Final Engagement

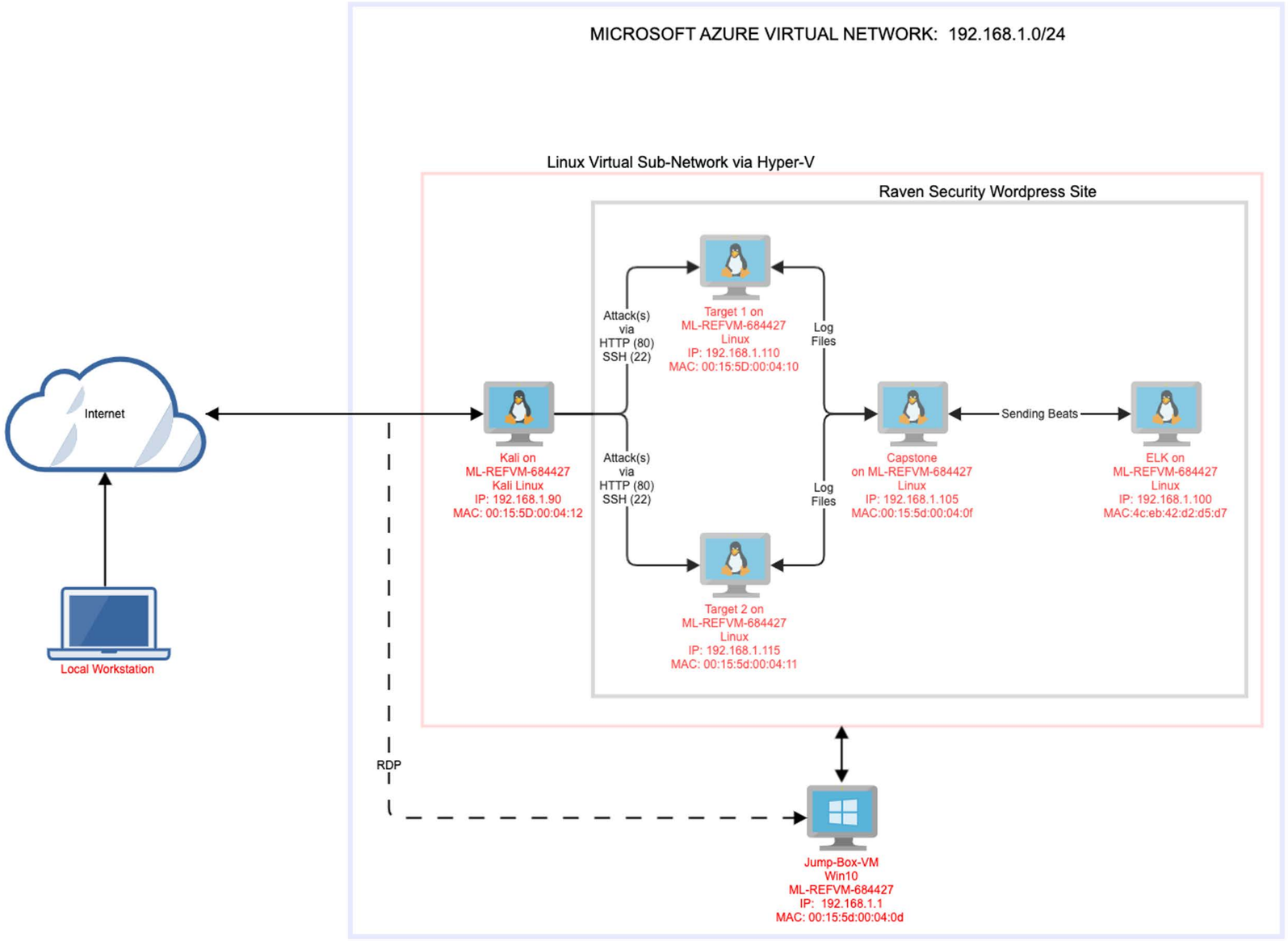
Attack, Defense & Analysis of the Raven Network

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: 192.168.1.110
OS: Linux 3.2 – 4.9
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux 3.2 – 4.9
Hostname: Target 2

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
User disclosure from 2 separate locations	Using either enum4linux or wpscan an attacker can gain access to usernames, enum4linux giving a more complete look	Gives the attacker a list of users to attempt to brute force the password.
Weak passwords	3 accounts have either a username for a password, or the reverse spelling of the username as the password	¾ user accounts are using default passwords or passwords that are otherwise easy to guess. Two of these accounts provide passwordless sudo use (root+vagrant) and the last was a no brainer to guess. The strongest password on the machine was cracked in under 5 minutes.
Port 22 Open	When port 22 is open it allows attackers to ssh and use brute force attacks on systems	Attacker can craft an attack method that exploits having ssh open such as a brute force
CWE-307: Brute Force attacks	Improper Restriction of Excessive Authentication Attempts	Gives attacker higher chance of success for brute force attacks

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Root password of the database in the wordpress configuration file	Database root password was stored in an application configuration file	This has a high impact because the threat can gain access to machine, the password will be easily available and they can quickly gain access to the database.
Privilege escalation by sudo python (CVE-2006-0151)	Allows local users to gain privileges by using a python script	provides root escalation this is very dangerous and impactful because it provides root to the threat actor

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
unprotected access to sensitive data	The vendor page is open to the public and contains vulnerability information and confirmation the version in use is vulnerable	The attacker was able to retrieve version information, vulnerability info (which included the current version and a link to a near ready to use exploit)
phpmailer **CVE-2016-10033**	Not only was the version in use vulnerable, the vulnerability was included in the documentation.	The mailSend function in the isMail transport in PHPMailer before 5.2.18 might allow remote attackers to pass extra parameters to the mail command and consequently execute arbitrary code via a \" (backslash double quote) in a crafted Sender property
Wp-config not locked down	Because the config file isn't locked down the attacker gained access to the mysql password	This has a high impact because the threat can gain access to machine, the password will be easily available and they can quicklygain access to the database**.**
Mysql UDF dynamic library Privilege escalation	A well documented exploit is available ready to compile that allows an unprivileged user to gain root	This allowed the attacker to jump from the www-data user to root

