Final Engagement

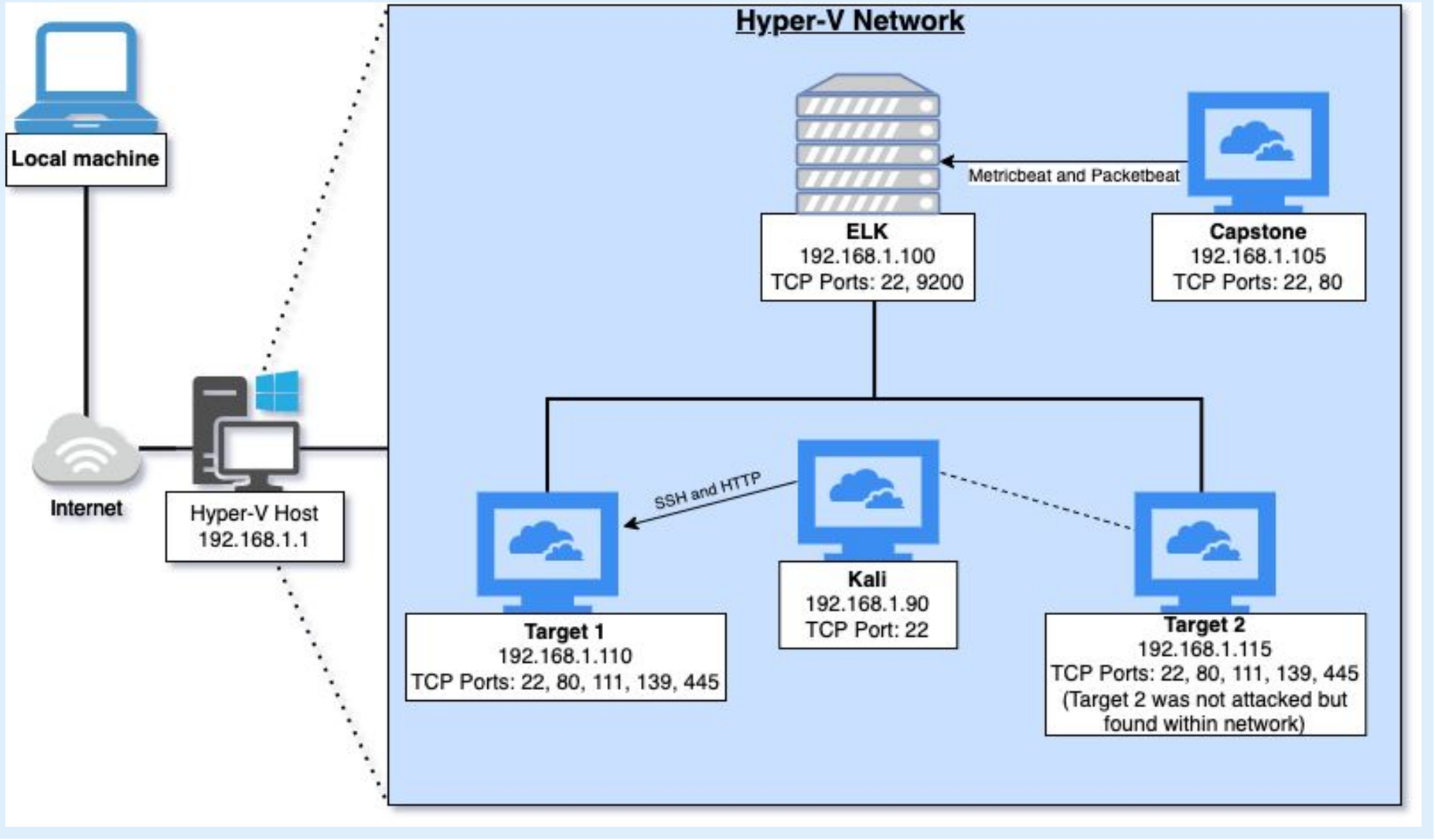
Attack, Defense & Analysis of a Vulnerable Network

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: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak password	After finding usernames into the ssh server through the wordpress scan we were able to GUESS the password into the server.	Without even having to bruteforce the password after 2-3 attempts the password for username michael was found. It is michael
Plain text information	Important information was found throughout the server whether in a file or html source page.	Found usernames and passwords inside the database/web server
Database Misconfiguration	Database has to be secure as far as who can run/open the database and how the database is setup.	Michael was able to open the database aside from only root which is fine if it is needed but in this case it caused a security breach.
Root Permissions	Permissions should be evaluated and given on an as needed case.	Steven was given root permissions on python and we were able to use those permissions to gain root access

