Penetration Test Report

Security Engineering Division: CSIRT

Attack, Defense, and Analysis

Table of Contents

Description

X-CORP is a comprehensive penetration testing and incident response project focused on evaluating the security posture of X-CORP's systems. This was done through simulating a Red Team and Blue Team environment, conducting a penetration test with modular, repeatable, and effective security testing processes while leveraging security tools for offensive security. As for defensive security, the tasks were to test, implement, and monitor alerts generated from SIEM and security tools to respond to security incidents. Also, to conduct network forensics to analyze and report on malicious activity not detected by security tools.

The project involves various stages of pentesting, including reconnaissance, vulnerability assessment, exploitation, and post-exploitation analysis. It also incorporates the incident response cycle, including preparation, identification, containment, eradication, recovery, and lessons learned. The primary goal was to identify potential vulnerabilities and threats within X-CORP systems, assess their impact, implement new alerts and detections using data analysis, and provide recommendations for mitigating these risks.

Overview

You are working as a Security Engineer for X-CORP, supporting the SOC infrastructure. The SOC Analysts have noticed some discrepancies with alerting in the Kibana system and the manager has asked the Security Engineering team to investigate. You will start by confirming that newly created Kibana alerts are working, after which you will monitor live traffic on the wire to detect any abnormalities that are not reflected in the alerting system. You are to report back all your findings to both the SOC Manager and the Engineering Manager with appropriate analysis.

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. The following sections were covered in this test:

- **Reconnaissance**: Gathering information on the target system, including network mapping and identifying potential attack vectors.
- **Vulnerability Assessment**: Scanning the system for vulnerabilities using industry-standard tools.
- **Exploitation**: Attempting to exploit identified vulnerabilities to assess the extent of the security risk.
- **Post-Exploitation**: Analyzing the impact of successful exploits, including privilege escalation and data exfiltration.
- **Reporting**: Documenting findings, providing a risk assessment, and recommending security improvements.

Blue Team – Defensive Security

- **Preparation**:
    - Establish a well-documented IR plan that has assigned roles and prepare tools,

such as SIEM, EDR, scanner, alert testing, etc.
    - Set objectives and goals for the Red Team (phishing, lateral movement, etc.), and Blue Team (Define KPIs like MTTR.)
- **Identification**:
    - Monitoring & Detection: Use monitoring tools like SIEM or IDS/IPS to detect unusual activity. Also, review logs, alerts, and network traffic for anomalies (e.g., spikes in outbound traffic, unauthorized access attempts).
    - Triage Alerts: Prioritize alerts based on severity and potential impact. Confirm whether it's a false positive or a legitimate incident.
- **Containment**:
    - Immediate Actions: Disconnect affected systems from the network to prevent further spread. Apply firewall rules to block malicious IPs or domains.
    - Short-Term Containment: Isolate compromised accounts, endpoints, or servers while maintaining evidence integrity.
- **Eradication**:
    - Root Cause Analysis: Identify the root cause of the incident (e.g., exploited vulnerability, misconfigured system). Investigate using tools like memory forensics or malware analysis.
    - Remove Threats: Patch vulnerabilities, update software, and remove malware. Reconfigure access controls and security policies.
- **Recovery**:
    - Restore Systems: Reinstall or restore affected systems using clean backups. Verify that systems are no longer compromised.
    - Monitor Post-Recovery: Watch for recurring signs of compromise. Increase monitoring on previously affected systems.
- **Lessons Learned**:
    - Debrief & Document: For Red Team, provide a detailed report of tactics, techniques, and procedures (TTPs) used. As for Blue Team, document findings, response actions, and lessons learned.
    - Improve Defenses: Update playbooks, tools, and configurations based on gaps identified during the exercise. Conduct follow-up training or simulations to reinforce learning.

- **Communication**:
    - Internal Reporting:
        - Keep key stakeholders informed of progress during incident.
        - Use clear channels of communication for efficient coordination.
    - External Communication (if applicable):
        - Plan communication with affected parties, regulators, or the public in real-world scenarios.