Critical Vulnerability Prevalent Server-Side Attacks / Target 1 & 2

- Attacks include but are not limited to:
  - **Website Defacement:** An Attack against a website that alters the appearance and information contained on a website
  - **HTTP response splitting** (CRLF injection attack): Server does not properly sanitize input values, such as character returns (CRs) and line feeds (LFs), allowing for attacks such as cross-site scripting.
  - **Web cache poisoning**: Involves replacing legitimate cached webpages with malicious content
  - **Parameter or URL tampering**: Manipulates parameters in a URL passed to a web server
  - **Path or directory traversal** (dot-dot-slash attack): Used to navigate into files and directories by using dot-dot-slash (../) as if navigating through directories in a terminal, with the aim of reaching folders outside of the root directory

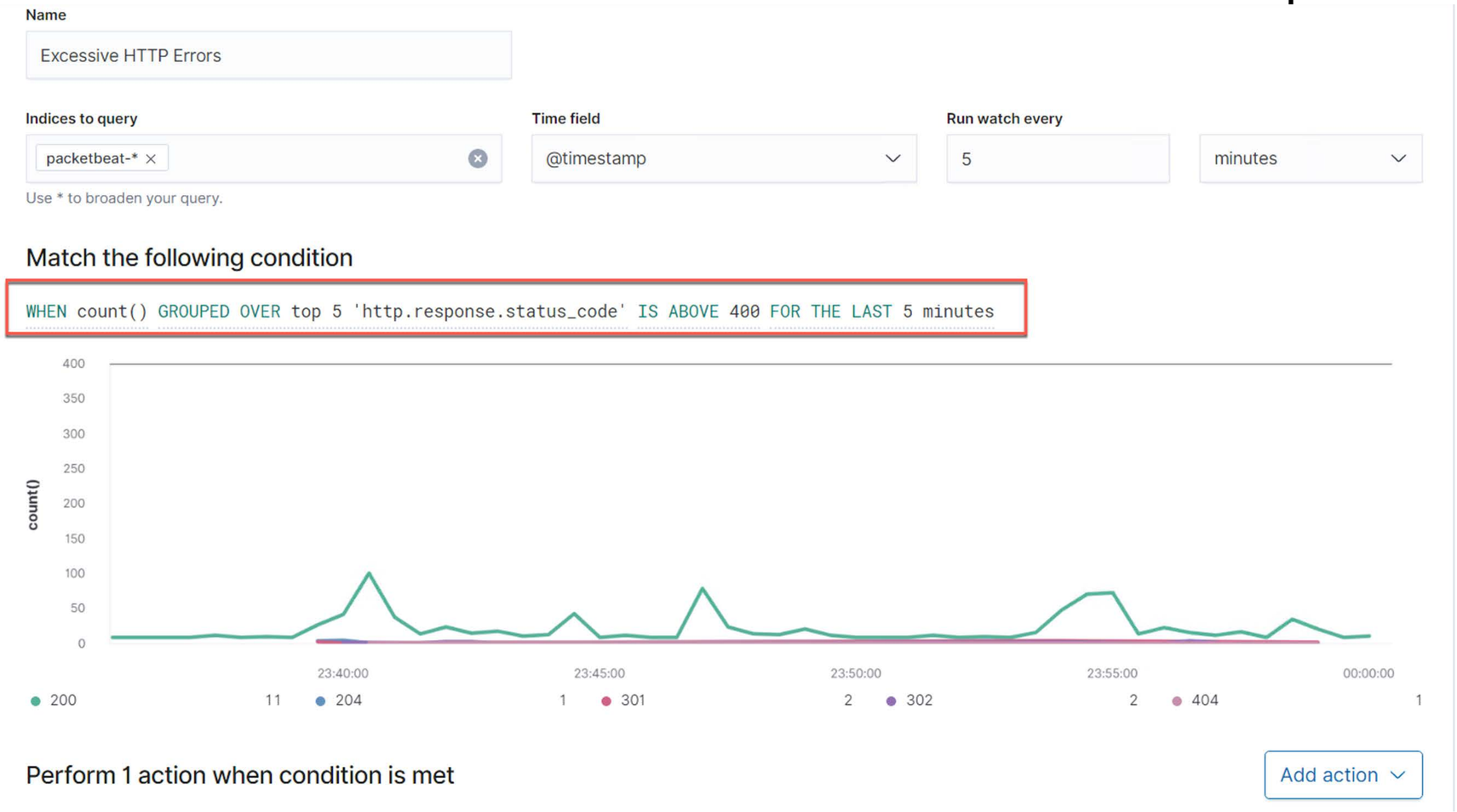
Critical Vulnerability No Defense In Depth / Target 1 & 2

- The network is not properly deployed and lacks technical security controls
  - These included but not limited to:
    - Firewalls
    - Intrusion Detection System (IDS)
    - Data loss prevention
    - Endpoint Security
    - Load Balancing
    - Proper Fail Over

