Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

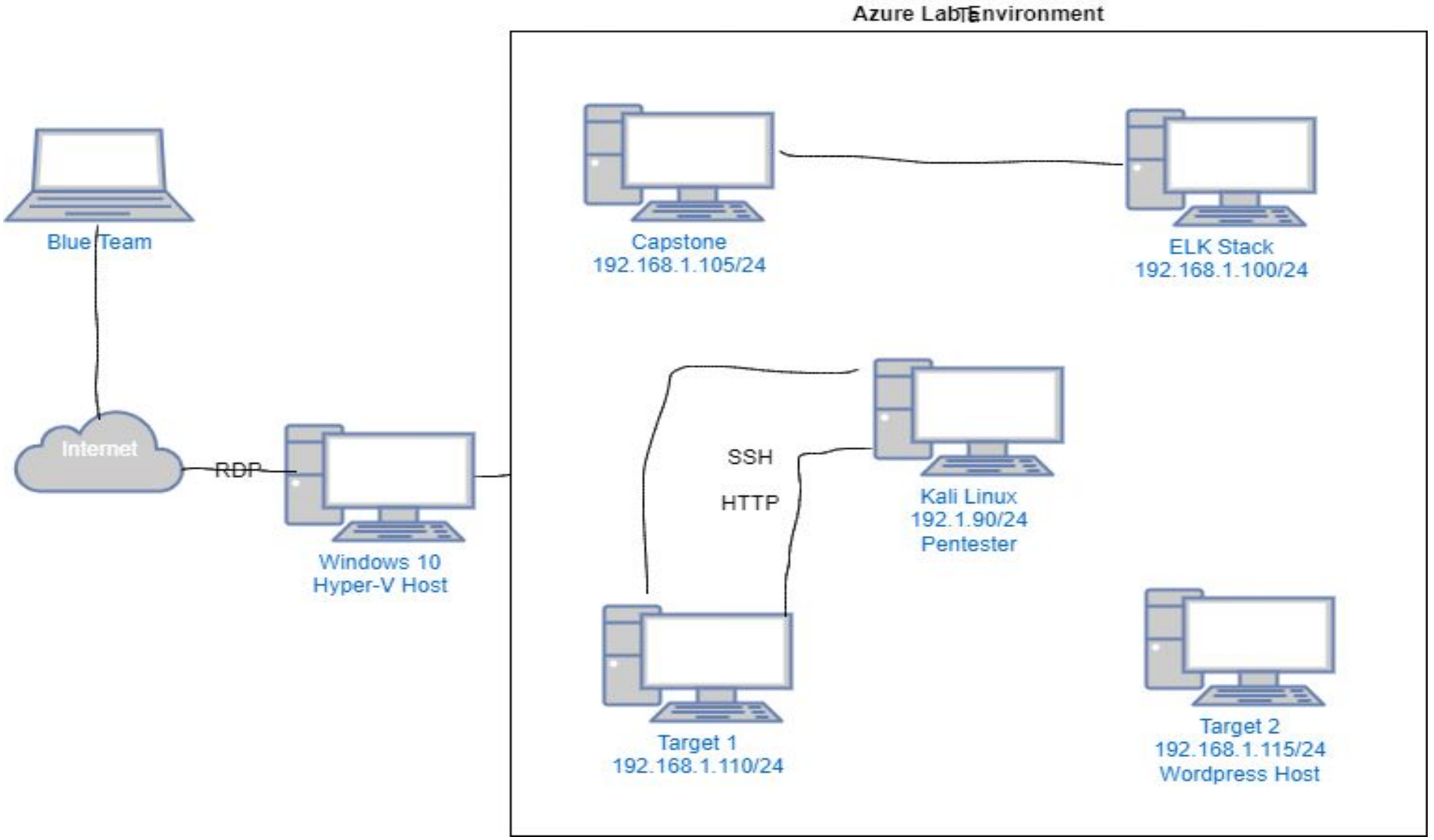
Auboni, Austin, Chris, Jeffrey, Kurt, Samson

