Software Security

UBNetDef, Spring 2024
Week 8

Presenters:
Dikshit Khandelwal [DK]
Ben Juliano

Agenda

- Three common vulnerabilities
  - Malformed Inputs
  - Poor implementation
  - Memory Management
- Common exploits
  - Web applications
  - Systems
- Common software vulnerabilities
  - Shared Libraries (CVEs)
  - Old Libraries
- Database security
  - Data minimization
- How to find vulnerabilities in code bases
- SIEM

What is Software Security

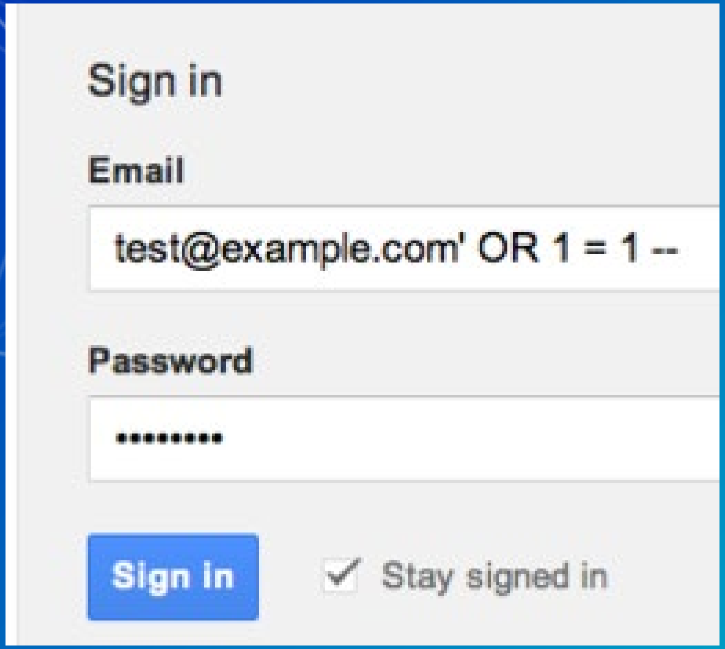
- Software security is a set of policies and controls designed to ensure that programmers design and implement secure applications.
- Secure applications are programs that do not break from their intended purpose or implement measures so that they don't inadvertently reveal any system information that may be useful for an attacker.

What’s a Software Vulnerability?

- A software vulnerability is a security flaw in an application that can be used to do malicious things to the application or the computer the application is running on.
- These vulnerabilities occur when an applications behaves in an unintended way.

Malformed Inputs

- A malformed input can occur when a user enters an input into an application that the application doesn’t know how to handle.
- Attackers can leverage malformed inputs to inject a malicious input in a system to perform some unknown activity.


Sign in

Email

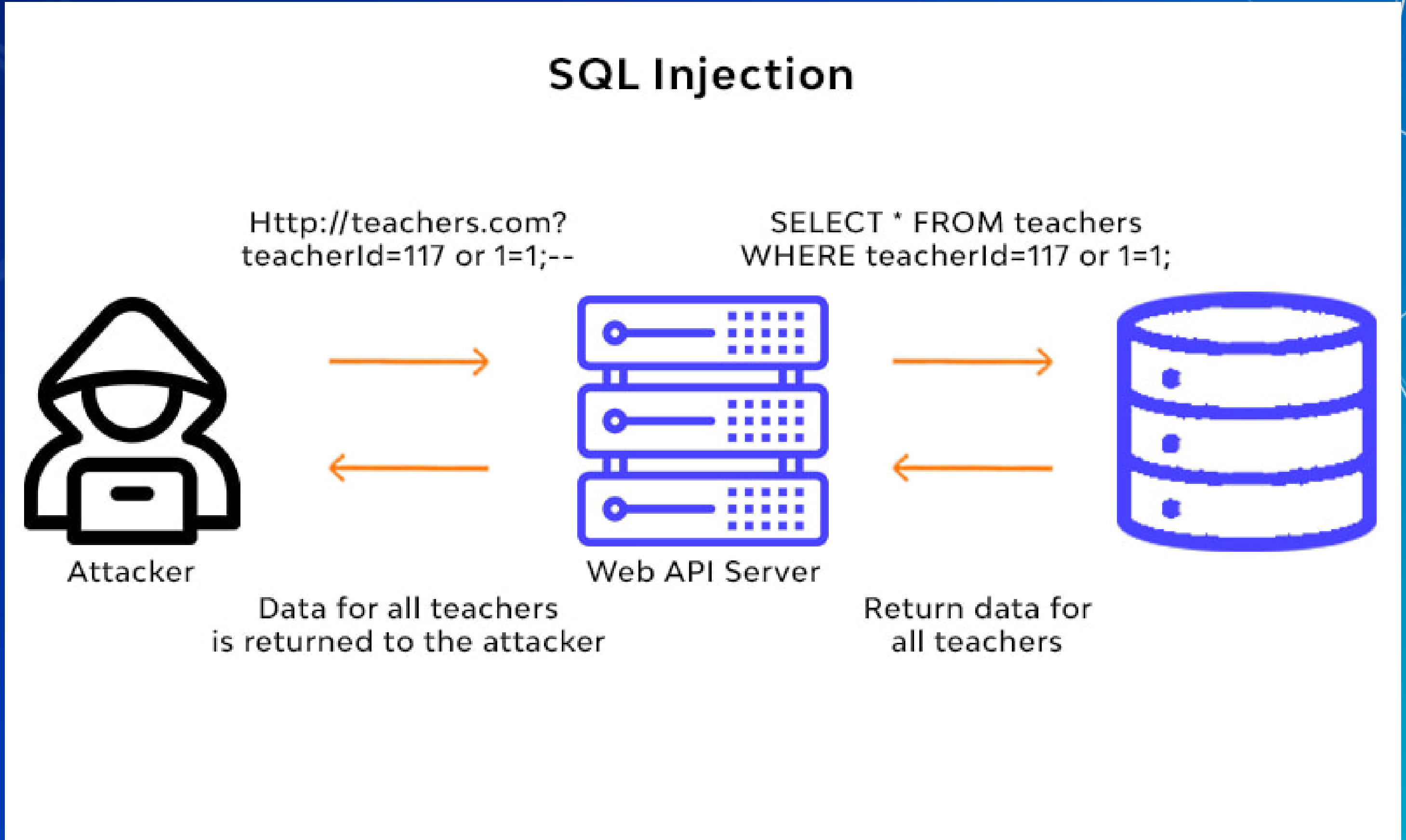
test@example.com' OR 1 = 1 --

Password

••••••••••

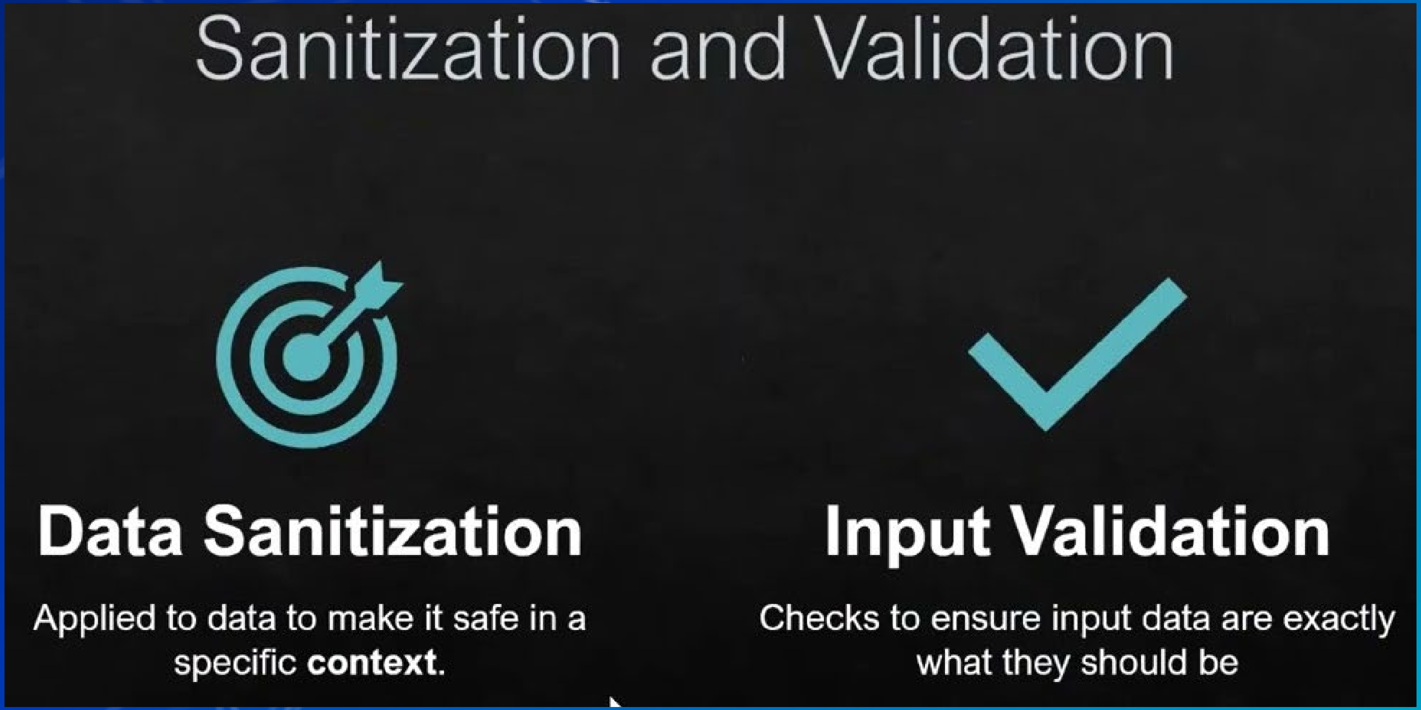
Malformed Input Attacks

- The most well know injection attacks are SQL injections and command injections.



