Introduction to Application Security and OWASP Top 10 Risks - Part 1 of 2

OWASP
The Open Web Application Security Project

Ralph Durkee
Durkee Consulting, Inc.
info@rd1.net

OWASP: About

- **OWASP = Open Web Application Security Project**
- Dedicated to making application security superior
- 200 Chapters, and Hundreds of Projects
- ROC Chapter **www.owasp.org/rochester**
- **2 mailing list**
  - Announcements Only
  - Discussion List
- Meetings approximately every quarter
- Individual OWASP membership is only $50/yr

Who's Ralph Durkee?

- **Principal Security Consultant**
  Durkee Consulting Inc.
- **Founder of Rochester OWASP**
- **Past President of Rochester ISSA**
- **Application Security Consulting**, development, auditing, advanced application penetration testing
- **Advanced Penetration Tester**, Security Trainer, Incident Handler and Auditor
- **Certifications**: GXPN, GPEN, GSEC, GCIH, GSNA, GCIA, C|EH, CISSP

Application Security is Really Hot

- Information Security is **Hot**

  Cisco estimates a million unfilled security jobs worldwide. *(Network World - Mar 2015)*

- Application Development is even **Hotter**

  10 hottest IT skills for 2015

  #1 = Programming/Application Development

  #4 = Security/Compliance Governance

  #5 = Web Development

- **Combine both: Application Security** if you really want to be in demand

Application Security is Really Hot (2)

Top 10 IT Security Jobs

- CIO.com survey based on DICE posting
- **#1 - Lead software security engineer**
  - Avg Salary $233,333
- **#10 - Application security manager**
  - Avg Salary $165,000

http://www.cio.com/article/2933173/salary/10-highest-paying-it-security-jobs.htm

Why is Application Security so Hot?

- Application Security is both critical and challenging
- Traditional security and Web App Firewalls are not sufficient
- Applications are often complex with many layers
- The attacker only needs one weakness in one component
- Application Security often isn't a priority
- Custom application requires custom security
- Application Security is relatively expensive, as it must be integrated into the development life-cycle.
- **Applications must protect themselves!**

The Key to Application Security!

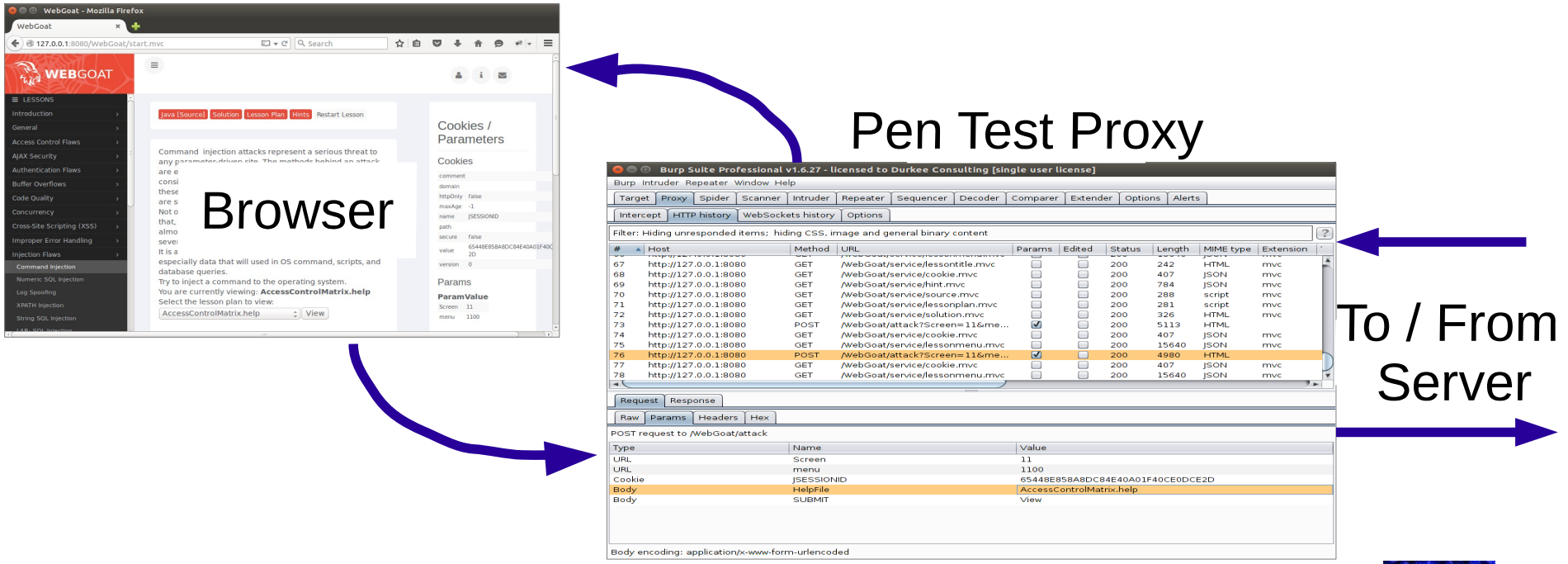
- What's the most important aspect of App Sec?
- One universal principle for getting it Right!
- Throughout the development cycle:
  Think like an Attacker!
  - Think about how can the application be misused?
  - Expect and handle the abnormal conditions
  - Anything received (or sent) might be malicious