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: 192.168.1.90
OS: Debian Kali 5.4.0
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux 8
Hostname: Target 1

Critical Vulnerabilities: Target 1

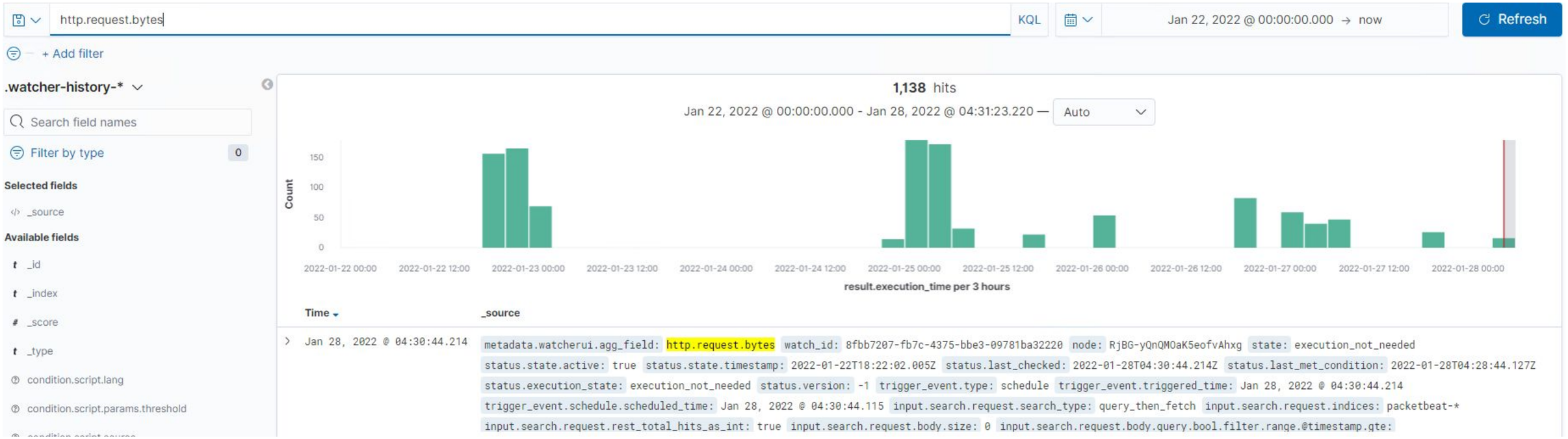
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Author ID Brute Force	Passwords can be brute forced using the Author ID, using non complex usernames and passwords	User accounts Steven and Michael's passwords can be brute forced due to the use of non complex passwords.
Open SSH	Port 22/tcp is open on 192.168.1.110	Attacker can gain ssh access to server from any source IP
User access to privilege escalation	User Steven's access to sudo Python ws used to escalate from "Steven" to "root"	# !/usr/bin/python allows python script to run to create a pseudo terminal to run commands as root
CVE-2012-6707 Weak MD5-based password hashing	Weak MD5-based hashing for passwords are being used for user accounts on the wordpress site.	Attackers can easily decrypt and determine cleartext values, and discover user passwords on the wordpress site.
Simple username and password	The username "michael" was discovered with wpscan, and password was found to also be "michael".	Enumerating users with wpscan revealed simple usernames and the password is easily guessed, granting access to a remote attacker through SSH.
Network Mapping and User Enumeration (WordPress site)	Nmap was used to discover open ports.	Able to discover open ports and plan their attacks accordingly.

