Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

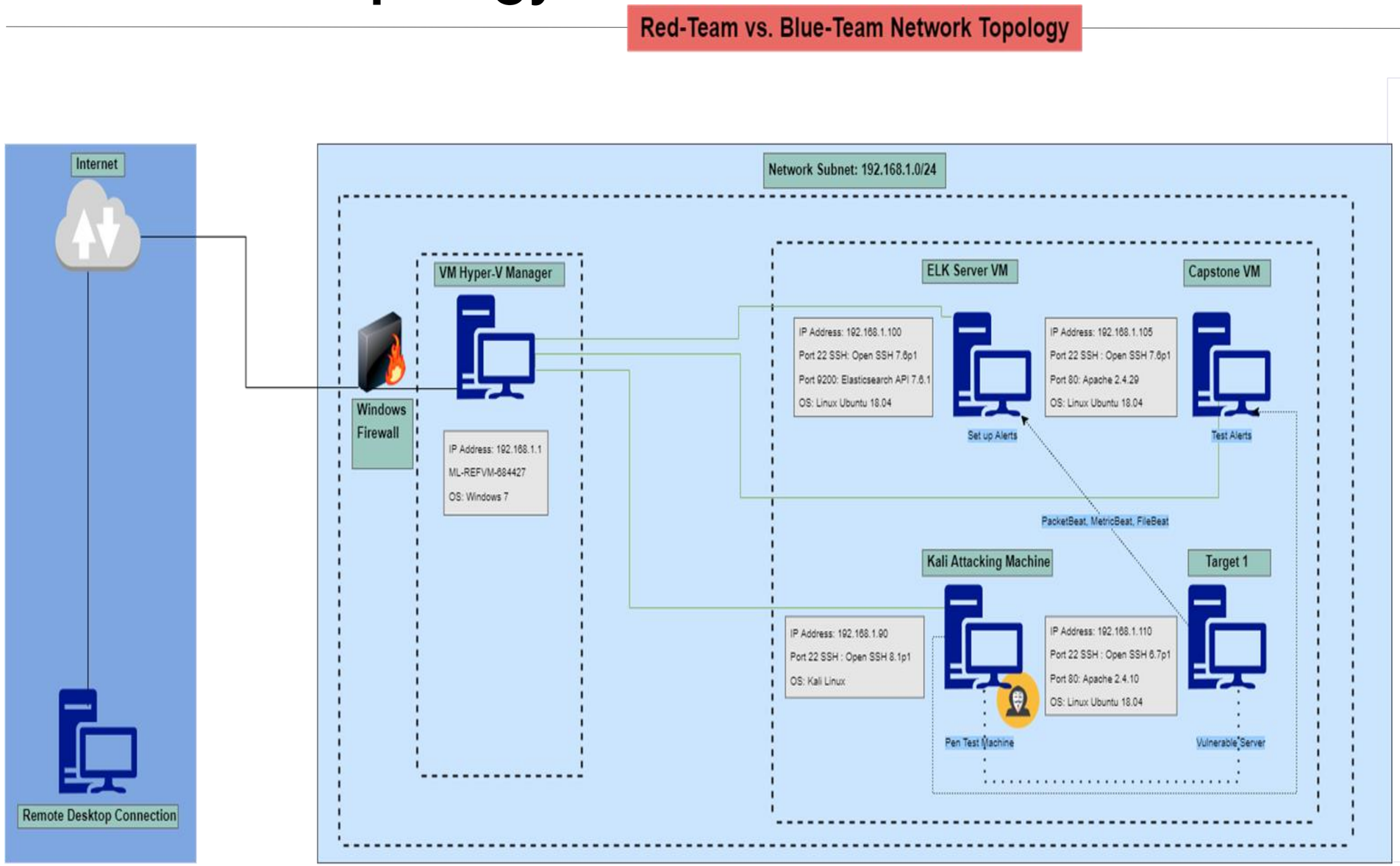
Presented by:
Travis Sherwood, Clement Yang, Olivia Stine, Joseph Arzeno, Omar Anbari, and Edward Cruz

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network

Address Range:192.168.1.1/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines

IPv4: 192.168.1.90
OS: Linux Kali 5.4.0
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux Ubuntu 18.04
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.110
OS: Debian GNU/ Linux 8
Hostname: Target 1

