S

Pomona Wellness Network

Spring 2023 Penetration Testing Report
4/8/2023

CONFIDENTIAL

Final

Disclosure

This engagement was performed in accordance with Pomona Wellness Network, and all procedures are limited to the scope provided by our client. All vulnerabilities were identified and recorded between the dates of March 25th, 2023, and April 11th, 2023. All findings and recommendations reflect only the information gathered within this time frame.

We would like to emphasize that due to the time-limited nature of the engagement, the evaluation may not cover all vulnerabilities. Therefore, our security researchers prioritize the most critical and common vulnerabilities that a threat actor may exploit. Hash Gatos recommends conducting regular engagements to ensure that all known security flaws are identified, corrected and to detect new issues if needed.

This report is intended solely for the information and internal use of Pomona Wellness Network.

Contents

Executive Summary

Hash Gatos was contracted by Pomona Wellness Network to conduct a comprehensive penetration testing of their website, which serves as the primary platform for patients and clinical staff. The objective of the test was to identify any potential vulnerabilities and non-compliance issues that may exist in the public facing side of the system.

After an initial assessment, Hash Gatos was able to compromise several components of the website using various open-source tools, leading to indications of security concerns and potential non-compliance with HIPAA regulations. The team was able to identify multiple attack vectors and gain unrestricted root access, which could allow an external threat actor to deface, steal, or deny access to the website.

In response to Pomona Wellness Network's request for a special emphasis on role-based access, our team engaged in intensive testing of privilege escalation, server misconfigurations, and the potential for confidential information leakage. Based on our findings, Hash Gatos has concluded that Pomona Wellness Network's website is not compliant with industry-standard security practices, and we recommend immediate remediation in the interest of confidentiality and HIPAA compliance. This will protect Pomona Wellness Network from multiple liabilities including potential legal risks, fines, lawsuits, and potential reputational damage in the event of a cyber threat compromising patient data.

Attached, our security researchers have provided detailed recommendations, including the implementation of thorough and regular penetration testing engagements to detect and address all found vulnerabilities. Our legal team has also provided compliance recommendations based on HIPAA and CCPA. It is our primary recommendation that Pomona Wellness Network implements these measures to ensure the security of their systems and prevent threat actors from violating the confidentiality, integrity and availability of Pomona Wellness Network’s website.

Strategic Recommendations

Mission and Objectives:

As a medical group within the healthcare sector, Pomona Wellness Network must provide proper safeguards for personally identifiable information and patient data. This also prevents Pomona Wellness Network from incurring any potential financial or legal risks posed due to improper holding of patient data.

Short-term Strategies:

1. Migrate to HTTPS protocol: This will encrypt all data transmitted between the web server and client's browser, providing an additional layer of security and protection against data interception or theft.
2. Implement stronger password policies: Passwords should be complex, unique, and regularly changed. Pomona Wellness Network should enforce password policies that require employees and users to use strong passwords, avoid password reuse across different accounts, and change their passwords frequently.
3. Update SSH protocols: Pomona Wellness Network should update their SSH protocols to the latest versions, ensuring that they use strong encryption algorithms and secure authentication methods. This will help to prevent unauthorized access to their systems and sensitive data. Multiple scans on Nmap concluded that the SSH protocols within Pomona Wellness Network's website were not sufficient and may allow for impersonation attacks by leveraging SSH login access.
4. Move away from storing passwords in MD5: MD5 is an outdated hashing algorithm that is no longer considered secure. Pomona Wellness Network should move away from using MD5 to store passwords and implement more secure hashing algorithms such as bcrypt or scrypt. Through the use of sqlmap, Hash Gatos was able to crack stored MD5 passwords.
5. Implement a Content Security Policy header: The remote web server on the target machine does not set a response header in all content responses. This exposes the website to potential click-acking attacks wherein an attacker can lure users to another page to perform fraudulent transactions.
6. Update to the latest software updates and patches: Pomona Wellness Network should ensure that all software applications and systems are updated to the latest versions and patches. This will help to address known vulnerabilities and protect against potential attacks.

Long-term Strategies:

1. Hash Gatos recommends for Pomona Wellness Network to follow NIST 800-53 as a comprehensive framework for best practices. This recommendation comes from the relative flexibility and customization of this framework to meet the needs of healthcare organizations. It is designed to complement existing regulations such as HIPAA and can help healthcare organizations identify privacy risks beyond regulatory compliance. Through the usage of this framework, Pomona Wellness Network should also establish a risk management program to identify, assess, and manage potential risks to their systems and data as part of their long-term strategy to prevent future threats.
2. Our security researchers encourage the usage of cybersecurity awareness training in order to further protect their sensitive patient data. By implementing these measures, Pomona Wellness Network will be much more prepared to identify and respond to threats, alongside demonstrating their commitment to ensure compliance for their stakeholders. This in turn protects Pomona Wellness Network from potential data leakage due to human error, and ensures they are knowledgeable about the best cybersecurity practices available.
3. Multi-factor authentication as an additional layer of security would reduce the risk of potential unauthorized access into Pomona Wellness Network's data. The usage of multi-factor authentication ensures that even if an attacker were to compromise a user's password – they would still require the user's phone or token to gain access to the system. This deterrent method works to greatly reduce the risk of unauthorized access even in the event of a data breach.
4. Hash Gatos heavily recommends in the investment of route security hygiene, which includes but is not limited to regular penetration tests, security audits and ensuring that Pomona Wellness Center is not liable to potential damages in the event of a cyber-attack.

Scope

Hash Gatos performed a penetration test on Pomona Wellness Network's website hosted at the IP address 192.168.1.2. This engagement will cover all public facing components of the patient portal including the calendar, patient data, appointment scheduling and all other functions. The primary goal of our client is to identify vulnerabilities, risks and weaknesses within the website's security controls to ensure that threat actors do not gain unauthorized access to confidential information. Alongside this, the assurance that this website is HIPAA compliant is especially important to our client and therefore takes upmost priority, and therefore testing will include an evaluation of access-based controls. Both automated and manual testing techniques are utilized alongside industry standard practices to ensure that this engagement is as thorough as possible given the constraints. Upon the request of our client, we will not be utilizing social engineering or physical security testing.

Methodology

Hash Gatos employs a combination of industry-standard frameworks to guide our penetration testing engagements. Specifically, we utilize the OWASP (Open Web Application Security Project) Top 10 for the year 2021 and the Penetration Testing Execution Standard (PTES) to ensure the most comprehensive and rigorous methodology within our allocated time frame.

Our focus on the OWASP Top 10 vulnerabilities ensures that we address the most critical issues first, following their prescribed order. These include:

1. Broken Object Level Authorization
2. Broken Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Server-Side Request Forgery
7. Security Misconfiguration
8. Lack of Protection from Automated Threats
9. Improper Asset Management
10. Unsafe Consumption of APIs

By following these rigorous methodologies, we ensure that our penetration testing engagements are conducted in a structured, comprehensive, and professional manner.

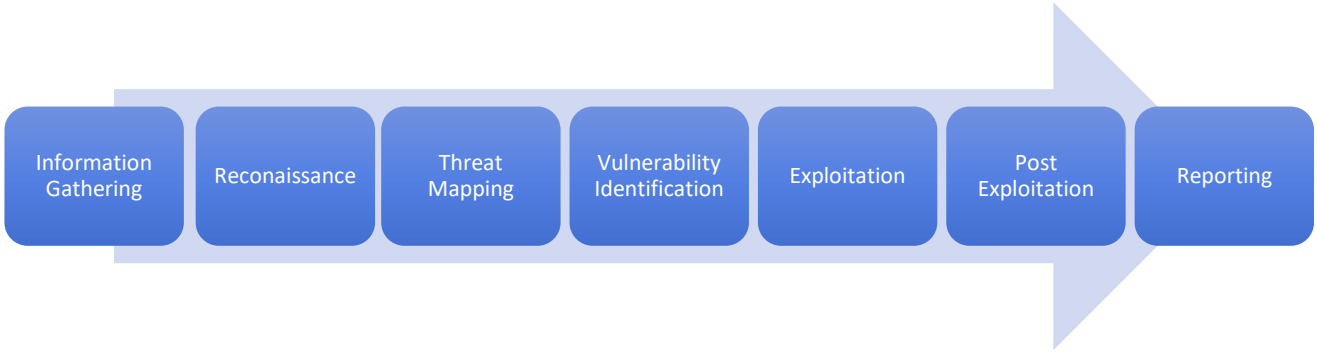


FIGURE 1: INFOGRAPHIC OF PTES PROCESS

The PTES consists of 7 primary components that we utilize to ensure proper rigor during our tests. They are described as such.

- Pre-engagement Interactions - Tooling, goals, rules of engagement and scope
- Intelligence Gathering - OSINT, Fingerprinting and Footprinting/enumeration
- Threat Modeling - Business Asset, Business Process Analysis, and Threat Agents
- Vulnerability Analysis - Active / Passive Analysis, Validation, Research
- Exploitation - Countermeasures, Evasion, Exploits
- Post Exploitation - Infrastructure Analysis, Exfiltration, Persistence, Follow-up
- Reporting - Executive Summary, Risk Rating

By following these rigorous methodologies, we ensure that our penetration testing engagements are conducted in a structured, comprehensive, and professional manner.

Risk Assessment

For our risk assessment we consider the traditional calculation of Risk using the product of Likelihood of Threat and Expected Loss of Impact. This approach is widely utilized as a standard measure, this is further detailed in the following section, Risk Scaling. By applying the principles detailed therein, we can establish an estimated quantifiable risk level for each vulnerability discovered. This should enable the business to make better decisions with regards to how to deal with the risks identified and presented in this report. Responses to risk typically fall under one of a handful of approaches, depending on the analysis of the identified risks: Avoidance, Mitigation, Transfer, Acceptance, Share, Contingency, Enhance, or Exploit.

To determine the appropriate response, it is important to first conduct a thorough risk analysis. There are several ways to conduct risk analysis and risk management, but most of these techniques generally follow the process of:

1. Identify existing risk
2. Assessing the risk
3. Develop an appropriate response
4. Develop preventative mechanisms for identified risks

The field of healthcare, however, presents some unique challenges with regards to privacy that must be considered when implementing a risk management strategy. For this reason, we strongly recommend implementing the NIST Privacy Framework and NIST Risk Management Framework (RMF) for the strategic management of organizational risk and the implementation of NIST 800-53 for starting to enact the technical controls needed to manage risk according to the needs of the organization. This ensures that Pomona Wellness Center adheres to it's objective in providing a secure environment for patient data while also deterring threats in the future.

1. The NIST Privacy Framework is a modular, jurisdiction agnostic framework that can meet the needs of any organization with regards to compliance with the regulation around data privacy. In this assessment we are testing the infrastructure of a health care facility that processes and stores sensitive personal health data of patients. This puts the organization under the obligation to comply with the Health Insurance Portability and Accountability Act (HIPAA) of 1996 and, where appropriate, the California Consumer Privacy Act (CCPA) and California Privacy Rights Act regulatory constraints.

   https://www.nist.gov/privacy-framework/privacy-framework

2. The NIST Risk Management Framework provides a process that integrates security, privacy, and cyber supply-chain risk management activities into the system development life cycle, making it an ideal choice for considering how to implement any additional controls that are chosen in response to the findings generated from this report. This framework expands upon the 4-step process above to provide a step-by-step approach to the organization's new and legacy systems. The chart is laid out as follows:

Step	Purpose
Prepare	Essential activities to prepare the organization to manage security and privacy risks.
Categorize	Inform organizational risk management processes and tasks by determining the adverse impact with respect to the loss of confidentiality, integrity, and availability of systems and the information processed, stored, and transmitted by those systems
Select	Select, tailor, and document the controls necessary to protect the system and organization commensurate with risk
Implement	Implement the controls in the security and privacy plans for the system and organization
Assess	Determine if the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security and privacy requirements for the system and the organization.
Authorize	Provide accountability by requiring a senior official to determine if the security and privacy risk based on the operation of a system or the use of common controls is acceptable
Monitor	Maintain ongoing situational awareness about the security and privacy posture of the system and organization to support risk management decisions.

FIGURE 2: NIST RISK MANAGEMENT FRAMEWORK

Table 1:
https://csrc.nist.gov/projects/risk-management/about-rmfCVE

3. Finally, NIST 800-53, titled Security and Privacy Controls for Information Systems and Organizations, provides a comprehensive guide on the implementation of the technical systems and controls needed to meet the security and privacy objectives of the organization. Unlike the two frameworks referenced above, the 800-53 framework does not provide a process for selecting and implementing controls but instead details a catalog of security controls for different functions that can be used to meet the objectives of these frameworks. In combination with either or both of the above frameworks, NIST 800-53 can greatly strengthen the organization's security posture, and the data of its patients.

   https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

Risk Scaling

Vulnerabilities are weaknesses in the environment, system architecture, design, or implementation; the organizational policies, procedures, or practices; and the management or administration of hardware, software, data, facility, or personnel resources. Vulnerabilities that are exploited may cause harm to the system or information processed, transported, or stored by the system. After analyzing system management, operational, and technical security controls for the system in its fielded environment, system vulnerabilities were then identified.

The analysis of the system's vulnerabilities, the threats associated with them, and the probable impact of that vulnerability exploitation resulted in a risk rating for each missing or partially implemented control. The risk level was determined on the following two factors:

1. Likelihood of Occurrence - The likelihood to which the threat can exploit a vulnerability given the system environment and other mitigating controls that are in place. This can be determined in part using Threat Modeling with particular attention to the trust boundaries.

2. Impact – The impact of the threat exploiting the vulnerability in terms of loss of tangible assets or resources and impact on the organization's mission, reputation, or interest. In the case of a health care facility this could also result in financial penalties or civil liability.

The likelihood that a threat will exploit a vulnerability and cause damage as described above was determined based on the following factors: the frequency of the threat and the existence of mitigation controls. The likelihood of occurrence was determined to be high, moderate or low in accordance with the following chart.

Value	Likelihood of Occurrence Description
High (3)	The threat-source is highly motivated and sufficiently capable, and controls to prevent the vulnerability from being exploited are ineffective.
Moderate (3)	The threat-source is motivated and capable, but controls are in place that may impede successful exploitation of the vulnerability.
Low (3)	The threat-source lacks motivation or capability, or controls are in place to prevent, or at least significantly impede, the vulnerability from being exploited.

FIGURE 3: LIKELIHOOD OF OCCURRENCE TABLE

An impact analysis prioritizes the impact levels associated with the compromise of an organization's information assets based on a qualitative or quantitative assessment of the sensitivity and criticality of those assets. The system and data sensitivity can be determined based on the level of protection required to maintain the system and data's availability, integrity, and confidentiality. The analysis first looked at how important the availability, integrity, and confidentiality of the system and/or its data are to the ability of the system to perform its function and the types of damage that could be caused by the exercise of each threat-vulnerability pair. Therefore, the adverse impact of a security event can be described in terms of loss or degradation of any, or a combination of any of the three security goals: integrity, availability, and confidentiality.

To determine overall risk levels, Hash Gatos determined how important the security goals of the CIA triad (confidentiality, integrity, and availability) of the system and/or its data are to the mission's ability to function as intended. The system sensitivity values of this report were mapped to the magnitude of impact qualitative values of high (100), moderate (50), and low (10) as defined in the NIST guidelines and shown in the following table

Impact Level/Value	Impact Description
High (3)	Exercise of the vulnerability (1) may result in the highly costly loss of major tangible assets or resources; (2) may significantly violate, harm, or impede an organization's mission, reputation, or interest; or (3) may result in human death or serious injury.
Moderate (2)	Exercise of the vulnerability (1) may result in the costly loss of major tangible assets or resources; (2) may violate, harm, or impede an organization's mission, reputation, or interest; or (3) may result in human injury.
Low (1)	Exercise of the vulnerability (1) may result in loss of some tangible assets or resources or (2) may noticeably affect an organization's mission, reputation, or interest.