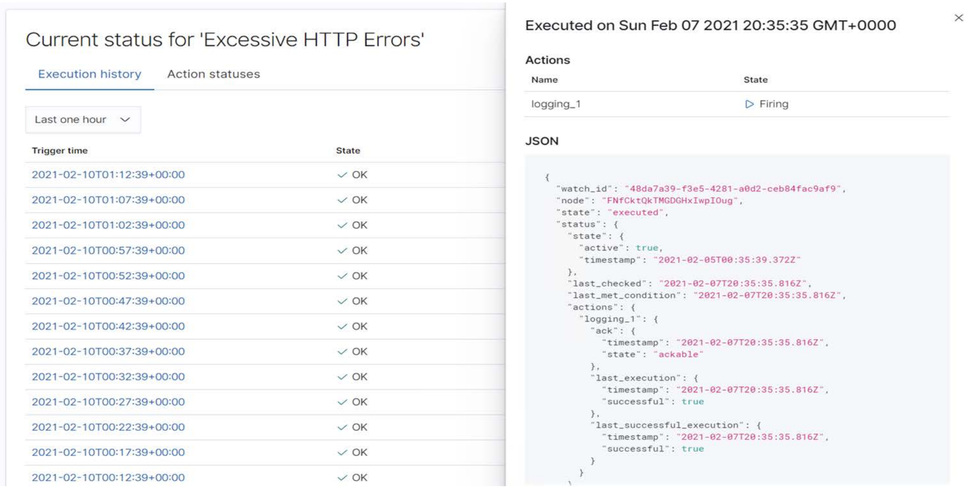
Alerts Implemented

Excessive HTTP Errors

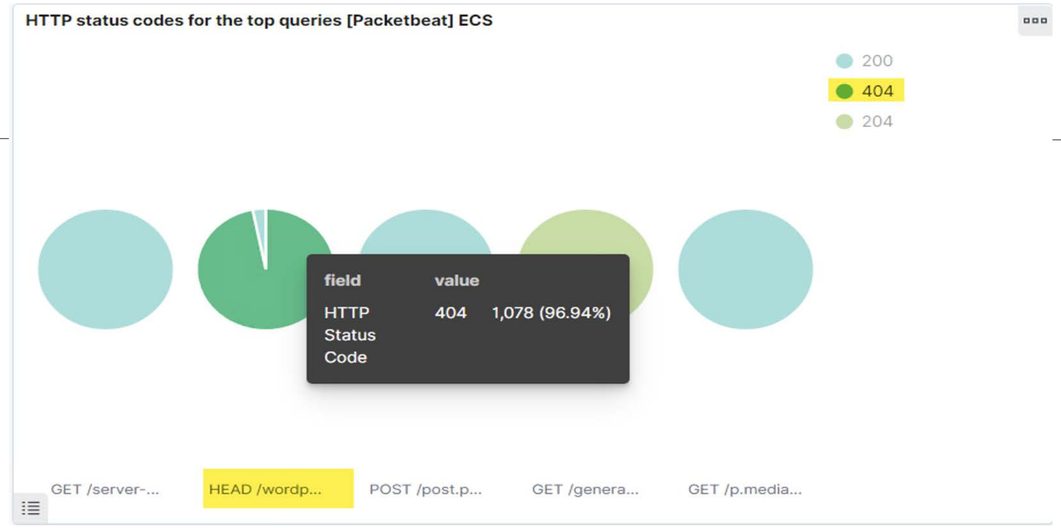
- This alert monitors the Packetbeat
- Threshold set if it fires more than 400 times in a 5 minute span

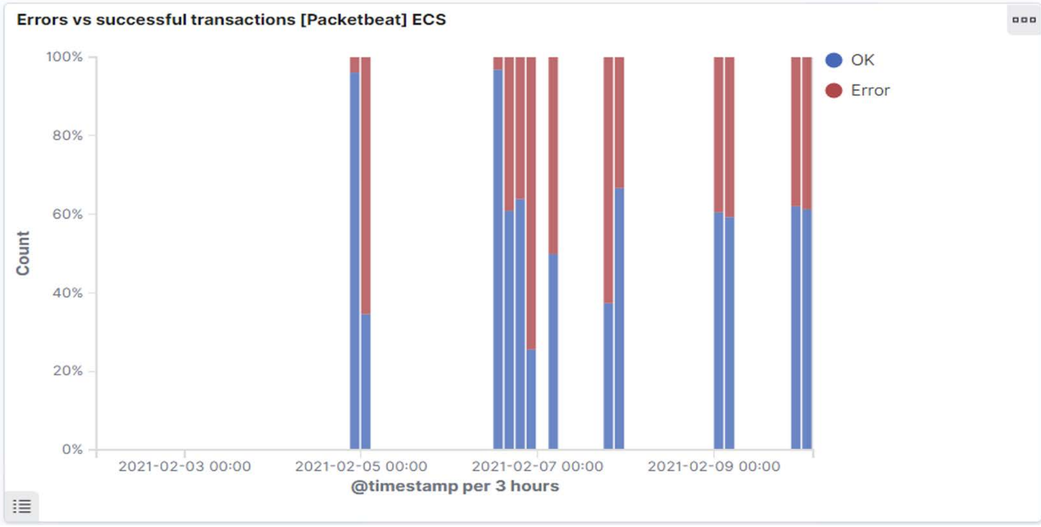


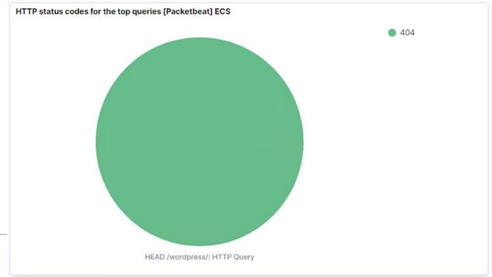
HTTP Error Monitoring

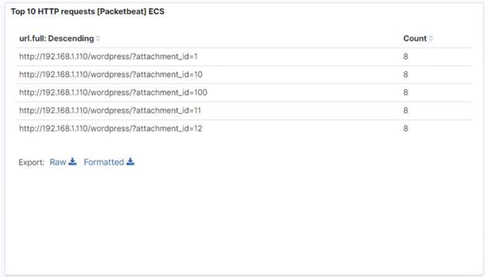


S.N.	Code and Description
1	**1xx: Informational** This means request received and continuing process.
2	**2xx: Success** This means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** This means further action must be taken in order to complete the request.
4	**4xx: Client Error** This means the request contains bad syntax or cannot be fulfilled
5	**5xx: Server Error** The server failed to fulfill an apparently valid request