Tools & Technologies

Tools used in this test include:

- Network & Vulnerability Scanners: Nmap, Netcat, and Nikto
- Exploitation Framework & Tools: Metasploit, John the Ripper, and Hydra
- Scripting Languages: Python, PHP, SQL, and XML

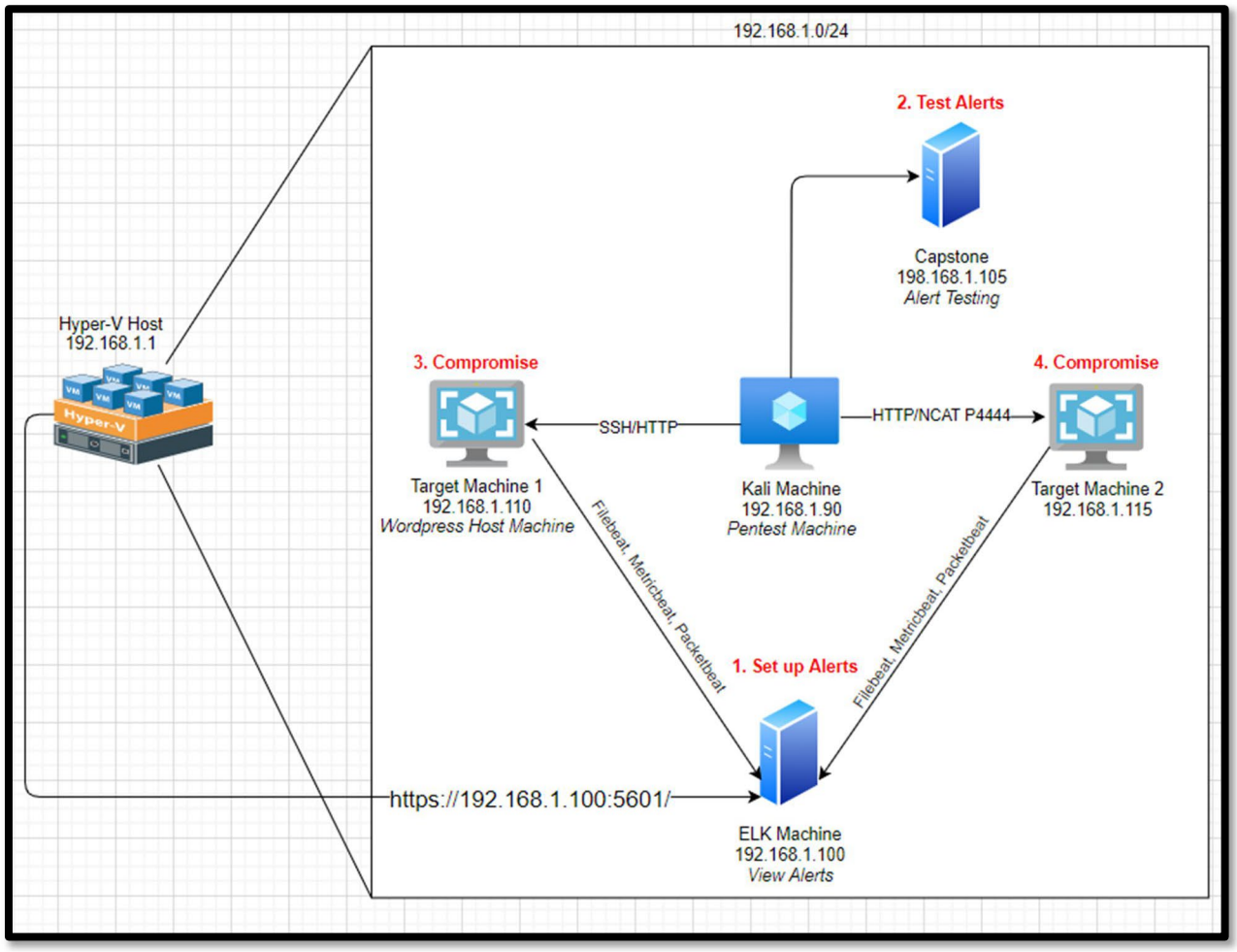
- SIEM Tools: ELK & Elastic
- Network & Forensics: Wireshark & VirusTotal