FIGURE 4: IMPACT LEVEL TABLE

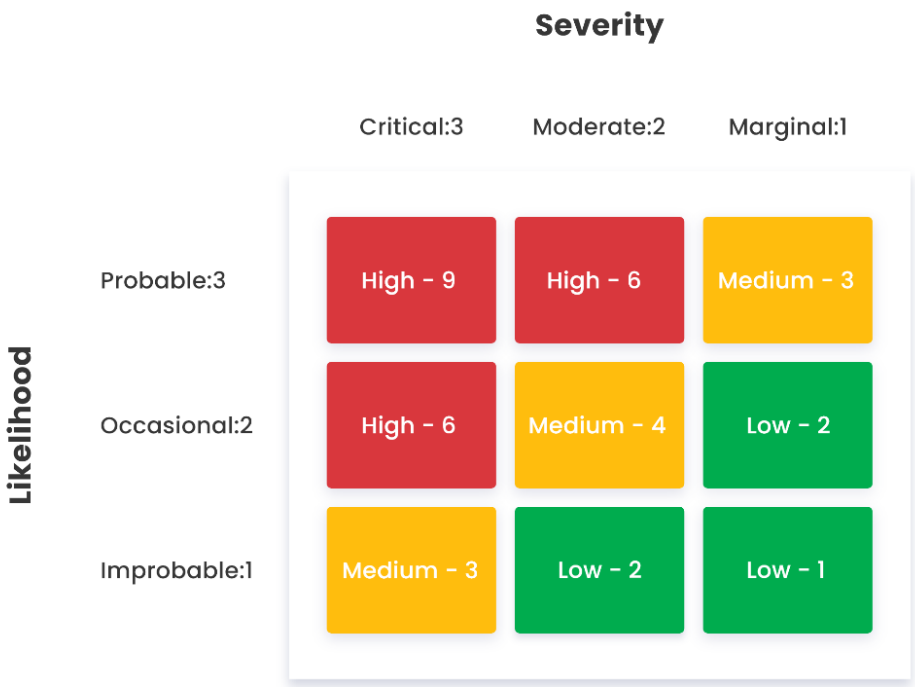


FIGURE 5: SEVERITY MATRIX

In the following section, each protection requirement is rated on a scale of High, Moderate, or Low, using the guidance from NIST *Guide for Developing Security Plans for Information Technology Systems*, SP 800-18, and FIPS 199, *Standards for Security Categorization of Federal Information and Information Systems*. This model is meant as a guide for information security policies and solutions.

Confidentiality: Prevents the disclosure of data to unauthorized parties while continuing to provide the lowest amount of information to third parties accessing information.

- **High:** The consequences of unauthorized disclosure or compromise of data or information in the system are **unacceptable**. Loss of confidentiality could be expected to cause a severe degradation in or loss of mission capability and client trust. This will harm AEFS's ability to deliver meals securely and discreetly to clients.
- **Moderate:** The consequences of unauthorized disclosure or compromise of data or information in the system are only **marginally acceptable**. Loss of confidentiality could be expected to cause a significant degradation in mission capability to an extent.
- **Low:** The consequences of unauthorized disclosure or compromise of data or information in the system are **generally acceptable**. Loss of confidentiality could be expected to cause degradation in mission capability to an extent and duration that AEFS is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced.

Integrity: The insurance that all data transmitted from one end to another is not tampered with in any form.

- **High:** The consequences of corruption or unauthorized modification of data or information in the system are **unacceptable**. Loss of integrity could be expected to cause a severe degradation in or loss of mission capability in correctly verifying all information for meal distribution and client information.
- **Moderate:** The consequences of corruption or unauthorized modification of data or information in the system are only **marginally acceptable**. Loss of integrity could be expected to cause a significant degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of functions is significantly reduced.
- **Low:** The consequences of corruption or unauthorized modification of data or information in the system are **generally acceptable**. Loss of integrity could be expected to cause degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced.

Availability: Ensuring that all data and critical functions for day-to-day operations are readily available 24/7

- **High:** The consequences of loss or disruption of access to system resources or to data or information in the system are **unacceptable**. Loss of availability could be expected to cause a loss of service to AEFS meal distribution in most areas.
- **Moderate:** The consequences of loss or disruption of access to system resources or to data or information in the system are only **marginally acceptable**. Loss of availability could be expected to cause a significant degradation in mission capability to an extent by causing a loss of service to only some areas.
- **Low:** The consequences of loss or disruption of access to system resources or to data or information in the system are **generally acceptable**. Loss of availability could be expected to cause degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, with service only being minimally reduced.
  ope

Vulnerability Assessment

In addition to likelihood and impact, the following vulnerabilities are accompanied by tone or more of the following identifiers:

1. The Reference ID as provided by The Web Application Security Consortium
2. The corresponding Common Weakness Enumeration (CWE)
3. Any corresponding Common Vulnerabilities and Exposures (CVE)

Open Ports

Target IP: 192.168.1.2

Port	Service	Version
22/tcp	SSH	OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
80/tcp	HTTP	Apache httpd
443/tcp	SSL/HTTP	Apache httpd
12320/tcp	SSL/HTTP	ShellInABox httpd
12321/tcp	SSL/HTTP	MiniServ 1.780 (Webmin httpd)
12322/tcp	HTTP	Apache httpd

FIGURE 6: OPEN PORTS AND SERVICES TABLE

Finding #1

Vulnerability Name: PHI | Reference ID
Likelihood: 3
Impact / Severity: 3
Risk Score: 9.0 (High)

While logged as the root user in the SSH server, we used the command “ls -la” which displayed the listed directories.

```
root@lamp /var/www# ls -la
total 328
drwxr-xr-x 10 root root  12288 Feb 16 05:14 .
drwxr-xr-x 13 root root   4096 Apr  8  2016 ..
-rw-r--r--  1 root root  67841 Feb 16 05:10 MISSA Logo.png
drwxrwxrwx  7 root root   4096 Feb 11  2017 MISSAITC2023
drwxr-xr-x  2 root root   4096 Feb 14 02:24 PHI
-rw-r--r--  1 root root  10163 Feb 14 01:19 Patient-File-MikaHamatani-CONFIDENTIAL.xlsx
-rw-r--r--  1 root root   2319 Dec 29  2016 README.txt
drwxr-xr-x  2 root root   4096 Apr  8  2016 cgi-bin
drwxr-xr-x  2 root root   4096 Apr  8  2016 css
drwxr-xr-x  2 root root   4096 Sep 30  2019 html
drwxrwxrwx  2 root root   4096 Apr  8  2016 images
-rw-r--r--  1 root root    990 Feb 16 05:13 index.php
drwxr-xr-x  2 root root   4096 Apr  8  2016 js
-rwxr-xr-x  1 root root    522 May 18  2016 license.txt
drwxr-xr-x  2 root root   4096 Feb 11 22:26 logs
-rwxr-xr-x  1 root root 106017 Jan 10  2017 online clinic management system.axp
-rwxr-xr-x  1 root root      0 Oct 27  2016 patient_botton.php~
-rwxr-xr-x  1 root root     28 Oct 27  2016 patients.gender.csv
-rwxr-xr-x  1 root root    234 Oct 27  2016 patients.state.csv
-rw-r--r--  1 root root  45883 Oct 13 18:12 privacyChecker.php
-rw-r--r--  1 root root  28316 Feb 13 05:15 prototype.js
```

FIGURE 1: PHI DIRECTORY

Since we were able to SSH into root and were given root (the highest level of access) we were able to access sensitive information. The PHI directory had confidential patient files, confidential medical records, and a file containing social security numbers. Given access to the root the attacker was given private and confidential information that can be maliciously exploited.

```
root@lamp www/PHI# ls -la
total 2860
drwxr-xr-x  2 root root    4096 Feb 14 02:24 .
drwxr-xr-x 10 root root   12288 Feb 16 05:14 ..
-rw-r--r--  1 root root  441915 Feb 14 01:44 CONFIDENTIAL MEDICAL RECORD Ima Nottingham MRN 157786138.pdf
-rw-r--r--  1 root root 1191983 Feb 14 02:05 MikaHamataniChestX-Ray.png
-rw-r--r--  1 root root  602516 Feb 14 02:00 Patient-File-ImaNottingham-CONFIDENTIAL.pdf
-rw-r--r--  1 root root   86533 Feb 14 01:44 Patient-File-ImaNottingham-CONFIDENTIAL.xlsx
-rw-r--r--  1 root root  549363 Feb 14 02:00 Patient-File-MikaHamatani-CONFIDENTIAL.pdf
-rw-r--r--  1 root root   10163 Feb 14 01:44 Patient-File-MikaHamatani-CONFIDENTIAL.xlsx
-rw-r--r--  1 root root   11021 Feb 14 02:24 SocialSecurityNumbers.xlsx
```

FIGURE 2: SOCIAL SECURITY NUMBERS FILE IN PHI DIRECTORY

With the information that we have, we can use URL manipulation to access the PHI directory containing the social security number file and the confidential medical records of patients. By typing in the IP address of the hosted site, along with the PHI directory, it displays the given information.

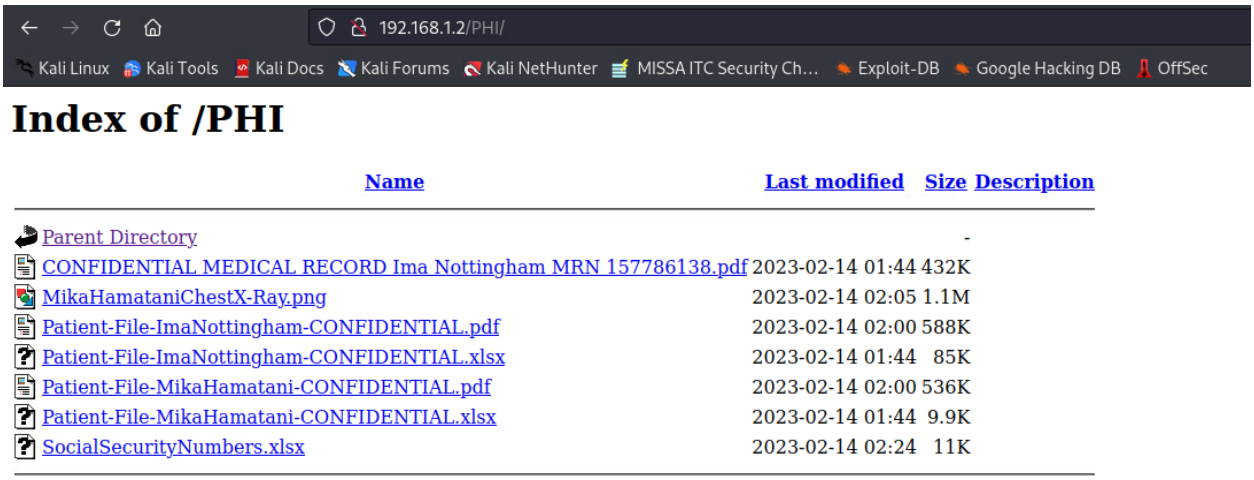


FIGURE 3: PHI DIRECTORY OF CONFIDENTIAL PATIENT INFORMATION AND SOCIAL SECURITY NUMBER.XLSX

Finding # 2

Vulnerability Name: SQL Injection | Reference ID. WASC-19
Likelihood: 2
Impact / Severity: 3
Risk Score: 6.0 (High)

The CWE-89 Vulnerability: Improper Sanitization of Special Elements used in an SQL Command ('SQL Injection') refers to the ability to construct all or part of an SQL command using externally influenced input from an upstream component, which does not sanitize or incorrectly sanitizes special elements that could modify the intended SQL command when it is sent to a downstream component.

SQL is a specialized programming language for sending queries to databases. An SQL Injection is an attack used to exploit applications that construct SQL statements from user-supplied input. This allows the attacker to obtain sensitive information and modify the logic of SQL statements. For this assessment we used a combination of BurpSuite and SQLMap to access patient data.

"Reports" has parameters to search for history, prescription, and diagnosis of patients which is vulnerable to SQL injections

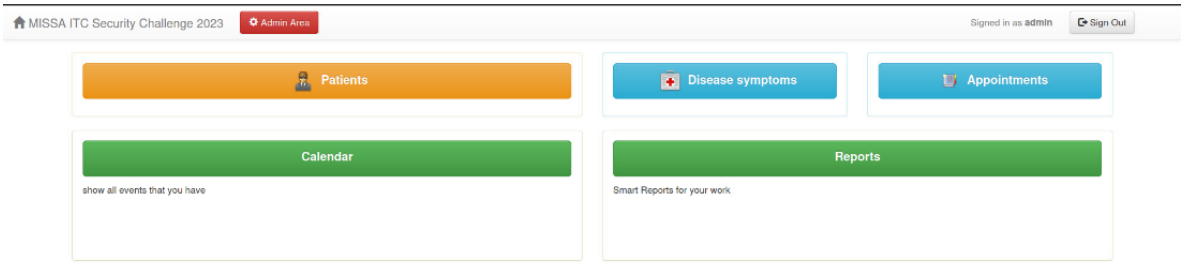


FIGURE 4: SCREENSHOT OF LOGIN AS ADMIN

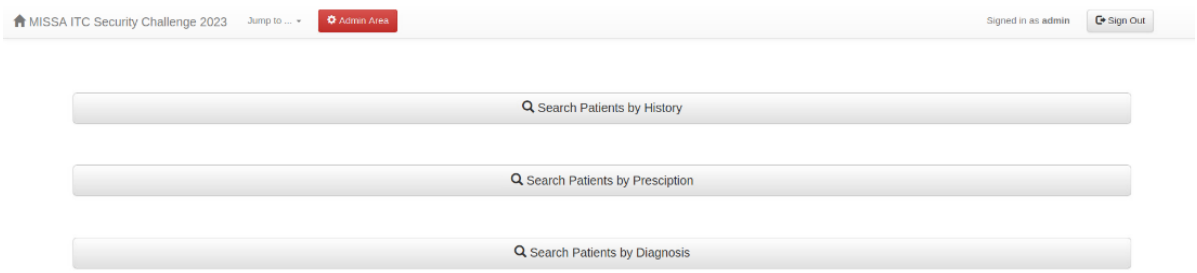


FIGURE 5: ADMIN AREA -SEARCH PATIENTS BY HISTORY

A request was intercepted through Burp Suite and allowed us to save the request as an XML file to be used in sqlmap

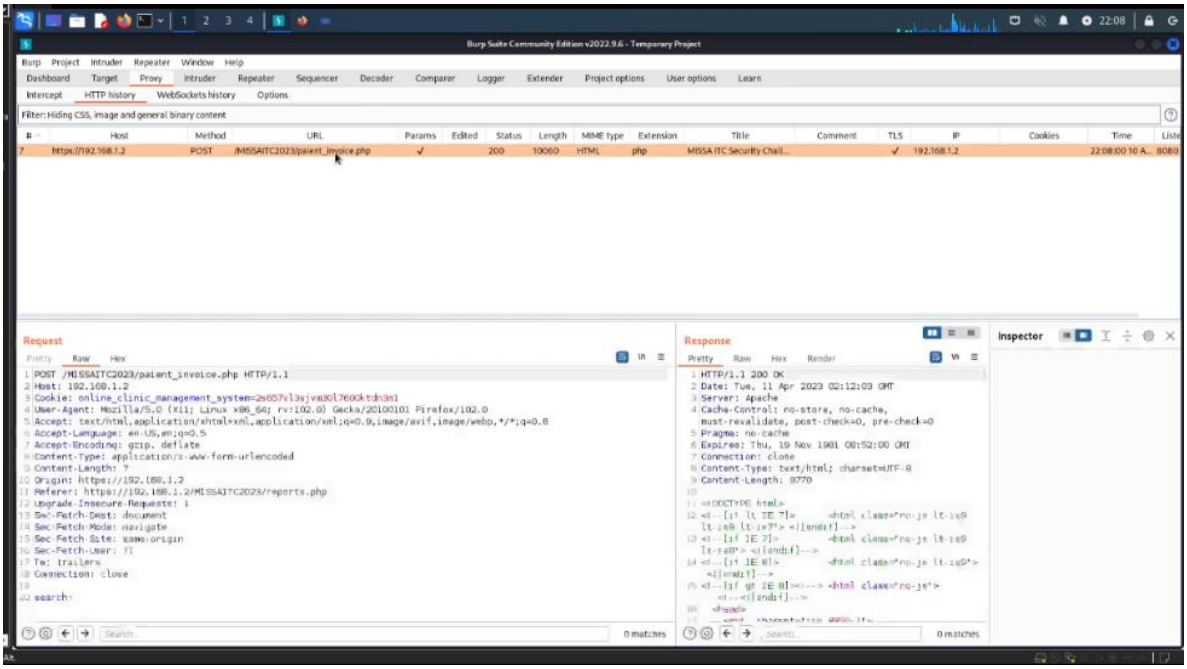


FIGURE 6: BURPSUITE XML FILE

Sqlmap was then used to automate the SQL Injection vulnerability by using the save XML file.