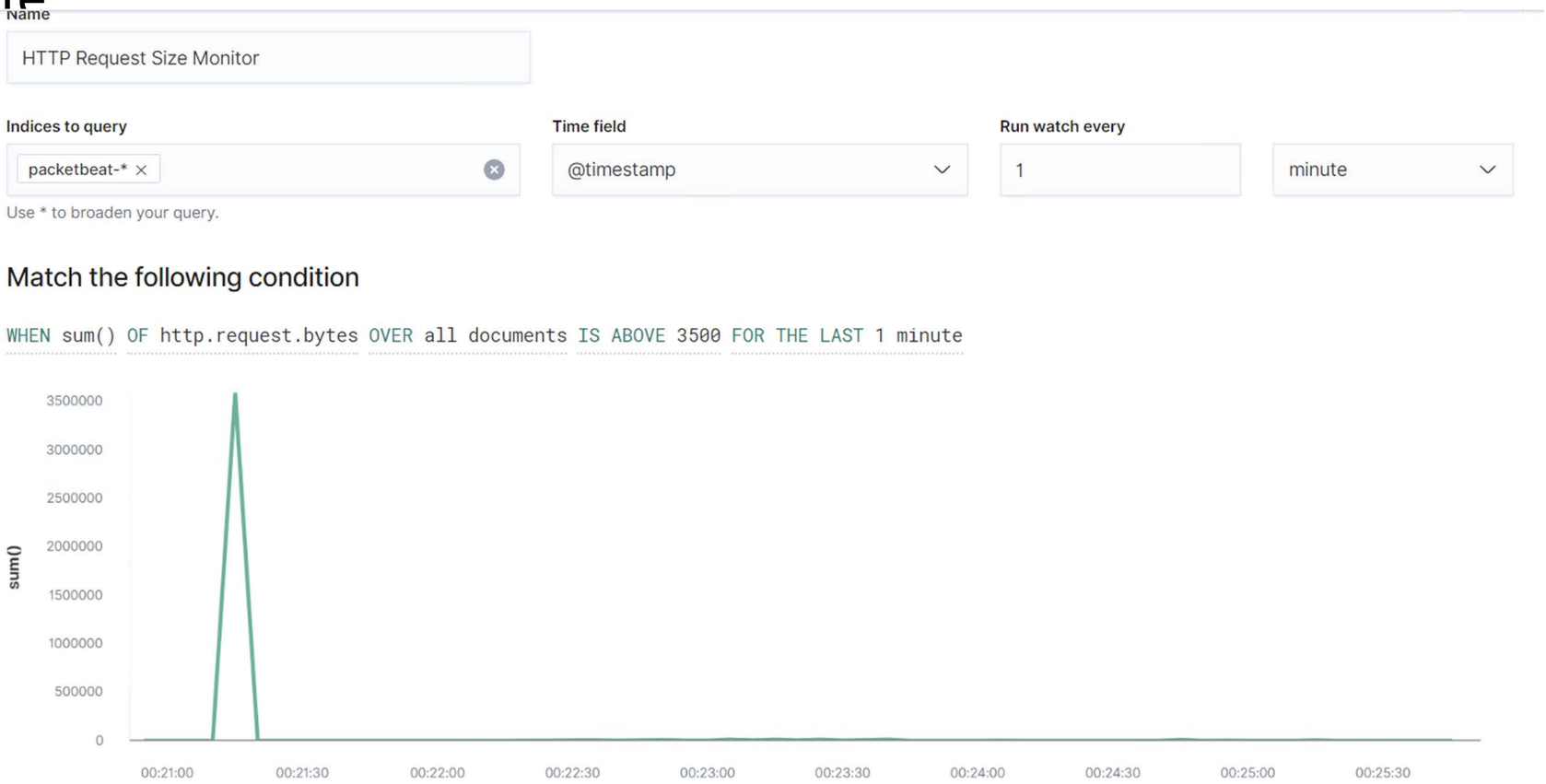




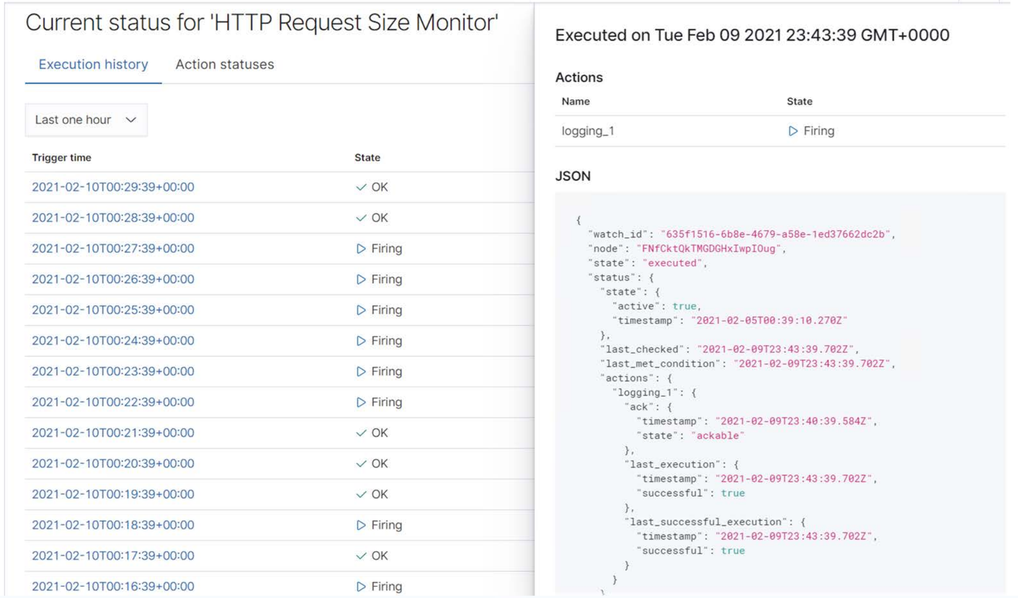


HTTP Request Size Monitor

- This alert monitors packetbeat
- This alert fires when the sum of bytes of any document is above 3500 in the last minute