Red Team – Critical Vulnerabilities & Threats

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
Target 1	192.168.1.110	Target Machine 1 (Hosts WordPress app)
Target 2	192.168.1.115	Target Machine 2 (Hosts web server)
ELK	192.168.1.100	ELK (Network & Infrastructure Monitoring)
Capstone	192.168.1.105	Alert Testing Machine

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**, user login information is easily accessible to external users with no security. This allows access to database authentication information and configuration information where root credentials for the SQL database can be obtained.
- **Brute Force 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 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.
- **Cloudflare Protection Bypass** is a vulnerability where WordPress uses Cloudflare to increase site speed with a content delivery network, ping commands and DNS tools 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.

- **Directory Path Traversal** in Target 2 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** refer to software elements (such as libraries, frameworks, modules, or plugins) within an application that have known security weaknesses, flaws, or software elements that are running on older versions, in which there may be newer releases or patches available. The organization is running an outdated version of WordPress.
- **Cryptographic Failures** are 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 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.

    - **Cross-Site Scripting (XSS)** is a type of security vulnerability commonly found in web applications which occur when malicious scripts are injected into web pages that are viewed by other users.
    - **SQL Injection** occurs when malicious SQL code is inserted into input fields or parameters used in SQL queries within a web application.
    - **Command Injection** involves the insertion or execution of malicious commands within a system or application where user input is directly passed to a system shell or interpreter without proper validation or sanitization.
    - **LDAP Injection** is the manipulation of Lightweight Directory Access Protocol queries within an application that interacts with LDAP directories, such as Windows Active Directory. LDAP is commonly used for accessing and managing directory information services, such as user authentication and authorization.
    - **XML Injection** occurs when untrusted data is inserted into eXtensible Markup Language documents without proper validation or sanitization. XML injection vulnerabilities can arise in applications that generate XML documents dynamically, such as web applications or web services like WordPress.

Critical Incidents discovered on organization's network:

- **Malware** – a Trojan Malware was discovered on the network that was downloaded locally and infected multiple hosts on network.

- **AD Domain** – a private Active Directory domain was created on corporate network to conduct torrenting, avoid detection, and stream videos online.
- **Torrenting** – a few users were discovered torrenting on the network by uploading and downloading copyrighted materials.

Attack Narrative – Vulnerability & Service Exploitation

Scanning Ports with Nmap

Network scans (**Nmap**) showed IP addresses, ranges, operating system information, running services, scripts, versions, open ports, and a trace route. Nmap 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 discovered that ports 22, 80, and 111 could be points of entry and port 4444 can be used to upload a malicious payload.

WPSCAN & Enumerating WordPress Service

Running a WPSCAN on WordPress service to enumerate a list of users to target, scans identify Michael and Steven as users of the WordPress application. Michael has weak user credentials allowing for easy access into 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, a user can SSH into his user account and navigate to restricted directories on the system. Also, users can navigate to restricted directories by manipulating variables that reference 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:~$
```

External users can locate the WordPress configuration directory which contains exposed sensitive and confidential information like authentication and configuration information. This information is in plaintext, unencrypted, and improperly secured. Once the restricted WordPress directory is located, it can be 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$
```

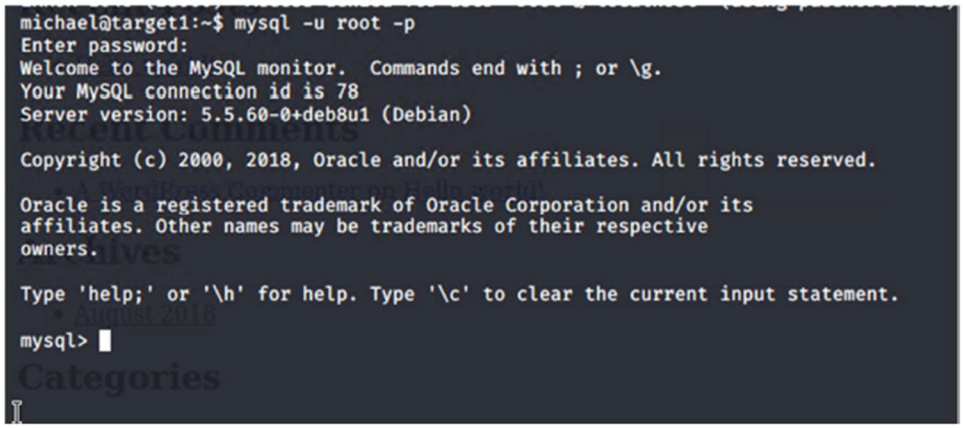
The WordPress configuration file is vulnerable and accessible via user on the target. The configuration file root credentials are stored in plain text in the WordPress directory which should not be accessible to a non-system administrator. The attacker is able to obtain administrative credentials logging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

