OWASP Top 10

Hacking Web Applications with Burp Suite

What?

- Yes
  - Intro to the OWASP Top 10
  - Burp Suite
  - Mutillidae
  - Sqlmap
  - Beef
- No
  - Comprehensive vulnerability discovery
  - Deep dive into SQL injection, what SQL is, how SQL injection works
  - Deep exploration of XSS
  - Etc.

Sorry. I could easily spend an hour on each of these subjects.

Why?

- Web application penetration testing
- Useful skill for any web application developer
- Bug bounty programs
- Help out in the open-source community
- Learn more about Burp Suite
- Brief introduction to the OWASP Top 10

Who am I?

- Full-stack Tech Lead
- Web developer for over 10 years
- **Twitter**: https://twitter.com/chadfurman
- **LinkedIn**: https://linkedin.com/in/chadfurman
- **GitHub**: https://github.com/chadfurman
- **Website**: https://chads.website

clevertech®

https://hire.clevertech.biz
100% Remote Full-Time Web Development

OWASP Top 10

1. Injection
2. Weak Authentication and Session Management
3. Cross Site Scripting (XSS)
4. Insecure Direct Object Reference
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirects and Forwards

Bold items in demo

SQL Injections

- **SELECT**
  - **In-band** (i.e. ‘ or 1=1 --)
  - **UNION select null, null, ..., null, null --**
  - **UNION select “a”, ..., null, null --**
  - **GROUP_CONCAT(SELECT * FROM ... ORDER BY ...)**
- **INSERT**/UPDATE/DELETE
  - **In-band**
  - **Error insertion**
    - nameconst() db version specific
    - updatexml() requires XPATH
  - Result-based
    - **conv(hex()) -- 8 characters at a time**
- **sqlmap**

Bold items in demo

Weak Authentication and Session Management

- Account Enumeration (Wrong Username vs Incorrect Credentials)
- HTTPS for login request (not just the login page...)
- Default passwords (admin/password?)
- Weak / No account lockout mechanism?
- Account creation (Can we make an admin account?)
- Weak password policy
- **Authentication bypass**
- **Cookie manipulation / replay**

Bold items in demo

Cross Site Scripting

- **Reflected**
  - User clicks a malicious link or button, gets directed to the page, XSS triggers
- **Stored**
  - JavaScript gets stored to the database and renders on pages unencoded
- **Steal cookies for replay attacks**
- Trigger actions / read nonces
- Social engineering / page rewrite
  - Enter credit card number
  - Download a file
  - Enter credentials
- **BeEF - Browser Exploitation Framework**

Bold items in demo

Insecure Direct Object Reference

- **Directory Traversal**
- **Local File Inclusion**
- Remote File Inclusion
- **ID enumeration**

Bold items in demo

Security Misconfiguration

- **Fingerprinting**
  - **Database (MySQL? MSSQL? version?)**
  - **Server (OS / Apache / Nginx / version)**
  - **Language (PHP? Ruby? ASP? version?)**
  - Framework (Laravel? WordPress? Drupal? Angular? etc...)
- **Forced Browsing?**
  - .htaccess, config.php, phpinfo.php...
  - .tar.gz, .zip, .bak, .old, .txt, .pdf...
- Enumerating admin interfaces / install scripts
- **Stack traces? Error messages? System / directory information?**

Bold items in demo

Sensitive Data Exposure

- Weak SSL ciphers / Protocols / Keys
- Padding Oracle
- **Unencrypted transport / storage**
- **HTTP Strict Transport Security (HSTS)**

Bold items in demo

Missing Function Level Access Control

- **Directory traversal**
- Bypassing authentication
  - **Direct page request (forced browsing)**
  - Parameter modification (/page.asp?authenticated=yes)
  - Session ID prediction (i.e. password reset tokens?)
  - **SQL injection**

Bold items in demo

Cross Site Request Forgery (CSRF)

- User clicks a malicious link/button
- Button triggers an action on another site
  - Create a blog post
  - Logout of their account
- CSRF is always possible if there's XSS
- Missing nonces allow this kind of attack

Bold items in demo

Using Components with Known Vulnerabilities

- CVEs out for the web server version
- **CVEs for the language interpreter**
- **Exploits for the framework**
- Exploits for framework plugins
- Imagick?

Bold items in demo

Unvalidated Redirects and Forwards

- **Client side redirects**
- Allow users to be maliciously redirected
- Enables social engineering attacks to look more legit
  - Clicking a long link with a redirect at the end
  - You might see the domain name and not notice where you end up

Live Demo

Questions?

https://chads.website

https://twitter.com/chadfurman

https://github.com/chadfurman

https://linkedin.com/in/chadfurman

https://portswigger.net/burp

https://www.owasp.org

https://hire.clevertech.biz

Special thanks to:

@irongeek

@kalilinux

@offsectraining

@Burp_Suite

@Clevertech

@2600