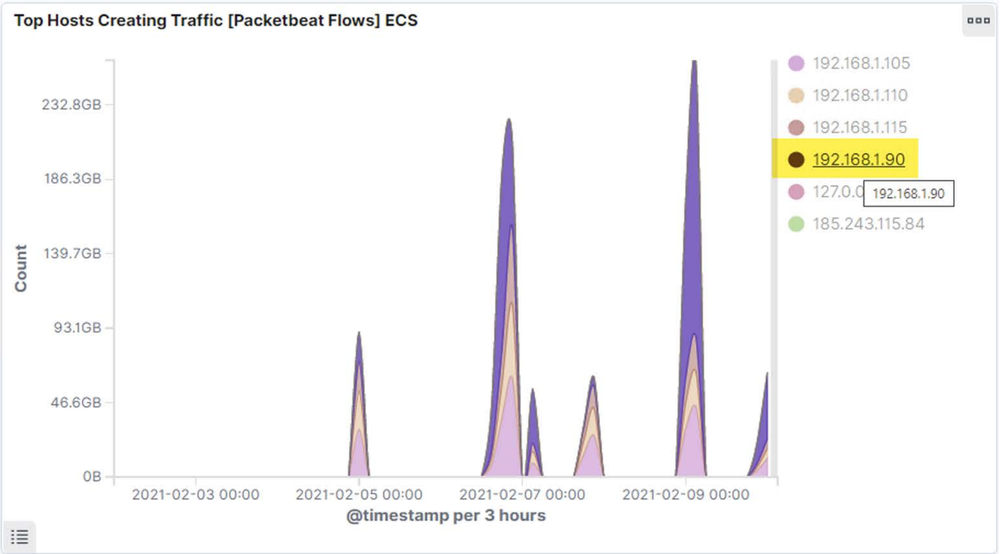
HTTP Size Monitoring



Network Traffic Between Hosts [Packetbeat Flows] ECS

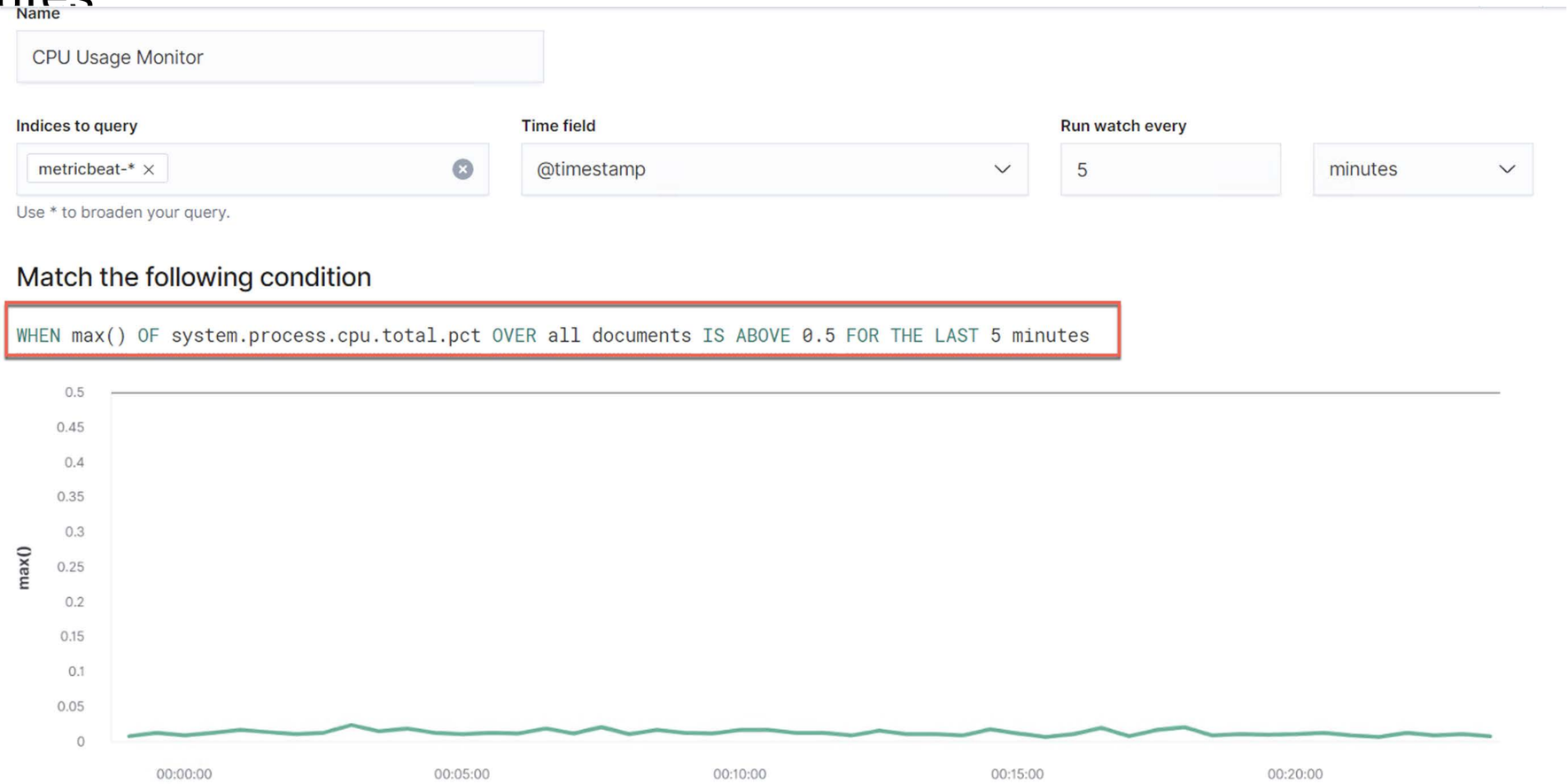
Source IP	Destination IP	Source Bytes	Destination Bytes	
192.168.1.90	192.168.1.100	591.1GB	13.4GB	ELK
192.168.1.90	192.168.1.110	212.6MB	211.1MB	Target 1
192.168.1.90	192.168.1.115	63MB	78.2MB	Target 2
192.168.1.90	192.168.1.1	2.7MB	101.8KB	
192.168.1.90	192.168.1.105	1.9MB	2MB	
192.168.1.105	192.168.1.100	276.6GB	22.4GB	
192.168.1.105	91.189.91.43	390.3KB	170.3MB	
192.168.1.105	91.189.88.152	378.2KB	87.9MB	
192.168.1.105	91.189.88.142	252KB	33.4MB	
192.168.1.105	91.189.92.20	102.8KB	4.4MB	

Export: Raw Formatted



Cpu Usage Monitor

- This alert monitors metricbeat
- This threshold fires when the max of any system process is above 0.5 in the last 5 minutes



Hardening

Hardening Against Excessive HTTP Errors on Target 1

- Thorough application backup of system and Database before attempting any fixes.

Set up special-security-package to the latest version.

- Special-security-package scans the system for viruses every day.
- To install it: apt-get upgrade

Hardening Against Apache HTTP on Target 1

- This Apache attack can be considered a slow HTTP Denial of Service attack utilizing the allowable connections. Due to there being so much data, the attackers opted to send it slowly. One way to mitigate this is by :
- Configure our ELK machine with docker that hold our Ansible playbooks to include but not limited to:
  - setting timeouts mod_reqtimeout
  - implementing controls assign different priorities to different HTTP Requests (mod_qos)
  - Implement a WAF to and set specific rules to block specific requests. mod_Security
  - apt-get install mod _reqtimeout mod_qos, mod_security,

Hardening Against Cpu Usage on Target 1

- Update the operating system
- Check for firmware updates.
- Update the browser.
- Update other software
- Set the system to consistently scan for malware.
- Identify and End all the processes that are causing the Cpu to spike.