Agenda

- Introduction
- Overview OWASP Top 10
- A2 – Broken Authentication and Session Management
- A4 – Insecure Direct Object References
- A5 – Security Misconfiguration
- A6 – Sensitive Data Exposure
- A7 – Missing Function Level Access Control
- A9 - Using Components with Known Vulnerabilities

Application Testing Basics: Using a Proxy

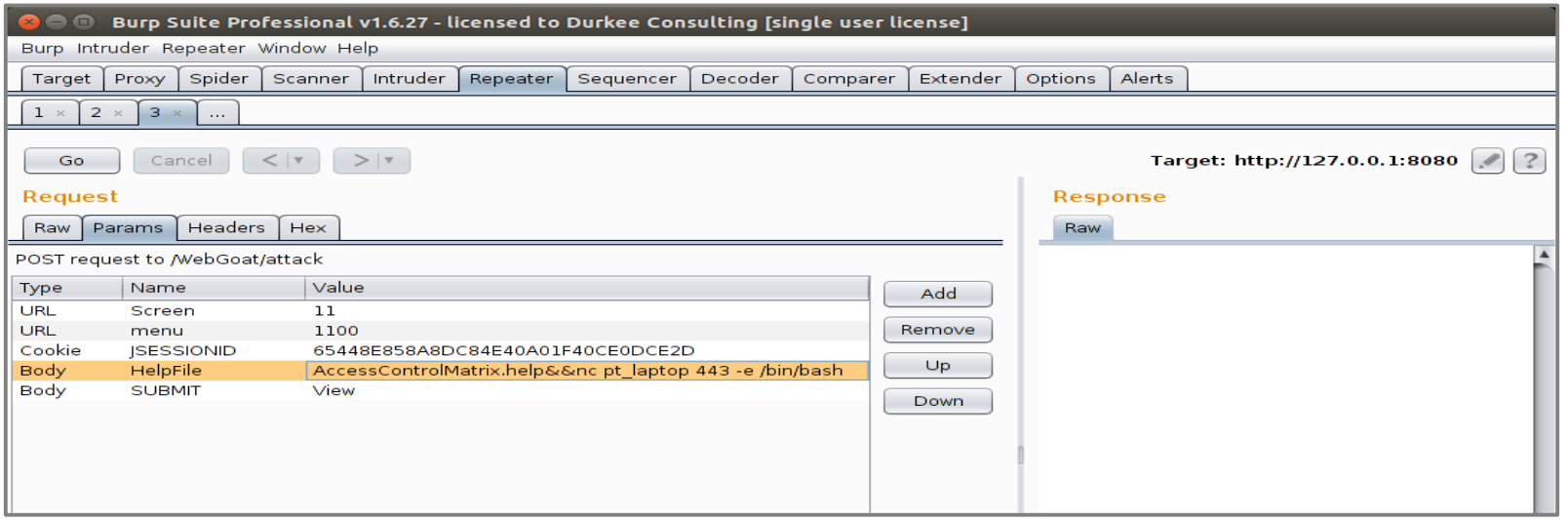
- Tool: Proxy such as Burp or OWASP ZAP
- Proxies between the Browser and Web App, or between the Mobile App and the Web Service



Application Testing Basics: Using a Proxy (2)

- Everything can be modified (cookies, headers, hidden and normal parameters)
- Look for what is assumed or what is trusted
- Example: change AccessControlMatrix.help to

