Part 1: Logging

Metrics:

Time spent: 3h 32min
True positives: 6
True positives per hours: 1.70 true positives/hour

8 Black Box Tests

For the sake of these black box tests, we are considering the event audit logger actions, using the EventAuditLogger.php functionality, as logs, since these are OpenEMR's auditable logs. Given that the audit logs are saved in the log table of the database, we refer to these steps to read the logs in the database.

Steps to Access Log Table in OpenEMR Database

All of these steps will apply if OpenEMR is run in a Docker container. If OpenEMR is not run in a Docker container, then access the machine with the OpenEMR database and follow the steps of this procedure, starting with step 8.

1. Open the Docker Desktop application and select Containers on the right hand side.
2. In the list of containers, use the search bar to find the mysql database container for OpenEMR. Type "mysql" into the search bar. My container is called mysql-1.
3. Select mysql-1.
4. In the next view, select the tab named Exec. It is in the row that includes other tabs including Logs, Inspect, Bind mounts, etc.. Select Exec.
5. If you do not want to use the Docker Desktop, you can use the command line.
6. Type "docker container ls" into the command line.
    a. Note the container ID of the image "mariadb:<version num>"
7. Type "docker exec -it <mariadb container id> /bin/sh
    a. <mariadb container id> is the value in the container ID column of the "docker container ls" command that is in the same row as the image "mariadb:<version num>"
8. Now that you have access to the Docker container for the OpenEMR database, use the following commands to log into the database.
9. Type in "mysql -u openemr -p openemr"
10. If it prompts "Enter password:" type "openemr".
11. The command line should say "MariaDB [openemr]>"

12. Now you are ready to query the database for logs.

6 Ws of Non-Repudiation

For reference, these are the 6 Ws of non-repudiation from the lecture slides.

• Who – who was involved, who performed the action?
• What – what happened, what resource was acted upon?
• When – when did the action happen?
• Where – where did the action happen, where was the source/destination of the action?
• Why – why did the action happen, why did the person need to perform the action?
• How – how did the action happen, how important is the event (priority)?

Test 1:

1. **ASVS V7.1 Log Content Requirements**
   Unique ID: 7.1.3-0 CWE 778: Insufficient Logging

   Repeatable steps:

   1. If not already running, start/run OpenEMR.

   2. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

   3. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.

   4. In the database, use the command **SELECT * FROM log WHERE category="login";** to see if there are logs for the login.

   You can use a clause like *data>”2024-3-26 12:00:00”* in the WHERE clause to help narrow down the results if necessary.

   Expected results:

   1. The logs show the login entry and satisfies the 6 W’s of non-repudiation.

Test 2:

2. **ASVS V7.1 Log Content Requirements**
   Unique ID: 7.1.3-1 CWE 778: Insufficient Logging

   Repeatable steps:

   1. If not already running, start/run OpenEMR.

2. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.
3. Log out of the account by going to the top right of OpenEMR and hovering over the person.
4. From the resulting menu, select Logout.
5. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.
6. Use the command **select * from log where category="logout";** to see if there are logs for the log out.

   You can use a clause like *data>”2024-3-26 12:00:00”* in the WHERE clause to help narrow down the results if necessary.

   Expected results:

   1. The logs show the logout entry for admin and satisfies the 6 W’s of non-repudiation.

Test 3:

3. ASVS V7.1 Log Content Requirements

Unique ID: 7.1.4-0 CWE 778: Insufficient Logging

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

3. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.

4. Use the command **select id, date, event, user from log WHERE date IS NOT NULL;** to see the logs. In addition use, **select id, date, event, user from log WHERE date IS NULL;** to see that there is no data without a timestamp.

You can use a clause like ***data>”2024-3-26 12:00:00”*** in the WHERE clause to help narrow down the results if necessary.

Expected results:

1. The logs show timestamps for the events logged for .

Test 4:

4. **ASVS V7.1 Log Content Requirements**
   Unique ID: 7.1.3-2 CWE 778: Insufficient Logging

   Repeatable steps:

   1. If not already running, start/run OpenEMR.

   2. Complete a failed login of the admin account, use the credentials “admin” for the username, and “admin” for the password.

   3. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.

   4. Use the command **select date, event, success from log WHERE event="login";** to see if there are logs for the failed login.

   You can use a clause like ***data>”2024-3-26 12:00:00”*** in the WHERE clause to help narrow down the results if necessary.

   Expected results:

   1. The logs show login failure for admin and satisfies the 6 W’s of non-repudiation.

Test 5:

5. **ASVS V7.1 Log Content Requirements**
   Unique ID: 7.1.3-3 CWE 778: Insufficient Logging

   Repeatable steps:

   1. If not already running, start/run OpenEMR.
   2. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.
   3. Select the Patient on the top of the screen. If Patient is not visible at the top of the screen, then select the icon with the 3 horizontal lines in the top left to bring up the menu, then select Patient.
   4. Select New/Search.
   5. In the Search or Add Patient tab. Add the following data.
      a. First Name: Testing
      b. Last Name: Testing2
      c. DOB: 2024-03-01
      d. Sex: Female
   6. Select Create New Patient.
   7. If a popup is shown, select Confirm Create New Patient.
   8. If any additional pop ups show, select OK.
   9. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.

10. Use the command **select *from log WHERE event="patient-record-insert";** to see if there are logs for patient inserts.

    You can use a clause like *data>"2024-3-26 12:00:00"* in the WHERE clause to help narrow down the results if necessary.

    Expected results:

    1. The logs show the inserts for the patient records and satisfies the 6 W's of non-repudiation.

Test 6:

6. ASVS V7.1 Log Content Requirements

Unique ID: 7.1.2-0 CWE 532: Insertion of Sensitive Information into Log File

Repeatable steps:

1. If not already running, start/run OpenEMR.
2. Login with your admin account, unless changed, the credentials should be "admin" for the username, and "pass" for the password.
3. If Patient is not an available menu at the top of the screen, then select the icon with the 3 horizontal lines to bring up the menu.
4. Select Patient and then New/Search.
5. In the Search or Add Patient tab. Add the following data.
    a. First Name: Test
    b. Last Name: Test2
    c. DOB: 2024-03-01
    d. Sex: Female
6. Select Create New Patient
7. If a popup is shown, select Confirm Create New Patient.
8. If any additional pop ups show, then select OK.
9. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.
10. Use the command **select *from log WHERE event="patient-record-insert";** to see if there are logs.

    You can use a clause like *data>"2024-3-26 12:00:00"* in the WHERE clause to help narrow down the results if necessary.

    Expected results:

    1. The logs show no sensitive data in the patient records.

Test 7:

7. ASVS V7.1 Log Content Requirements

Unique ID: 7.1.3-4 CWE 778: Insufficient Logging

Repeatable steps:

1. If not already running, start/run OpenEMR.
2. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.
3. If the Patient menu is not available on the top of the window, then select the icon with the 3 horizontal lines to bring up the menu.
4. Select Patient and then New/Search.
5. In the Search or Add Patient tab. Add the following data, if the patient is not already created.
    a. First Name: Testing
    b. Last Name: Testing2
    c. DOB: 2024-03-01
    d. Sex: Female
6. Select Create New Patient.
7. If a popup is shown, select Confirm Create New Patient.
8. If additional pop ups occur, then select OK.
9. If the Calendar menu is not available in the top bar, then select the icon with the 3 horizontal lines to bring up the menu.
10. Select Calendar.
11. Press the plus button.
12. Use patient Testing Testing2 if no patient is currently inputted.
13. Select Save.
14. If “Provider not available” popup shows select OK.
15. Verify the new appointment is on the calendar.
16. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.
17. Use the command **select *from log WHERE event="scheduling-insert";** to see if there are logs.

    You can use a clause like *data>”2024-3-26 12:00:00”* in the WHERE clause to help narrow down the results if necessary.