Identify the Ip that might be the root cause of the spike and blacklist it.

- Always have an antivirus active and running.
- Doing this make the system less vulnerable to attacks.

Hardening Against phpmailer on Target 2

- Turn on the UseCanonicalName directive. With UseCanonicalName On Apache httpd will use the hostname and port specified in the ServerName directive to construct the canonical name for the server, and the attacker won't be able to inject code anymore.
  - Avoid wildcards (*) on your ServerName configuration. Using wildcards could be leveraged by an attacker to perform header injection attacks on websites and should be avoided whenever possible.
- It is important to remark that depending on the environment, these configurations could impact the normal functioning of the website and should be tested on a non-production instance before hand.

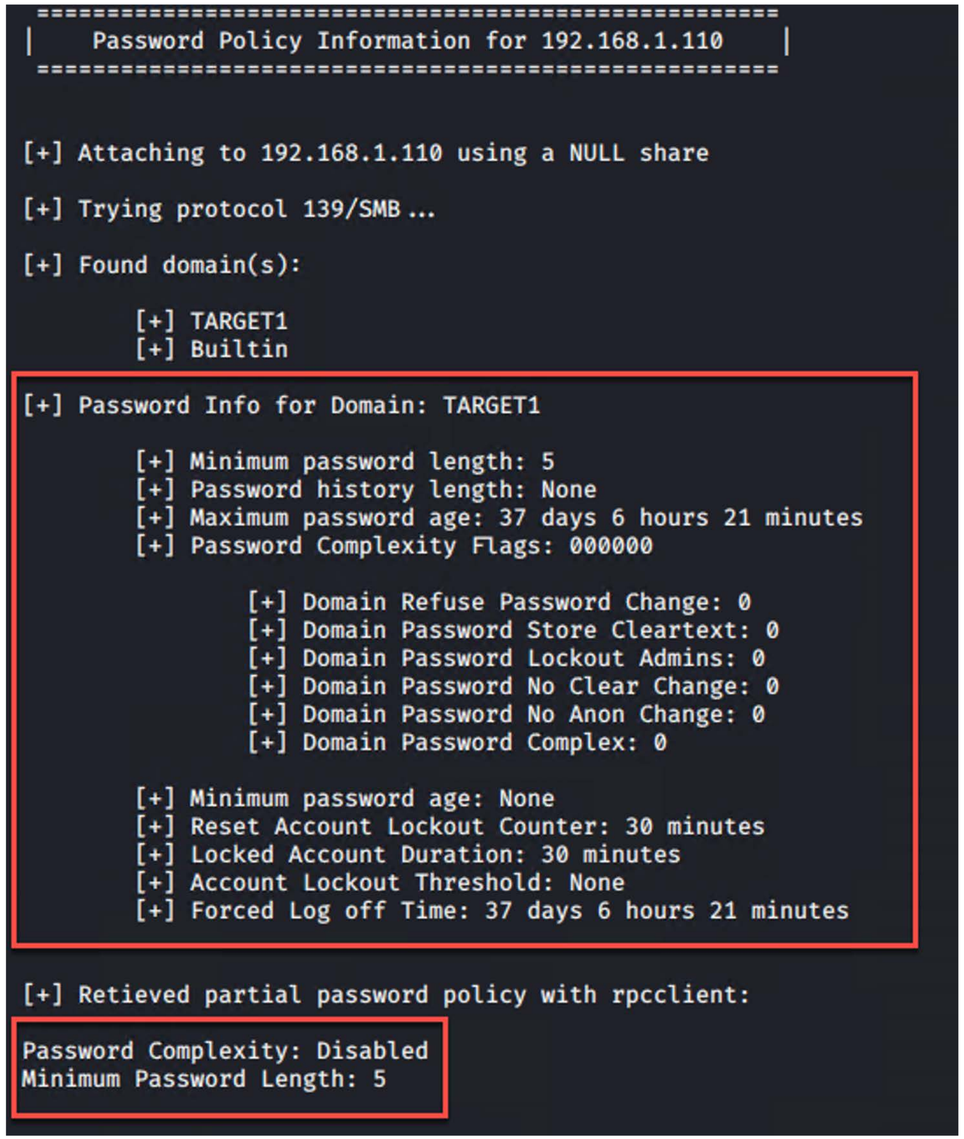
Hardening Against Prevalent Server-Side Attacks / Target 1 & 2

- Mitigation Include but not limited to:
  - **Input sanitation:** the process by which the web application (server side) actively modifies user input to an acceptable format or blocks the user's request
  - **Input validation:** the process that checks if a user's input is in an acceptable format
    - Restricted numeric range
    - Restricted character sequence and patterns
    - Restricted use of Null
    - Restricted parameter use of unused form fields
  - **Content Security Policy (CSP)**: Detects and mitigates specific types of attacks including XSS and various other injection attacks (Not all browsers support CSP)
  - Ensure that all files and folders on the local server have **proper access controls** in place
  - Instituting a **quality assurance** program that verifies that web developers are not storing private sensitive information in the web root directory ((i.e. Lock down intellectual property, like DevOps code)