Malformed Input Solutions

- The ways of preventing malformed inputs from posing a threat are:
  - Input sanitization
  - Input validation



Sanitization and Validation

Data Sanitization

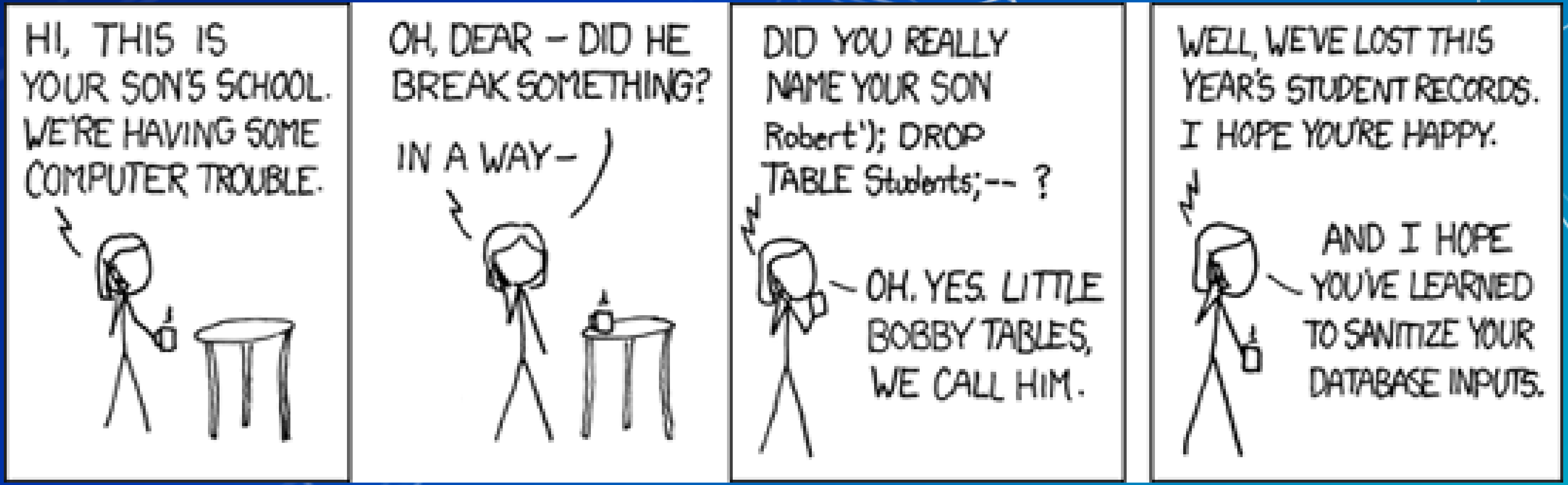
Applied to data to make it safe in a specific **context**.

Input Validation

Checks to ensure input data are exactly what they should be

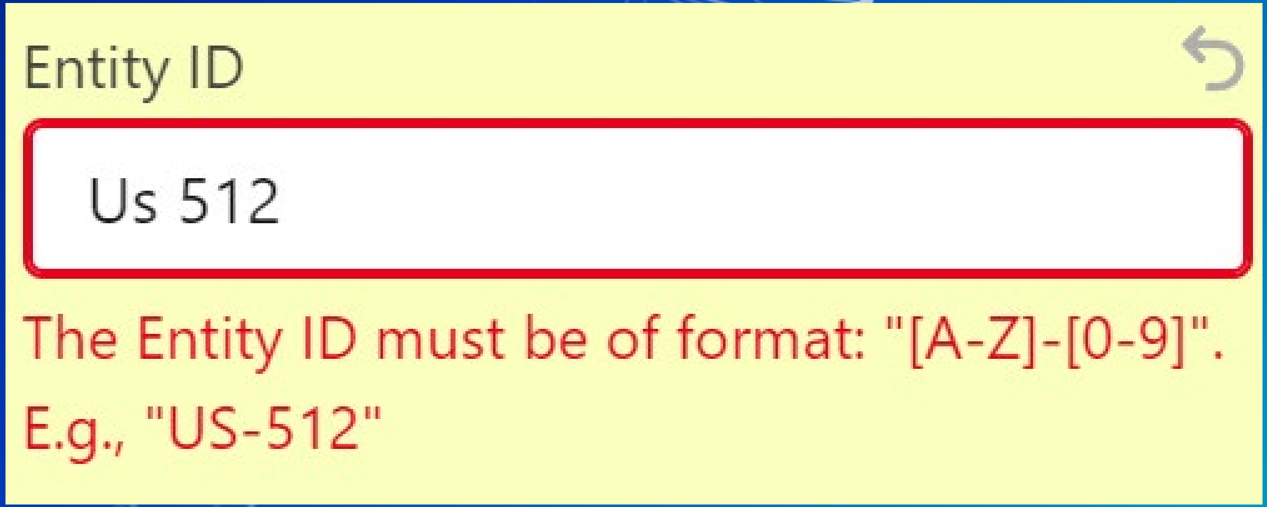
Input Sanitization

- Input is sanitized by modifying or removing any disallowed characters from the input.
- An example is adding a ‘ if there is only one in the text or changing all ‘ to “.
- After undergoing sanitization, the data is checked with input validation to confirm the validity.


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Input validation

- Input validation is the process of confirming the data is correct.
- Input is well formed if it meets a set of criteria.
- It test for things such as:
  - Length
  - Formatting
  - Allowed characters



Entity ID

Us 512

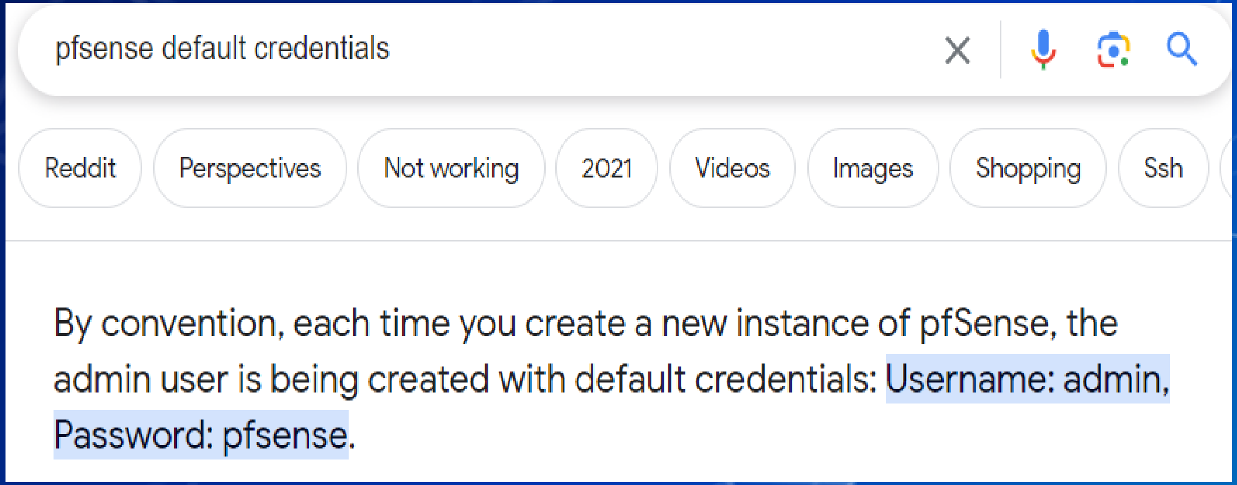
The Entity ID must be of format: "[A-Z]-[0-9]". E.g., "US-512"

System Misconfiguration

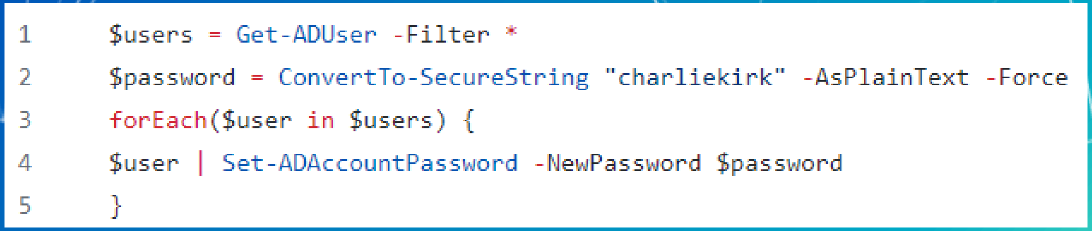
- Misconfigured systems are a common driver of vulnerabilities to an organization.
- Some common system misconfigurations are:
  - Leaking server-side information
  - Default credentials
  - Setuid binaries
  - Services running with elevated privileges
  - Access controls
  - Systems left in demo/testing mode
  - ......Several more

Default Credentials

- Default credentials preconfigured on hardware devices or software applications by manufacturers or vendors are often left unchanged by users or administrators, creating a security vulnerability.
- Finding default credentials is as easy as searching "[SOFTWARE] default password".



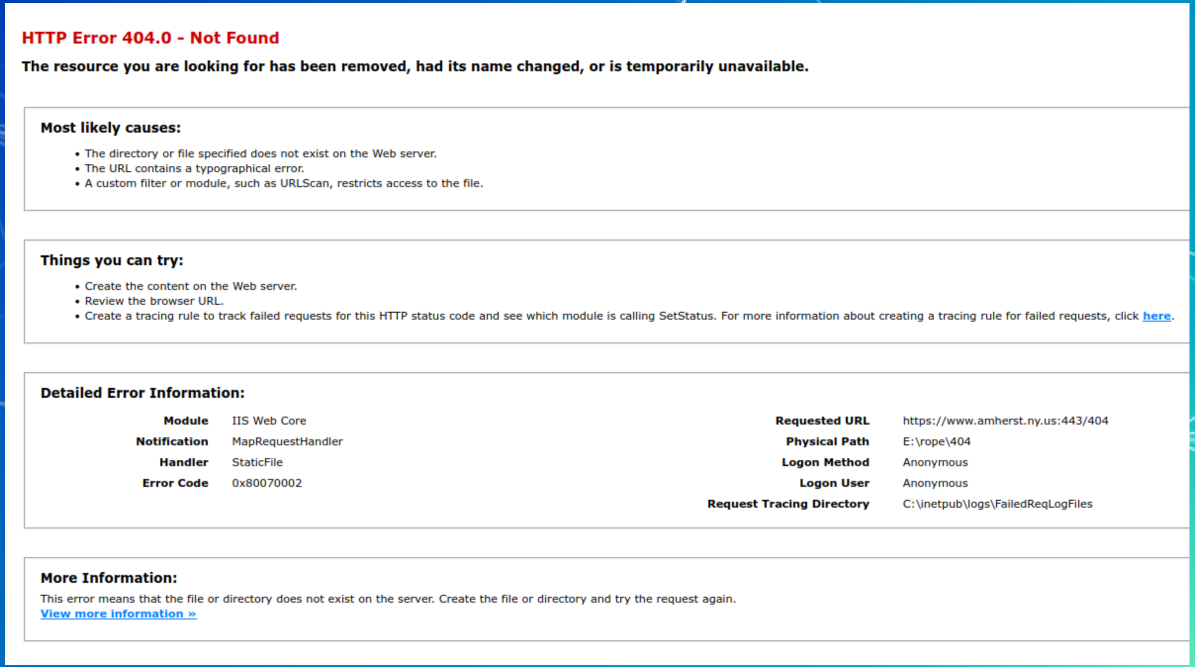
By convention, each time you create a new instance of pfSense, the admin user is being created with default credentials: Username: admin, Password: pfsense.



```
$users = Get-ADUser -Filter *
$password = ConvertTo-SecureString "charliekirk" -AsPlainText -Force
forEach($user in $users) {
$user | Set-ADAccountPassword -NewPassword $password
}
```

Leaking server-side information

- Leaking server-side information can have major implications for an attacker.
- This information could be used by an attacker to plan out their attack.
- Some information that needs to be kept secure is:
  - Operating systems.
  - System/Software version number.
  - Relevant paths.
- Why is leaking this information bad?



