Version: 4.0.30319
  X-Powered-By: ASP.NET
  Date: Wed, 25 Aug 2021 08:15:46 GMT
  Content-Length: 4241
┌─[root@parrot]─[~]
└──╼ #
```

Figure 9.39: Screenshot of GNU Wget



Module Summary

- This module has discussed the secure application design and architecture
- Further, this module introduced various secure coding practices
- The module also explained the importance of software security standards, models, and frameworks in detail
- Moreover, the secure application, development, deployment, and automation have been explained
- Finally, this module presented an overview of various application security testing techniques and tools
- In the next module, we will discuss virtualization and cloud computing concepts in detail

Module Summary

This module has discussed the secure application design and architecture. Further, this module introduced various secure coding practices. The module also explained the importance of software security standards, models, and frameworks in detail. Moreover, the secure application, development, deployment, and automation have been explained. Finally, this module presented an overview of various application security testing techniques and tools.

In the next module, we will discuss virtualization and cloud computing concepts in detail.