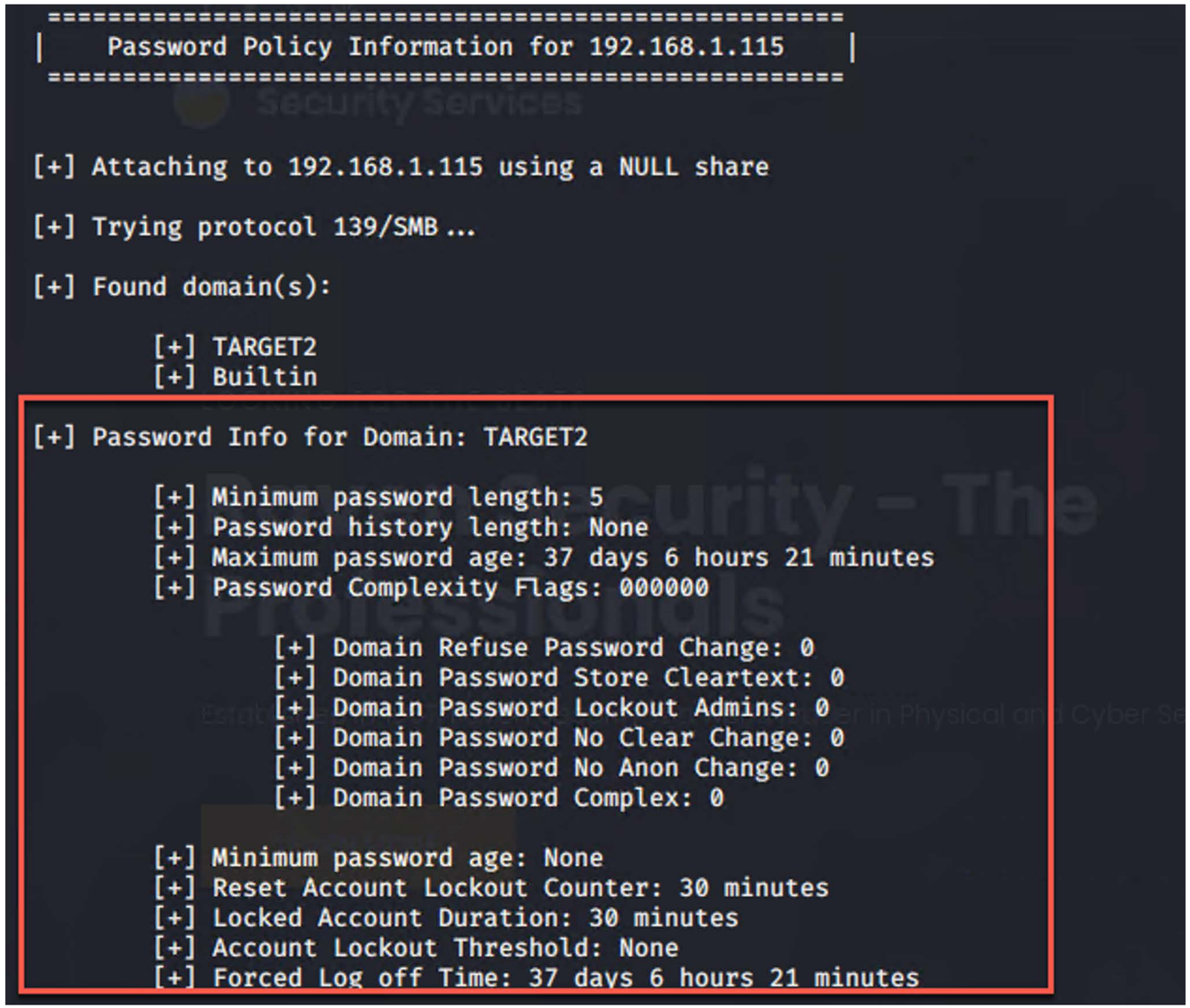
Hardening Against Unauthorized SSH Access on Target 1 & 2

- We will harden SSH access by editing the sshd configuration file:
  - **Change the default TCP** port where SSH daemon is listening. An example is changing it from port 22 to something much higher like 24596. We want to ensure we do not use a port number that is easy to guess, such as 222, 2222 or 22222
  - **Use SSH key pairs for authentication** to enable passwordless SSH login. It is safer and also allows logging in without the need to use a password which is faster and more convenient.
  - **Disable password-based logins** on the server. This eliminates the possibility of passwords getting cracked and used for login.
  - **Disable root access to the server and utilize a service account** with the su - command to switch to a root user once properly authenticated and logged in.
  - **We can restrict access to certain IP addresses or hostnames** via TCP wrappers and editing the /etc/hosts.allow and etc/hosts.deny files.

Hardening Against Weak Passwords on Target 1



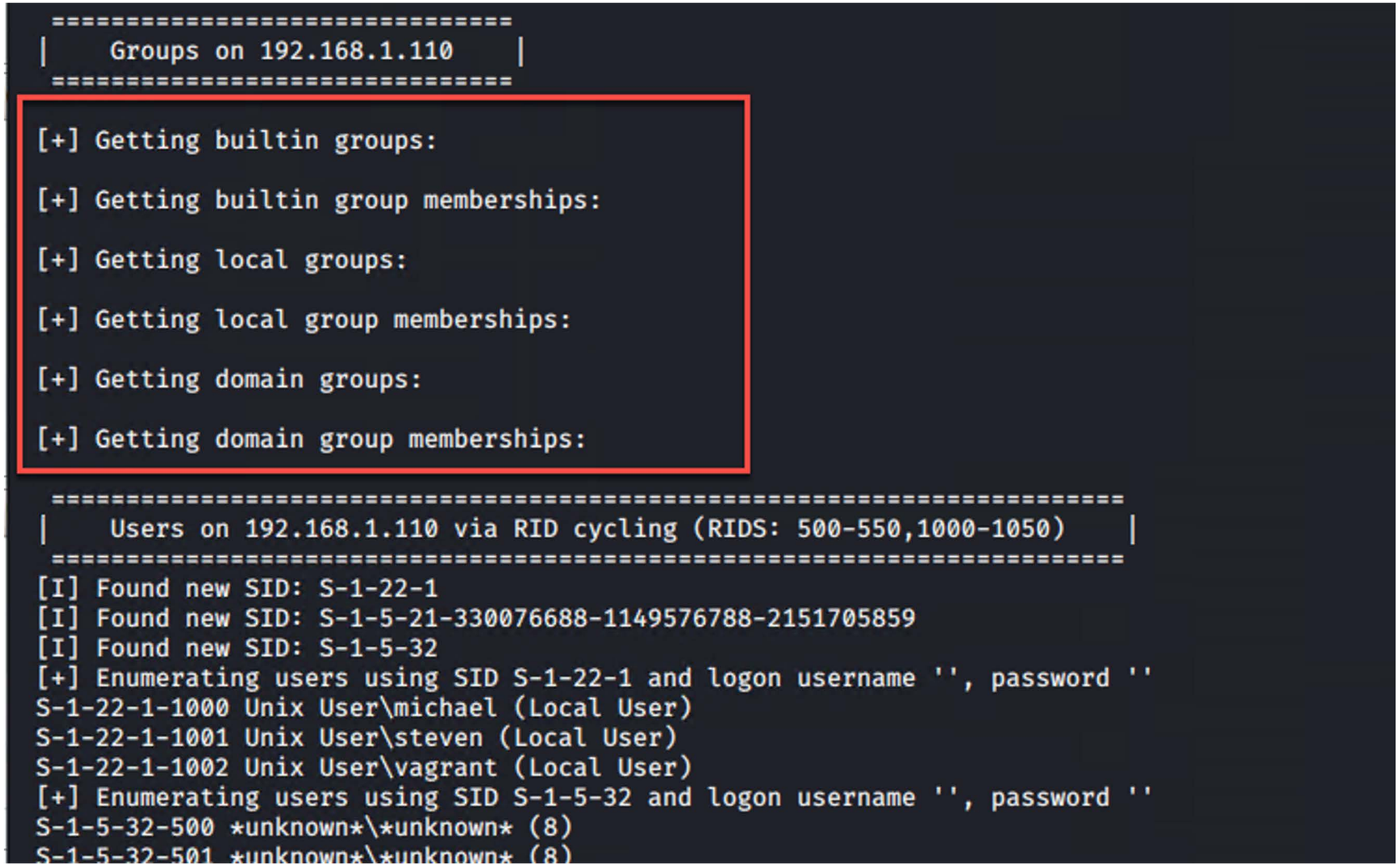
Hardening Against Weak Passwords on Target 2