HTTP Error 404.0 - Not Found

The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.

Most likely causes:

- The directory or file specified does not exist on the Web server.
- The URL contains a typographical error.
- A custom filter or module, such as URLScan, restricts access to the file.

Things you can try:

- Create the content on the Web server.
- Review the browser URL.
- Create a tracing rule to track failed requests for this HTTP status code and see which module is calling SetStatus. For more information about creating a tracing rule for failed requests, click here.

Detailed Error Information:

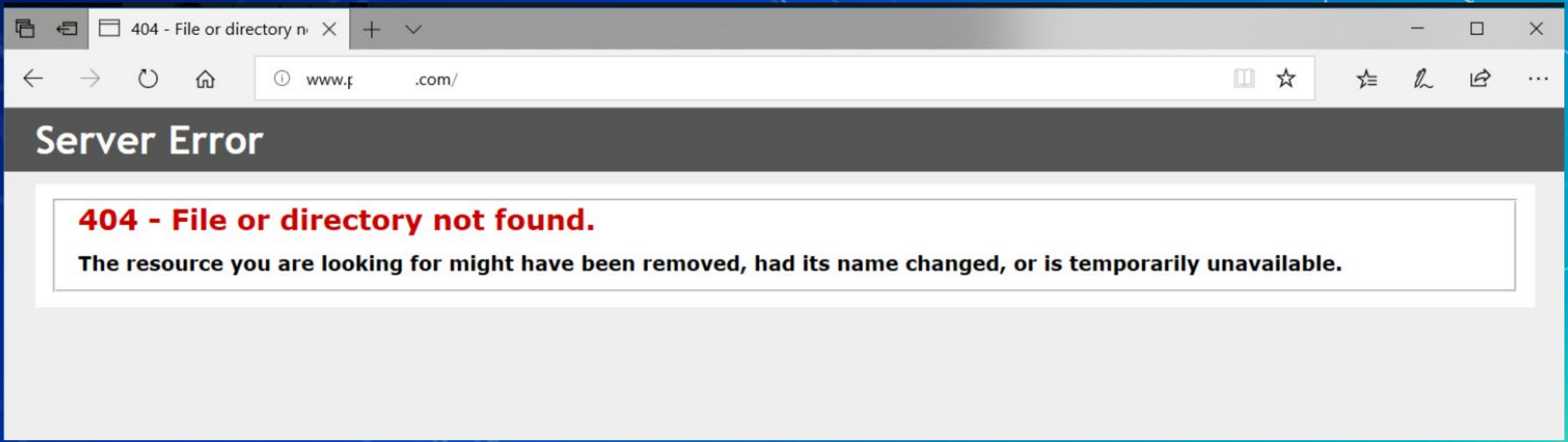
Module	IIS Web Core	**Requested URL**	https://www.amherst.ny.us:443/404
Notification	MapRequestHandler	**Physical Path**	E:\rope\404
Handler	StaticFile	**Logon Method**	Anonymous
Error Code	0x80070002	**Logon User**	Anonymous
		Request Tracing Directory	C:\inetpub\logs\FailedReqLogFiles

More Information:

This error means that the file or directory does not exist on the server. Create the file or directory and try the request again.
View more information »

Leaking server-side information

- To mitigate this issue limit error message to show the minimal amount of information for a user.
- Here is an example of a hardened IIS server .



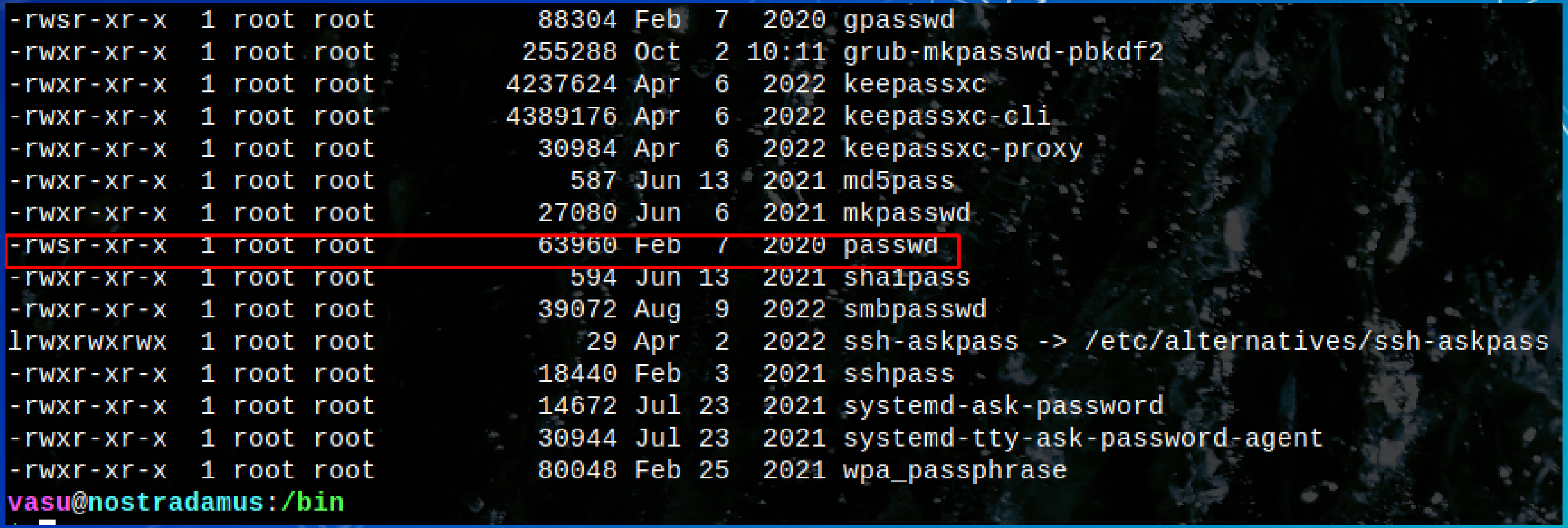
Server Error

404 - File or directory not found.

The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Setuid Binaries

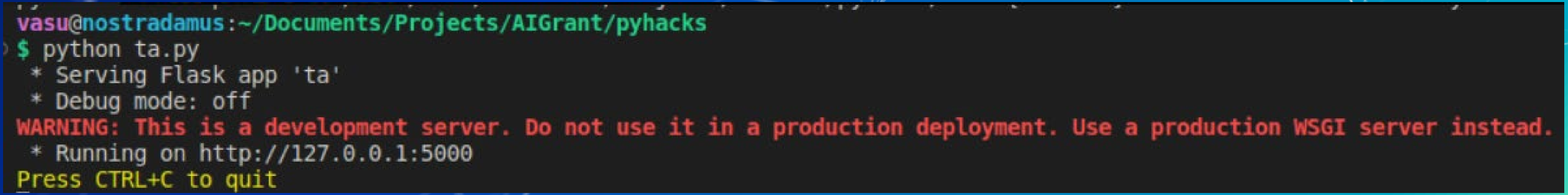
- Setuid binaries only exist on Linux machines these types of files allow the user to run the file with the permissions of the owner.
- If root is the owner, then when running the file it will run with root permissions.
- These files can allow an unprivileged user to do more than they are normally allowed to do.
- An attacker could target these files and try to escalate their privileges.



```
-rwsr-xr-x  1 root root         88304 Feb  7  2020 gpasswd
-rwxr-xr-x  1 root root        255288 Oct  2 10:11 grub-mkpasswd-pbkdf2
-rwxr-xr-x  1 root root       4237624 Apr  6  2022 keepassxc
-rwxr-xr-x  1 root root       4389176 Apr  6  2022 keepassxc-cli
-rwxr-xr-x  1 root root         30984 Apr  6  2022 keepassxc-proxy
-rwxr-xr-x  1 root root           587 Jun 13  2021 md5pass
-rwxr-xr-x  1 root root         27080 Jun  6  2021 mkpasswd
-rwsr-xr-x  1 root root         63960 Feb  7  2020 passwd
-rwxr-xr-x  1 root root           594 Jun 13  2021 sha1pass
-rwxr-xr-x  1 root root         39072 Aug  9  2022 smbpasswd
lrwxrwxrwx  1 root root            29 Apr  2  2022 ssh-askpass -> /etc/alternatives/ssh-askpass
-rwxr-xr-x  1 root root         18440 Feb  3  2021 sshpass
-rwxr-xr-x  1 root root         14672 Jul 23  2021 systemd-ask-password
-rwxr-xr-x  1 root root         30944 Jul 23  2021 systemd-tty-ask-password-agent
-rwxr-xr-x  1 root root         80048 Feb 25  2021 wpa_passphrase
vasu@nostradamus:/bin
```