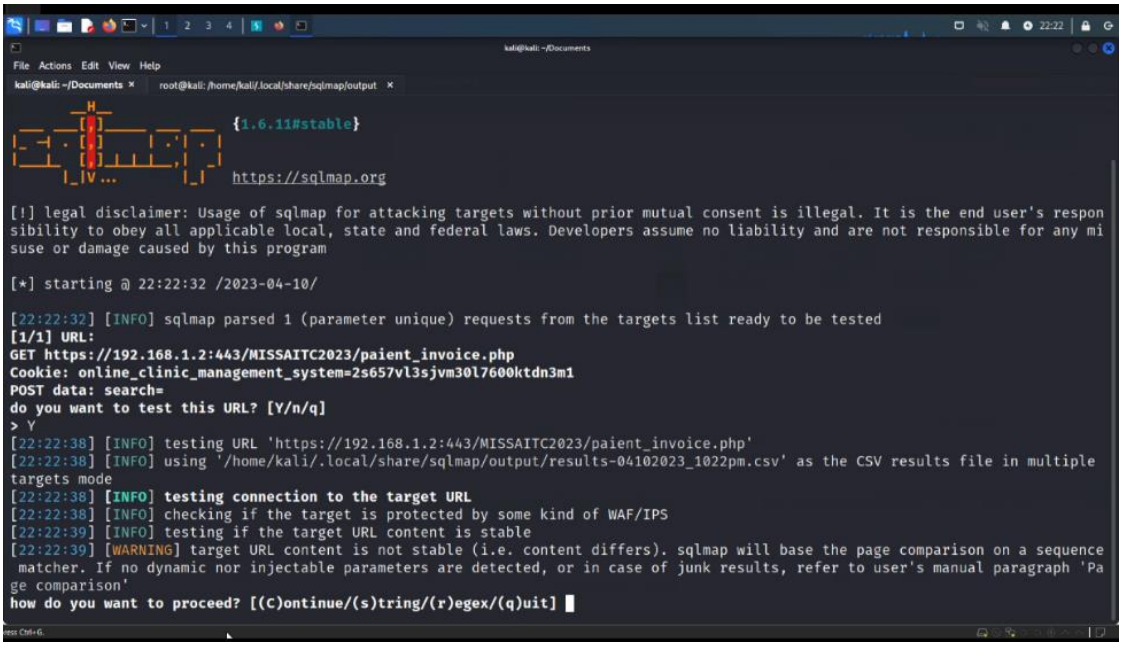
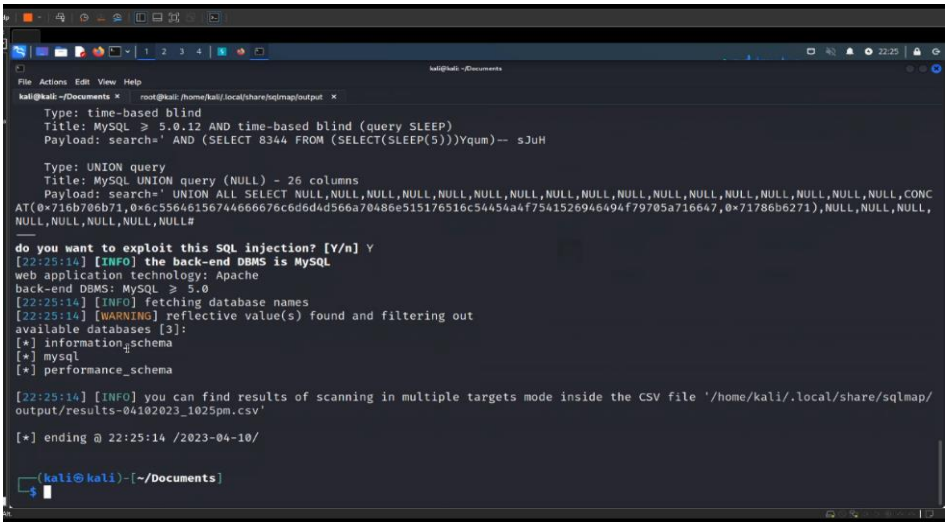


FIGURE 7: SQLMAP TESTING CONNECTION

FIGURE 8: THREE DATABASES

SQLMap found three databases.



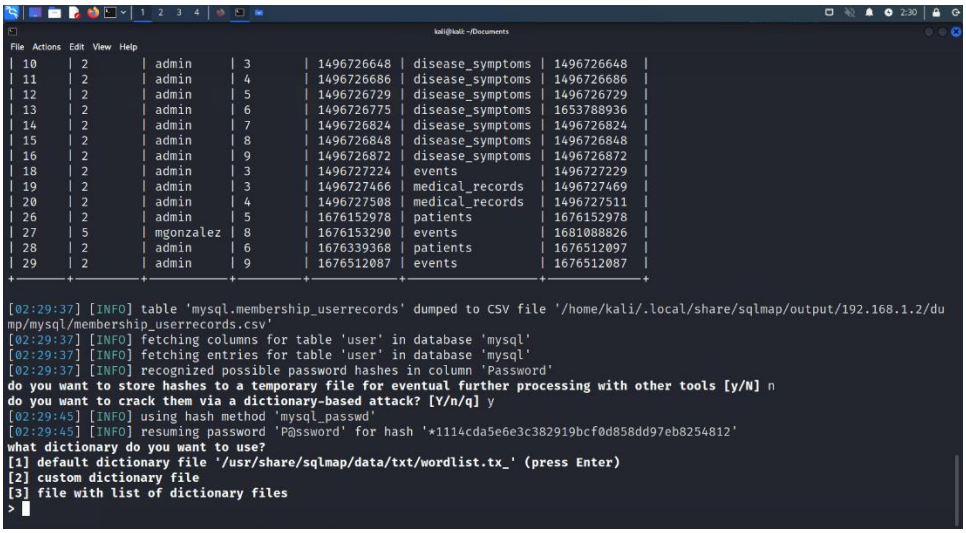


FIGURE 9: ROOT PASSWORD FOUND

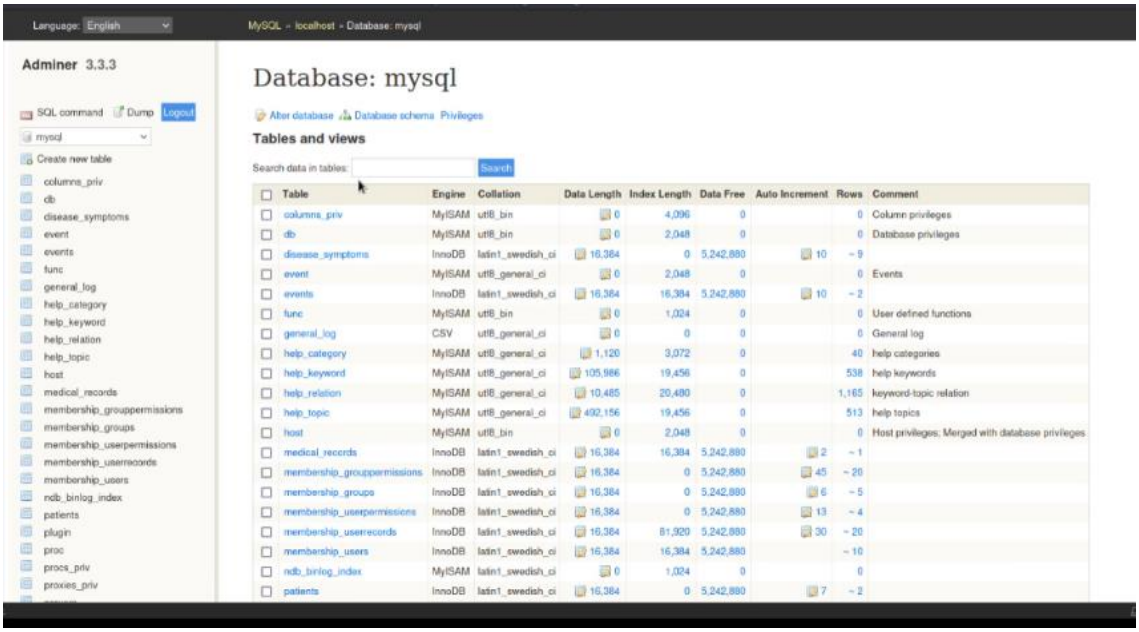


FIGURE 10: EXPOSED MYSQL DATABASE

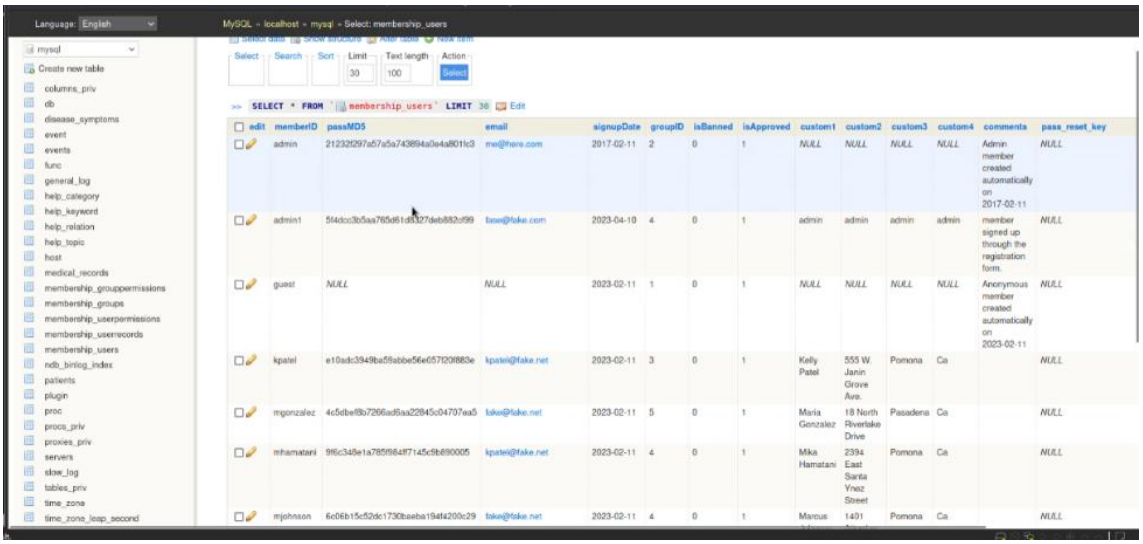


FIGURE 11: EDIT PERMISSIONS OPENED ON THE DATABASE

Finding # 3

Vulnerability Name: Insufficient Authorization | Reference ID. WASC-02
Likelihood: 2
Impact / Severity: 3
Risk Score: 6.0 (High)

Appointments are accessible even after logging out by going to calendar.php

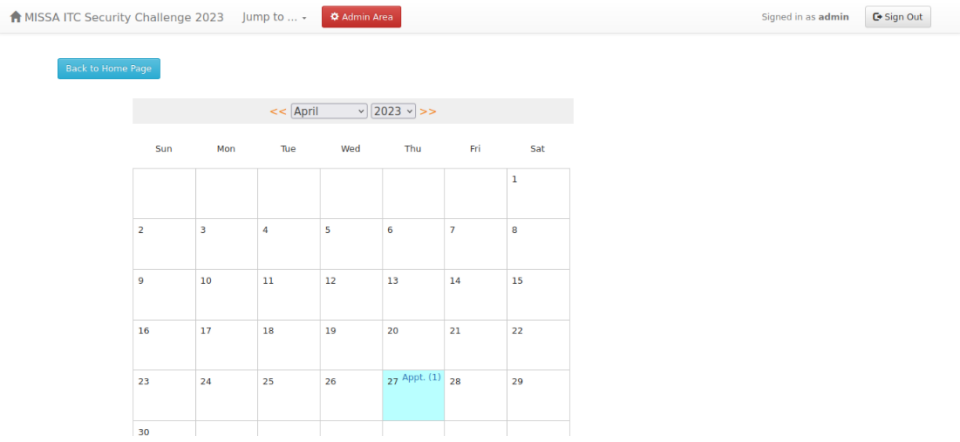


FIGURE 20: OPEN PORTS AND SERVICES TABLE

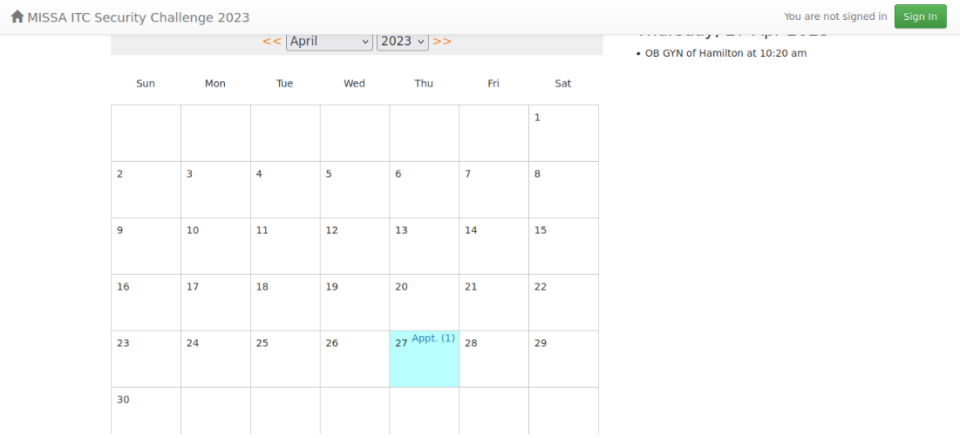


FIGURE 21: OPEN PORTS AND SERVICES TABLE

Remediation: Pomona Wellness Network should implement role-based access controls that restrict access to users based on their permission levels. This includes removing all session data

associated with a user after clearing, which includes temporary data such as calendar information.

Finding # 4

Vulnerability Name: Security Misconfiguration or Access Control Issue| Reference ID
Likelihood: 2
Impact / Severity: 2
Risk Score: 4.0 (Medium)

While being logged in as a patient with only patient credentials our team was able to find a vulnerability by typing single letters in the “Patient History” search bar. By typing the singular letter “a” it returned two patients and their private information.