Expected results:

1. The logs show the inserts for the schedule and satisfies the 6 W’s of non-repudiation.

Test 8:

8. ASVS V7.1 Log Content Requirements

Unique ID: 7.1.3-5 CWE 778: Insufficient Logging

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.
3. If Patient is not available in the top menu, then select the icon with the 3 horizontal lines to bring up the menu.
4. Select Patient and then New/Search.
5. In the Search or Add Patient tab. Add the following data, if the patient is not already created.
    a. First Name: Testing
    b. Last Name: Testing2
    c. DOB: 2024-03-01
    d. Sex: Female
6. Select Create New Patient.
7. If a popup is shown, select Confirm Create New Patient.
8. If any additional pop ups show, then select OK.
9. If Calendar is not available in the top bar, then select the icon with the 3 horizontal lines to bring up the menu.
10. Select Calendar.
11. Press the plus button.
12. Use patient Testing Testing2 if no patient is currently inputted.
13. Select Save.
14. If “Provider not available” popup shows select OK.
15. Verify the new appointment is on the calendar.
16. Double click on the new appointment. A pop up with appointment details should show.
17. Select Delete.
18. Select OK in the confirmation popup.
19. Verify the appointment is deleted from the calendar.
20. Use the steps from Steps to Access Log Table in OpenEMR Database to log into the OpenEMR database.
21. Use the command **select *from log WHERE event="scheduling-delete";** to see if there are logs.

    You can use a clause like *data>”2024-3-26 12:00:00”* in the WHERE clause to help narrow down the results if necessary.

    Expected results:

    1. The logs show the appointment was deleted from the schedule and the log shows the 6 W’s of non-repudiation.

Commentary on the Adequacy of OpenEMR’s Logging

Based on the 8 tests that I have conducted, OpenEMR has been always been successful in the who, what and when, but is often not successful on the where, why and how. There is a chance that this information is only available as part of the encrypted comments, but it should be more easily accessible than that. There are timestamps on all of the logs, and no personal information is logged in clear text.

2. Attack Trees

Section 1

1.
    a. EXOTIC LILY
    b. FIN4
    c. Fox Kitten
    d. Leviathan
    e. menuPass
2.
    a. EXOTIC LILY
        i. Acquire Infrastructure: Domains
        ii. Establish Accounts: Social Media Accounts
        iii. Exploitation for Client Execution
        iv. Gather Victim Identity Information: Email Addresses
        v. Phishing: Spearphishing Attachment
        vi. Search Closed Sources
        vii. Search Open Websites/Domains: Social Media
        viii. Search Victim-Owned Websites
        ix. Stage Capabilities: Upload Malware
        x. Web Service
    b. FIN4
        i. Application Layer Protocol: Web Protocols
        ii. Command and Scripting Interpreter: Visual Basic
        iii. Email Collection: Remote Email Collection
        iv. Hide Artifacts: Email Hiding Rules
        v. Input Capture: Keylogging
        vi. Input Capture: GUI Input Capture
        vii. Phishing: Spearphishing Attachment
        viii. Phishing: Spearphishing Link
        ix. Proxy: Multi-hop Proxy
        x. User Execution: Malicious Link
    c. Fox Kitten
        i. Account Discovery: Local Account
        ii. Account Discovery: Domain Account
        iii. Archive Collected Data: Archive via Utility
        iv. Browser Information Discovery
        v. Brute Force

vi. Command and Scripting Interpreter
vii. Data from Cloud Storage
viii. Data from Information Repositories
ix. Data from Local System
x. Data from Network Shared Drive

d. Leviathan
i. Acquire Infrastructure: Domains
ii. Archive Collected Data
iii. BITS Jobs
iv. Data Staged: Local Data Staging
v. Deobfuscate/Decode Files or Information
vi. Drive-by Compromise
vii. Exploitation for Client Execution
viii. External Remote Services
ix. Ingress Tool Transfer
x. Internal Spearphishing

e. menuPass
i. Account Discovery: Domain Account
ii. Archive Collected Data
iii. Automated Collection
iv. Data from Local System
v. Data Staged: Local Data Staging
vi. File and Directory Discovery
vii. Rename System Utilities
viii. Native API
ix. Network Service Discovery
x. Obfuscated Files or Information

3.

	EXOTIC LILY	**FIN4**	**Fox Kitten**	**Leviathan**	**menuPass**
Abuse Elevation Control Mechanism					
Access Token Manipulation					
Account Access Removal					
Account Discovery			x		x
Account Manipulation					
Acquire Access					
Acquire Infrastructure	x			x	x

Active Scanning					
Adversary-in-the-Middle					
Application Layer Protocol		x			
Application Window Discovery					
Archive Collected Data			x	x	x
Audio Capture					
Automated Collection					x
Automated Exfiltration					
BITS Jobs				x	
Boot or Logon Autostart Execution				x	
Boot or Logon Initialization Scripts					
Browser Extensions					
Browser Information Discovery			x		
Browser Session Hijacking					
Brute Force			x		
Build Image on Host					
Clipboard Data					
Cloud Administration Command					
Cloud Infrastructure Discovery					
Cloud Service Dashboard					
Cloud Service Discovery					
Cloud Storage Object Discovery					
Command and Scripting Interpreter		x		x	x
Communication Through Removable Media					
Compromise Accounts				x	
Compromise Client Software Binary					
Compromise Infrastructure					
Container Administration Command					
Container and Resource Discovery					
Content Injection					

Create Account					
Create or Modify System Process					
Credentials from Password Stores					
Data Destruction					
Data Encoding					
Data Encrypted for Impact					
Data from Cloud Storage			x		
Data from Configuration Repository					
Data from Information Repositories			x		
Data from Local System			x		x
Data from Network Shared Drive			x		
Data from Removable Media					
Data Manipulation					
Data Obfuscation					
Data Staged				x	x
Data Transfer Size Limits					
Debugger Evasion					
Defacement					
Deobfuscate/Decode Files or Information				x	x
Deploy Container					
Develop Capabilities					
Device Driver Discovery					
Direct Volume Access					
Disk Wipe					
Domain Policy Modification					
Domain Trust Discovery					
Drive-by Compromise				x	
Dynamic Resolution					x
Email Collection		x			
Encrypted Channel					

Endpoint Denial of Service					
Establish Accounts	x		x	x	
Event Triggered Execution				x	
Execution Guardrails					
Exfiltration Over Alternative Protocol					
Exfiltration Over C2 Channel				x	
Exfiltration Over Other Network Medium					
Exfiltration Over Physical Medium					
Exfiltration Over Web Service				x	
Exploitation for Client Execution	x			x	
Exploitation for Credential Access					
Exploitation for Defense Evasion					
Exploitation of Remote Services					
Exploit Public-Facing Application			x		x
External Remote Services			x	x	
Fallback Channels					
File and Directory Discovery			x		x
File and Directory Permissions Modification					
Financial Theft					
Firmware Corruption					
Forced Authentication					
Forge Web Credentials					
Gather Victim Host Information	x				
Gather Victim Identity Information				x	
Gather Victim Network Information					
Gather Victim Org Information					
Group Policy Discovery					
Hardware Additions					

Hide Artifacts		x			
Hijack Execution Flow					x
Impair Defenses					
Impersonation					
Implant Internal Image					
Indicator Removal					x
Indirect Command Execution					
Ingress Tool Transfer				x	
Inhibit System Recovery					
Input Capture		x			x
Inter-Process Communication				x	
Internal Spearphishing				x	
Lateral Tool Transfer					
Log Enumeration					
Masquerading			x		x
Modify Authentication Process					
Modify Cloud Compute Infrastructure					
Modify Registry					
Modify System Image					
Multi-Factor Authentication Interception					
Multi-Factor Authentication Request Generation					
Multi-Stage Channels					
Native API					x
Network Boundary Bridging					
Network Denial of Service					
Network Service Discovery			x		
Network Share Discovery					
Network Sniffing					
Non-Application Layer Protocol					
Non-Standard Port					
Obfuscated Files or Information			x	x	x
Obtain Capabilities					

Office Application Startup					
OS Credential Dumping			x	x	x
Password Policy Discovery					
Peripheral Device Discovery					
Permission Groups Discovery					
Phishing	x	x		x	x
Phishing for Information					
Plist File Modification					
Power Settings					
Pre-OS Boot					
Process Discovery					
Process Injection				x	x
Protocol Tunneling			x	x	
Proxy		x		x	x
Query Registry			x		
Reflective Code Loading					
Remote Access Software					x
Remote Services			x	x	
Remote Service Session Hijacking					
Remote System Discovery					
Replication Through Removable Media					
Resource Hijacking					
Rogue Domain Controller					
Rootkit					
Scheduled Task/Job			x		x
Scheduled Transfer					
Screen Capture					
Search Closed Sources	x				
Search Open Technical Databases					
Search Open Websites/Domains	x				
Search Victim-Owned Websites	x				
Serverless Execution					