Demo/Testing Mode

- When a new update is going to be rolled out to an applications developers will test the updates on the application first.
- These demo/test modes normally aren't configured to have every security measure implemented leaving them vulnerable to multiple attacks if they are not properly disposed of after testing.
- An attacker could find one of these test environments and pivot off it to get deeper into a network or escalate their privileges.



```
vasu@nostradamus:~/Documents/Projects/AIGrant/pyhacks
$ python ta.py
 * Serving Flask app 'ta'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Access Controls

- Access control enforces policy such that users cannot act outside of their intended permissions.
- Only the authorized users should be allowed to see/use the specific data they need to complete their work.
- Violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone.

Services and Executables

- Some services will require higher levels of privileges to function properly.
- These services poses a major security risk, if a service needs higher privileges to function compensatory controls will need to be put in place to help protect against an attack that originates from the service.
- With extra controls in place the service still functions as needed, and we reduce the risk of the service being exploited and dealing a greater amount of damage because of the privileges it has.
- Many services are remotely provisioned.
- Local services often interact system level functionality.

Memory Management

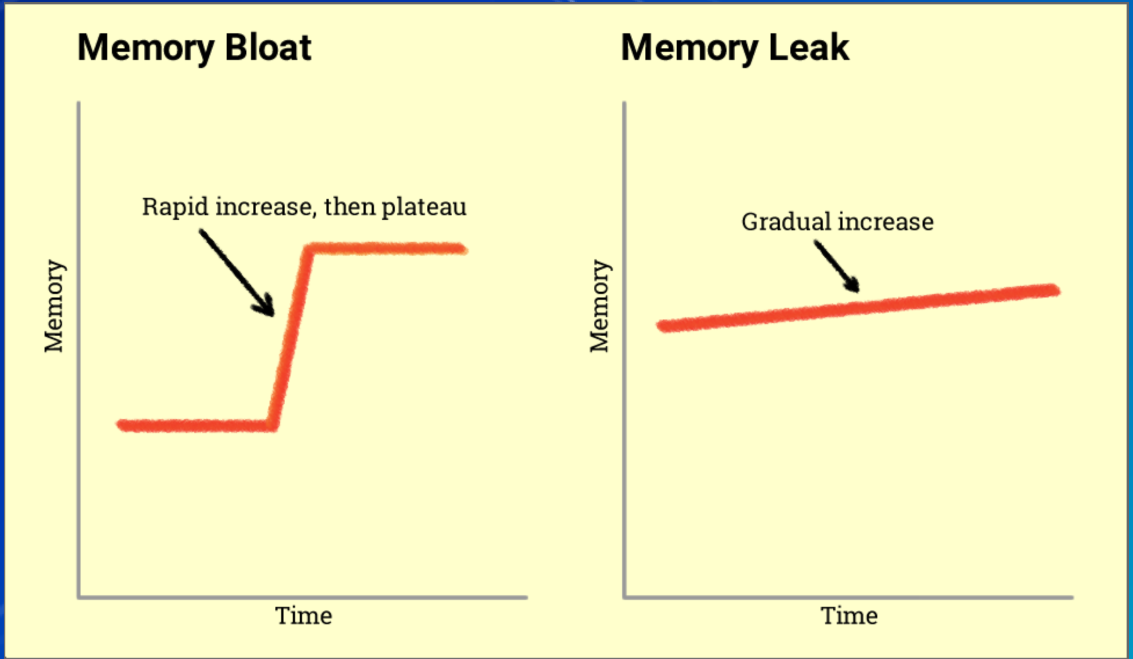
- Memory management refers to how software developers manage the memory they use in their applications.
- Poor memory management can lead to:
  - Buffer overflows
  - Memory leaks

Memory Management Mitigations

- To prevent buffer overflows
  - Stack smashing tools
  - Canary values
  - Safe system and libraries calls
  - Specifying data lengths
  - Address randomization
  - Complier time
  - Error trapping/code hygiene

Memory Management mitigation

- To prevent memory leaks developers, need to properly free all the memory they allocate for a program.
- To help developers code can be tested by automated tools that will look for and flag any memory leaks found in the program.



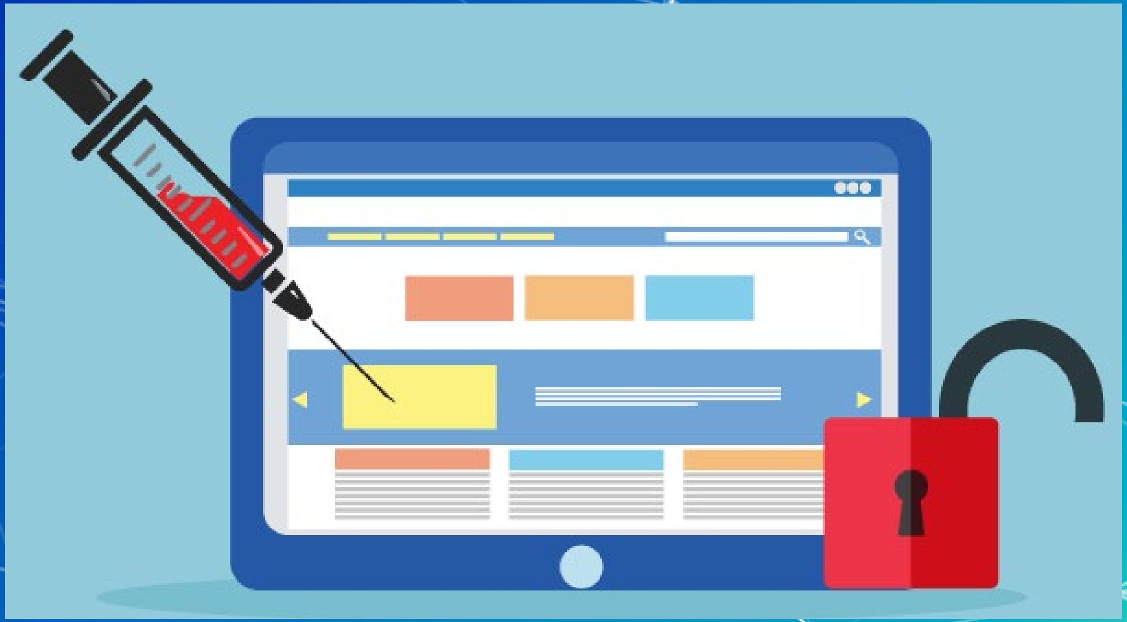
Please return in 10 minutes

Agenda

- Three common vulnerabilities
  - Malformed Inputs
  - System Misconfiguration
  - Memory Management
- Common exploits
  - Web applications
  - Systems
- Common software vulnerabilities
  - Shared Libraries
  - Old Libraries
- Database security
  - Data minimization