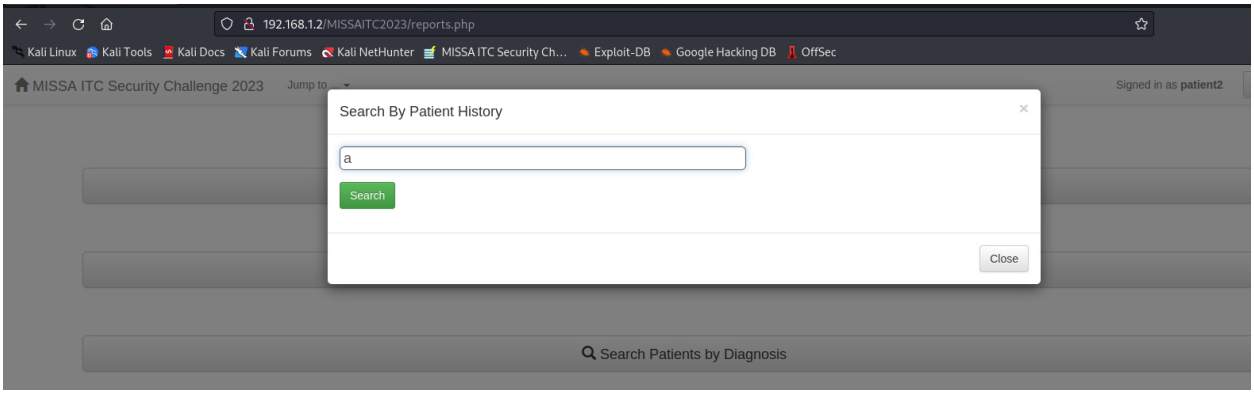
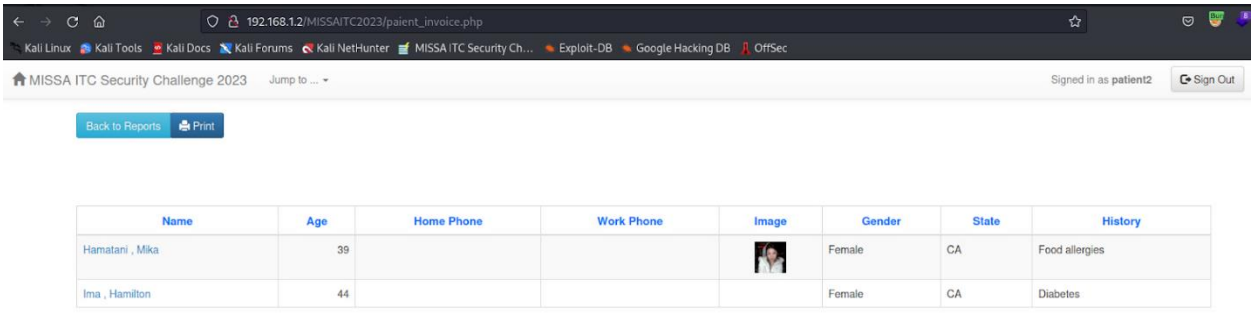


FIGURE 23: OPEN PORTS AND SERVICES TABLE



Below is the same example of changing the payload to be “a”, in which an attacker can do a brute force attack and see other single letters that may return information. Below with the payload of “a” we also see the verification of the page fully displaying there are “2 matching results” for the patients. Logged in as a patient account we should not be able to access or be able to see other patients as only admin has those privileges.

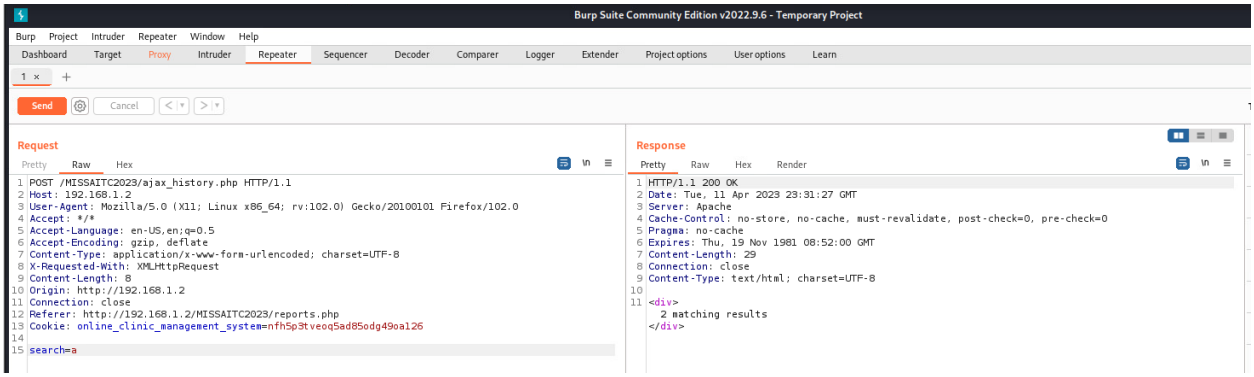
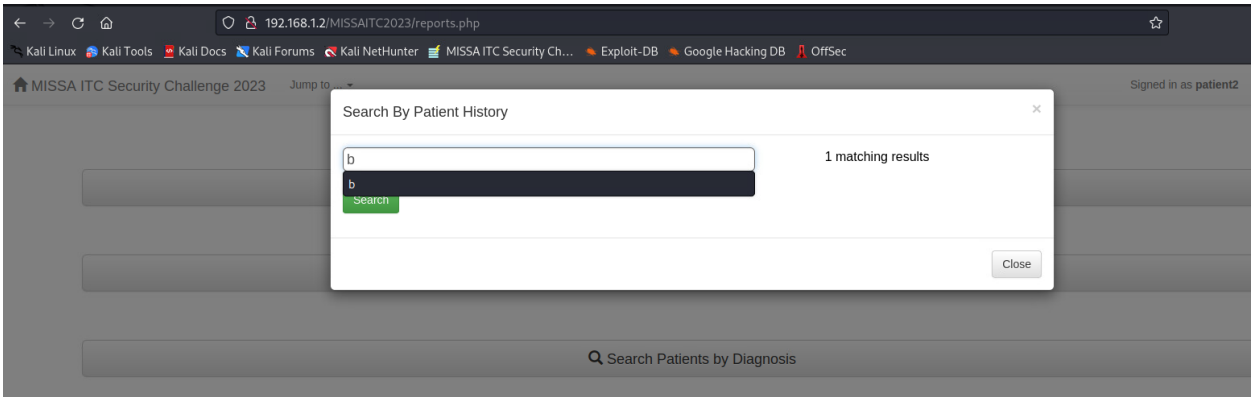
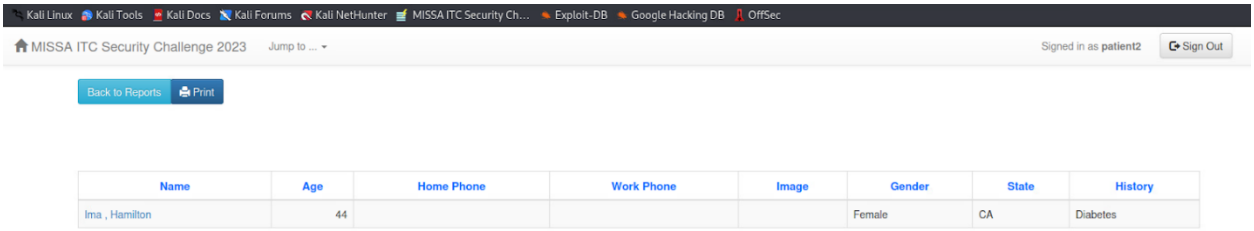


FIGURE 24: OPEN PORTS AND SERVICES TABLE

Below is an example of inputting the single letter “b” to return “1 matching result”. Because this user does not have administrative permissions, this is a breach of confidentiality to have these results viewable.





Like how our team mentioned in the sentence above, that this vulnerability works with singular letters, but not all. Below provides the example that it had no matching results when we used the letter “c”.

c example:

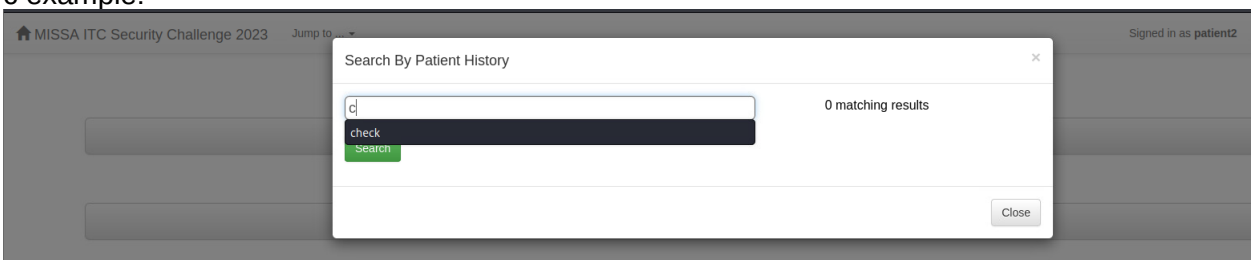
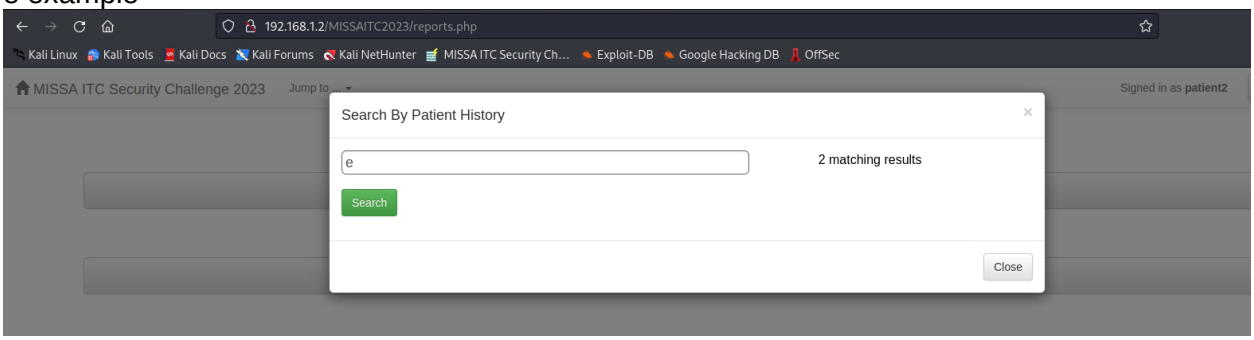
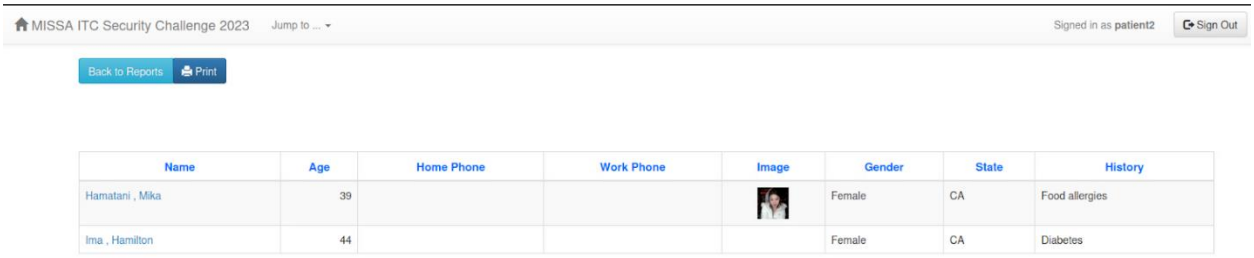


FIGURE 25: OPEN PORTS AND SERVICES TABLE

Like example a, when inputting the letter “e” we were able to get two matching results with the returning information of the two patients and their confidential information.

e example

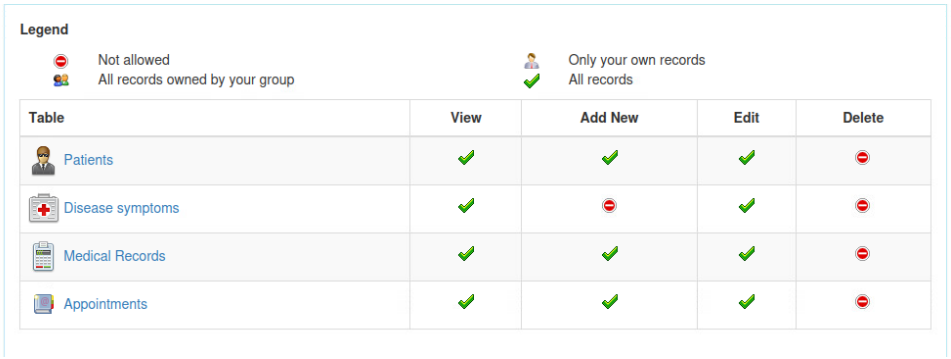




Finding # 5

Vulnerability Name: | Reference ID. WASC-14
Likelihood: 3
Impact / Severity: 2
Risk Score: 6 (High)

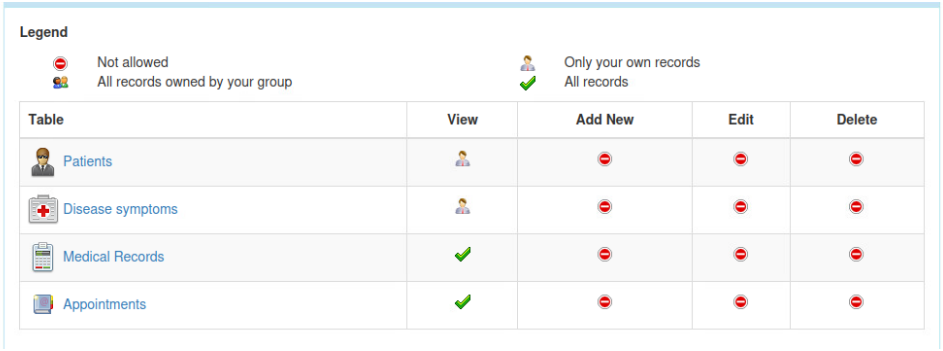
This organization has poor implementation of RBAC controls. The server contained accounts with elevated privileges that are not in accordance with the needs of their respective roles.