AccessControlMatrix.help"%26nc%20pt_laptop%20443|%20"/bin/bash



OWASP Top 10 Application Security Risks

A1 – Injection	**A6** -- Sensitive Data Exposure
A2 - Broken Authentication and Session Management	**A7** – Missing Function Level Access Control
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9** - Using Components with Known Vulnerabilities
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

OWASP Top 10 Application Security Risks

- **A1 – Injection** Injection flaws, such as SQL, OS, and LDAP injection occur when hostile data tricks the interpreter into executing unintended commands or accessing data without authorization.
- **A2 – Broken Authentication and Session Management** Authentication and session management may allow attackers to compromise passwords, keys, or session tokens, to assume other users' identities.
- **A3 – Cross-Site Scripting (XSS)** XSS flaws occur when untrusted data is sent to a web browser without validation or escaping. XSS allows attackers to execute scripts in the victim's browser.
- **A4 – Insecure Direct Object References** Application exposes an internal object, such as a file, directory, or an ID, without an access control check.

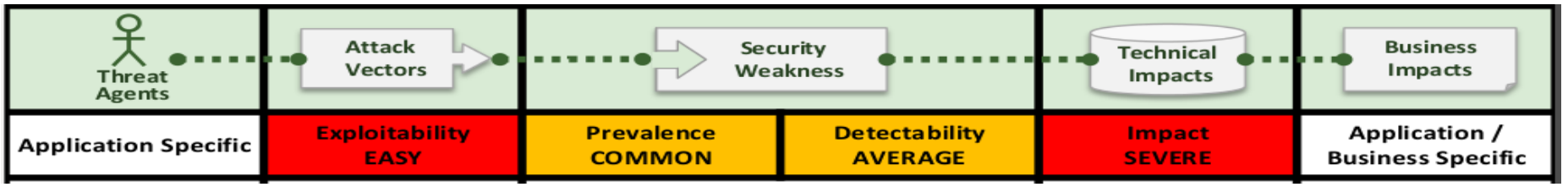
OWASP Top 10 Application Security Risks

- **A5 - Security Misconfiguration** Good security requires a secure configuration for the application, frameworks, and all servers. Additionally, software should be kept up to date.
- **A6 – Sensitive Data Exposure** Protect sensitive data, such as credit cards, tax IDs, and authentication credentials with encryption at rest or in transit.
- **A7 – Missing Function Level Access Control** Most applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same checks on the server when each function is accessed.

OWASP Top 10 Application Security Risks

- **A8 - Cross-Site Request Forgery (CSRF)** A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.
- **A9 – Using Components with Known Vulnerabilities** Components, such as libraries, frameworks, and software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.
- **A10 – Unvalidated Redirects and Forwards** Applications often redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites or use forwards to access unauthorized pages.

OWASP Top 10 – PDF Document



- Each Risk has a Graphical Header
  - Threat Agents
  - Exploitability
  - Prevalence
  - Detectability
  - Technical and Business Impact
- Am I Vulnerable?
- How Do I Prevent?
- Example Attack Scenarios & References

OWASP Top 10 Application Security Risks

A1 – Injection	**A6** -- Sensitive Data Exposure
A2 - Broken Authentication and Session Management	**A7** – Missing Function Level Access Control
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9** - Using Components with Known Vulnerabilities
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

OWASP Top 10 – A2: Broken Auth and Session Management

A2 RISKS

- Authentication and session management are often not implemented correctly
- Attackers may steal, discover, guess, or fix the session ID value.
- Having the session ID allows the attacker to assume the users' identity and privileges
- Broken Authentication may also disclose credentials, or allow passwords to be changed.

OWASP Top 10 – A2: Broken Auth and Session Management

A2 ATTACKS

- Guessing of Session IDs can be easily automated
- Sessions IDs can be disclosed via
  - Placed in a URL
  - In a session on shared computer
  - Cross-site scripting to send the session ID to the attacker
  - Over the network via clear text HTTP
  - By MITM due to weak HTTPS configurations
- Sessions IDs can be predetermined via Session Fixation attacks. *(The attacker provides a session ID via phishing, XSS, or malicious website)*

OWASP Top 10 – A2: Broken Auth and Session Management

A2 DEFENSES (*Authentication*)