Alerts Implemented

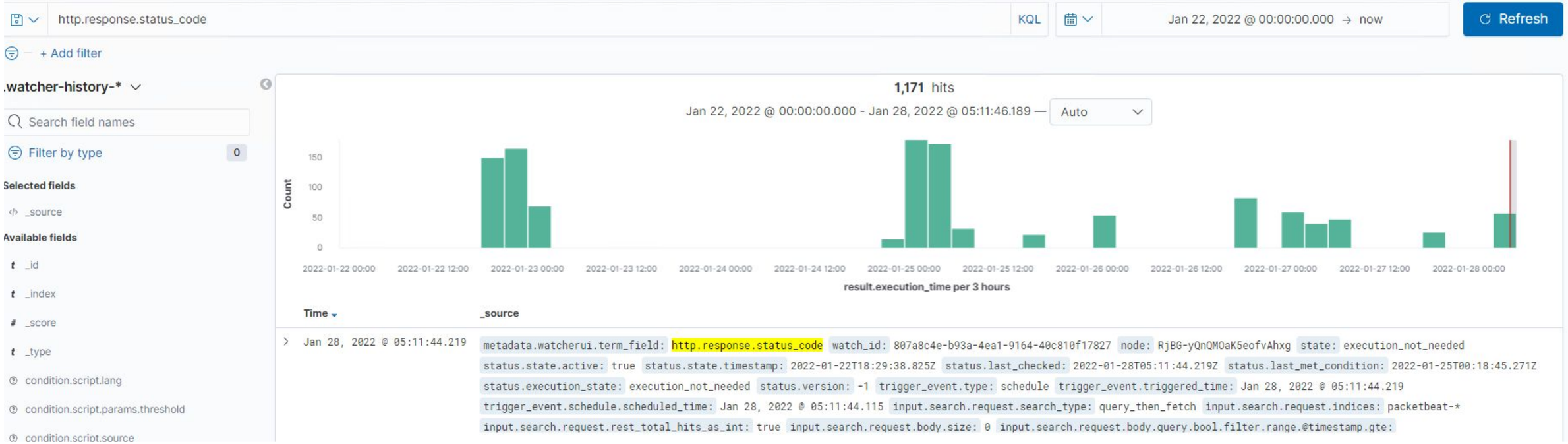
Alert 1: HTTP Request Size Monitor

- Which **metric** does this alert monitor?
  - The metric that this alert monitors is ‘WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 60 seconds’
- What is the **threshold** it fires at?
  - ABOVE 3500



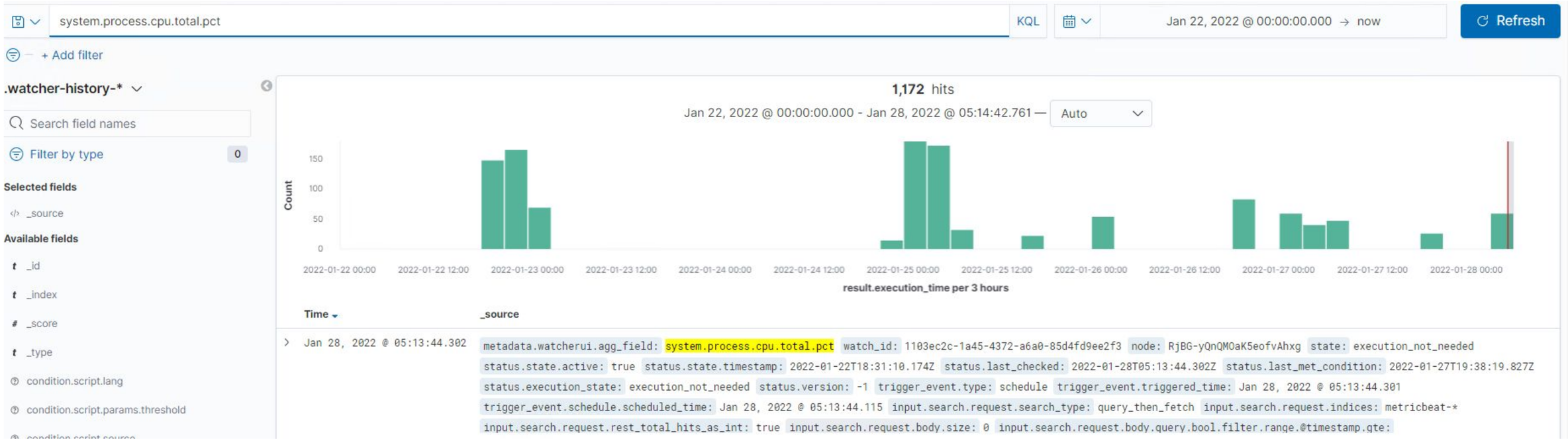
Alert 2: Excessive HTTP Errors

- Which **metric** does this alert monitor?
  - The metric this alert monitors is ‘WHEN count() GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 minutes
- What is the **threshold** it fires at?
  - ABOVE 400