Legend

Not allowed
All records owned by your group
Only your own records
All records

Table	View	Add New	Edit	Delete
Patients	✓	✓	✓	⛔
Disease symptoms	✓	⛔	✓	⛔
Medical Records	✓	✓	✓	⛔
Appointments	✓	✓	✓	⛔



Legend

Not allowed
All records owned by your group
Only your own records
All records

Table	View	Add New	Edit	Delete
Patients	👤	⛔	⛔	⛔
Disease symptoms	👤	⛔	⛔	⛔
Medical Records	✓	⛔	⛔	⛔
Appointments	✓	⛔	⛔	⛔

Need to check the privileges of all roles to ensure they are properly in line with their respective roles

CONTEXT	PATIENT INFORMATION			DISEASE SYMPTOMS			APPOINTMENTS			CALENDAR			REPORTS		ADMIN AREA	
	View	Del	Edit	View	Del	Edit	View	Del	Edit	View	Del	Edit	View	Edit	View	Edit
UNAUTHENTICATED																
RECEPTIONIST							✓		✓	✓		✓				
PATIENT	Own						Own			Own		Own	Own			
NURSE	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
NURSE	✓		✓	✓	✓	✓	✓	✓	✓	✓						
ADMIN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Our findings of privileges when tested for each user:

Context	Patient Information			Disease Symptonms			Appointments			Calendar			Reports			Admin Area		
	View	Del	Edit	View	Del	Edit	View	Del	Edit	View	Del	Edit	View	Del	Edit	View	Del	Edit
Unauthenticated																		
Receptionist	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	NA		
Patient	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	NA		
Nurse	✓	✗	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	NA		
Nurse	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	NA		
Admin	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			

Finding # 6

Vulnerability Name: OpenSSH Vulnerability
Likelihood: 2
Impact / Severity: 3
Risk Score: 6 (High)

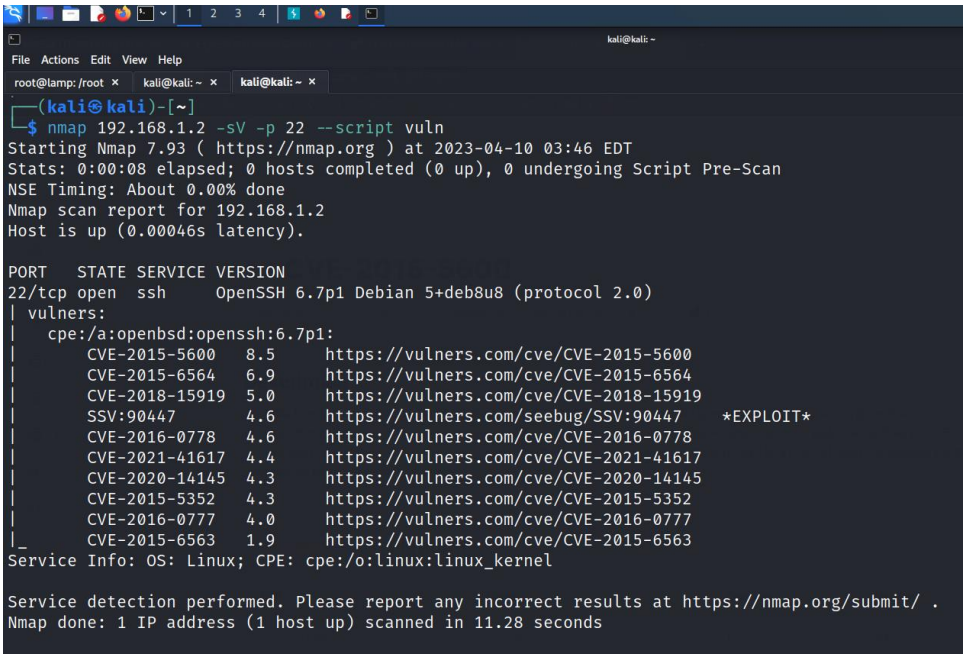
CVE-2020-14145 details the client-side application in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached by the client). NOTE: some reports state that 8.5 and 8.6 are also affected.

Remediation: This vulnerability is found on older versions of OpenSSH. By updating to the latest version, you can eliminate this threat.

Finding # 7

Vulnerability Name: Brute Force | Reference ID. WASC-11
Likelihood: 2
Impact / Severity: 3
Risk Score: 6.0 (High)

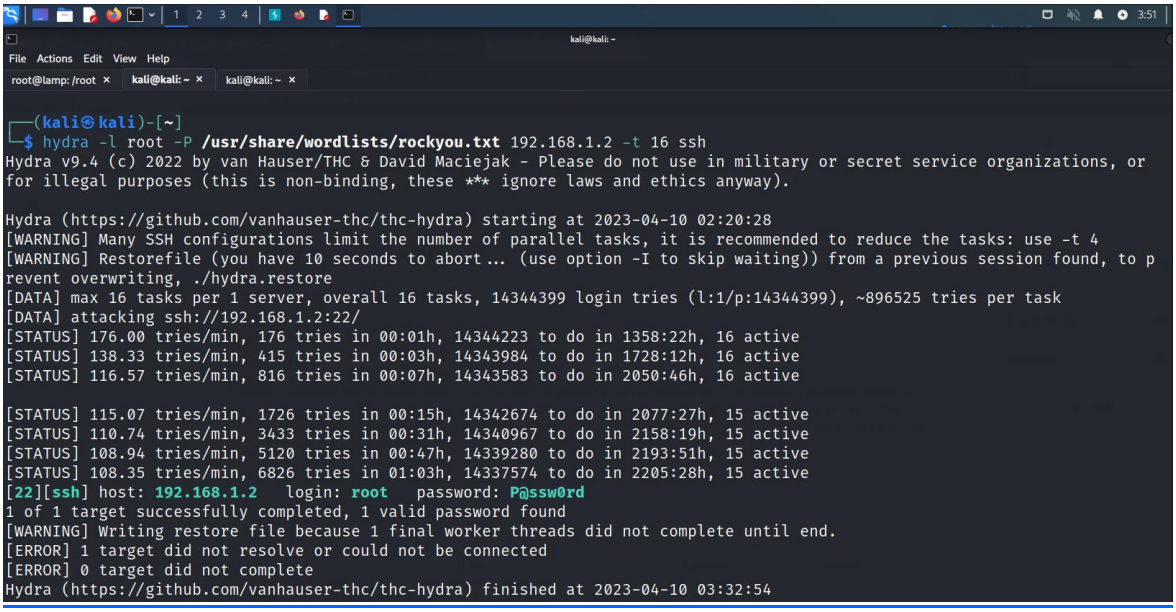
After we ran a Nmap vulnerability scan on port 22, using the –sV –p 22 –script vuln flags, we were able to identify additional CVEs that would be effective exploits against SSH.



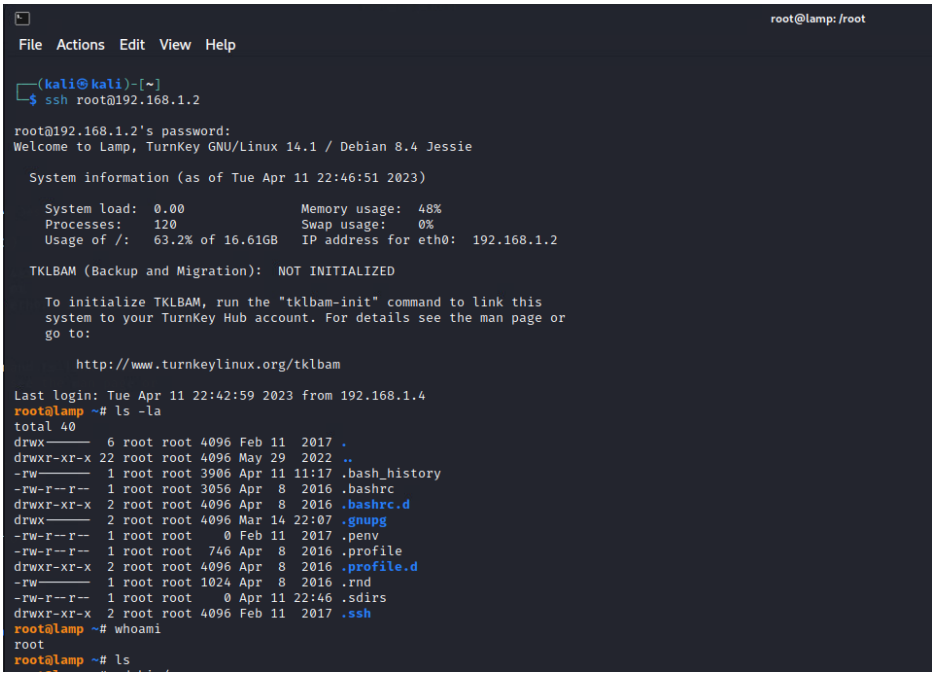
CVE-2015-5600 refers to a vulnerability wherein a function in the code for sshd does not properly restrict keyboard-interactive devices within a single connection, making it easy to conduct a brute force attack. This was attempted using hydra with the following command, revealing the password to be P@ssw0rd and resulting in a successful exploitation of SSH.

hydra -l root -P /usr/share/wordlists/rockyou.txt 192.168.1.2 -t 16 ssh

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5600



Remediation: This vulnerability is found on older versions of OpenSSH. By updating to the latest version, you can eliminate this threat



Since we used Hydra and were able to get the username and password of root, our team was able to ssh into root. By being able to ssh into root where we now have full administrative

privileges, we were able to access and manipulate sensitive data on the system, potentially causing significant damage or data loss if done maliciously.

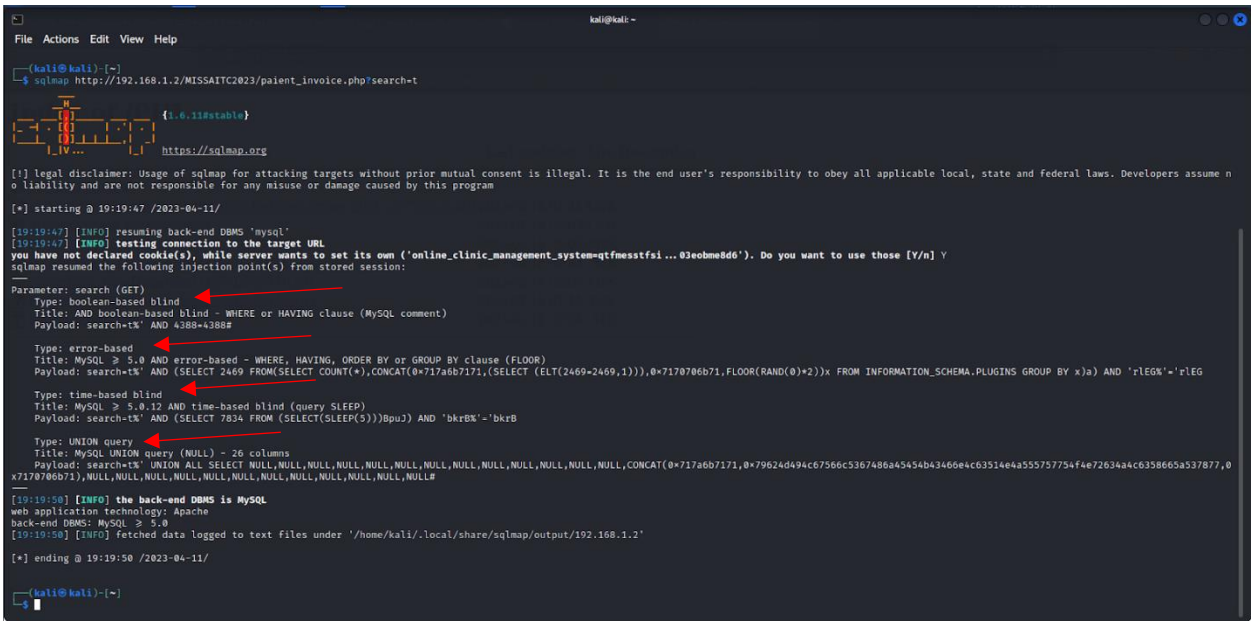
```
root@lamp /etc# cat shadow
root:$6$bhrJwOrf$LopY.3QfgSkJOqJ8MlRcI1D1e8PsrYyI4q6tECjGur9fVRSDf/minMr9udZtwG6d9N7f2ydCdLO.TvUUIQwID0:17208:0:99999:7:::
daemon:*:16560:0:99999:7:::
bin:*:16560:0:99999:7:::
sys:*:16560:0:99999:7:::
sync:*:16560:0:99999:7:::
games:*:16560:0:99999:7:::
man:*:16560:0:99999:7:::
lp:*:16560:0:99999:7:::
mail:*:16560:0:99999:7:::
news:*:16560:0:99999:7:::
uucp:*:16560:0:99999:7:::
proxy:*:16560:0:99999:7:::
www-data:*:16560:0:99999:7:::
backup:*:16560:0:99999:7:::
list:*:16560:0:99999:7:::
irc:*:16560:0:99999:7:::
gnats:*:16560:0:99999:7:::
nobody:*:16560:0:99999:7:::
systemd-timesync:*:16899:0:99999:7:::
systemd-network:*:16899:0:99999:7:::
systemd-resolve:*:16899:0:99999:7:::
systemd-bus-proxy:*:16899:0:99999:7:::
mysql:!:16899:0:99999:7:::
ntp:*:16899:0:99999:7:::
stunnel4:!:16899:0:99999:7:::
uuidd:*:16899:0:99999:7:::
sshd:*:16899:0:99999:7:::
postfix:*:16899:0:99999:7:::
shellinabox:*:16899:0:99999:7:::
root@lamp /etc#
```

Access to /etc/shadow file is a system file that stores password hashes of user accounts.

Finding # 8

Vulnerability Name SQL Injection| Reference ID
Likelihood: 2
Impact / Severity: 2
Risk Score: 4.0 (Medium)

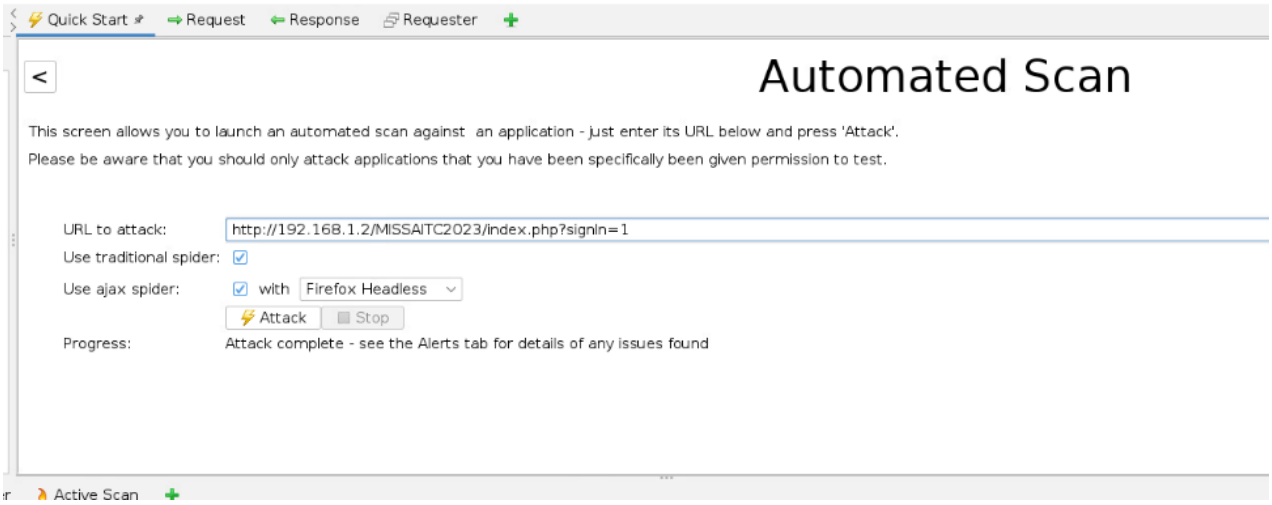
Running a nmap scan using the command
"sqlmap http://192.168.1.2/MISSAITC2023/paient_invoice.php?search=t"
While being logged in as a patient and in the section of "Patient History" we were given the http patient link. In the search bar is where the SQLMap injection took place. SQLMap found Boolean-based blind, error based, time-based blind, and UNION query, parameter vulnerabilities.