Administrative credentials →
Username: **root**
Password: **R@v3nSecurity**

```
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', '');
```

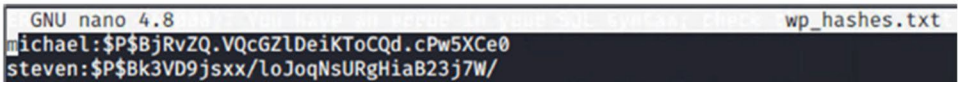
Using discovered root credentials, users can exploit the SQL database revealing the password hashes for the site's users (Steven & Michael). Navigating across SQL, the user hash table can be dumped. An external user can dump the hashes (exfiltrate) into a plain text file using nano and John-Ripper to crack hashes and obtain additional credentials (Steven's password).

Unauthorized Access to SQL:

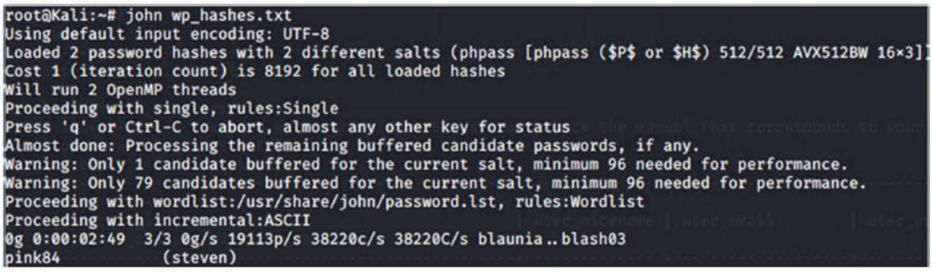


Dumping Hashes (hash exfiltration):

Copying hashes to plain text file using nano:



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

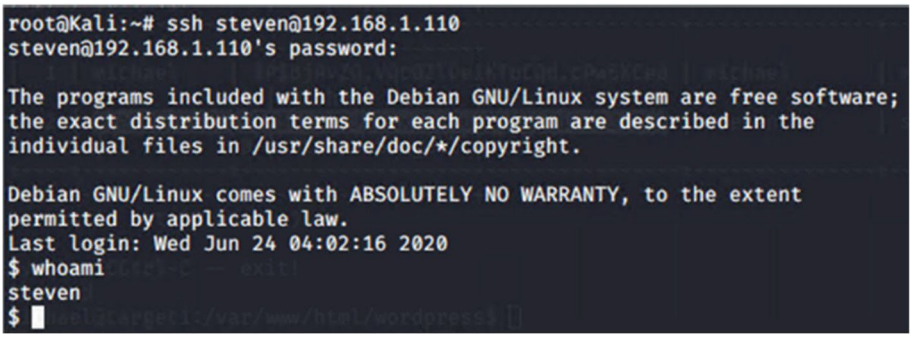


Privilege Escalation & Rooting Target

Once Steven's credentials are discovered, it can be used to access his account via remote command execution (SSH). Steven's account can be used to spawn a root shell with a simple python script via sudo to escalate privileges to root user. Once the shell is spawned, the target is successfully rooted, and full system control can be gained with the ability to place malware which could grant persistence turning this into an advanced persistent threat (APT).