Alert 3: CPU Usage Monitor

- Which **metric** does this alert monitor?
  - This metric that this alert monitors is ‘WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes’
- What is the **threshold** it fires at?
  - ABOVE 0.5% of CPU usage



Hardening

Hardening Against Author ID Brute Force on Target 1

- **Security Recommendations**
  - **Patch:**
    - **Install Firewall plugin**
    - **Update wordpress**
    - **Use stronger passwords**
    - **Block/disable wpscans**
      - Adding code to the .htaccess file (hypertext file) in the WordPress root directory.
  - Why this works:
    - It reduces the probability of a bad actor gaining access to vulnerable files and user account information to exploit the system.
    - Specifically blocking / disabling wpscans by adding code to the .htaccess file (hypertext file), which holds configuration data for the WP site, in the WordPress root directory. This will block scanning of crucial data in the file, when using WPscan

Hardening Against Open SSH Force on Target 1

- **Security Recommendations**
  - **Patch:**
    - **Use Stronger Passwords**
    - **Update SSH**
    - **Disable root Login**
    - **Block port 22 from scans**
    - **Setup automatic email alerts for unusual attempts to login**
  - **Why this works:** Employing these actions will offer stong mitigation tactics to reduce elevation of root privileges and exploitation of sensitive files.

Hardening Against Enumeration and Brute Force Attacks

- **Security Recommendations**
  - **Patch: WordPress Hardening**
    - Lock out accounts after a predetermined number of failed attempts and implement multi-factor authentication (MFA)
    - Disable the WordPress REST API and XML-RPC if it’s not needed and configure the web server to block request to /? author=
    - Prohibits exposure of /wp-admin and /wp-login.php
  - **Why it works:**
    - Accounts lock outs will be mitigated credential stuffing and multi-factor authentication will mitigate password spraying attacks
    - WPScan use Rest API to enumerate users, and XML-RPC uses HTTP as it s transport mechanism for data
    - WordPress permalinks can be set to include an author and prevent exposure of WordPress login portals will help mitigate brute force attacks

Hardening Against Weak MD5-based password hashing (CVE-2012-6707) on Target 1

Security Recommendations:

- **Patch: Upgrade Wordpress to latest version**
  - Latest revision of Wordpress is 5.9
    - Versions 3.7 and up are easier to update from the wordpress site with just a click of a button.
- **Require more complex username and password requirements for logins**
- **Use PHP hashing on top of MD5**
- **Harden the wp-config.php file**
  - Change the permissions on the wp-config.php to ‘read only’ for root users.
    - Command:
      $ *sudo chmod 400 /path/to/wp-config.php*

- **Why It Works:**
  - Updating wordpress to the latest version uses better versions of PHP hashing, PHP hashing uses bcrypt, an adaptive function that salts passwords to protect against more advanced cracking methods like using a rainbow table
  - Hardening the ‘wp-config.php’ file by changing permissions to ‘400’ will only allow root users to access the file.

Hardening Against Privilege Escalation on Target 1

- **Security Recommendations**

  Administrator permissions should be limited to essential personnel with privilege given to individuals temporarily for specific assignments

  - Be aware of hidden administrators
    - The local administrator account on workstations and servers
    - Service accounts with weak or unchanging passwords
  - User privilege specification
    - while several users can have sudo access, limiting who can have root access prevents unnecessary escalation

Why this works:

- limiting access to permissions allows for accountability; in the case of compromise, the attacker will not be able to escalate

How to install it:

- auditd to find any compromised user accounts
- ensure proper configuration to the sudoers files.