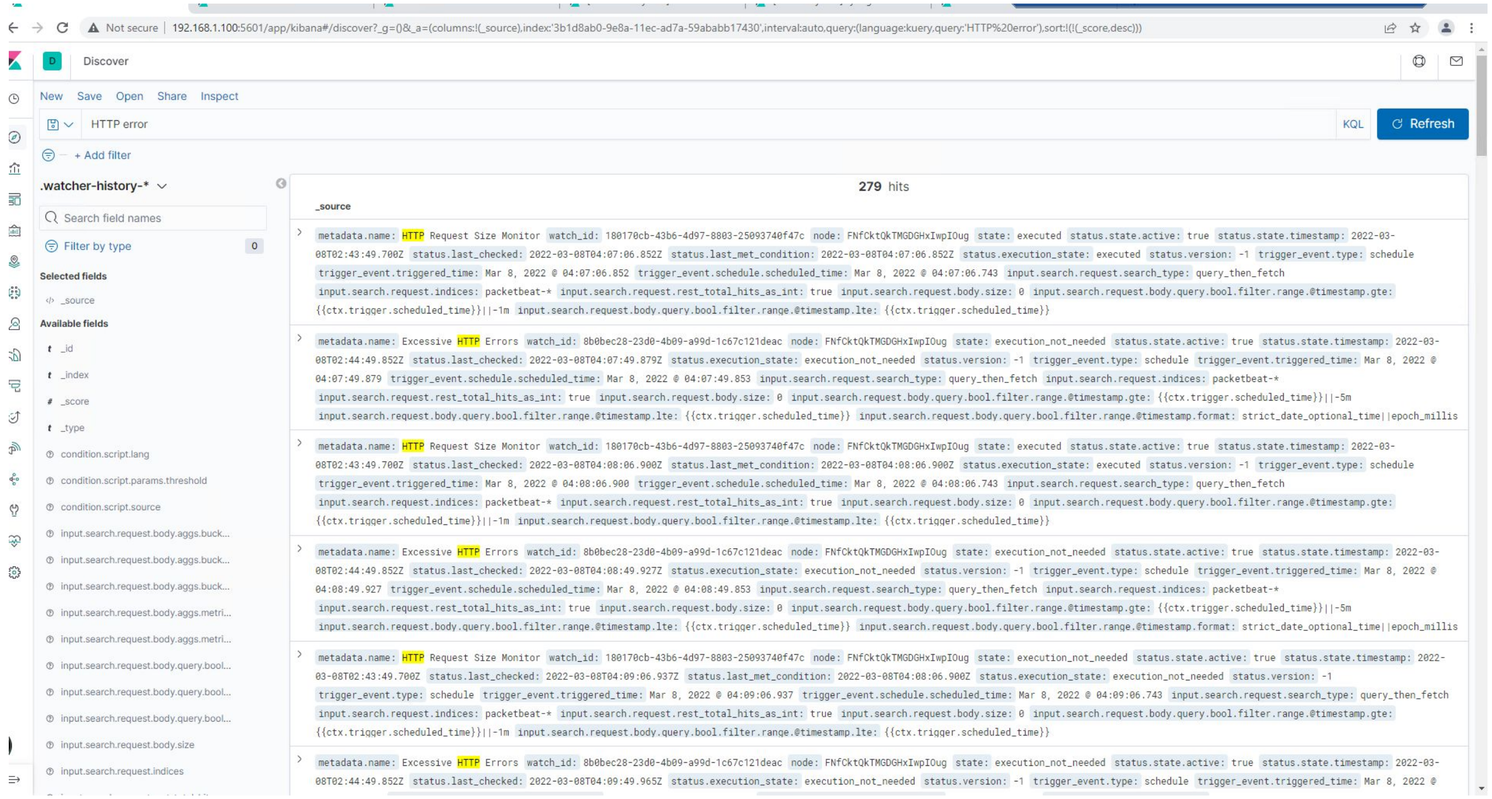
Alerts Implemented

CPU Usage Monitor

- It monitors total CPU usage in percentages.
- It alerts at any percentages that are higher than 50% of total cpu usage in the past 5 minutes.