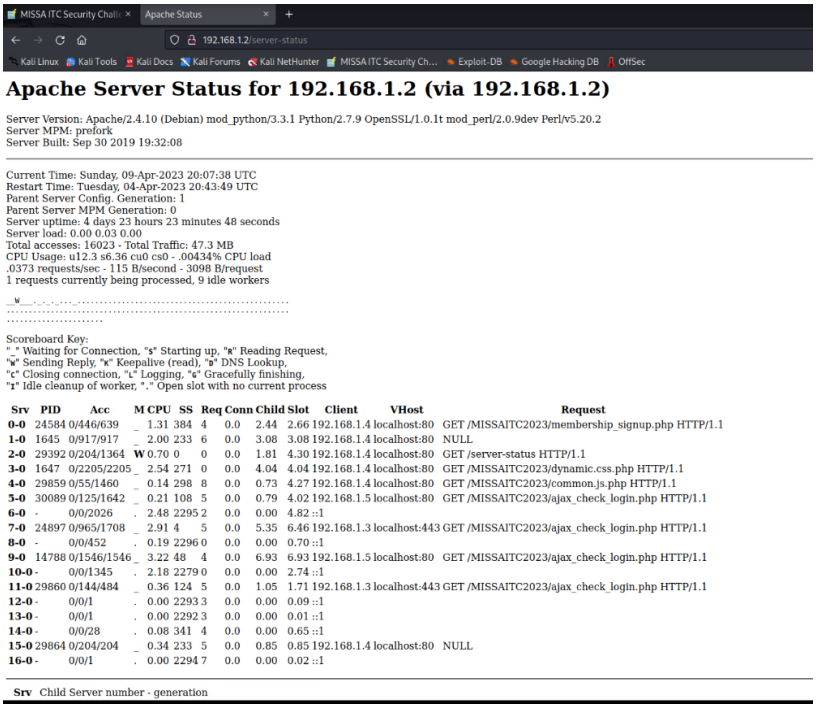
Finding # 9

Vulnerability Name: Information Leakage| Reference ID. WASC-13
Likelihood: 2
Impact / Severity: 2
Risk Score: 4.0 (Medium)

As previously mentioned in the Directory Indexing Vulnerability, using OWASP Zap tool we were able to detect the server status directory by investigating the results of dirbuster but also running the attack with OWASP ZAP using an Automated Scan on the URL:
http://192.168.1.2/MISSAITC2023/index.php?signIn=1



This level of information could be discovered through a directory attack with several capable and accessible tools. This can provide reconnaissance information by leaking administrative, configuration and version information. While this may offer some utility for remote troubleshooting, a more secure way to gauge this information is through a secure remote connection to the server and a process list using a command like top.



MISSA ITC Security Challe | Apache Status

192.168.1.2/server-status

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | MISSA ITC Security Ch... | Exploit-DB | Google Hacking DB | OffSec

Apache Server Status for 192.168.1.2 (via 192.168.1.2)

Server Version: Apache/2.4.10 (Debian) mod_python/3.3.1 Python/2.7.9 OpenSSL/1.0.1t mod_perl/2.0.9dev Perl/v5.20.2
Server MPM: prefork
Server Built: Sep 30 2019 19:32:08

Current Time: Sunday, 09-Apr-2023 20:07:38 UTC
Restart Time: Tuesday, 04-Apr-2023 20:43:49 UTC
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 4 days 23 hours 23 minutes 48 seconds
Server load: 0.00 0.03 0.00
Total accesses: 16023 - Total Traffic: 47.3 MB
CPU Usage: u12.3 s6.36 cu0 cs0 - .00434% CPU load
.0373 requests/sec - 115 B/second - 3098 B/request
1 requests currently being processed, 9 idle workers

_W___._._._.___.....
.....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "r" Reading Request,
"w" Sending Reply, "k" Keepalive (read), "d" DNS Lookup,
"c" Closing connection, "l" Logging, "g" Gracefully finishing,
"i" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	24584	0/446/639	_	1.31	384	4	0.0	2.44	2.66	192.168.1.4	localhost:80	GET /MISSAITC2023/membership_signup.php HTTP/1.1
1-0	1645	0/917/917	_	2.00	233	6	0.0	3.08	3.08	192.168.1.4	localhost:80	NULL
2-0	29392	0/204/1364	W	0.70	0	0	0.0	1.81	4.30	192.168.1.4	localhost:80	GET /server-status HTTP/1.1
3-0	1647	0/2205/2205	_	2.54	271	0	0.0	4.04	4.04	192.168.1.4	localhost:80	GET /MISSAITC2023/dynamic.css.php HTTP/1.1
4-0	29859	0/55/1460	_	0.14	298	8	0.0	0.73	4.27	192.168.1.4	localhost:80	GET /MISSAITC2023/common.js.php HTTP/1.1
5-0	30089	0/125/1642	_	0.21	108	5	0.0	0.79	4.02	192.168.1.5	localhost:80	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
6-0	-	0/0/2026	.	2.48	2295	2	0.0	0.00	4.82	::1		
7-0	24897	0/965/1708	_	2.91	4	5	0.0	5.35	6.46	192.168.1.3	localhost:443	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
8-0	-	0/0/452	.	0.19	2296	0	0.0	0.00	0.70	::1		
9-0	14788	0/1546/1546	_	3.22	48	4	0.0	6.93	6.93	192.168.1.5	localhost:80	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
10-0	-	0/0/1345	.	2.18	2279	0	0.0	0.00	2.74	::1		
11-0	29860	0/144/484	_	0.36	124	5	0.0	1.05	1.71	192.168.1.3	localhost:443	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
12-0	-	0/0/1	.	0.00	2293	3	0.0	0.00	0.09	::1		
13-0	-	0/0/1	.	0.00	2292	3	0.0	0.00	0.01	::1		
14-0	-	0/0/28	.	0.08	341	4	0.0	0.00	0.65	::1		
15-0	29864	0/204/204	_	0.34	233	5	0.0	0.85	0.85	192.168.1.4	localhost:80	NULL
16-0	-	0/0/1	.	0.00	2294	7	0.0	0.00	0.02	::1		

Srv Child Server number - generation

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	24584	0/446/639	_	1.31	384	4	0.0	2.44	2.66	192.168.1.4	localhost:80	GET /MISSAITC2023/membership_signup.php HTTP/1.1
1-0	1645	0/917/917	_	2.00	233	6	0.0	3.08	3.08	192.168.1.4	localhost:80	NULL
2-0	29392	0/204/1364	W	0.70	0	0	0.0	1.81	4.30	192.168.1.4	localhost:80	GET /server-status HTTP/1.1
3-0	1647	0/2205/2205	_	2.54	271	0	0.0	4.04	4.04	192.168.1.4	localhost:80	GET /MISSAITC2023/dynamic.css.php HTTP/1.1
4-0	29859	0/55/1460	_	0.14	298	8	0.0	0.73	4.27	192.168.1.4	localhost:80	GET /MISSAITC2023/common.js.php HTTP/1.1
5-0	30089	0/125/1642	_	0.21	108	5	0.0	0.79	4.02	192.168.1.5	localhost:80	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
6-0	-	0/0/2026	.	2.48	2295	2	0.0	0.00	4.82	::1		
7-0	24897	0/965/1708	_	2.91	4	5	0.0	5.35	6.46	192.168.1.3	localhost:443	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
8-0	-	0/0/452	.	0.19	2296	0	0.0	0.00	0.70	::1		
9-0	14788	0/1546/1546	_	3.22	48	4	0.0	6.93	6.93	192.168.1.5	localhost:80	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
10-0	-	0/0/1345	.	2.18	2279	0	0.0	0.00	2.74	::1		
11-0	29860	0/144/484	_	0.36	124	5	0.0	1.05	1.71	192.168.1.3	localhost:443	GET /MISSAITC2023/ajax_check_login.php HTTP/1.1
12-0	-	0/0/1	.	0.00	2293	3	0.0	0.00	0.09	::1		
13-0	-	0/0/1	.	0.00	2292	3	0.0	0.00	0.01	::1		
14-0	-	0/0/28	.	0.08	341	4	0.0	0.00	0.65	::1		
15-0	29864	0/204/204	_	0.34	233	5	0.0	0.85	0.85	192.168.1.4	localhost:80	NULL
16-0	-	0/0/1	.	0.00	2294	7	0.0	0.00	0.02	::1		

Srv Child Server number - generation
PID OS process ID
Acc Number of accesses this connection / this child / this slot
M Mode of operation
CPU CPU usage, number of seconds
SS Seconds since beginning of most recent request
Req Milliseconds required to process most recent request
Conn Kilobytes transferred this connection
Child Megabytes transferred this child
Slot Total megabytes transferred this slot

SSL/TLS Session Cache Status:
cache type: **SHMCB**, shared memory: **512000** bytes, current entries: **0**
subcaches: **32**, indexes per subcache: **88**
index usage: **0%**, cache usage: **0%**
total entries stored since starting: **0**
total entries replaced since starting: **0**
total entries expired since starting: **0**
total (pre-expiry) entries scrolled out of the cache: **0**
total retrieves since starting: **0** hit, **803** miss
total removes since starting: **0** hit, **0** miss

Remediation: Depending on the root cause of the information leakage, if related to Vulnerability No. WASC-14 then a misconfiguration of the web server is likely responsible, calling for a review of the security controls and if necessary, a migration to a server that can provide the adequate level of security.

Finding # 10

Vulnerability Name: OpenSSH Vulnerability
Likelihood: 2
Impact / Severity: 2
Risk Score: 4.0 (Medium)

CVE-2015-6564 details a use-after-free vulnerability on a function that can be found in the monitor.c file of sshd in OpenSSH versions prior to 7.0. This would allow users to gain privileges by leveraging control of the sshd uid to send an unexpectedly early MONITOR_REQ_PAM_FREE_CTX request.
https://cve.mitre.org/cgi-bin/cvename.cgi?name=publ-2015-6564

Remediation: This vulnerability is found on older versions of OpenSSH. By updating to the latest version, you can eliminate this threat.

Finding # 11

Vulnerability Name: Fingerprinting | Reference ID. WASC-45
Likelihood: 2
Impact / Severity: 2
Risk Score: 4.0 (Medium)

Running Nmap scan with flags –sV and –p- reveal 6 open ports.

```
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
80/tcp    open  http     Apache httpd
443/tcp   open  ssl/http Apache httpd
12320/tcp open  ssl/http ShellInABox httpd
12321/tcp open  ssl/http MiniServ 1.780 (Webmin httpd)
12322/tcp open  http     Apache httpd
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

After researching MiniServ 1.780, a lightweight web server, we came across a potential vulnerability CVE-2022-35132.

CVE-2022-35132 details a vulnerability that allows for OS command injection by remote users

Here since were logged in as root user, by using the “arp” command we can see the other machines and router. With this command we can see their machines’ IP addresses used. As a company you would want to keep your machine’s IP addresses private, so attackers cannot have the ability of running an Nmap scan for open port vulnerabilities against the IP’s.