- Use centralized authentication
- Use approved, platform provided authentication module
- All authentication controls enforced on the server side.
- Log all successful and failed authentications.
- Automatic blocking after too many failed logins
- All ~~credentials~~ **communications** sent over HTTPS link
- Authenticate the authentication server
- Authenticate requests for password reset, change email etc

OWASP Top 10 – A2: Broken Auth and Session Management

A2 DEFENSES (*Session*)

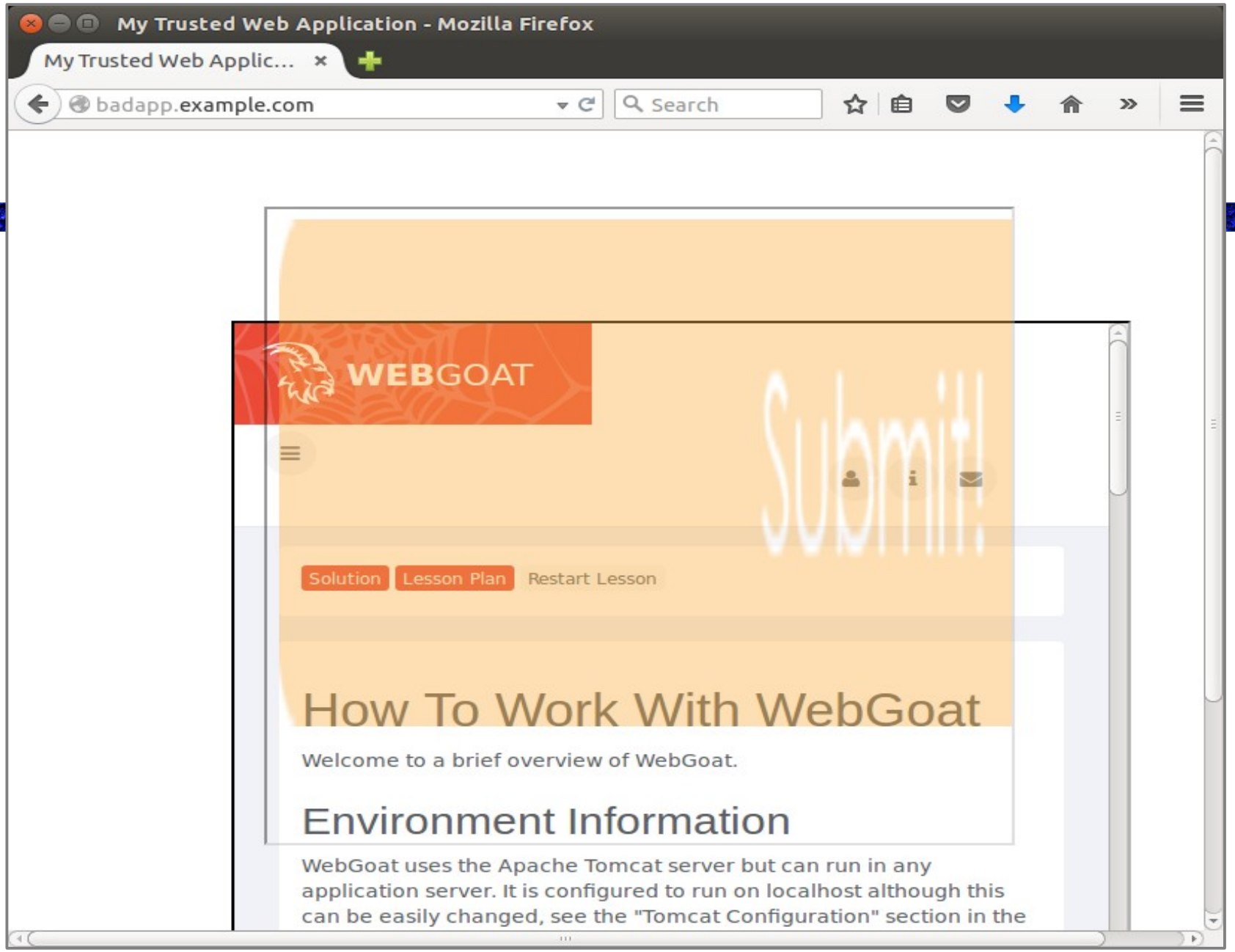
- Do NOT roll-your-own;
  Use a built-in session management.
- Session IDs must be long and unpredictable (>= 128 bits / 16 bytes)
- Server side session timeout and invalidation
- Change value when privileges change (such as login) to prevent session fixation.
- Stored in cookie with Secure & HttpOnly flags & limited domain and path attributes, expires at end of the session

Man-In-The-Middle (MITM) Attacks

MITM & Spoofing Attacks

- Arp Spoofing, DNS Spoofing, Bogus website, SSL MITM
- Allows the attacker to see and modify requests and response, and perform actions on behalf of the victim.