Implementing Patches

Implementing Patches with Ansible

```
michael@target1:~$ mysql -V
mysql  Ver 14.14 Distrib 5.5.60, for debian-linux-gnu (x86_64) using readline 6.3
```

Playbook Overview

```
GNU nano 4.8                                    Target1_Pa
---
- name: Target 1 VM Vulnerability Patching
  hosts: webservers
  become: true
  tasks:


  - name: backup html files
    archive:
      path:  /var/www/html
      dest: "/home/wordpress-bck-{{ansible_date_time.iso8601}}.tgz"
      format: gz
    become: true


  - name: get latest wordpress
    unarchive:
      src: https://wordpress.org/latest.zip
      dest: /tmp/
      remote_src: yes
    become: true

  - name: Wait for wordpress to download
    wait_for:
      path: /tmp/wordpress/index.php
      state: present

  - name: copy wordpress to website
    shell: /bin/cp -rf /tmp/wordpress/* /var/www/html/
    become: true

  - name: delete tmp wordpress
    file:
      path: /tmp/wordpress
      state: absent
    become: true
```

- **target1 playbooks used to update to latest version of wordpress to better mitigate against** *wpscan* **vulnerabilities such as user enumeration**
- Consists of backing up wordpress database, downloading latest version, and updating the new wp_version 5.9 with the backed up database

```
 * @global string $wp_version
$wp_version = '4.8.7';
```

```
root@target1:/var/www/html# grep wp_version wp-includes/version.php
 * @global string $wp_version
$wp_version = '5.9';
```

```
GNU nano 4.8                                    Target1_MySQL.yml
---
- name: Target 1 VM MySQL Version Update
  hosts: webservers
  become: true
  tasks:

  - name: stop MySQL
    service:
      name: mysql
      state: stopped

  - name: Download MySQL Repository
    command: curl -L -O https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb

  - name: Install MySQL Package
    command: dpkg -i mysql-apt-config_0.8.22-1_all.deb

  - name: Update Package Information from MySQL APT Repository
    shell: apt-get update

  - name: Upgrade MySQL server
    command: apt-get install mysql-server

  - name: restart MySQL
    service:
      name: mysql
      state: started
```

```
root@Kali:/etc/ansible# ansible-playbook -kK Target1_MySQL.yml
SSH password:
BECOME password[defaults to SSH password]:

PLAY [Target 1 VM MySQL Version Update] ****************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************
ok: [192.168.1.110]

TASK [stop MySQL] **************************************************************************************************************************************************
ok: [192.168.1.110]

TASK [Download MySQL Repository] ***********************************************************************************************************************************
[WARNING]: Consider using the get_url or uri module rather than running 'curl'.  If you need to use command because get_url or uri is
insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [192.168.1.110]

TASK [Install MySQL Package] ***************************************************************************************************************************************
^C [ERROR]: User interrupted execution
```

```
root@Kali:/etc/ansible# ansible-playbook -kK Target1_Patch.yml
SSH password:
BECOME password[defaults to SSH password]:

PLAY [Target 1 VM Vulnerability Patching] **************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************
[WARNING]: Platform linux on host 192.168.1.110 is using the discovered Python interpreter at /usr/bin/python, but future installation of
another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for
more information.
ok: [192.168.1.110]

TASK [backup html files] *******************************************************************************************************************************************
changed: [192.168.1.110]

TASK [get latest wordpress] ****************************************************************************************************************************************
changed: [192.168.1.110]

TASK [Wait for wordpress to download] ******************************************************************************************************************************
ok: [192.168.1.110]

TASK [copy wordpress to website] ***********************************************************************************************************************************
changed: [192.168.1.110]

TASK [delete tmp wordpress] ****************************************************************************************************************************************
changed: [192.168.1.110]

PLAY RECAP *********************************************************************************************************************************************************
192.168.1.110              : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- **A second target1 playbook used to update MySQL to at least version 5.6 to utilize new sha256_password plugin to patch MD5 Hash Vulnerability**
- Consists of full server backup along with /etc/mysql/my.cnf file, downloading and installing MySQL APT repo, and then upgrading MySQL server and databases