Server Software Component			x	x	
Service Stop					
Shared Modules					
Software Deployment Tools					
Software Discovery					
Stage Capabilities	x				
Steal Application Access Token					
Steal or Forge Authentication Certificates					
Steal or Forge Kerberos Tickets					
Steal Web Session Cookie					
Subvert Trust Controls				x	x
Supply Chain Compromise					
System Binary Proxy Execution				x	
System Information Discovery					
System Location Discovery					
System Network Configuration Discovery					x
System Network Connections Discovery					
System Owner/User Discovery					
System Script Proxy Execution					
System Service Discovery					
System Services					
System Shutdown/Reboot					
System Time Discovery					
Taint Shared Content					
Template Injection					
Traffic Signaling					
Transfer Data to Cloud Account					
Trusted Developer Utilities Proxy Execution					
Trusted Relationship					x
Unsecured Credentials			x		
Unused/Unsupported Cloud Regions					

Use Alternate Authentication Material					
User Execution	x	x		x	x
Valid Accounts		x	x	x	x
Video Capture					
Virtualization/Sandbox Evasion					
Weaken Encryption					
Web Service	x		x	x	
Windows Management Instrumentation				x	x
XSL Script Processing					

4.

	Persistence	Privilege Escalation	Defense Evasion	Initial Access
T1197 BITS Jobs	x		x	
T1547 Boot or Logon Autostart Execution	x	x		
T1546 Event Triggered Execution	x	x		
T1133 External Remote Services	x			x
T1078 Valid Accounts	x	x	x	x

5.

	Exfiltration	**Reconnaissance**	**Credential Access**	**Execution**	**Persistenc**	**Privilege E**	**Defense Evasion**	**Discovery**	**Resource Development**	**Command and Control**	**Collection**	**Initial Access**	**Lateral Movement**

					e	**s c al at io n**							
T1087 Account Discovery								M1028 Operating System Configuration					
T1583 Acquire Infrastructure									M1056 Pre-compromise				
T1071 Application Layer Protocol										M1031 Network Intrusion Prevention			
T1560 Archive Collected Data											M1047 Audit		
T1119 Automated Collection											M1041 Encrypt Sensitive Information		
T1197					M		M103						

BITS Jobs					1037 Filter Network Traffic		7 Filter Network Traffic						
T1547 Boot or Logon Autostart Execution					DS0017 Command Command Execution	DS0017 Command Command Execution							
T1217 Browser								DS0022					

Information Discovery								File File Access					
T1110 Brute Force			M1032 Multi-factor Authentication										
T1059 Command and Scripting Interpreter				M1049 Antivirus/Antimalware									
T1586 Compromise Accounts									M1056 Pre-compromise				
T1530 Data from Cloud Storage											M1047 Audit		
T1213 Data from Information Repositories											M1047 Audit		
T1005 Data from Local System											M1057 Data Loss Prevention		
T1039 Data											N/A		

from Network Shared Drive													
T1074 Data Staged											N/A		
T1140 Deobfuscate/Decode Files or Information											N/A		
T1189 Drive-by Compromise												M1048 Application Isolation and Sandboxing	
T1568 Dynamic Resolution										M1031 Network Intrusion Prevention			
T1585 Establish Accounts									M1056 Pre-compromise				
T1546 Event Triggered Execution					N/A	N/A							

T1041 Exfiltration Over C2 Channel	M1057 Data Loss Prevention												
T1567 Exfiltration Over Web Service	M1021 Restrict Web-Based Content												
T1203 Exploitation for Client Execution				M1048 Application Isolation and Sandboxing									
T1190 Exploit Public-Facing Application												M1050 Exploit Protection	
T1133 External Remote Services					M1042 Disable or Remove Featur							M1042 Disable or Remove Feature or Program	

					e or Program								
T1008 Fallback Channels										M1031 Network Intrusion Prevention			
T1083 File and Directory Discovery								N/A					
T1589 Gather Victim Identity Information		M1056 Pre-compromise											
T1564 Hide Artifacts							N/A						
T1574 Hijack Execution Flow					M10013 Application Code	M10013 Application Code	M1013 Application Developer Guidance						

					veloperGuidance	veloperGuidance							
T1070 Indicator Removal							M1041 Encrypt Sensitive Information						
T1105 Ingress Tool Transfer										M1031 Network Intrusion Prevention			
T1056 Input Capture			N/A								N/A		
T1559 Inter-Process Communication				M1013 Application Developer Guidance									
T1534 Internal Spearphishing													N/A

T1036 Masquerading							M1049 Antivirus/Antimalware						
T1106 Native API				M1040 Behavior Prevention on Endpoint									
T1046 Network Service Discovery								M1042 Disable or Remove Feature or Program					
T1027 Obfuscated Files or Information							M1049 Antivirus/Antimalware						
T1003 OS Credential Dumping			M1015 Active Directory Configuration										
T1566 Phishing												M1049 Antivirus/Antimalware	

T1055 Process Injection						M1040 Behavior Prevention on Endpoint	M1040 Behavior Prevention on Endpoint						
T1572 Protocol Tunneling										M1037 Filter Network Traffic			
T1090 Proxy										M1037 Filter Network Traffic			

T1012 Query Registry								N/A					
T1219 Remote Access Software								M1038 Execution Prevention					
T1021 Remote Services													M1042 Disable or Remove Feature or Program
T1053 Scheduled Task/Job					M1047 Audit	M1047 Audit	M1047 Audit						
T1597 Search Closed Sources		M1056 Pre-compromise											
T1593 Search Open Websites/Domains		M1013 Application Developer Guidance											

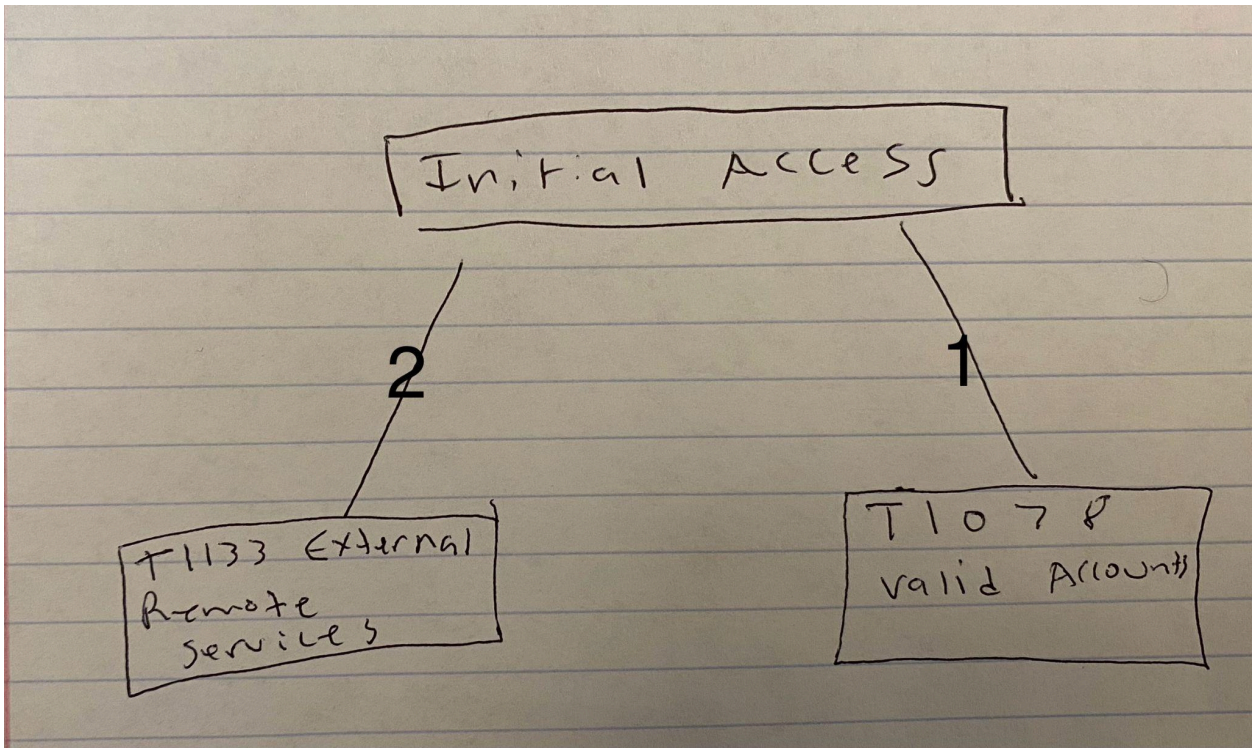
T1594 Search Victim-Owned Websites		M1056 Pre-compromise											
T1505 Server Software Component					M1047 Audit								
T1608 Stage Capabilities									M1056 Pre-compromise				
T1553 Subvert Trust Controls							M1038 Execution Prevention						
T1195 Supply Chain Compromise												M1051 Update Software	