Hardening Against Weak Passwords on Target 1 & 2

- Make the password expire after a set time, so users will have to reset it
- We can also set the length, number and character requirements for the password:
  - Tools: (1) the command “chage” (to change when a password expires (2) the command “nano” (to edit the settings inside the /etc/security/pwquality.conf file)
  - Recommendation(s)
    - Set the maximum number of days between password changes to 90
    - Require passwords to have a minimum of 16 x characters
    - Require complex passwords: 2 x numbers - 2 x upper case - lower case - 2 x special char.
    - Require new passwords to differ from the last 5 previous password
    - Limit incorrect password by setting account lockout threshold: 10 in a 10 minute time frame
    - Set forced log off time to 24 hours
    - Force immediate password expiration for all users on the server to institute new policy

Hardening Against Escalating Privileges on Target 1 & 2



Hardening Against Escalating Privileges on Target 1 & 2

- We will implement a strategy around practicing the principle of least privilege
- We will inspect all users on the web server
  - check every user’s UID and GID
  - remove all non-admin users from having access to the server
  - We will create an admin group and ensure only proper admin users are assigned to this group
- We will ensure that all admin users would have to employ the following in order to connect remotely to the network:
  - require all servers to only be accessed via 256 bit cryptographic RSA VPN
  - **SSH key pairs for authentication**
  - **Disable root access to the server and utilize a service account**

Hardening Against Unfiltered Ports on Target 1 & 2

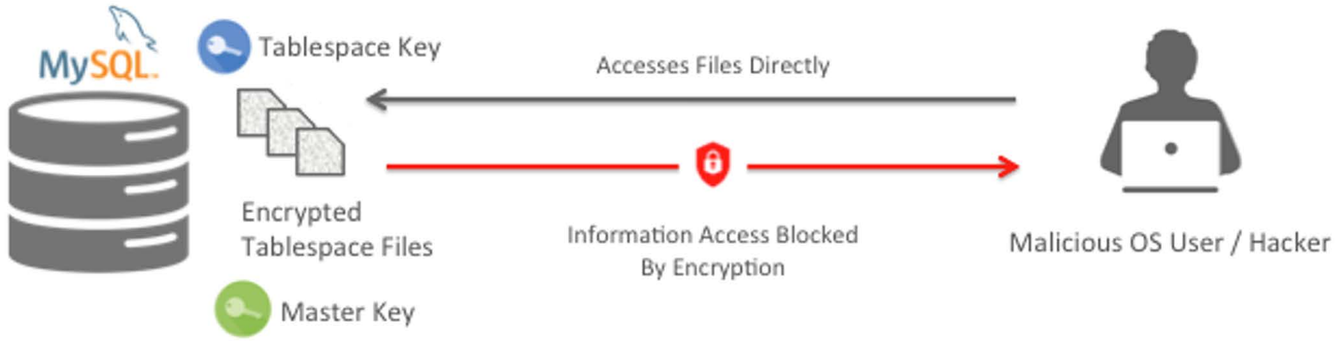
```
root@Kali:~# nmap -sA 192.168.1.110 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-12 16:48 PST
Nmap scan report for raven.local (192.168.1.110)
Host is up (0.0010s latency).
All 1000 scanned ports on raven.local (192.168.1.110) are unfiltered
MAC Address: 00:15:5D:00:04:10 (Microsoft)

Nmap scan report for 192.168.1.115
Host is up (0.0010s latency).
All 1000 scanned ports on 192.168.1.115 are unfiltered
MAC Address: 00:15:5D:00:04:11 (Microsoft)

Nmap done: 2 IP addresses (2 hosts up) scanned in 0.38 seconds
root@Kali:~#
```

Hardening Against Unfiltered Ports on Target 1 & 2

- We will implement a cloud based web application firewall which is a technical security control that would be placed in front of the target 1 and 2 apache web servers to block malicious internet traffic from entering the network. Thus providing the ability to:
  - Low cost, subscription based, turnkey product that requires minimal resources.
  - Protects applications across a variety of hosting locations that protect against application layer attacks.
  - Cloud service providers use the most current threat intelligence to help identify and block new application security threats.
  - We will implement a “Hybrid” model to analyze and filter traffic based on combination of both allow (whitelisting) and deny (blacklisting) lists.
  - Consideration for WAF such as Wafw00f

Hardening Unsecure MySQL Database on Target 1 & 2

- MySQL Enterprise Transparent Data Encryption (TDE) protects your critical data by enabling data-at-rest encryption in the database. It protects the privacy of your information, prevents data breaches and helps meet regulatory requirements including:
  - Payment Card Industry Data Security Standard (PCI DSS)
  - Health Insurance Portability and Accountability Act (HIPAA)
  - General Data Protection Regulation (GDPR)
  - California Consumer Protection Act (CCPA)
  - And more



MySQL Enterprise Transparent Data Encryption (TDE)

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

Explain which vulnerability each task in the playbook patches.

We will implement provisioners to run the tasks. The tasks performed will:

- Update browser
- Upgrade security package to newer version
- Block specific request
- Automatically conduct service updates
- Implement quality of service controls (i.e.)
  - prioritizing what information will be collected
  - set timeout thresholds

Thank You! Up Next (Network Team)

Cybersecurity