```
root@lamp ~# ls -la
total 40
drwx------  6 root root 4096 Feb 11  2017 .
drwxr-xr-x 22 root root 4096 May 29  2022 ..
-rw-------  1 root root 3906 Apr 11 11:17 .bash_history
-rw-r--r--  1 root root 3056 Apr  8  2016 .bashrc
drwxr-xr-x  2 root root 4096 Apr  8  2016 .bashrc.d
drwx------  2 root root 4096 Mar 14 22:07 .gnupg
-rw-r--r--  1 root root    0 Feb 11  2017 .penv
-rw-r--r--  1 root root  746 Apr  8  2016 .profile
drwxr-xr-x  2 root root 4096 Apr  8  2016 .profile.d
-rw-------  1 root root 1024 Apr  8  2016 .rnd
-rw-r--r--  1 root root    0 Apr 12 00:43 .sdirs
drwxr-xr-x  2 root root 4096 Feb 11  2017 .ssh
root@lamp ~# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.1.5              ether   00:50:56:97:5c:cc   C                     eth0
pfSense.home.arpa        ether   00:50:56:97:78:a8   C                     eth0
192.168.1.4              ether   00:50:56:97:0a:0b   C                     eth0
192.168.1.3              ether   00:50:56:97:15:b5   C                     eth0
```

Finding # 12

Vulnerability Name: Directory Indexing | Reference ID. WASC-16
Likelihood: 2
Impact / Severity: 1
Risk Score: 2 (Low)

By running Nmap we were able to obtain access to multiple subdomains that potentially expose sensitive information and cross-site scripting vulnerabilities using the following command: This resulted in 7 directories in which logs, README, and server-status contained information that may help an attacker find further information.

nmap 192.168.1.2 -p 80 -sV --script=discovery

This resulted in the exposure of 7 subdirectories.

1. /logs
2. /README
3. /css
4. /html
5. /images
6. /js
7. /server-status

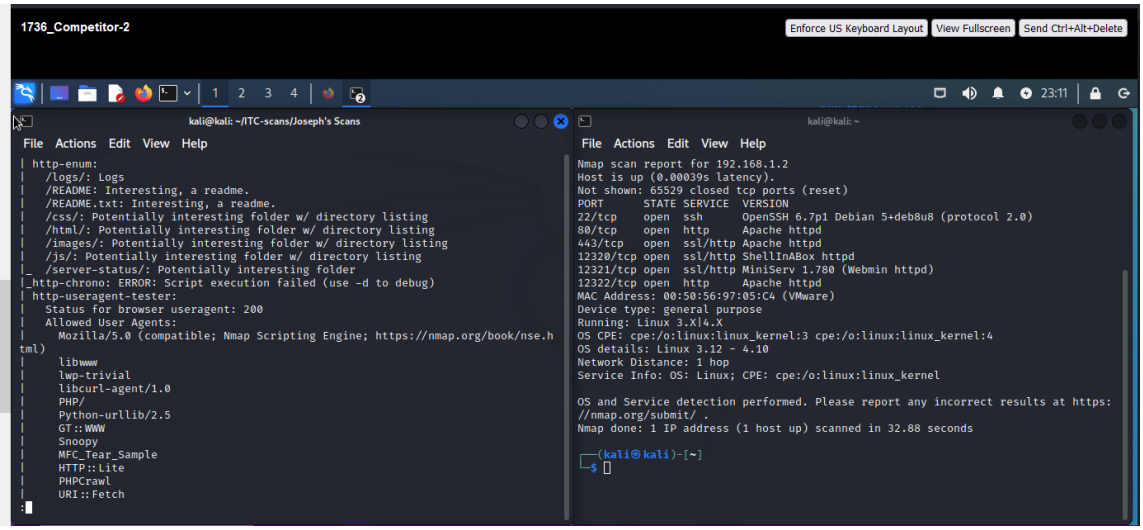


FIGURE 7: LIST OF SUBDIRECTORIES AND OPEN PORTS

The first of these, the log subdirectory, contained a file called Patient File-MikaHamatani-CONFIDENTIAL.xlsx which held sensitive data including health data and social security numbers of patients. This puts the organization at risk of incurring financial penalties in violation of the applicable sections of the Health Insurance Portability and Accountability Act (HIPAA): HIPAA Privacy Rule - §§ 262, 264 & HIPAA Security Rule - § 1173.

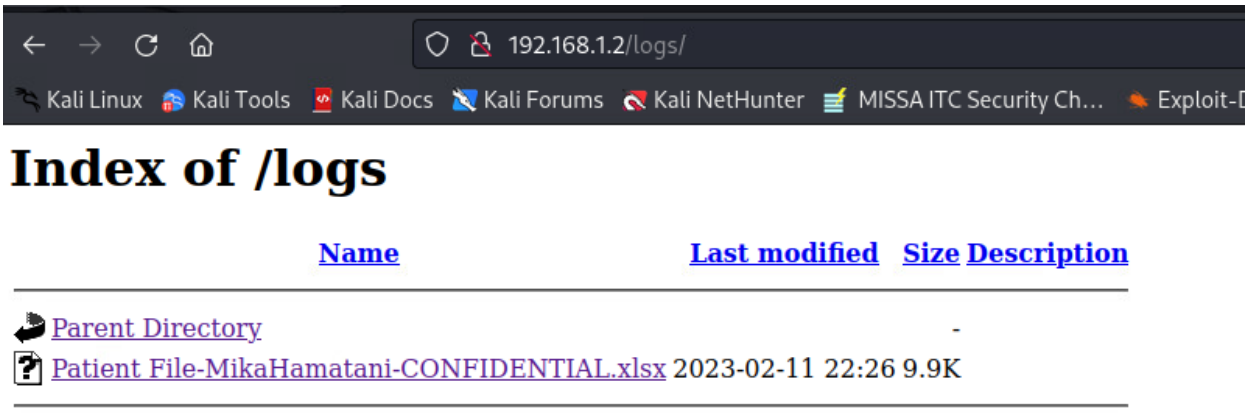
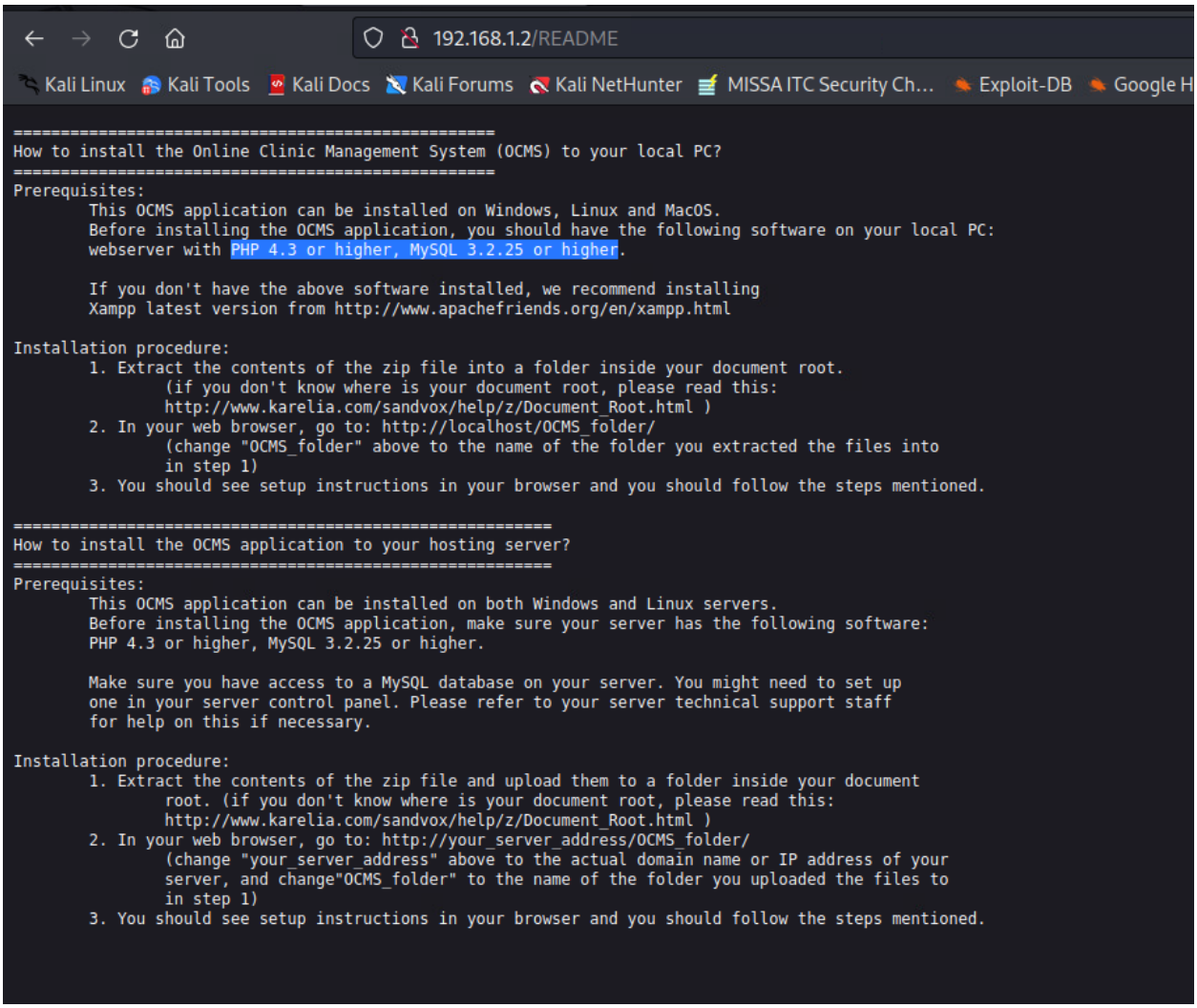
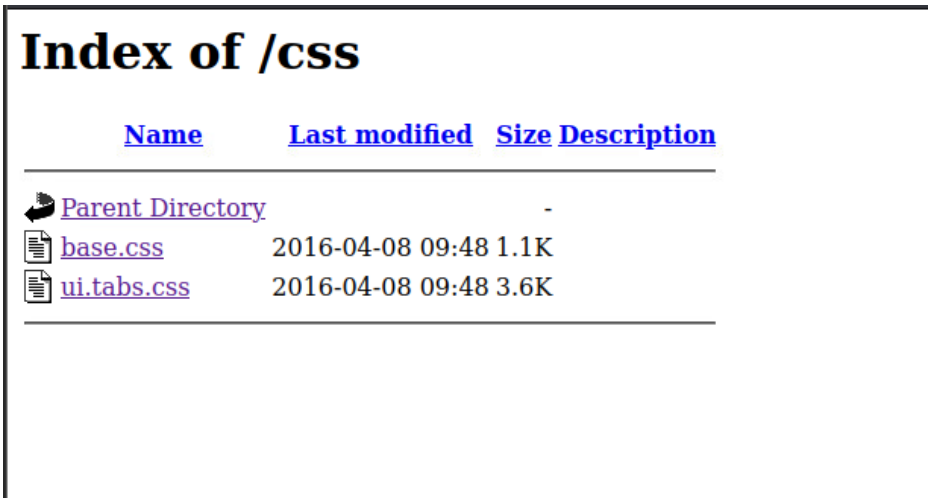


FIGURE 8: SUBDIRECTORY CONTAINING PATIENT INFORMATION

/README also contains a file with sensitive data about how to successfully install itself as root. Among other interesting bits of information, it mentions a version of MySQL that is required to successfully deploy the application on the server. This corroborates our findings from the nmap scan which indicated that a MySQL database was installed, giving us a lead for our vulnerability research and ultimately resulting in a successful SQL Injection attack.


192.168.1.2/README
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
MISSA ITC Security Ch...
Exploit-DB
Google H
==
How to install the Online Clinic Management System (OCMS) to your local PC?
==
Prerequisites:
This OCMS application can be installed on Windows, Linux and MacOS.
Before installing the OCMS application, you should have the following software on your local PC:
webserver with PHP 4.3 or higher, MySQL 3.2.25 or higher.
If you don't have the above software installed, we recommend installing
Xampp latest version from http://www.apachefriends.org/en/xampp.html
Installation procedure:
1. Extract the contents of the zip file into a folder inside your document root.
(if you don't know where is your document root, please read this:
http://www.karelia.com/sandvox/help/z/Document_Root.html)
2. In your web browser, go to: http://localhost/OCMS_folder/
(change "OCMS_folder" above to the name of the folder you extracted the files into
in step 1)
3. You should see setup instructions in your browser and you should follow the steps mentioned.
==========
How to install the OCMS application to your hosting server?
==========
Prerequisites:
This OCMS application can be installed on both Windows and Linux servers.
Before installing the OCMS application, make sure your server has the following software:
PHP 4.3 or higher, MySQL 3.2.25 or higher.
Make sure you have access to a MySQL database on your server. You might need to set up
one in your server control panel. Please refer to your server technical support staff
for help on this if necessary.
Installation procedure:
1. Extract the contents of the zip file and upload them to a folder inside your document
root. (if you don't know where is your document root, please read this:
http://www.karelia.com/sandvox/help/z/Document_Root.html)
2. In your web browser, go to: http://your_server_address/OCMS_folder/
(change "your_server_address" above to the actual domain name or IP address of your
server, and change"OCMS_folder" to the name of the folder you uploaded the files to
in step 1)
3. You should see setup instructions in your browser and you should follow the steps mentioned.

/css - contains the css file which can provide a scrupulous attacker with intel on what components are referenced on that page and a bit about the backend functionality of the site. This can be problematic when, for example, using descriptive variables that might tip off the presence of password reset functionality. In our example, at the bottom of the code in the ui.tabs.css file, there are a pair of comments about bug fixes which can inspire future attack vectors.




```
root@lamp www/css# nano bass.css
root@lamp www/css# nano base.css
root@lamp www/css# nano base.css
root@lamp www/css#
```

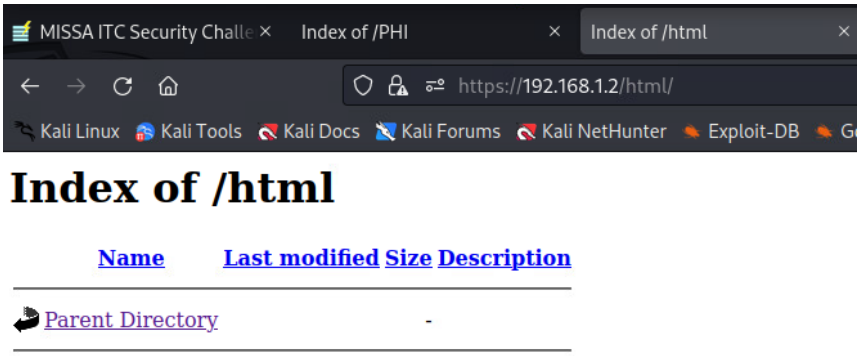
```
kali@kali: ~ ×   root@lamp: /var/www/css ×
 GNU nano 2.2.6