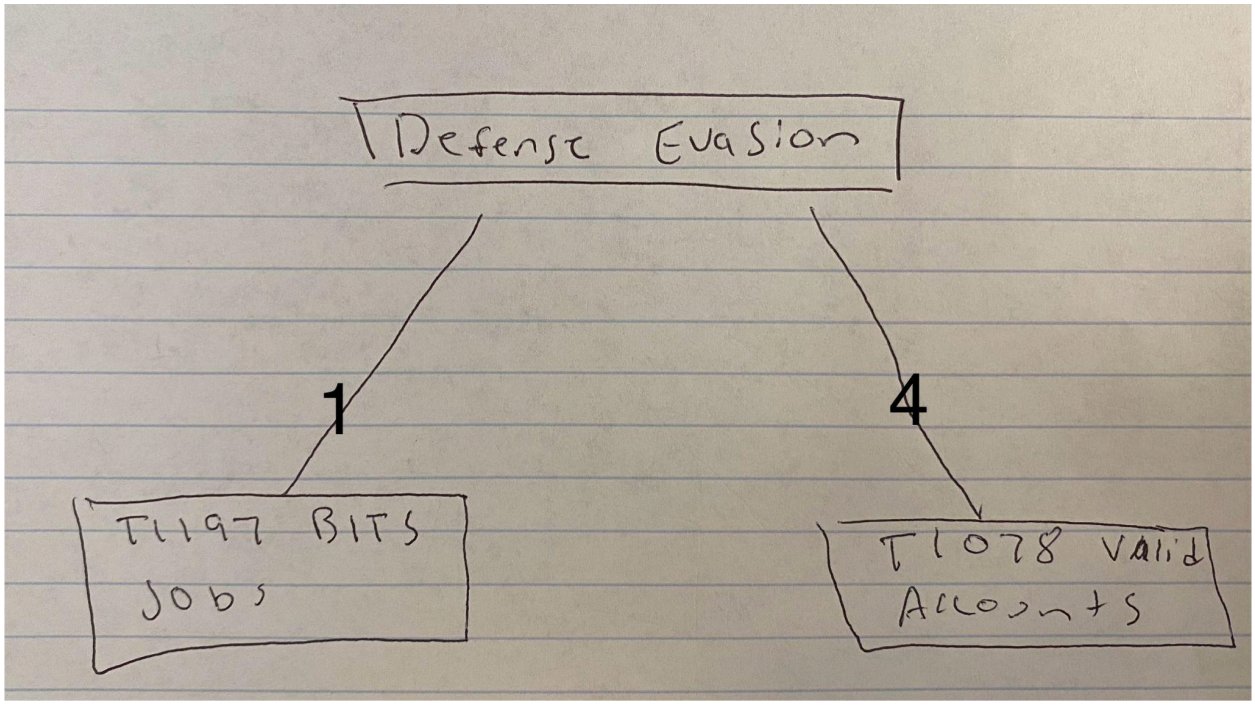
T1218 System Binary Proxy Execution							M1042 Disable or Remove Feature or Program						
T1016 System Network Configuration Discovery								N/A					
T1199 Trusted Relationship												M1032 Multi-factor Authentication	
T1552 Unsecured Credentials			M1015 Active Directory Configuration										
T1204 User Execution			M1040 Behavior Prevention on Endpoint										
T1078 Valid Accounts						M1036	M1036 Account Use	M1036 Account Use				M1036 Account Use	

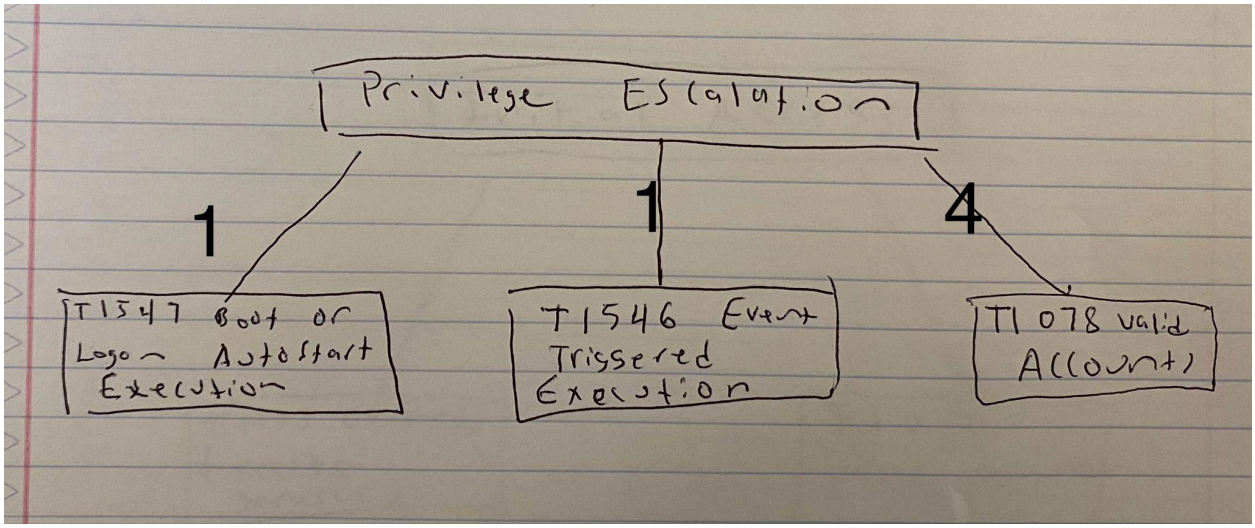
						Account Use Policies	Policies	Policies				Policies	
T1102 Web Service										M1031 Network Intrusion Prevention			
T1047 Windows Management Instrumentation				M1040 Behavior Prevention on Endpoint									

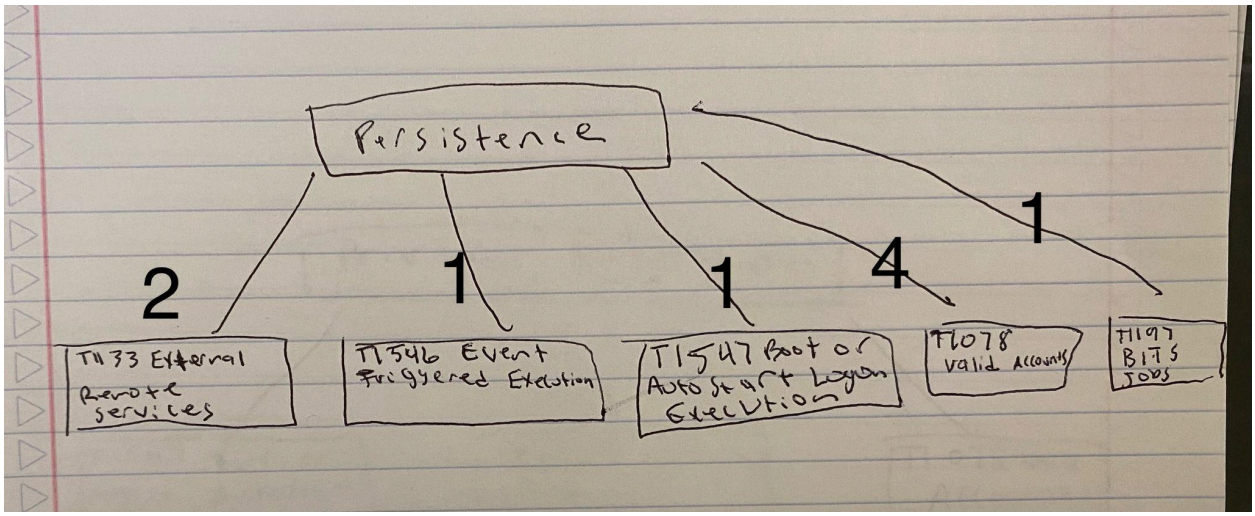
Section 2

6.