Excessive HTTP Errors

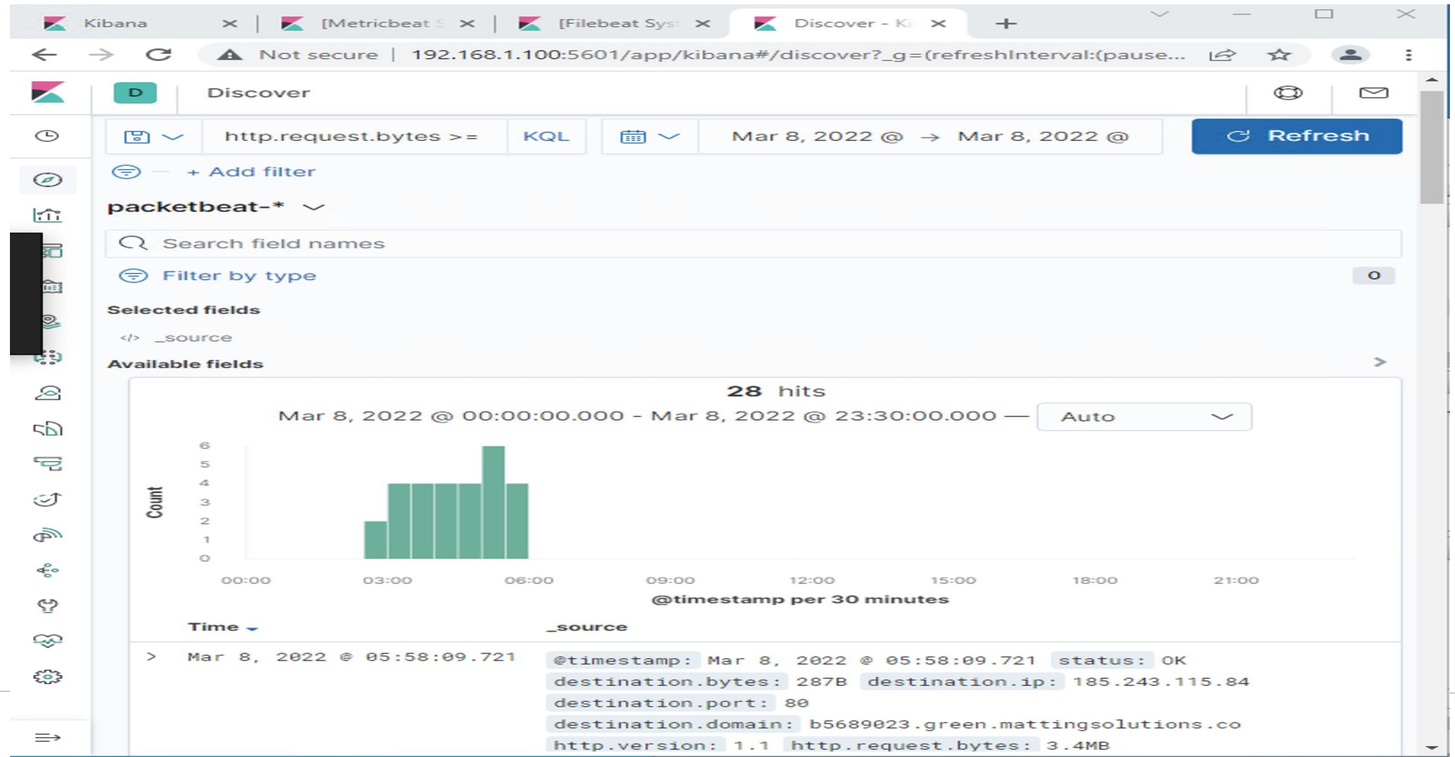
- It monitors HTTP Errors that are over 400 because those types of errors are client and server side errors.
- The alert triggers when a group of the top 5 http response status codes are all 400 or higher error codes in the past 5 minutes.



HTTP Request Size Monitor

Summarize the following:

- **metric**: WHEN sum ( ) of http.request.bytes OVER all documents
- **threshold: IS ABOVE 3500 FOR THE LAST 1 minute**
- Vulnerability Mitigation: What this is, is a measurement of events with hightraffic in a short period of time which is an indicator of an attack.



Hardening

Vulnerability 1: Weak Password

- Adopt best practices in regards to password creations and storage
- We recommend multi factor authentication
- In addition to a password we recommend to use a token or HOTP verification
- Lock out policy

Vulnerability 2: Plain Text password Exposure

- Improper exposure of password in public files
- Re-configure the server to use only HTTPS certificate
- Set up the system to redirect any HTTP request to HTTPS site
- All passwords have to be encrypted in storage

Vulnerability 3: Database Misconfiguration

- Encrypt `wp-config` files that are stored locally
- Establish RBAC for sensitive documents
- Limit access to the MySQL database
  - Remove Michael’s access
  - Ensure only specific users get admin/root privledges

Vulnerability 4: Root Permissions

- Evaluate root/admin privileges of all users
- Implement RBAC to allow only necessary root access
- Whitelist MAC/IP’s from internal network to prevent privilege escalation from outside devices

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

- 1st Use: Update and manage our configurations which should help mitigate future risks
- 2nd Use: Patch and update the system automatically by creating a playbook
- 3rd Use: Create a cron job to keep certain tools up to date and run a system wide update
- Main Reason for Use: To run then entire system