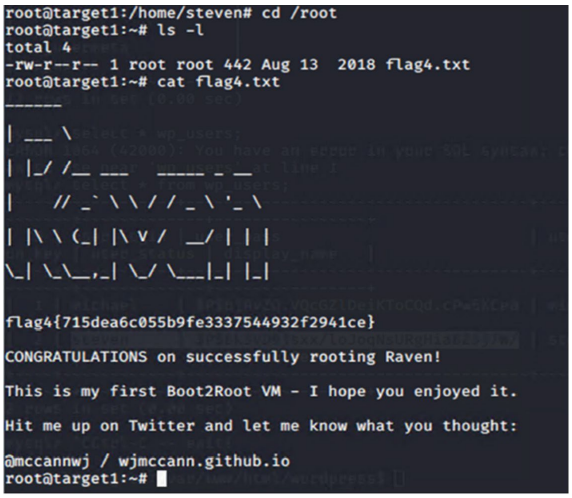
Python Script:

SSH & WHOAMI:



Privilege escalation and Root shell spawning with Python:

Steven's machine successfully rooted:



Avoiding Detection

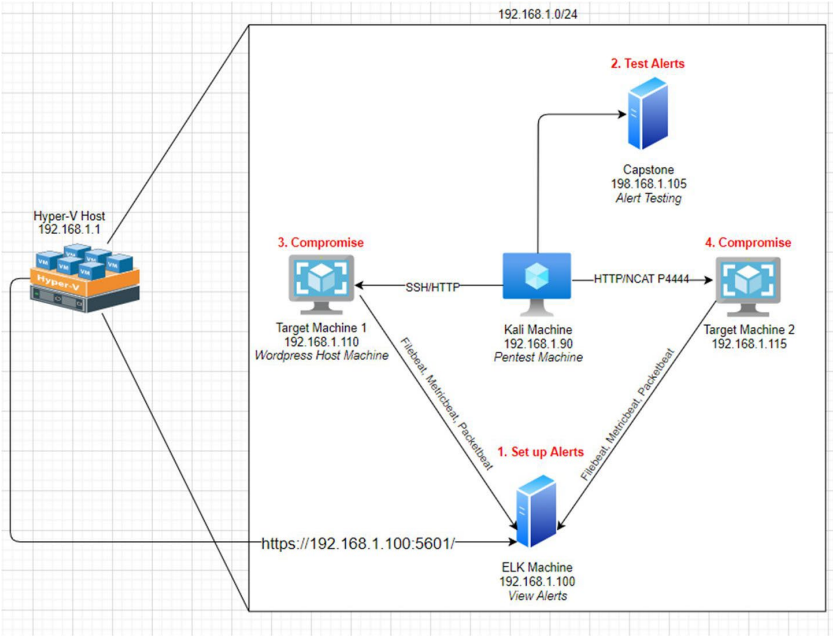
- Stealth Exploitation – Port Scanning:
    - Monitoring Overview:
        - HTTP Request Size Monitor Alert
        - Sum of request bytes exceed 3500 per minute
        - Alert triggered after 3500 bytes
- Mitigating Detection:
    - Alert triggered after 3500 bytes.
    - Slower scans by adjusting timing parameters.
    - Fragmentation by obscuring contents of the packets.
    - Proxy chains and anonymization by rerouting scan traffic to obfuscate the source of the scans.
    - Protocol-Based scanning using less common protocols like ICMP or SCTP.
- Stealth Exploitation – Enumeration
    - Monitoring Overview:
        - Excessive HTTP Errors Alert
        - Top 5 response status codes exceed 400 every 5 minutes
        - Alert triggered after 400 threshold exceeded
- Mitigating Detection:
    - Enumerating is noisy
    - Stagger attempts
    - Reduce scan intensity
    - Passive enumeration techniques
    - Stealthier scans by randomizing timing, encrypting channels, and limiting concurrent connections.
- Stealth Exploitation – LFI
    - Monitoring Overview:
        - CPU Usage Monitor Alert
        - CPU usage process exceeds 0.5/minute
        - Alert is triggered after 50%
- Mitigating Detection:
    - Alert is highly reliable
    - Fileless Malware & LOTL
    - Evade detection through refactoring
    - Obfuscation
    - Request splitting
    - User-agent spoofing
    - IP address rotation
    - Slow and Low technique
    - Bypass logging
    - Custom exploits