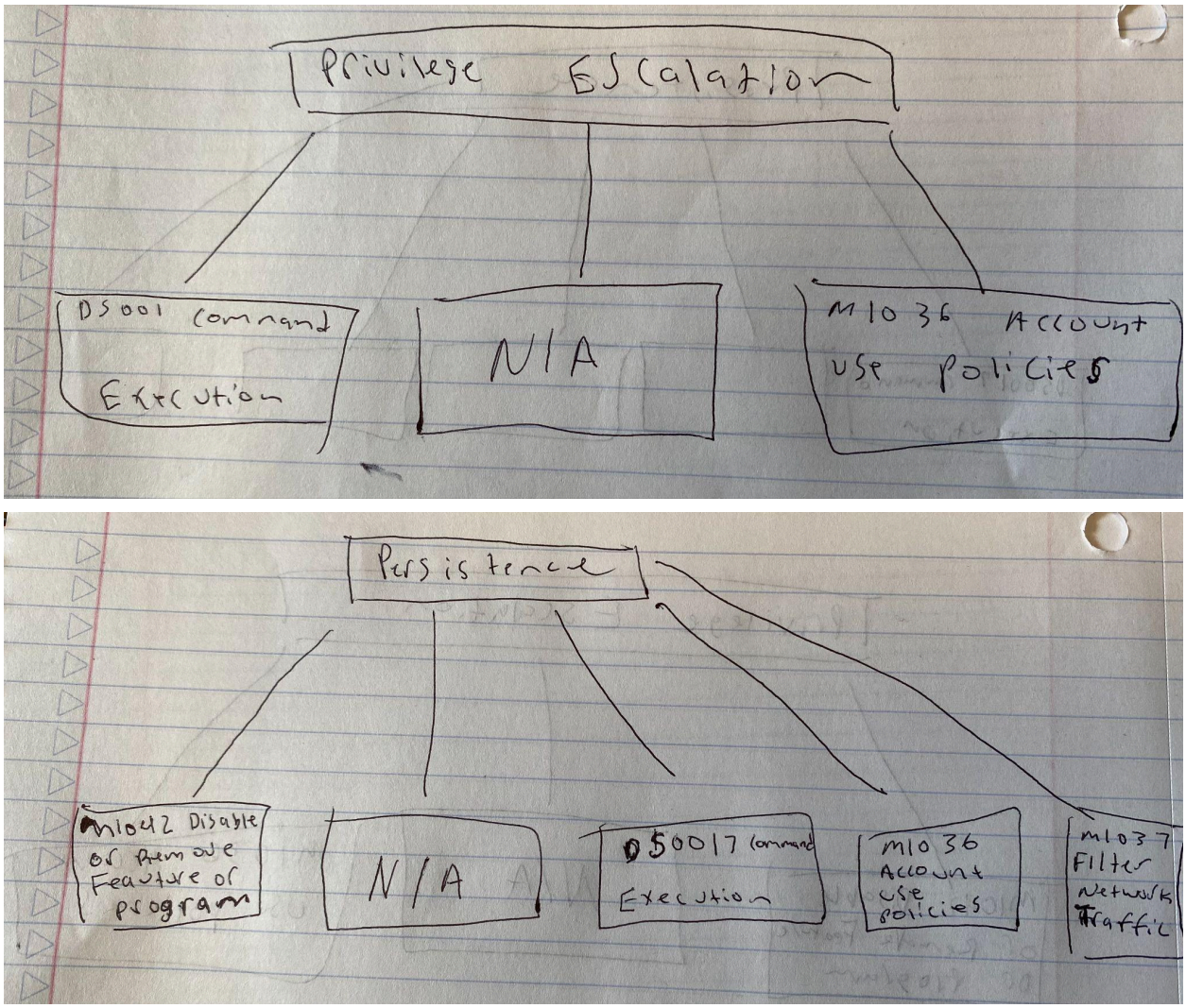

Initial Access
2
1
T1133 External Remote Services
T1078 Valid Accounts

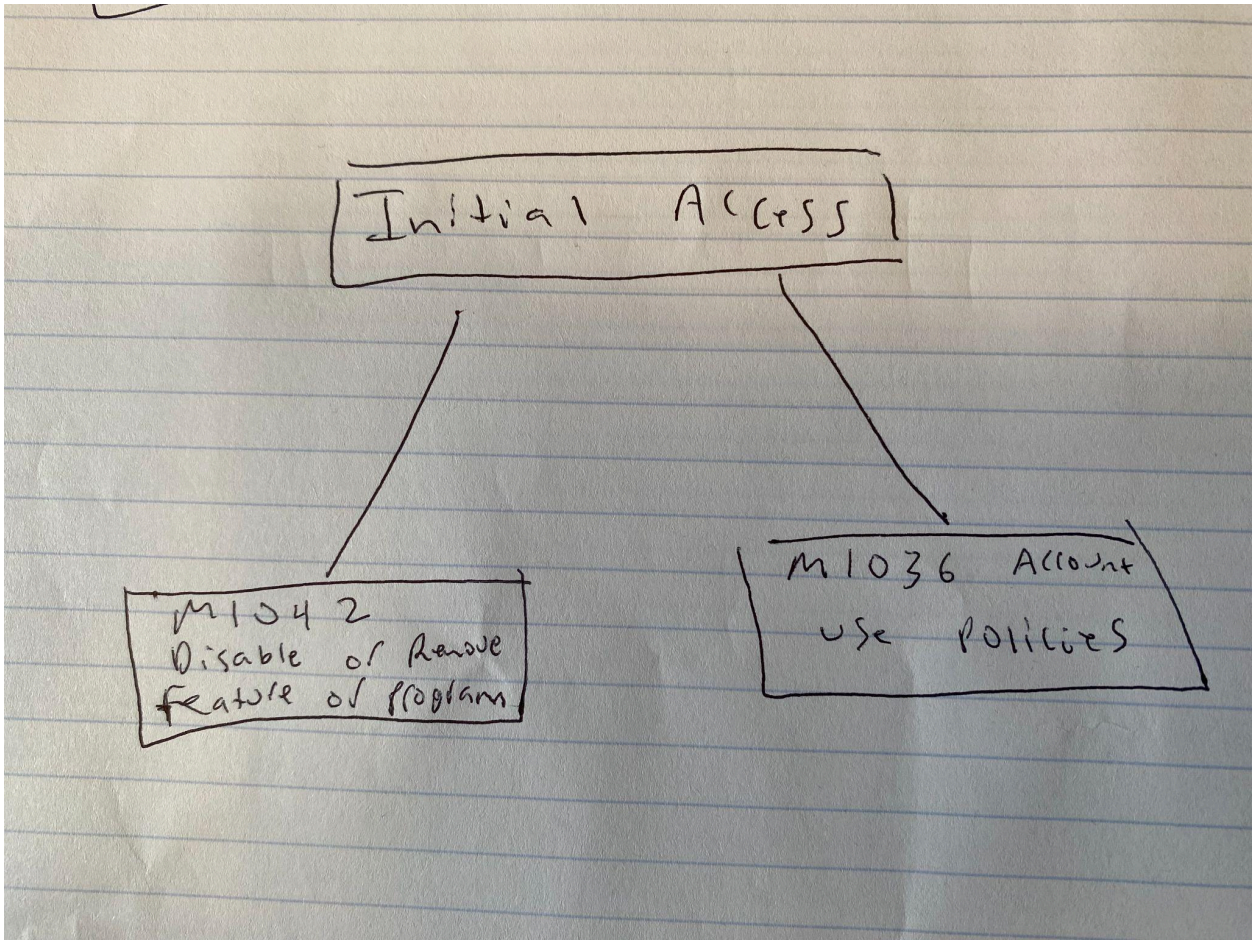

Defense Evasion
1
4
T1197 BITS Jobs
T1078 Valid Accounts

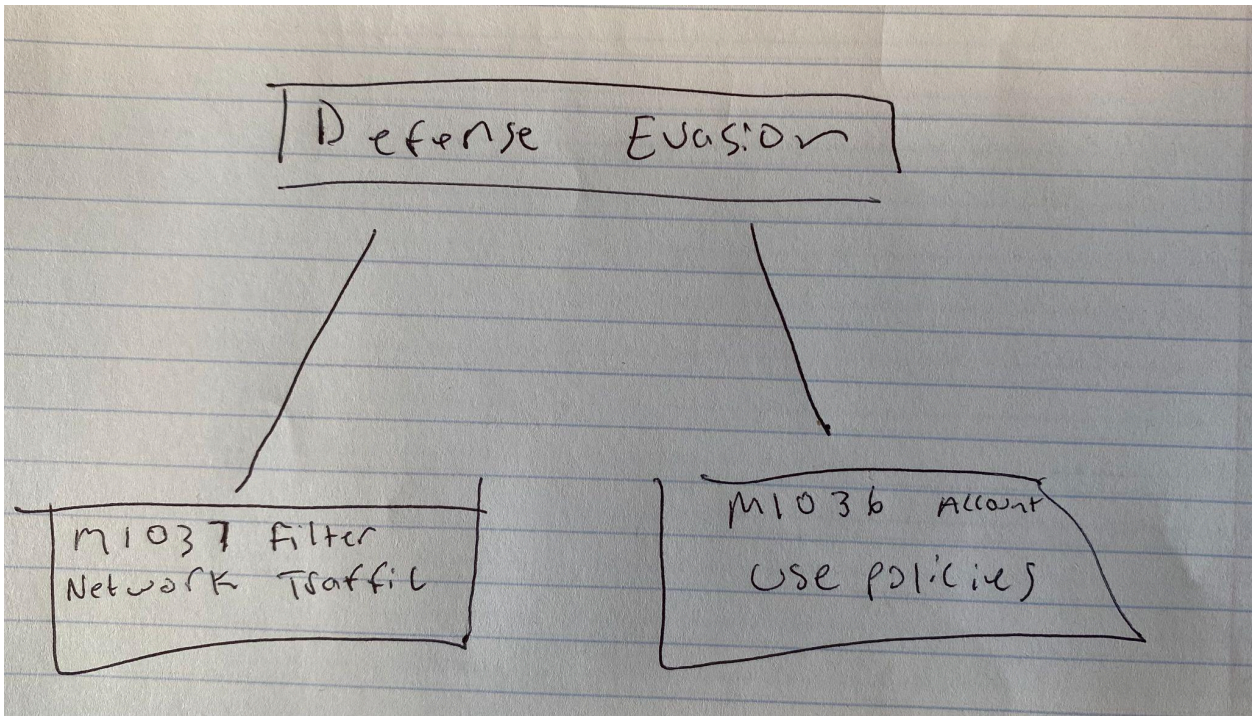

Privilege Escalation
1
1
4
T1547 Boot or Logon Autostart Execution
T1546 Event Trissered Execution
T1078 valid Account


Persistence
2
1
1
4
1
T1133 External Remote services
T1546 Event Triggered Execution
T1547 Boot or Autostart Logon Execution
T1078 valid Accounts
T1197 BITS Jobs

7.


Privilege Escalation
D5001 command Execution
N/A