- How to find vulnerabilities in code bases
- SIEM

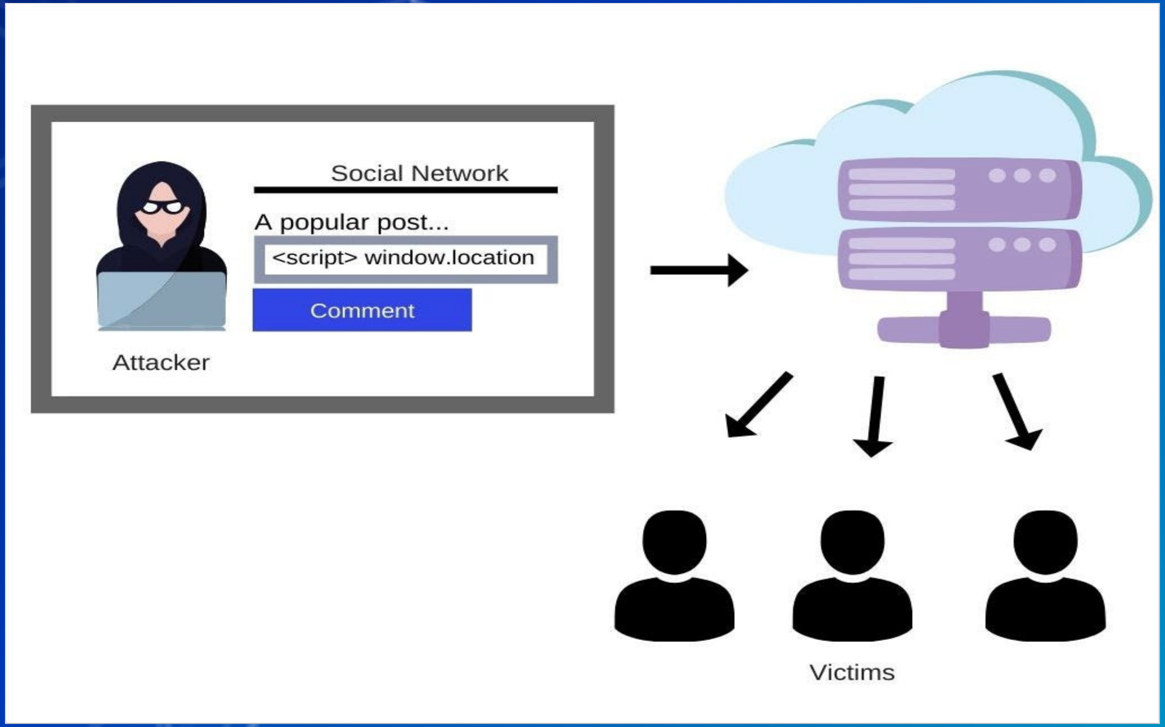
Web Application Exploits

- Some common web application exploits are:
  - Cross site scripting (XSS)
  - SQL injections
  - Cross site forgery request (CSFR)



Cross-site scripting (XSS)

- Cross site scripting - occurs when untrusted data is included in a web page without validation. This can allow attackers to inject malicious code into the web application and execute it on the client side.



Demo

Cross Site Scripting: Samy Worm

- MySpace gave users a lot of freedom to customize their profiles, allowing the use of HTML code.
- Samy Kamkar was a 19 year who used my space he knew how to code and wanted to see if he could make the site do things it wasn't supposed to.
- First, he found a way to edit his relationship status box to "in a hot relationship." the relation ship box had a drop-down menu of inputs the user could select from, but Samy found away around this.

myspace

Cross Site Scripting: Samy Worm

- About a month later he decided to write a script that would force everyone who visited his profile to add him as a friend. The script would also add a line to the person's profile, under the "my heroes" category: "but most of all, Samy is my hero."
- He also made it so the code would copy to other users profiles once someone visited a profile the worm had infected they too would add Samy and be infected.
- Around midnight on October 4, 2005, in Los Angeles, when Samy Kamkar, then a 19-year-old hacker, released what has come to be known as the "Samy worm," perhaps the fastest-spreading computer virus of all time.

SQL Injection

- SQL injection is one of the most common web hacking techniques.
- SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

Demo

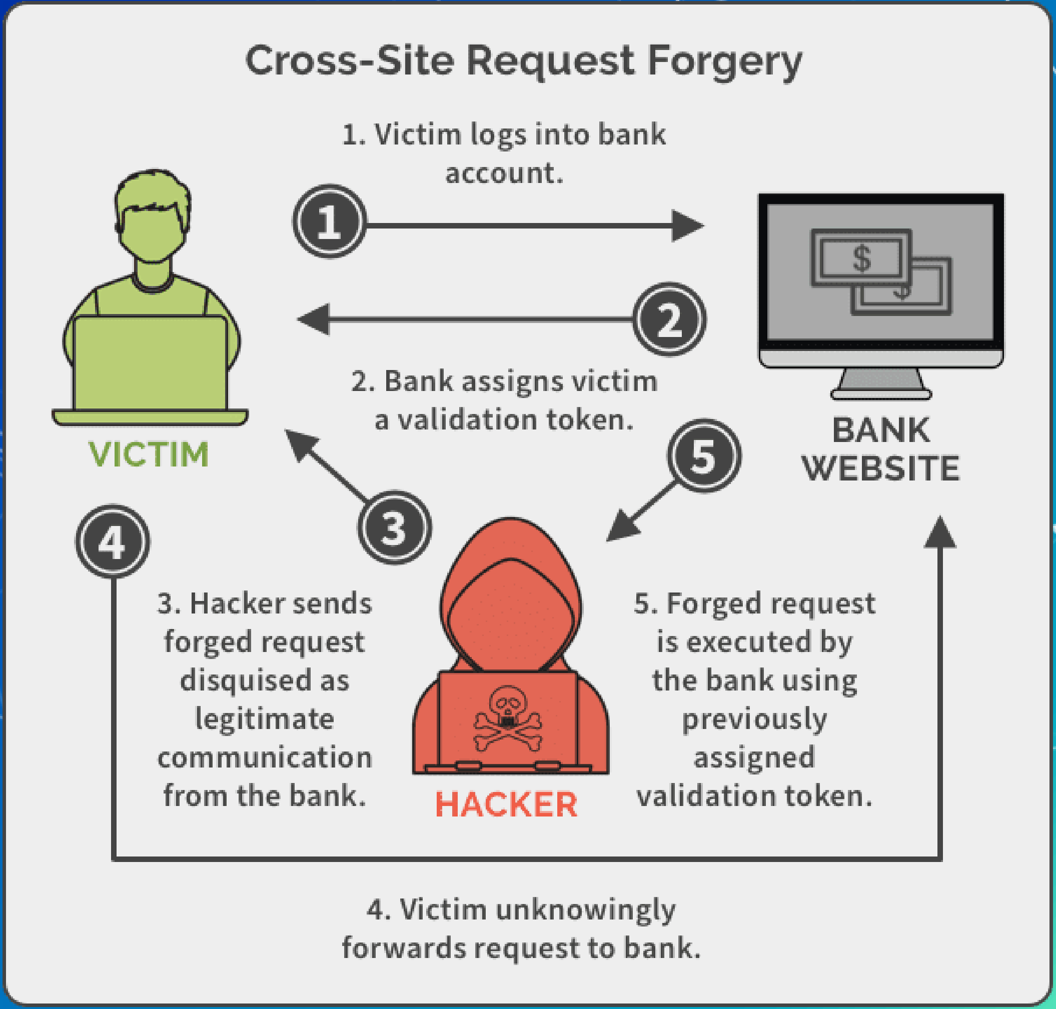
SQL Injection: Epic Games

- In 2016 Epic games suffered a breach that impacted 800,000 accounts due to SQL injections.
- The compromise occurred from Unreal Engine and Unreal Tournament forms.
- Attackers leveraged a SQL injection vulnerability in outdated vBulletin forum software the company used and were able to dump the entire database.

Epic Games forums hacked again, over 800,000 users affected

Cross Site Forgery Request (CSFR)

- Cross-Site Request Forgery is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.



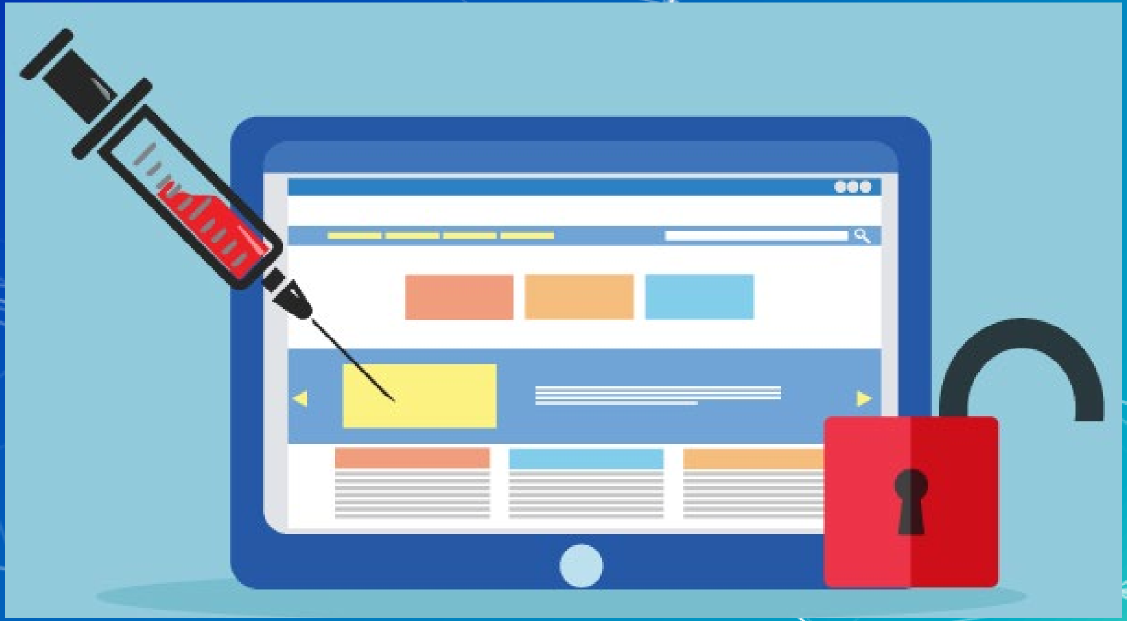
Cross-Site Request Forgery
1. Victim logs into bank account.
1
2
2. Bank assigns victim a validation token.
VICTIM
BANK WEBSITE
3
4
5
3. Hacker sends forged request disguised as legitimate communication from the bank.
HACKER