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 primary focus of this attack is **Target 1**, identified by the IP address 192.168.1.110. Both Target 1 and Target 2 are Apache web servers configured with Secure Shell (SSH) access, which means that ports 80 and 22 represent potential points of entry for unauthorized access or malicious activity. To address this risk and enhance monitoring capabilities, several critical alerts have been implemented to keep track of traffic patterns and identify any suspicious behavior. These alerts include the monitoring of excessive HTTP errors, which may indicate probing or exploitation attempts; HTTP request size, to detect unusually large or malformed requests that could be part of an attack; and CPU usage, as unexpected spikes might signal resource-intensive malicious activity. Together, these measures aim to provide robust oversight and early warning of potential threats targeting the servers.

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 by:

- Implementing input validation, query parameterization, and a web application firewall.
- Hardening 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, such as using AI when dealing with extremely large volumes of data.
- Implementing proper security controls and baseline configurations.
- Setting alerts and implementing 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: CVSS Score & Playbook – Vulnerability & Mitigation

Exploitation of Open Ports

Rating: **High**

CVSS Score: ~7.5 (High)
Explanation: Exposed ports can allow attackers to access services and exploit vulnerabilities. If these services are critical, the impact can escalate to availability or confidentiality breaches.

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.

Enumeration (CWE-23)

Rating: **Medium**

CVSS Score: ~5.5 (Medium)
Explanation: Account enumeration may allow attackers to identify valid usernames or system structures, aiding brute-force or phishing attacks. It's less impactful unless chained with other exploits.

Impact: User enumeration is what allowed targets to be discovered on the WordPress application.

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: **Medium**

CVSS Score: ~6.5 (Medium)

Explanation: Weak passwords are easy to crack, allowing unauthorized access. If high-privilege accounts use weak credentials, the score can increase.

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: **High**

CVSS Score: ~7.2 (High)

Explanation: Misconfigurations, such as default settings or improper permissions, leave systems vulnerable to unauthorized access or data leaks.

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: **Medium**

CVSS Score: ~6.4 (Medium)

Explanation: Improper parsing can enable injection attacks or abuse of the XML-RPC interface for unauthorized actions, like DoS or data leakage.

Impact: WordPress XML-RPC parsing is susceptible to DoS attacks by executing pingback.ping commands and XML injection. Several affected WordPress installations can launch a botnet level attack.

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

XML-RPC Ping

Rating: **Medium**

CVSS Score: ~5.3 (Medium)

Explanation: Vulnerable pings may enable DDoS amplification or unauthorized access, but the direct impact is often limited without chaining.

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**

CVSS Score: ~8.7 (High)

Explanation: Bypassing Cloudflare can expose services directly, leading to DDoS, data exposure, or privilege escalation attacks.

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**

CVSS Score: ~7.5 (High)

Explanation: LFI allows attackers to access sensitive files or execute unauthorized code. If chained with RCE (Remote Code Execution), the score can escalate.

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 and unexpected file inclusion output

Insecure Design – ***OWASP Top 10***

Rating: **High**

CVSS Score: ~7.0 (High)

Explanation: Fundamental flaws in architecture can lead to widespread exploitation, though impact varies based on specific implementation.

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: **High**

CVSS Score: ~8.0 (High)

Explanation: Using outdated software can expose systems to known exploits, significantly increasing attack surfaces.

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: **High**

CVSS Score: ~8.2 (High)

Explanation: 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 (XSS, SQL, Command, LDAP, XML) – ***OWASP Top 10***

Rating: **CRITICAL**

CVSS Score: ~9.0 (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/Elastic to monitor activity
- Beats
  - **Filebeat** – helps collect file logs with real-time insight into log data.
  - **Metricbeat** – collects system metrics and ingests system metric logs.
  - **Packetbeat** – helps with packet data analysis.
  - **Heartbeat** – helps with monitoring the health of systems.
  - **Auditbeat** – collects and ingests audit data, such as user activity, process changes, and file integrity monitoring from servers to Elastic.