- Often combined with other attacks like CSRF, Clickjacking, or client side exploits
- Example with Web Goat



A2 & A6 DEFENSE - TLS Configuration

- Using SSL / TLS to protect
  - A2: Broken Auth and Session Management
  - A6: Sensitive Data Exposure
- Need for Authentication server certificate
  - No self signed or invalid certificates
  - Certificate Signed by trusted Authority
  - Browsers to Server
  - App to Server
- Configure Strong TLS Protocols and ciphers

OWASP Top 10 Application Security Risks

A1 – Injection	**A6** -- Sensitive Data Exposure
A2 - Broken Authentication and Session Management	**A7** – Missing Function Level Access Control
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9** - Using Components with Known Vulnerabilities
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

A4 – Insecure Direct Object References

A4 RISKS

- Attacker controlled data value references a server side object, like account number, file or directory name, or user name or role.
- Allows the attacker to specify a different file or different account to be accessed.
- Server fails to check access authorization
- Example:

```
https://example.com/app?acct=92834
https://example.com/apphelp?file=help.html
```

A4 – Insecure Direct Object References

A4 DEFENSE

1. Use a per user or per session indirect or mapped IDs instead of direct object reference.
   - If users have multiple accounts, they mapped and referenced as 1,2,3 etc.
   - Random numbers may also be used.
2. Validate user is authorized to access to all objects at the time of access.

OWASP Top 10 Application Security Risks

A1 – Injection	**A6** -- Sensitive Data Exposure
A2 - Broken Authentication and Session Management	**A7** – Missing Function Level Access Control
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9** - Using Components with Known Vulnerabilities
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

A5 – Security Misconfiguration

A5 RISKS

- Application components such as frameworks, libraries, middleware, databases and operating systems may be vulnerable to attack
- Exploitation of a component vulnerability generally gives a high level of access.
- Security hardening and updating components often skipped or delayed.
- Unnecessary services tools or debug functionality on a production server increase the risks.

A5 – Security Misconfiguration

A5 DEFENSE

- Security standards and hardening process for each component
- Dev, QA and Production should be configured the same.
- Different passwords and access for Dev, QA & Production.
- Keep components up to date with patches
- Authenticated Security Scans

OWASP Top 10 Application Security Risks

A1 – Injection	**A6 -- Sensitive Data Exposure**
A2 - Broken Authentication and Session Management	**A7** – Missing Function Level Access Control
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9** - Using Components with Known Vulnerabilities
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

A6 – Sensitive Data Exposure

A6 RISKS

- Sensitive data may stored in the clear.
- Application keys or passwords stored in code or scripts.
- Clear text communication exposes data
- Week cryptographic keys or poor key protection
- Mixed usage of HTTP and HTTPS

A6 – Sensitive Data Exposure

A6 DEFENSE

- Consistently use HTTPS for all communications
- Encrypt or hash the storage of sensitive information
- Use strong crypto algorithms and key lengths
- Use strong key management for proper key generation and protection of private keys.
- Disable autocomplete and caching on forms to discourage browser storage of sensitive information.

OWASP Top 10 Application Security Risks

A1 – Injection	**A6** -- Sensitive Data Exposure
A2 - Broken Authentication and Session Management	**A7 – Missing Function Level Access Control**
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9** - Using Components with Known Vulnerabilities
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