5. Forged request is executed by the bank using previously assigned validation token.
4. Victim unknowingly forwards request to bank.

Cross Site Forgery Request: Glass door

- In 2020 a bug bounty hunter was inspecting glass door a website for job seekers and posting anonymous reviews.
- The vulnerability allowed attackers to hijack accounts from logged in victims and create new admin accounts or employer accounts.
- An attacker could delete information on job seekers and employers as well as creating admin accounts on the site.
- When the security team investigated, they found that the tokens did trigger and exception but didn’t fail and in turn logged the exception and allowed the operation to continue.

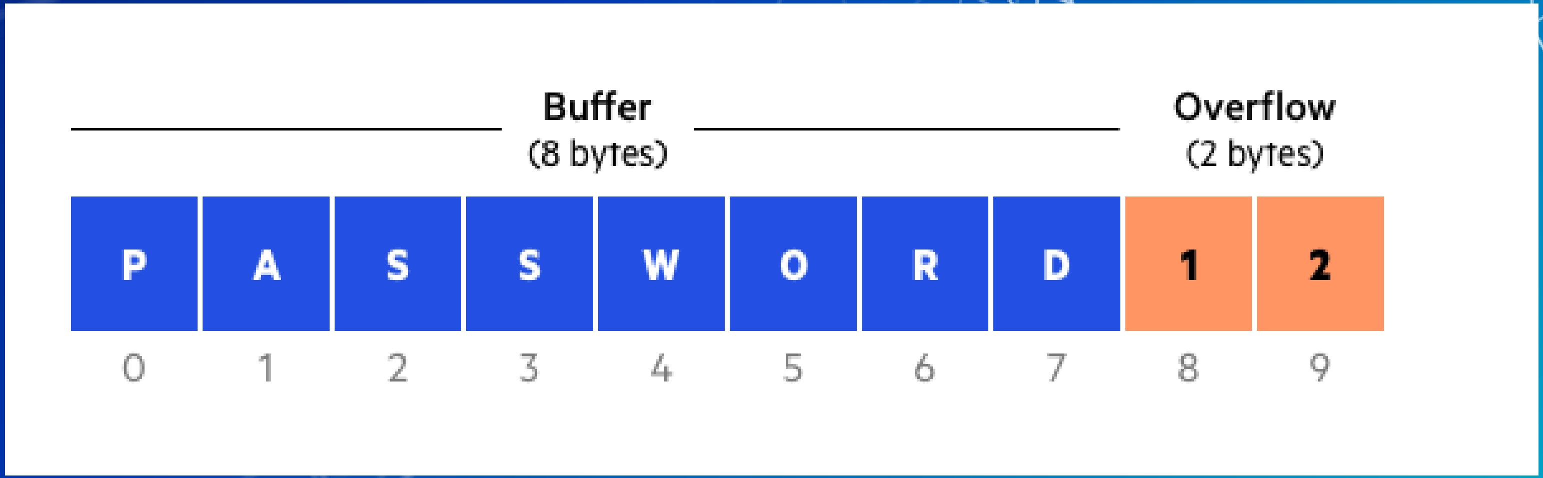
System Exploits

- Some common web application exploits are:
  - Buffer overflow
  - Setuid
  - Path attacks
  - Library injection



Buffer Overflow

- A buffer overflow occurs when the volume of data exceeds the storage capacity of a memory buffer
- Ex.) If a buffer is of size 8 and the data stored into the buffer is 10 there would be a buffer overflow



Demo

Path Traversal

- Path traversal attacks can be performed by automated tools that will try to go to different files on a webserver.
- An attacker can find hidden files on a webserver through these attacks possibly finding a vulnerable page.
- The attacker can then exploit this vulnerable page to get further access to the web server.

Library injection

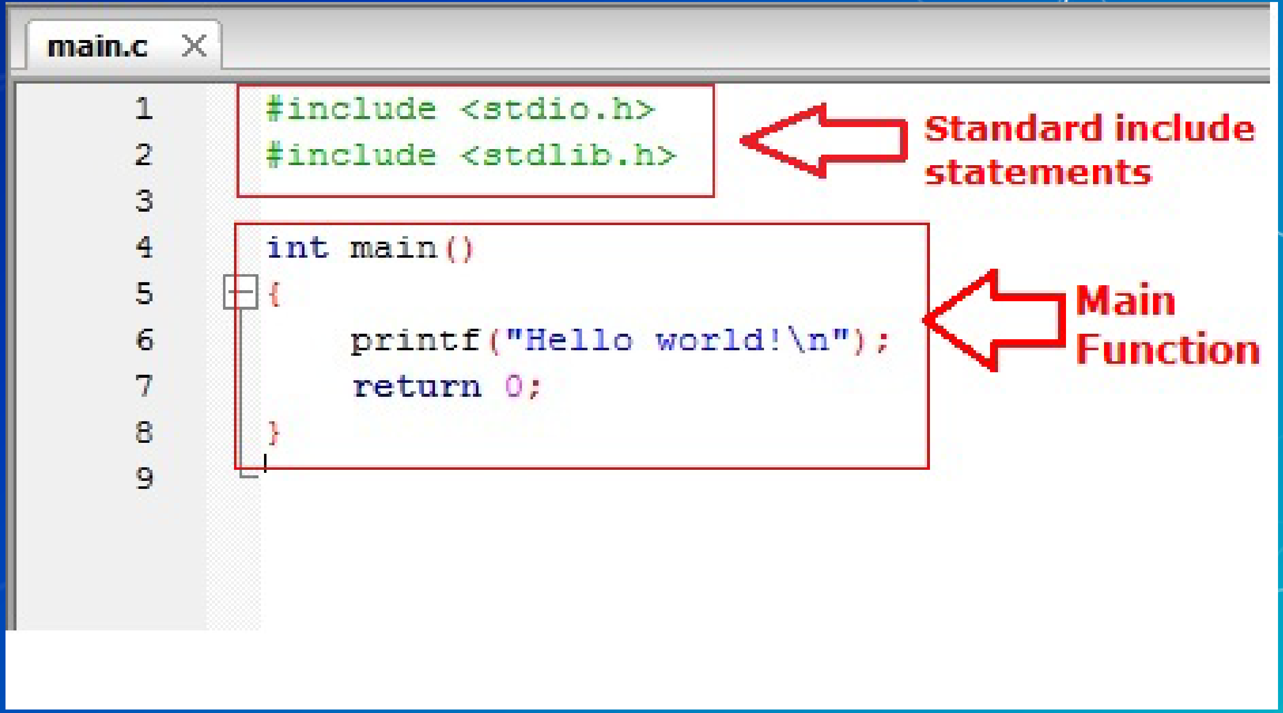
- An attacker can infect reusable libraries and inject them to different applications, altering the behavior without having to modify the source code.
- An example is replacing existing code with your own implementations (e.g. replace a function that should give random numbers).
- By injecting malicious libraries an attacker can create functions named the same as ones being used to manipulate the existing code.

Agenda

- Three common vulnerabilities
    - Malformed Inputs
    - System Misconfiguration
    - Memory Management
- Common exploits
    - Web applications
    - Systems
- Common software vulnerabilities
    - Shared Libraries
    - Old Libraries
- Database security
    - Data minimization
- Firmware Code
- How to find vulnerabilities in code bases
- SIEM

Shared Libraries

- Shared libraries are files that contain executable code to be used by multiple applications
- Most of these shared libraries also call other shared libraries inside their code, a lot of code relies on other code creating a lot of points for failure
- Examples are
  - `Import` in python
  - `Include` in C

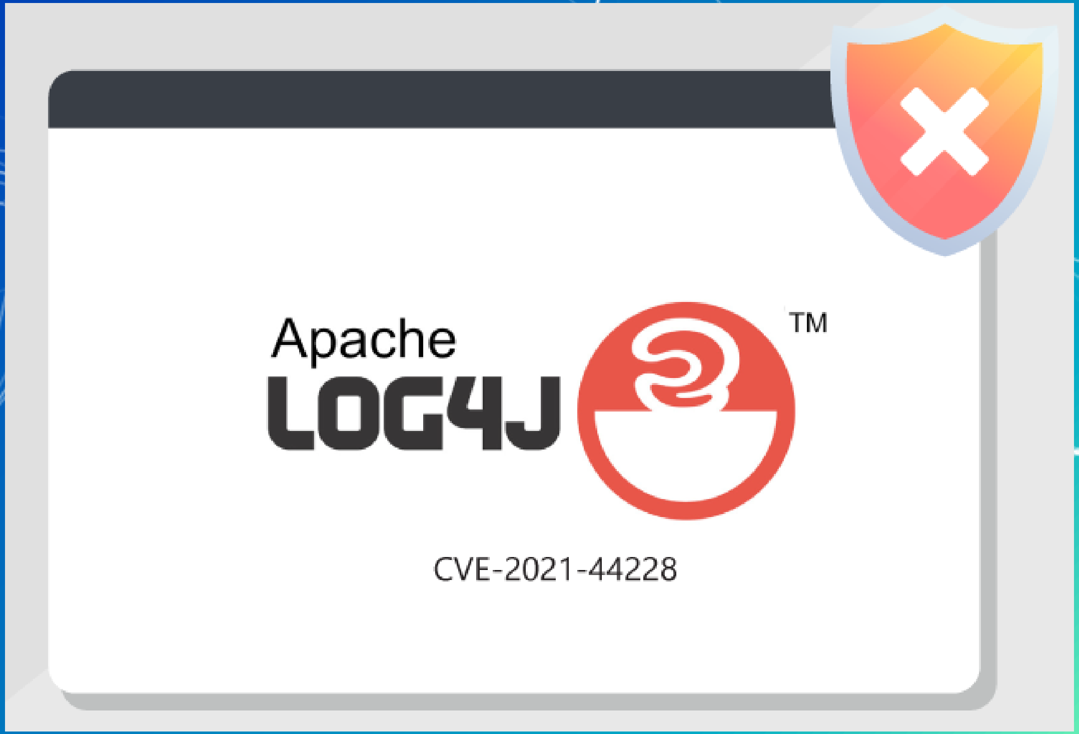