M1036 Account use policies
Persistence
M1042 Disable or Remove Feature or program
N/A
D50017 command Execution
M1036 Account use policies
M1037 Filter Network Traffic


Initial Access
M1042
Disable or Remove
feature or program
M1036 Account
use Policies


Defense Evasion
M1037 filter
Network Traffic
M1036 Account
use policies

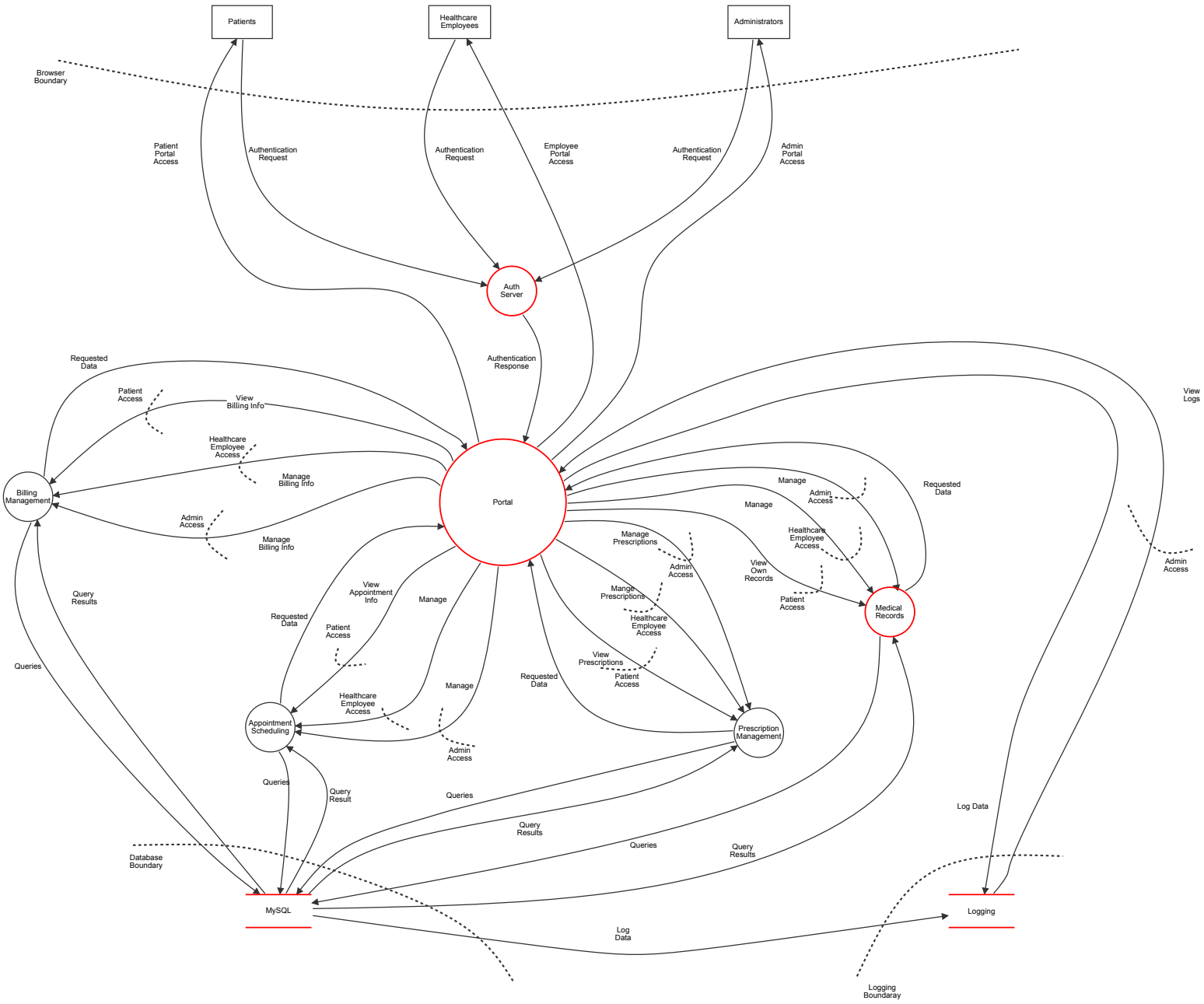
3. Threat Model

High level system description

Not provided

Summary

Total Threats	18
Total Mitigated	0
Not Mitigated	18
Open / High Priority	0
Open / Medium Priority	18
Open / Low Priority	0
Open / Unknown Priority	0