    display: block;
    margin: 0pt auto;
    border: 0pt none;
    border-style: none;
}
.fragment-content .icon {
    float: left;
    width: 64px;
    height: 80px;
    padding: 0pt 8px;
}
pre {
    margin-top: 1em;
    margin-bottom: 1em;
    background-color: #eee;
    padding: 0.75em 1.5em;
    font-size: 12px;
    border: 1px solid #ddd;
    border-right: #aaa 1px solid;
    border-bottom: #aaa 1px solid;
    overflow: auto;
}
#turnkey-credit {
    display: none;
}
```

https://cheatsheetseries.owasp.org/cheatsheets/Securing_Cascading_Style_Sheets_Cheat_Sheet.html

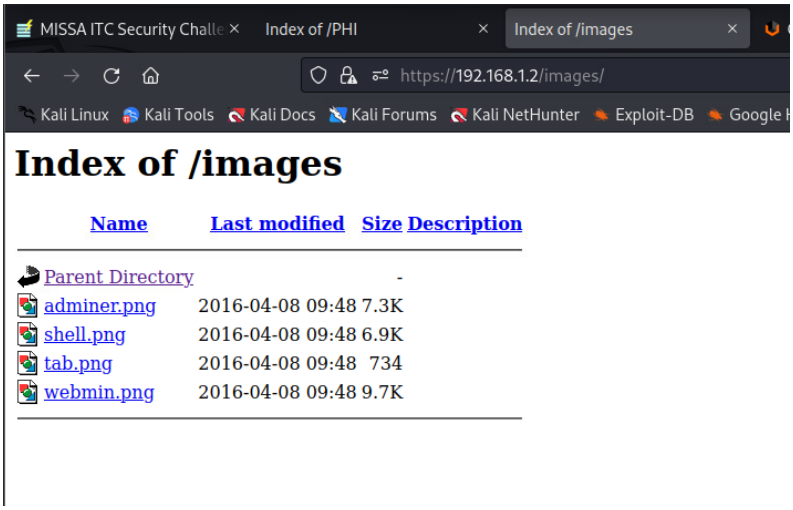
Risk: Low

/html - an empty directory. Contrary to its namesake, it doesn't contain any data on the html of the website.



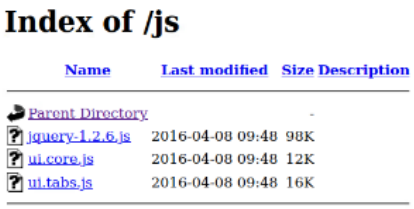
Risk: Low

/images - contains image files that give away a bit of information such as the adminer.png which demonstrates a database logo and webmin.png which might be indicative of an admin page.



Risk: Low

/js - contains among other things a .js file that contained comments regarding functions that seemed to be able to perform stack manipulation, making it vulnerable to a buffer overflow of the site. This could have a negative impact on server availability if followed up by a secondary attack that makes use of this feature in the code.

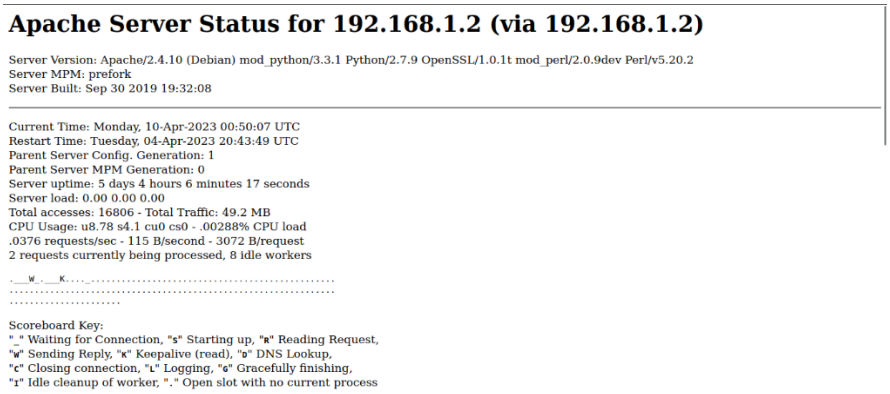


Index of /js

Name	Last modified	Size	Description
Parent Directory		-	
jquery-1.2.6.js	2016-04-08 09:48	98K	
ui.core.js	2016-04-08 09:48	12K	
ui.tabs.js	2016-04-08 09:48	16K	

Risk: Medium

/server-status - displays sensitive infrastructure information which could be used to deploy a follow up attack. The resource utilization and name of the programs being lend themselves well to some kind of brute force attack. This is further detailed later in the output of OWASP ZAP during another assessment.



Apache Server Status for 192.168.1.2 (via 192.168.1.2)

Server Version: Apache/2.4.10 (Debian) mod_python/3.3.1 Python/2.7.9 OpenSSL/1.0.1t mod_perl/2.0.9dev Perl/v5.20.2
Server MPM: prefork
Server Built: Sep 30 2019 19:32:08

Current Time: Monday, 10-Apr-2023 00:50:07 UTC
Restart Time: Tuesday, 04-Apr-2023 20:43:49 UTC
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 5 days 4 hours 6 minutes 17 seconds
Server load: 0.00 0.00 0.00
Total accesses: 16806 - Total Traffic: 49.2 MB
CPU Usage: u8.78 s4.1 cu0 cs0 - .00288% CPU load
.0376 requests/sec - 115 B/second - 3072 B/request
2 requests currently being processed, 8 idle workers

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

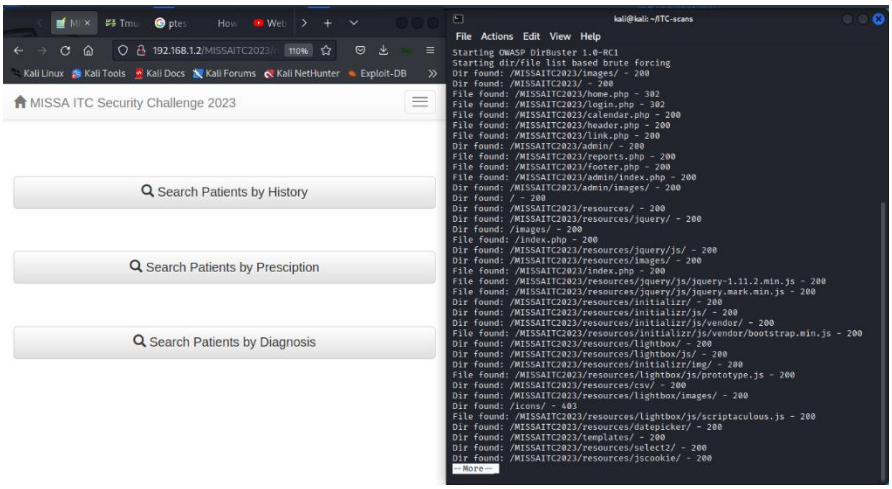
.__W_.__K.....
.....
.....

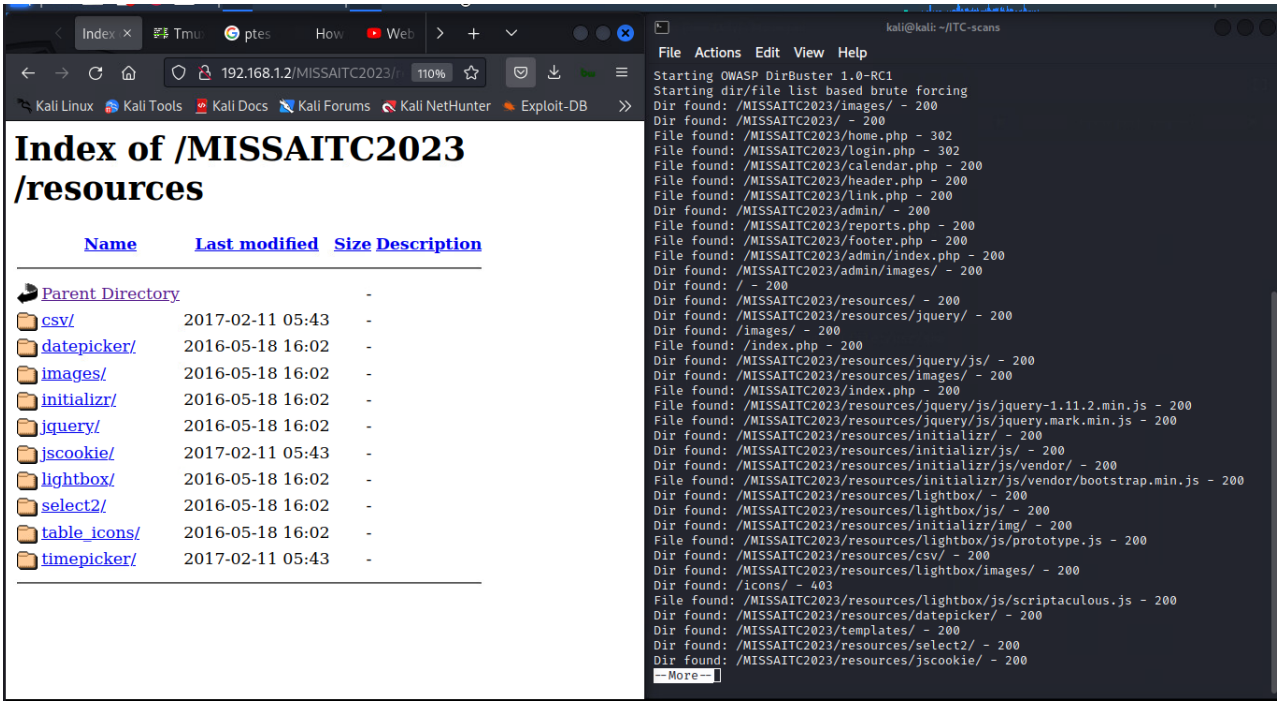
Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

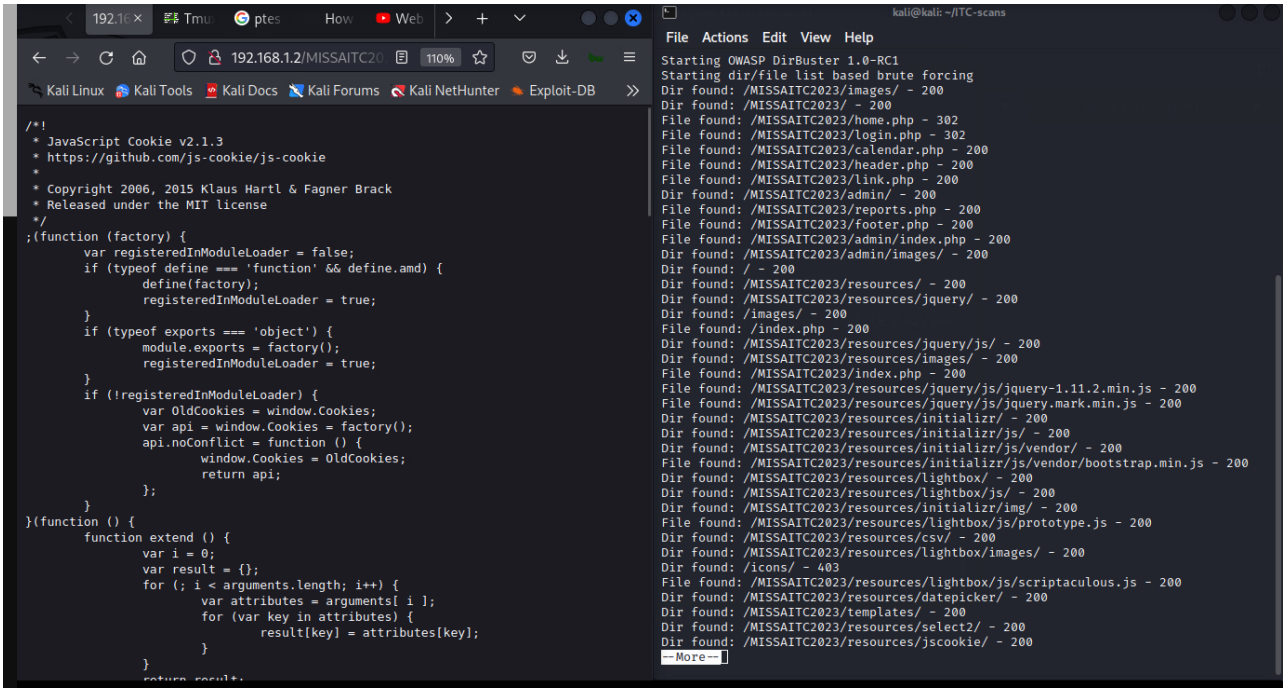
Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	-	0/0/734	.	1.66	1241	126	0.0	0.00	2.88	::1		
1-0	1645	0/976/976	_	2.25	122	1	0.0	3.21	3.21	192.168.1.3	localhost:443	GET /server-status HTTP/1.1
2-0	31559	0/34/1435	_	0.13	137	0	0.0	0.12	4.47	192.168.1.3	localhost:443	GET /html/ HTTP/1.1
3-0	31587	0/22/2274	_	0.08	15	26	0.0	0.06	4.18	192.168.1.3	localhost:443	NULL
4-0	29859	0/145/1550	W	0.51	0	0	0.0	0.94	4.48	192.168.1.5	localhost:80	GET /server-status/download/contact.php HTTP/1.1
5-0	32003	0/1/1707	_	0.01	1123	12	0.0	0.00	4.14	192.168.1.3	localhost:443	NULL
6-0	-	0/0/2034	.	0.04	4890	37	0.0	0.00	4.85	::1		
7-0	24897	0/1032/1775	_	3.19	130	0	0.0	5.50	6.61	192.168.1.3	localhost:443	GET /icons/image2.gif HTTP/1.1
8-0	31590	0/12/464	_	0.04	546	0	0.0	0.02	0.73	192.168.1.3	localhost:443	GET /icons/text.gif HTTP/1.1
9-0	14788	0/1621/1621	_	3.48	440	34	0.0	7.07	7.07	192.168.1.3	localhost:443	GET /css/base.css HTTP/1.1
10-0	31591	2/18/1363	K	0.06	0	6	11.4	0.04	2.78	192.168.1.3	localhost:443	GET /MISSAITC2023/index.php?signIn=1 HTTP/1.1
11-0	-	0/0/546	.	0.66	4908	14	0.0	0.00	1.81	::1		
12-0	-	0/0/1	.	0.00	19242	3	0.0	0.00	0.09	::1		
13-0	-	0/0/1	.	0.00	19241	3	0.0	0.00	0.01	::1		

Risk: Medium

/resources - By running Dirbuster we were able to get a recursive view of these subdirectories, the nested directories, and the files contained therein, including a resources subdirectory that contained files and folders that seem to be related to the calendaring functionality.cc




Index of /MISSAITC2023/resources
Name Last modified Size Description
Parent Directory -
csv/ 2017-02-11 05:43 -
datepicker/ 2016-05-18 16:02 -
images/ 2016-05-18 16:02 -
initializr/ 2016-05-18 16:02 -
jquery/ 2016-05-18 16:02 -
jscookie/ 2017-02-11 05:43 -
lightbox/ 2016-05-18 16:02 -
select2/ 2016-05-18 16:02 -
table_icons/ 2016-05-18 16:02 -
timepicker/ 2017-02-11 05:43 -
kali@kali: ~/ITC-scans
File Actions Edit View Help
Starting OWASP DirBuster 1.0-RC1
Starting dir/file list based brute forcing
Dir found: /MISSAITC2023/images/ - 200
Dir found: /MISSAITC2023/ - 200
File found: /MISSAITC2023/home.php - 302
File found: /MISSAITC2023/login.php - 302
File found: /MISSAITC2023/calendar.php - 200
File found: /MISSAITC2023/header.php - 200
File found: /MISSAITC2023/link.php - 200
Dir found: /MISSAITC2023/admin/ - 200
File found: /MISSAITC2023/reports.php - 200
File found: /MISSAITC2023/footer.php - 200
File found: /MISSAITC2023/admin/index.php - 200
Dir found: /MISSAITC2023/admin/images/ - 200
Dir found: / - 200
Dir found: /MISSAITC2023/resources/ - 200
Dir found: /MISSAITC2023/resources/jquery/ - 200
Dir found: /images/ - 200
File found: /index.php - 200
Dir found: /MISSAITC2023/resources/jquery/js/ - 200
Dir found: /MISSAITC2023/resources/images/ - 200
File found: /MISSAITC2023/index.php - 200
File found: /MISSAITC2023/resources/jquery/js/jquery-1.11.2.min.js - 200
File found: /MISSAITC2023/resources/jquery/js/jquery.mark.min.js - 200
Dir found: /MISSAITC2023/resources/initializr/ - 200
Dir found: /MISSAITC2023/resources/initializr/js/ - 200
Dir found: /MISSAITC2023/resources/initializr/js/vendor/ - 200
File found: /MISSAITC2023/resources/initializr/js/vendor/bootstrap.min.js - 200
Dir found: /MISSAITC2023/resources/lightbox/ - 200
Dir found: /MISSAITC2023/resources/lightbox/js/ - 200
Dir found: /MISSAITC2023/resources/initializr/img/ - 200
File found: /MISSAITC2023/resources/lightbox/js/prototype.js - 200
Dir found: /MISSAITC2023/resources/csv/ - 200
Dir found: /MISSAITC2023/resources/lightbox/images/ - 200
Dir found: /icons/ - 403
File found: /MISSAITC2023/resources/lightbox/js/scriptaculous.js - 200
Dir found: /MISSAITC2023/resources/datepicker/ - 200
Dir found: /MISSAITC2023/templates/ - 200
Dir found: /MISSAITC2023/resources/select2/ - 200
Dir found: /MISSAITC2023/resources/jscookie/ - 200
--More--



Remediation: Make sure that browsable directories do not leak confidential information or give access to sensitive resources. Additionally, use access restrictions or disable directory indexing. Begin by moving the data from the sensitive locations to designated infrastructure with added security controls like a CDN. Over time it would be beneficial to apply the NIST Privacy Framework to the infrastructure to mitigate the risk of breaches of sensitive data.

Here since were logged in as root user, by using the “arp” command we are able to see the other machines and router. With this command we are able to see their machines IP addresses used. As a company you would want to keep your machines IP addresses private, so attackers can not have the ability of running an Nmap scan for open port vulnerabilities against the IP’s.

Appendix A: Legal Memorandum

Hash Gatos Internal Counsel
3801 West Temple Avenue
Pomona, CA 91768

Legal Memorandum
HIPAA & CCPA in the context of Data Security Vulnerability Assessment

---

Facts: An anonymous healthcare provider has enlisted a group of cybersecurity penetration testers from California Polytechnic Institute of Pomona's Forensics and Security Technology Club. This group, hereinafter referred to as the Hash Gatos, has been contracted to perform a security assessment of their data security measures and privacy posture. As a health care facility in the state of California, this organization falls under the jurisdiction of the Health Insurance and Accountability Act of 1996 and the California Consumer Privacy Act ____. The organization will need to ensure compliance with these requirements to keep the data of their users secure and private. Failure to do so would incur liability for wrongful disclosure of personally identifiable information (PII). Hash Gatos will need to perform their assessment according to the rules of engagement provided by the facility and the applicable sections of the regulations in question to make any recommendations necessary to preserve the security and privacy of patient data.

Issues: What types of security controls for applicable sections of HIPAA, CCPA and any other legal requirements pertaining to the security and privacy of PII and ePHI (electronic Protected Health Information), will need to be tested when conducting a penetration test assessment?

Rules: The matter of ePHI security and privacy for a health care facility falls under multiple levels of regulatory control that should be considered.

HIPAA

1. The Health Insurance Portability and Accountability Act of 1996 (HIPPA: Pub.L. 104–191, 110 Stat. 1936, enacted August 21, 1996) with particular attention to
    i. a. HIPAA Privacy Rule - §§ 262, 264
    ii. b. HIPAA Security Rule - § 1173
2. Title 12 of the Code of Federal Regulations – Electronic Fund Transfers with special attention to Chapter 2, Subchapter A, Part 205.
3. Title 42 of the US Code, Chapter 7, Subchapter IX, Part C – Administrative simplification, with particular attention to § 1320d-6 **"Wrongful disclosure of individually identifiable information" pertaining to Social Security data standards.**

4. Title 42 of the Electronic Code of Federal Regulations – Public Health, with particular attention to 42 CFR § 403.812 "**HIPAA privacy, security, administrative data standards, and national identifiers" pertaining to data standards for Centers for Medicare & Medicaid Services**
5. Title 45 of the Electronic Code of Federal Regulations with particular attention to 45 CFR (A)(C) §§ 160.101 – 160.552 pertaining to **Public Welfare data standards**.

CCPA

6. The California Consumer Privacy Act of 2018 (CCPA: Civil Code Division 3, Part 4, Title 1.81.5 [§§1798.100 - 1798.199.100]) notably California Civil Code § 1798.145(c)(1)(A) (**Medical information governed by the Confidentiality of Medical Information Act**)

Application: The rules in question provide several factors to take into consideration.

Regulatory considerations:

The NIST Privacy Framework can provide a strong foundation for determining the security and privacy controls that would enable an organization to meet its obligations to its users, build trust and comply with the requirements of HIPAA and CCPA. The voluntary NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management (Privacy Framework) is intended to be widely usable by organizations of all sizes and agnostic to any particular technology, sector, law, or jurisdiction. This and its modular approach make it an ideal model to select for the purposes of framing this assessment. By following this framework, in addition to improving trust relations and adhering to compliance obligations, as well as the emergent cost savings, the following benefits will emerge:

- Taking privacy into account as they (re)design and deploy systems, products, and services that affect individuals;
- Communicating about their privacy practices; and
- Encouraging cross-organizational workforce collaboration—for example, among executives, legal, and information technology (IT)—through the development of Profiles, selection of Tiers, and achievement of outcomes.

Appendix B: Tools

Nmap: “Network Mapper” is an open-source tool that is used for network discovery, primarily to find vulnerable ports

SQLMap: An open-source tool that automates SQL injections

OWASP Zap: Searches directories for possible vulnerabilities and categorizes them based on severity

BurpSuite: Perform security testing of web applications from initial mapping to analysis

Hydra: Remote password cracking tool

Appendix C: Graph of Risks