```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello world!\n");
    return 0;
}
```

Shared Libraries: Log4j

- Log4j is an open-source logging library used by millions of computers worldwide.
- At the end of 2021, a major vulnerability was found in the library that attackers could leverage to break into systems, and steal data
- The vulnerability granted an attacker total control over a device running the unpatched version of log4j
- CISA release a repo on GitHub tracking all the products affected by Log4j repo

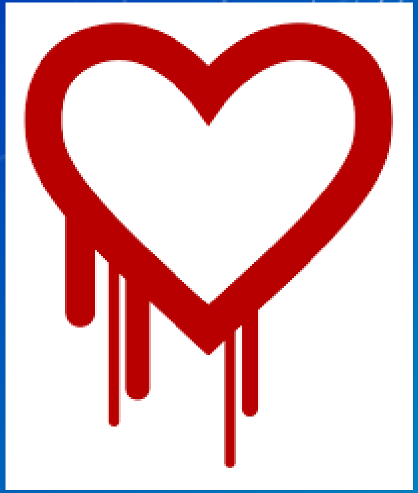


Shared Libraries: OpenSSL Heartbleed

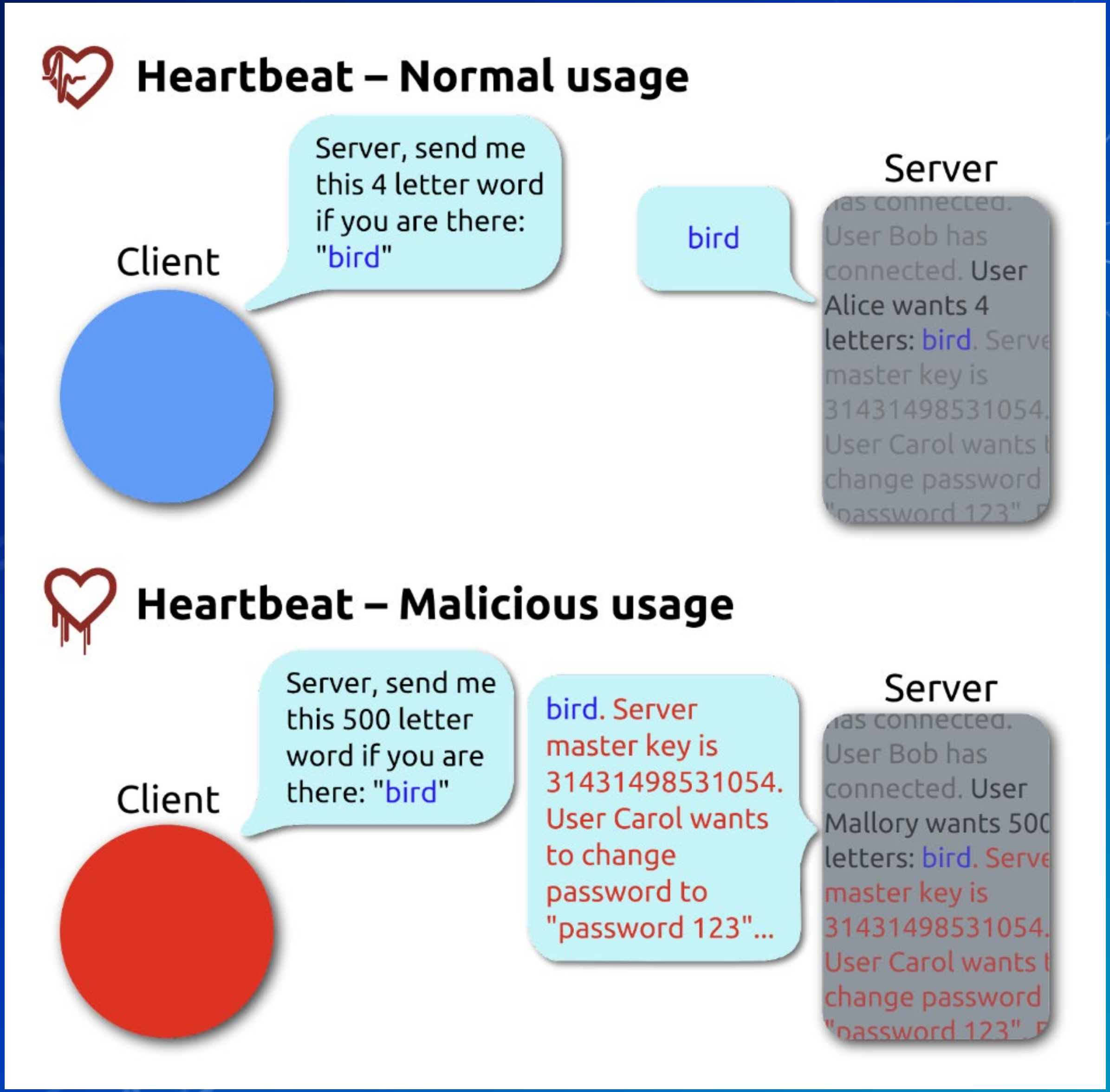
- OpenSSL is a free library that anyone can use, it helps to encrypt data sent over web applications.
- Session management is a key component of secure web applications, there is a need to know when to end the session
- Should we end it every 3hrs, 10hrs, or 24hrs? This is a hard question to answer.
- The SSL standard includes a heartbeat option, which allows a computer at one end of an SSL connection to send a short message to verify that the other computer is still online and get a response back which keeps the session alive.

Shared Libraries: OpenSSL Heartbleed

- The vulnerability was discovered in April of 2014 and involved the heartbeat function to check if the session was still in use.
- An attacker could send a small string to the server but would change the length value to be much larger than it was.
- This causes a buffer overflow to happen on the webserver which could return valuable information to the attacker.
- Some of the major companies that were affected included Tumblr, Google, Yahoo, Intuit (TurboTax), Dropbox, Netflix, and Facebook.



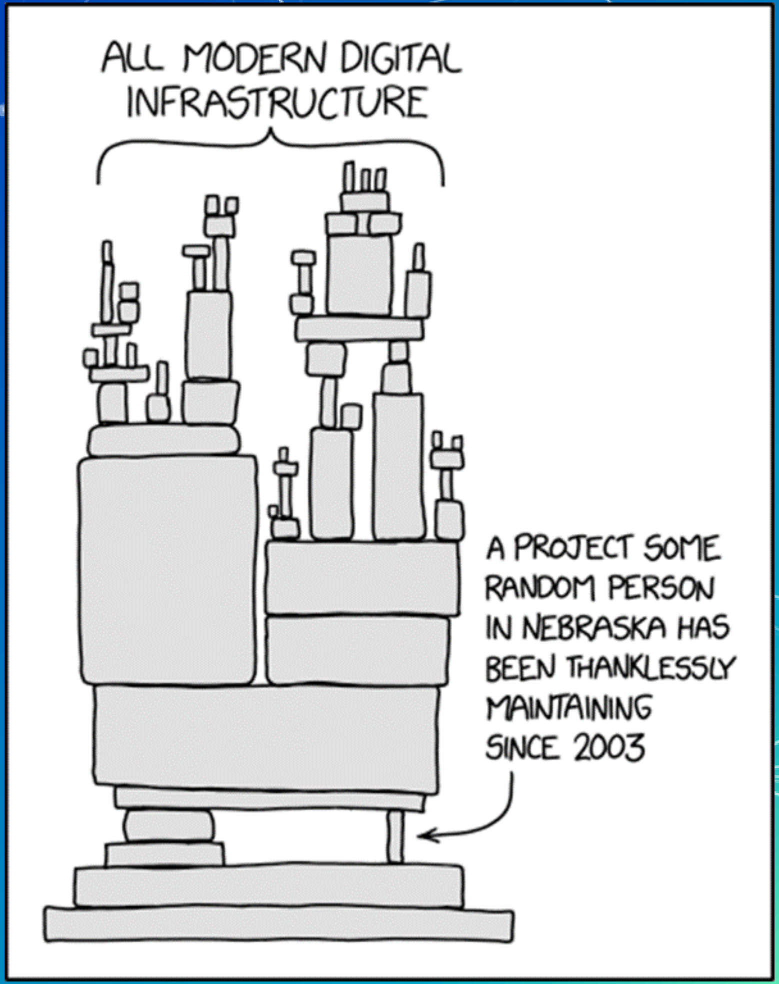
Shared Libraries: OpenSSL Heartbleed


Heartbeat – Normal usage
Client
Server, send me this 4 letter word if you are there: "bird"
bird
Server
has connected. User Bob has connected. User Alice wants 4 letters: bird. Serve master key is 31431498531054. User Carol wants t change password "password 123".
Heartbeat – Malicious usage
Client
Server, send me this 500 letter word if you are there: "bird"
bird. Server master key is 31431498531054. User Carol wants to change password to "password 123"...
Server
has connected. User Bob has connected. User Mallory wants 500 letters: bird. Serve master key is 31431498531054. User Carol wants t change password "password 123".

Old Libraries

- PyCrypto is an old vulnerable library with hashing functions.
- This library is no longer being maintained the hashing functions used in it are brute forceable at this point.
- The two options for a company that use an old library are:

1. Move on to a new library that is still being maintain by the developers.
2. Start maintaining the library own their own.

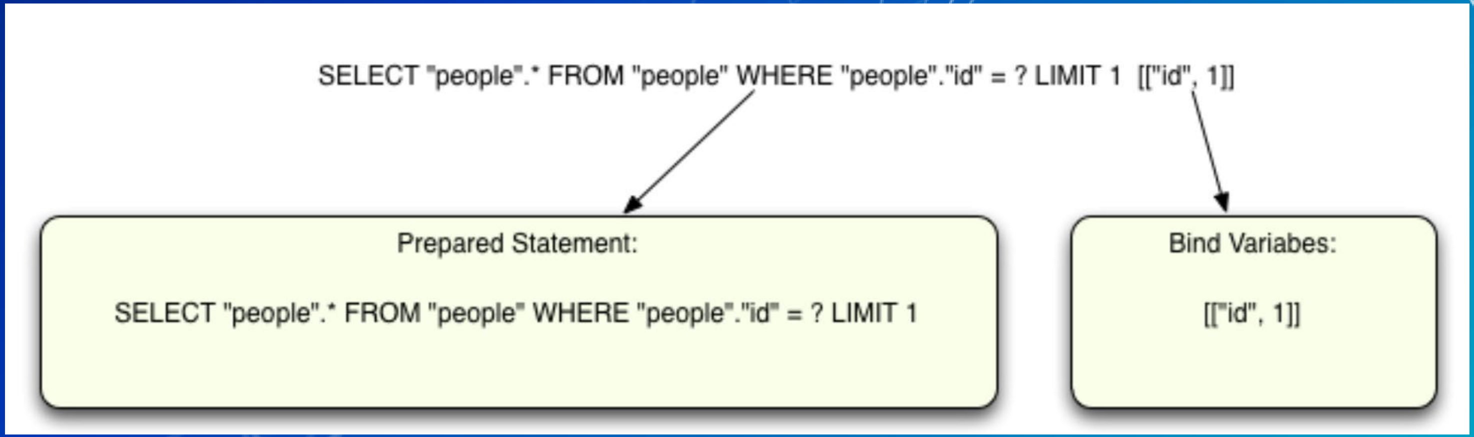

ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Agenda

- Three common vulnerabilities
  - Malformed Inputs
  - System Misconfiguration
  - Memory Management
- Common exploits
  - Web applications
  - Systems
- Common software vulnerabilities
  - Shared Libraries
  - Old Libraries
- Database security
  - Data minimization
- How to find vulnerabilities in code bases
- SIEM

Database Security

- Application should use the lowest possible level of privilege when accessing the database.
- Disable all default accounts that aren't required.
- Remove all unnecessary database functionality, install the minimum required features.
  - Including any unnecessary vendor content
- Use prepared statements


SELECT "people".* FROM "people" WHERE "people"."id" = ? LIMIT 1 [["id", 1]]
Prepared Statement:
SELECT "people".* FROM "people" WHERE "people"."id" = ? LIMIT 1
Bind Variabes:
[["id", 1]]

Data Minimization

- Data minimization is a policy of only collecting information that is needed
- This should be implemented across all applicable processes
- Benefits include:
  - Reduces risk exposure
  - Reduces storage cost

Please return in 10 minutes

Agenda

- Three common vulnerabilities
  - Malformed Inputs
  - System Misconfiguration
  - Memory Management
- Common exploits
  - Web applications
  - Systems
- Common software vulnerabilities
  - Shared Libraries
  - Old Libraries
- Database security
  - Data minimization
- How to find vulnerabilities in code bases
- SIEM

Testing Agenda

- Types of Testing
- Code Scanning
- What is Git?
- Git Requests

Types of Testing

Analysis without running the code

Analysis while running the code with inputs and expecting resulting outputs