OWASP Top 10 – A7: Missing Functional Level Access Control

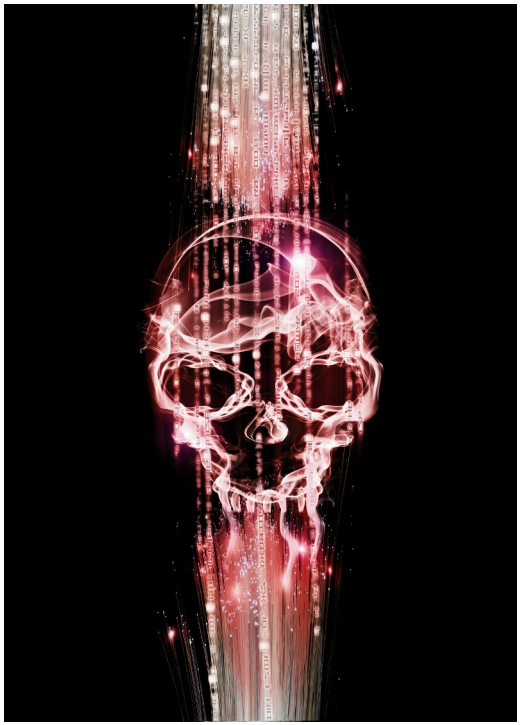
A7 RISKS

- Access rights are checked before the functionality is made visible in the UI.
- Then the application fails to verify the same access rights on the server when the function is accessed.
- Attackers may access privileged functions by forceful browsing or by a forged requests without proper authorization.

OWASP Top 10 – A7: Missing Functional Level Access Control

A7 ATTACKS

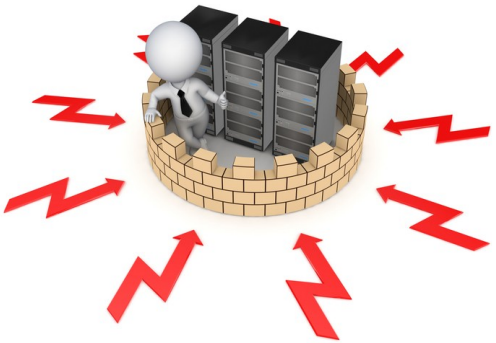
- **Forceful Browsing** - Just entering the correct URL in the browser to access the privileged operations
- **Parameter Tampering** – May change parameters values beyond the listed options.
- **Forged Request** – Just because a form wasn't provided doesn't mean a request can't be submitted.



OWASP Top 10 – A7: Missing Functional Level Access Control

A7 DEFENSES

- Authorization module should be consistent and easy to analyze so that it can be audited
- Authorization should deny access by default.
- Authorization module should be external to the main code, so that it's not hard coded in each page or function.
- For example some application frameworks may check privileges before access to specific paths or directories.



OWASP Top 10 Application Security Risks

A1 – Injection	**A6** -- Sensitive Data Exposure
A2 - Broken Authentication and Session Management	**A7** – Missing Function Level Access Control
A3 – Cross-Site Scripting (XSS)	**A8** - Cross-Site Request Forgery (CSRF)
A4 – Insecure Direct Object References	**A9 - Using Components with Known Vulnerabilities**
A5 – Security Misconfiguration	**A10** - Unvalidated Redirects and Forwards

A9 - Using Components with Known Vulnerabilities

A9 RISKS

- Vulnerable components (framework, libraries, etc) are very common
- Application components are not up-to-date and not included in security updates process
- Wide range of attacks often possible, including remote code execution
- Many components are not considered to be part of the application, or not my responsibility
- Example: *log analyzer buffer overflow*

A9 - Using Components with Known Vulnerabilities

A9 DEFENSE

- Keep inventory of necessary components and interdependencies.
- Remove unnecessary components and services
- Monitor vulnerability distribution lists for reports for new vulnerabilities
- Keep components up to date with patches by including in an organizational security update process
- Perform authenticated security scans

OWASP Application Security Summary

- Application Security is very challenging and often not well understood
- Knowing how to build secure and break insecure applications are important, in-demand skills
- Use OWASP Resources like ZAP, and WebGoat to train yourself
- Attend a OWASP chapter meetings and conferences.
- Sign-up on Rochester OWASP Chapter mailing lists **https://www.OWASP.org/rochester**

OWASP Top 10 Application Security Risks

THANK YOU!

Ralph Durkee
Durkee Consulting, Inc.
info@rd1.net

Resources - Non-Profit Groups & Events

OWASP Rochester Chapter

https://www.OWASP.org/rochester

OWASP Top 10

https://www.owasp.org/index.php/Top10

OWASP Application Security Verification Standard

https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

OWASP Secure Coding Principles

https://www.owasp.org/index.php/Secure_Coding_Principles

OWASP Testing Guide (large free ebook)

https://www.owasp.org/index.php/OWASP_Testing_Project

Rochester Security Summit

https://www.RochesterSecurity.org