Patients
Healthcare Employees
Administrators
Browser Boundary
Patient Portal Access
Authentication Request
Authentication Request
Employee Portal Access
Authentication Request
Admin Portal Access
Auth Server
Authentication Response
Requested Data
Patient Access
View Billing Info
Healthcare Employee Access
Manage Billing Info
Billing Management
Admin Access
Manage Billing Info
Portal
Manage
Admin Access
Manage
Requested Data
Manage Prescriptions
Admin Access
Healthcare Employee Access
View Own Records
Patient Access
Medical Records
View Logs
Admin Access
Query Results
Queries
Requested Data
View Appointment Info
Patient Access
Manage
Healthcare Employee Access
Admin Access
Manage
Requested Data
Manage Prescriptions
Healthcare Employee Access
View Prescriptions
Patient Access
Appointment Scheduling
Prescription Management
Queries
Query Result
Queries
Query Results
Queries
Query Results
Log Data
Database Boundary
MySQL
Logging
Log Data
Logging Boundaray

Auth
Server (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
1	Elevation of Privilege Game - Card (S8)	Spoofing	Medium	Open		Elevation of Privilege Game - Card (S8) Card Description: An attacker could steal credentials stored on the server and reuse them (for example, a key is stored in a world-readable file). Reasoning: If an attacker were to gain access to credentials improper file permissions, they could gain unauthorized access to sensitive health records and personal patient information. How to mitigate: Implement strict file permissions and use access control lists as well as filesystem permissions to restrict access.	
10	Cornucopia - Authentication - Weak Password Policy (Card A7):	Spoofing	Medium	Open		Cornucopia - Authentication - Weak Password Policy (Card A7): Card Description: Threat: Cecilia can use brute force and dictionary attacks against accounts due to insufficient complexity, length, expiration, and re-use requirements for passwords. Reasoning: Weak password requirements can allow unauthorized access to the application which can lead to data breaches or changes to the store data. How to mitigate: Implement a strong password policy that requires complexity, length, and require that the 2-factor implementation is required. Also implement locked accounts after so many attempts or require a timeout so attempts can not be done back-to-back.	
11	Cornucopia - Session Management - Insecure Session Handling (Card S9):	Spoofing	Medium	Open		Cornucopia - Session Management - Insecure Session Handling (Card S9): Card Description: Threat: Ivan can steal session identifiers because they are transmitted over insecure channels or included in URLs. Reasoning: If session hijacking occurs it could allow an attack to impersonate a legitimate user which could then provide the attacker access to sensitive information or allow them to perform unauthorized actions. How to mitigate: Implement secure cookies with “HttpOnly” and “Secure” flags. Ensure all traffic is moved to HTTPs.	
12	Cornucopia - Authorization - Excessive Privileges (Card A5):	Elevation of privilege	Medium	Open		Cornucopia - Authorization - Excessive Privileges (Card A5): Card Description: Threat: Chad can access resources he should not be able to due to missing authorization checks or excessive privileges. Reasoning: If an attacker can obtain more privilege than necessary due to missing checks, than they could potentially have access to sensitive information or system configurations as well as the ability to modify the data. How to mitigate: Implement the principle of least privilege to all users and ensure that there are regular reviews and audits to check user permissions.	

Number	Title	Type	Priority	Status	Score	Description	Mitigations
13	Cornucopia - Cryptography - Inadequate Encryption (Card C8):	Spoofing	Medium	Open		Cornucopia - Cryptography - Inadequate Encryption (Card C8): Card Description: Threat: Eoin can access stored business data because it is not securely encrypted or hashed. Reasoning: Unencrypted sensitive data can be easily accessed and exploited. This includes data like passwords, personal identifiable information, and session identifiers. How to mitigate: Implement strong cryptographic algorithms of encrypting and hashing the data. Ensure that the encryption keys used are stored and managed securely with limited access.	
18	LINDDUN - Non-repudiation (Nr1 - Credentials Non-repudiation):	Repudiation	Medium	Open		LINDDUN - Non-repudiation (Nr1 - Credentials Non-repudiation): Card Description: Threat: Person cannot deny having authenticated to a service. Reasoning: Users may not be willing to access sensitive health data, or provide it, if their data can be linked back to their personal identifiable information. How to mitigate: Implement anonymous or pseudonymous access methods where non-repudiation is not required.	

Medical Records (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
15	LINDDUN - Non-compliance (Nc1 - Disproportionate Collection):	Information disclosure	Medium	Open		LINDDUN - Non-compliance (Nc1 - Disproportionate Collection): Card Description: Threat: More personal data are being collected than required for the purpose. Reasoning: Collecting excessive and unneeded data can lead to privacy risks and be non-compliant with HIPAA. How to mitigate: Implement the principle of data minimization and regularly review the data collection practices so that only the essential data is being collected.	
16	LINDDUN - Unawareness (U1 - No Transparency):	Information disclosure	Medium	Open		LINDDUN - Unawareness (U1 - No Transparency): Card Description: Threat: Insufficient information provided to data subjects about the collection and processing of their data. Reasoning: Users may not fully be aware of how their data is used or shared which can lead to lack of trust or legal issues. How to mitigate: Implement clear and accessible information notices and consent mechanisms to users so they are aware of the data and privacy practices.	

Portal (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
5	Elevation of Privilege Game - Card (I9)	Information disclosure	Medium	Open		Elevation of Privilege Game - Card (I9) Card Description: An attacker can read sensitive information in a file with permissive permissions. Reasoning: Files with permissive permissions could allow unauthorized access to private patient information. How to mitigate: Implement regular audits of file permissions and enforce the principle of least privilege. Ensure encryption of the data in rest and in transit.	
8	Elevation of Privilege Game - Card (E4)	Elevation of privilege	Medium	Open		Elevation of Privilege Game - Card (E4) Card Description: An attacker can escape from a container or other sandbox. Reasoning: If the software is ran on a containerized environment, such like how we are using it with Docker, if the attacker escapes the container they could gain higher-level access which could lead to a full system compromise. How to mitigate: Implement hardened container images with up-to-date patches. Ensure that the principle of least privilege is used within the container environments.	
9	Cornucopia - Data Validation & Encoding Card (D4)	Tampering	Medium	Open		Cornucopia - Data Validation & Encoding Card (D4) Card Description: Threat: Dave can input malicious field names or data because it is not being checked within the context of the current user and process. Reasoning: An injection attack with malicious data injected into the application can manipulate the data or behavior. This can potentially compromise patient data. How to mitigate: Implement input validation that checks all inputs on the backend. Use a combination of different rule sets to inspect for different types of incoming user data. Also ensure parameterized queries are used to prevent SQL injection on all inputs.	
14	Cornucopia - Error Handling - Inconsistent Exception Handling (Card E6):	Information disclosure	Medium	Open		Cornucopia - Error Handling - Inconsistent Exception Handling (Card E6): Card Description: Threat: Aaron can bypass controls because error/exception handling is missing, inconsistent, or partially implemented, and does not deny access by default. Reasoning: Inconsistent error handling can expose sensitive information through the error messages. This can allow an attacker to infer information about the system or its structure to exploit other potential vulnerabilities. How to mitigate: Implement consistent and secure error handling strategies that do not disclose any sensitive information. The errors should be logged to the logging server and the users/frontend should be presented with generic error messages.	
17	LINDDUN - Detectability (D1 - Detectable Credentials):	Information disclosure	Medium	Open		LINDDUN - Detectability (D1 - Detectable Credentials): Card Description: Threat: Response of a request allows detection of the existence of a user. Reasoning: This may reveal to unauthorized actors' personal information about an individual's health record even if the content remains secure. How to mitigate: Implement standardized error messages to avoid revealing personal information or the existence of personal information of a user.	

MySQL (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
3	Elevation of Privilege Game - Card (T6)	Tampering	Medium	Open		Elevation of Privilege Game - Card (T6) Card Description: An attacker can write to a data store your code relies on. Reasoning: Tampering with the data store could corrupt patient data, alter prescriptions, or falsify medical records. How to mitigate: Implement parameterized queries to prevent unauthorized changes to the data. Use checksums or cryptographic hashes to detect if data has been tampered with. Ensure access control is implemented so only authorized individuals can modify the sensitive data and that any access for CRUD is logged with appropriate data.	
6	Elevation of Privilege Game - Card (D10)	Denial of service	Medium	Open		Elevation of Privilege Game - Card (D10) Card Description: An attacker can make a server unavailable or unusable without ever authenticating and the problem persists after the attacker goes away (server, anon, persist). Reasoning: A denial of service attack and disrupt access to patient data. How to mitigate: Implement rate limiting and filtering as well as use load balances and redundant systems so traffic can be distributed or still accessed if a server goes down.	
19	LINDDUN - Linkability (L3 - Linkability of Inbound Data):	Repudiation	Medium	Open		LINDDUN - Linkability (L3 - Linkability of Inbound Data): Card Description: Threat: Data sent to the system are linked to already collected data, making it possible to profile users. Reasoning: If data is no anonymized, or only partially anonymized, it could link different data sets that could then be used to re-identify the individual. How to mitigate: Implement data minimization principles so that only the data that is necessary is collected. Also implement strict access controls and anonymization that prevents data from being linked to the actual individual.	
20	LINDDUN - Identifiability (I1 - Identifying Credentials):	Repudiation	Medium	Open		LINDDUN - Identifiability (I1 - Identifying Credentials): Card Description: Threat: The use of non-anonymous credentials allows the identification of users. Reasoning: User’s health data can be linked to their real identities which could lead to privacy violations. How to mitigate: Implement techniques where the users’ identities are replaced with pseudonyms to prevent direct identification from unauthorized actors. Also ensure that personal defining details are removed from logging.	