Static Testing

- Analyzing source code without executing the program
- Manual Static Testing:
  - Inspections
  - Walkthroughs
  - Technical Reviews
- Automatic Static Testing
  - Control Flow Analysis
  - Data Flow Analysis
  - Failure Detection
- Beneficial because lots of time can be saved if defects are detected early on rather than during the later testing process

Examples of Automated Static Testing

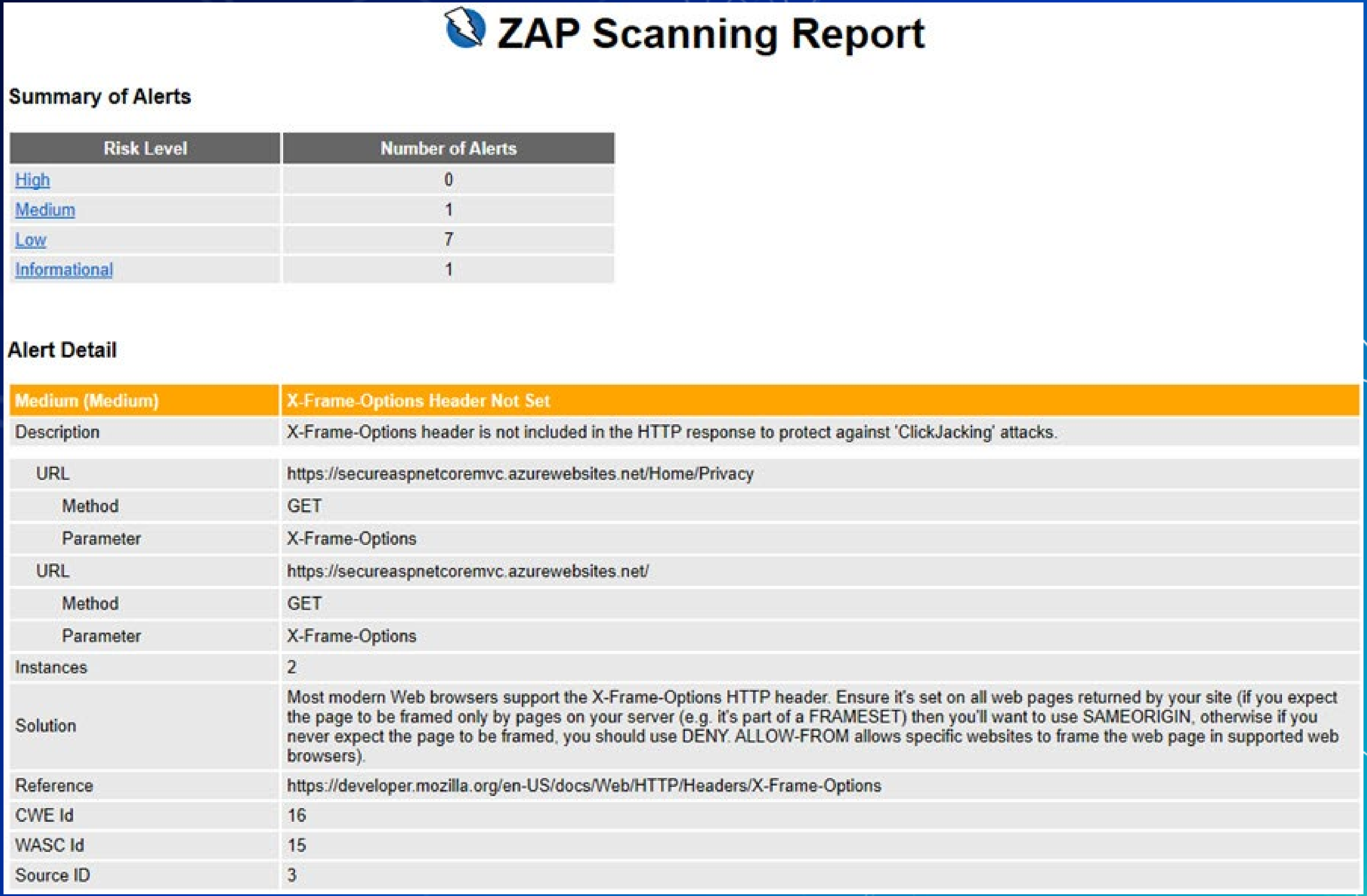
- Veracode
  - Standalone Application
  - CI/CD Integration
- SonarQube
  - Standalone Application
  - CI/CD Integration

Dynamic Testing

- Testing code by running the program and providing inputs to then verify
- Setting up simple test cases that meet basic requirements
- Fuzzing – Entering in random incorrect data until something goes wrong
- Re-running test cases over and over to ensure reliability
  - Tests may be executed in a random sequence to ensure they are not influenced by the system's state.
- Very beneficial to find defects with quality testing in real-world environments

Examples of Automated Dynamic Testing

- Astra PenTest
  - Used with testing:
    - Web and Mobile Applications, Cloud Infrastructure, API, and Networks
  - Standalone Application and CI/CD integration
- OWASP Zap
  - Used with testing:
    - Web application security testing, network ports, and API testing
  - Standalone Application and CI/CD integration



ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	7
Informational	1

Alert Detail

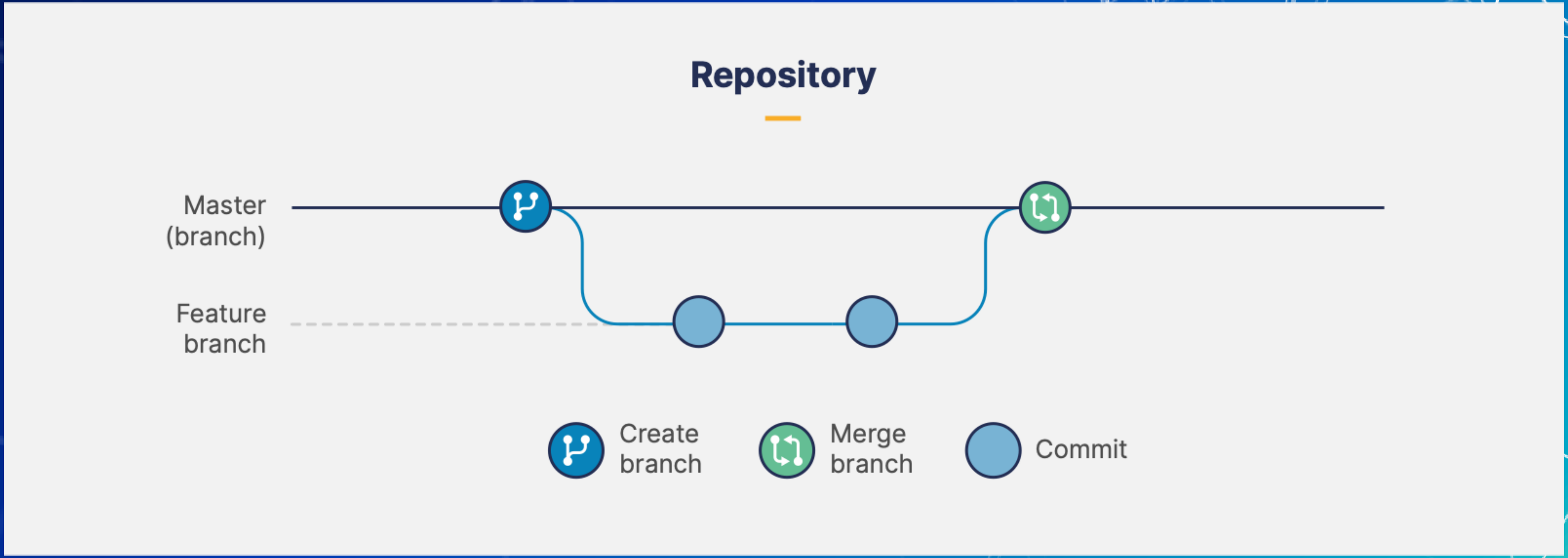
Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	https://secureaspnetcoremvc.azurewebsites.net/Home/Privacy
Method	GET
Parameter	X-Frame-Options
URL	https://secureaspnetcoremvc.azurewebsites.net/
Method	GET
Parameter	X-Frame-Options
Instances	2
Solution	Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	16
WASC Id	15
Source ID	3

What is Git?

- Git is a distributed version control system that tracks changes in any set of computer files
- Used in organizations to have multiple software engineers be able to work on the same files
- Examples:
  - GitHub
  - GitLab
  - Azure DevOps (Newly Popular)

Version Control

How does it work?



Types of Requests

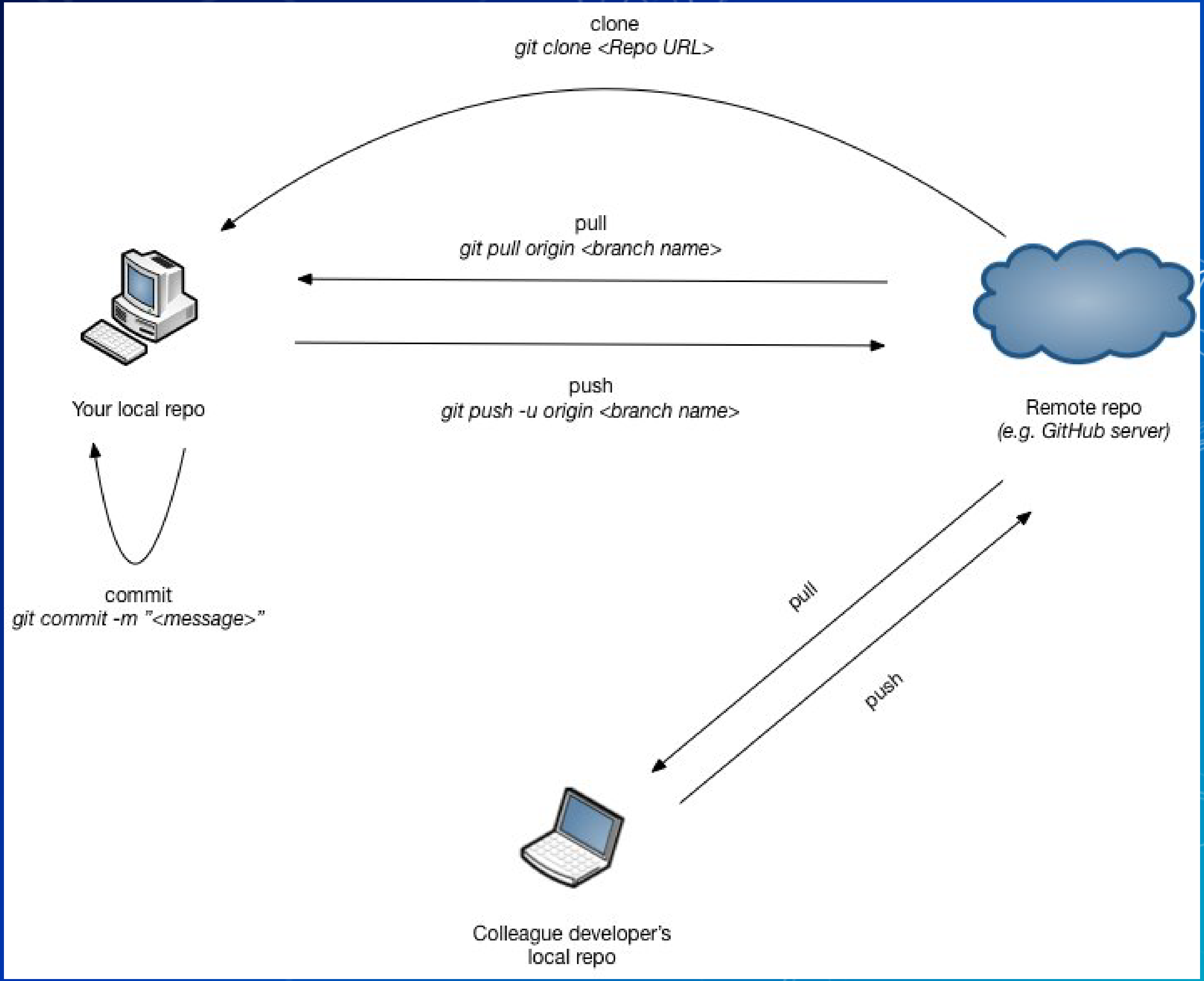
Clone: A method used to create a local copy of a specified repository

Pull: The process of receiving any new code changes on a branch repository

Push: The process of sending new code changes to a branch repository

Merge: The process of combining one branch with another

Fork: A copy of an existing repository in which the new owner disconnects the codebase from previous committers


clone
git clone <Repo URL>
pull
git pull origin <branch name>
push
git push -u origin <branch name>
Your local repo
Remote repo
(e.g. GitHub server)
commit
git commit -m "<message>"
pull
push
Colleague developer's
local repo

Agenda

- Common software vulnerabilities
  - Shared Libraries (CVEs)
  - Old Libraries
- Three common vulnerabilities
  - Malformed Inputs
  - Poor implementation
  - Memory Management
- Understanding the root cause of common vulnerabilities.
- Common exploits
  - Web applications
  - Systems
- Database security
  - Data minimization
- How to find vulnerabilities in code bases
- SIEM

What is a SIEM?

- Security Information and Event Management (SIEM)
- Used to identify, analyze, and respond to security threats
- They pull data from devices, servers, applications

Why use a SIEM?

- Compliance and reporting
- Centralization
- Search and reporting
- Event correlation
- Threat intelligence
- Incident response

Some popular “SIEMs”:

splunk>

LOGPOINT

wazuh.

elastic

LogRhythm

graylog

solarwinds

...many more

Components of a SIEM

- All SIEM implementations *generally* have 3 components
  - This may vary slightly or have different naming schemes on different SIEM brands; however, the general roles remain the same
  - Forwarder (sometimes referred to as an agent)
    - These exist on endpoints and forward data to indexers
  - Indexer (sometimes referred to as a server)
    - Collect, process, store, and query data received from agents
  - Dashboard
    - A web server and graphical user interface used to interact with SIEM implementation

Homework

wazuh.

- You will be configuring a Wazuh SIEM to digest logs on your network
- This Wazuh SIEM will have every component of the SIEM installed on the same device
    - This implementation will also require agents to be installed on most endpoints on your network