Logging (Store)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
4	Elevation of Privilege Game - Card (R7)	Repudiation	Medium	Open		Elevation of Privilege Game - Card (R7) Card Description: An attacker can make the logs wrap around and lose data. Reasoning: Tampering with the logs would make it difficult to trace unauthorized access or changes to data. How to mitigate: Implement a log solution that stores digital signatures to ensure integrity of the entries. Employ a centralized logging system that has secure backups to help prevent log tampering.	

4. Test Coverage:

1. ASVS V7.1: Provide repeatable steps offering comprehensive and moderate logging coverage, with a focus on successful login event logging and log retrieval.

2. Black Box Tests:

Test 1:

ASVS V6.2 Algorithms

Unique ID: 6.2.1

CWE 310: Weaknesses in this category is related to the use of cryptography.

Repeatable steps:

1. Logout of the application if you are logged in and access the sign in page. 2. Enter the username and an incorrect password and submit the form.

3. Inspect the network traffic and the request that shows "login.php?site=default"

Expected results:

1. You should not see any identifying information from the server such as: encrypted user information, error logs or messages providing insight from the server, etc. In the response, we should see the html page being returned.

Test 2:

ASVS V3.1 Fundamental Session Management Security

Unique ID: 3.1.1

CWE 598: Use of GET Request Method With Sensitive Query Strings

Repeatable steps:

1. Log into the application with a username and password

2. Check the address bar for any visible session tokens

Expected results:

1. After a login, no session tokens should be in the url.

Test 3:

ASVS V5.1 Input Validation

Unique ID: 5.1.3 CWE 20: Improper Input Validation

Repeatable steps:

1. Log into the application with a username and password
2. On the calendar, click on the 8:00 time
3. Highlight the date in the Date field
4. Type letters and press enter
5. Verify that you are not allowed to add letters in the date field

Expected results:

1. Letters are not allowed in the Date field

Test 4:

ASVS V5.2 Sanitizing and Sandboxing

Unique ID: 5.2.4 CWE 95: Improper Neutralization of special elements

Repeatable steps:

1. Log into the application using a valid username and password.
2. Choose the "Patient" option and click on "NEW/Search."
3. In the "NAME" and "LastName" registration fields, type the characters "(abdf)."
4. Click on "Create a new patient."
5. Verify that the patient is displayed with the inputted unusual characters and ensure the system recognizes and displays the patient as valid

Expected results:
Valid inputs should be processed successfully. However, inputs containing potentially harmful features, such as 'eval ()' or other dynamic code execution features, should be rejected to prevent security risks

Test 5:

ASVS V8.3 Sensitive Private Data

Unique ID: 8.3.2

CWE 212: Improper Removal of Sensitive Information Before Storage or Transfer

Repeatable steps:

1. Log into the application using a valid username and password.
2. Choose the "Patient" option and click on NEW/SEARCH.
3. Click on the Search button.
4. Select the intended patient.
5. Choose "DOCUMENTS" from the header toolbar. The patient can delete or remove previously uploaded documents by pressing the DELETE button.

Expected results:

Should indicate a method for patients to remove information or uploaded documents.

Test 6:

ASVS V14.4 HTTP Security Headers

Unique ID: 14.4.7

CWE 1021: Improper Restriction of Rendered UI Layers or Frames.

Repeatable steps:

1. Open your Network tab in the developer options.
2. Change the request to the “Doc” tab, or depending on your browser, to the “Headers” tab.
3. Manually access the page by entering into your URL bar: localhost:80.
4. You should see under the “File” tab in the “Network” menu of developer options a file named “login.php?site=default”.
5. Click on the file named “login.php?site=default” and look in the Headers tab of that request for "Content-Security-Policy" and "X-Frame-Options". You should

see "frame-ancestors 'none'" in Content-Security-Policy and "DENY" in X-Frame Options.

6. Create a simple text file on your computer and enter the following:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Clickjacking Test</title>
 </head>
 <body>
 <iframe src="http://localhost" width="800" height="600"></iframe>
</body>
</html>
```

7. Save the file as something like "test.html". The extension must be .html 8.

Open the file so that it opens in your web browser. The iframe should not load.

Expected results:

That the iframe fails to load on the webpage.

Test 7:

ASVS V4.2 Operation Level Access Control

Unique ID: 4.2.2 CWE 352: Cross-Site Request Forgery (CSRF)

Repeatable steps:

1. Log in to the application and navigate to a state-changing operation, in this case changing the password.

2. Submit an empty password change form.

3. Examine the request to obtain the CSRF token. You can find this by inspecting the “Network” tab under “Developer Options”. In the listed requests under “File” find the one named “user_info_ajax.php” and click it. Then, in the menu for that request, change to the “Request” tab to see the “Form data” which will contain the field “csrf_token_form”. (Make sure to copy it down as it is needed in step).

4. Log out and then log back into the application and navigate back to the password change form and submit an empty password form again.

5. Examine the request, following the steps we took in step 3, to ensure the CSRF token is different.

6. Test the previous CSRF token to ensure it is not accepted by going to the console tab and entering the following (make sure to enter the current password, the password you are wanting to change it to, and the CSRF token from your previous login attempt):

```
fetch('http://localhost/interface/usergroup/user_info_ajax.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded',
  },
    body: 'curPass=[current_password]&newPass=[new_password]

 &newPass2=[new_password_repeat]&csrf_token_form=[old_csrf_token]
 ' ,
  credentials: 'include'
 }) .then(response => response.text())
  .then(data => console.log(data))
  .catch((error) => console.error('Error:', error));
```

Expected results:

The console output in the browser should show the application rejecting the request for state-changing operations that do not include a valid CSRF token.

3.

Time spent: 2h 8min.
Total Vulnerabilities = 7 vulnerabilities
Vulnerabilities per Hour = Total Vulnerabilities / Total Time (in hours)
7 vulnerabilities / 2.08 hours ≈ 3.37 vulnerabilities/hour

4.

ASVS V6.2: provided repeatable steps offer decent coverage of vulnerabilities related to cryptography, focusing on identifying information leakage through network traffic inspection.

ASVS V3.1: provided test scenarios offer comprehensive coverage of session management security requirements outlined in (Unique ID: 3.1.1) with CWE 598 (Use of GET Request Method with Sensitive Query Strings)

ASVS V5.1: provided test scenarios offer comprehensive coverage of input validation requirements outlined in (Unique ID: 5.1.3) with CWE 20 (Improper Input Validation)

ASVS V5.2: provided test scenarios offer comprehensive coverage of sanitization and sandboxing requirements outlined in (Unique ID: 5.2.4) with CWE 95 (Improper Neutralization of Special Elements)
ASVS V8.3: The provided test scenarios offer comprehensive coverage of sensitive data handling requirements outlined in (Unique ID: 8.3.2) with CWE 212 (Improper Removal of Sensitive Information Before Storage or Transfer).

ASVS V14.4: The provided test scenarios offer comprehensive coverage of HTTP security headers requirements outlined in (Unique ID: 14.4.7) with CWE 1021 (Improper Restriction of Rendered UI Layers or Frames)

ASVS V4.2.2: The provided test scenarios offer comprehensive coverage of CSRF protection requirements outlined in (Unique ID: 4.2.2) with CWE 352 (Cross-Site Request Forgery).