Critical Vulnerabilities: Target 1

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

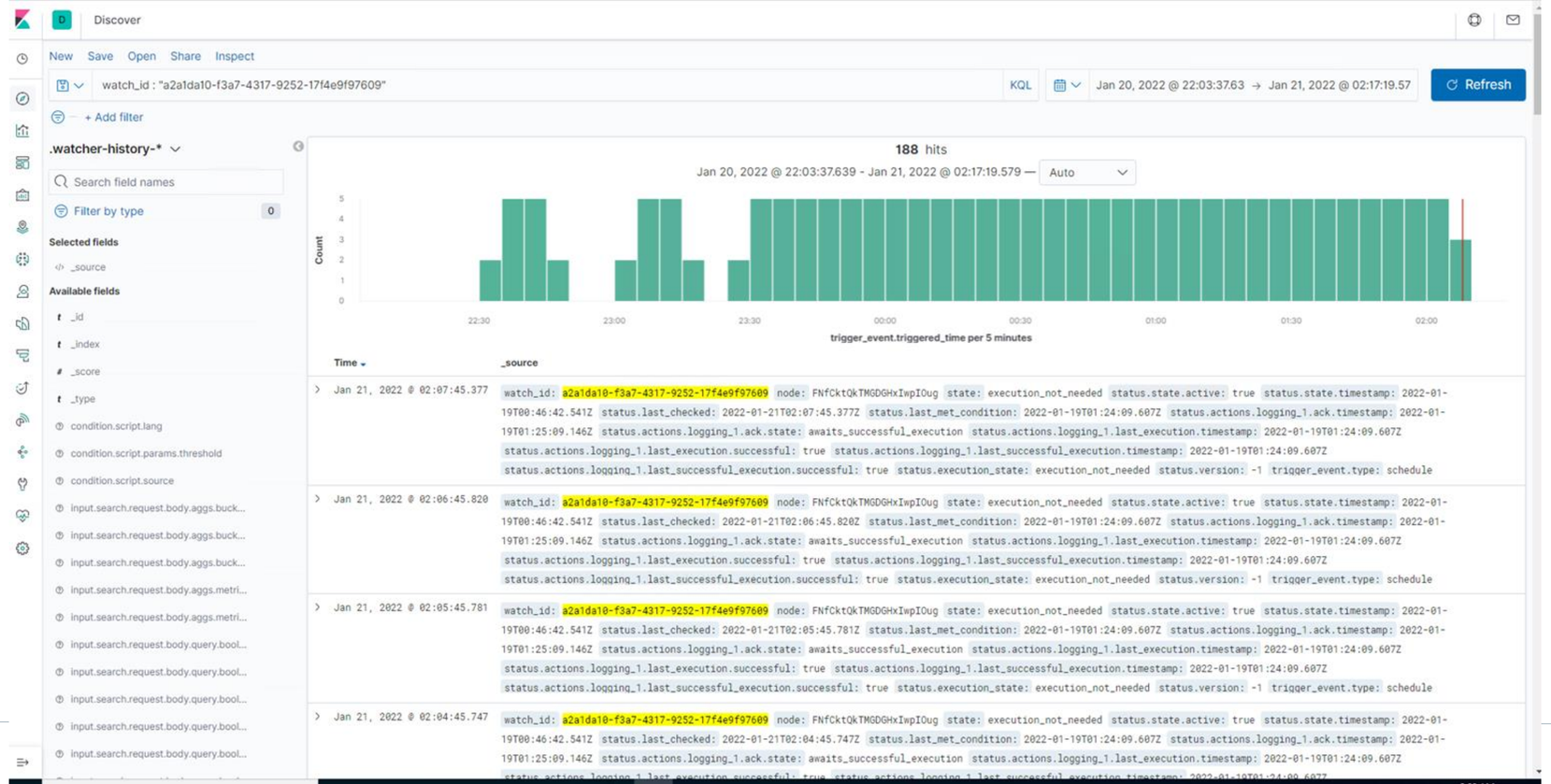
Vulnerability	Description	Impact
Port Scanning Vulnerability	Used by attackers to identify open ports and network services running on a host.	Attacker identified ports (HTTP)80 and (SSH)22 as open.
Weak Password Policy Vulnerability	A weak password policy leaves a network vulnerable to password cracking attacks where attackers are able gain access to privileged resources.	Attackers were able to use the Hydra password cracker and John the Ripper to gain access to user accounts.
MySQL Access Control Vulnerability	Allows attacker to run a sql command (mysql -u root -p) to access the database.	Attacker gained access to sql database and extracted hashed passwords.
Privilege Escalation Vulnerability	Network attack used to obtain unauthorized access to systems within the security perimeter.	Attacker was able to exploit python root escalation privileges to obtain root access.

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

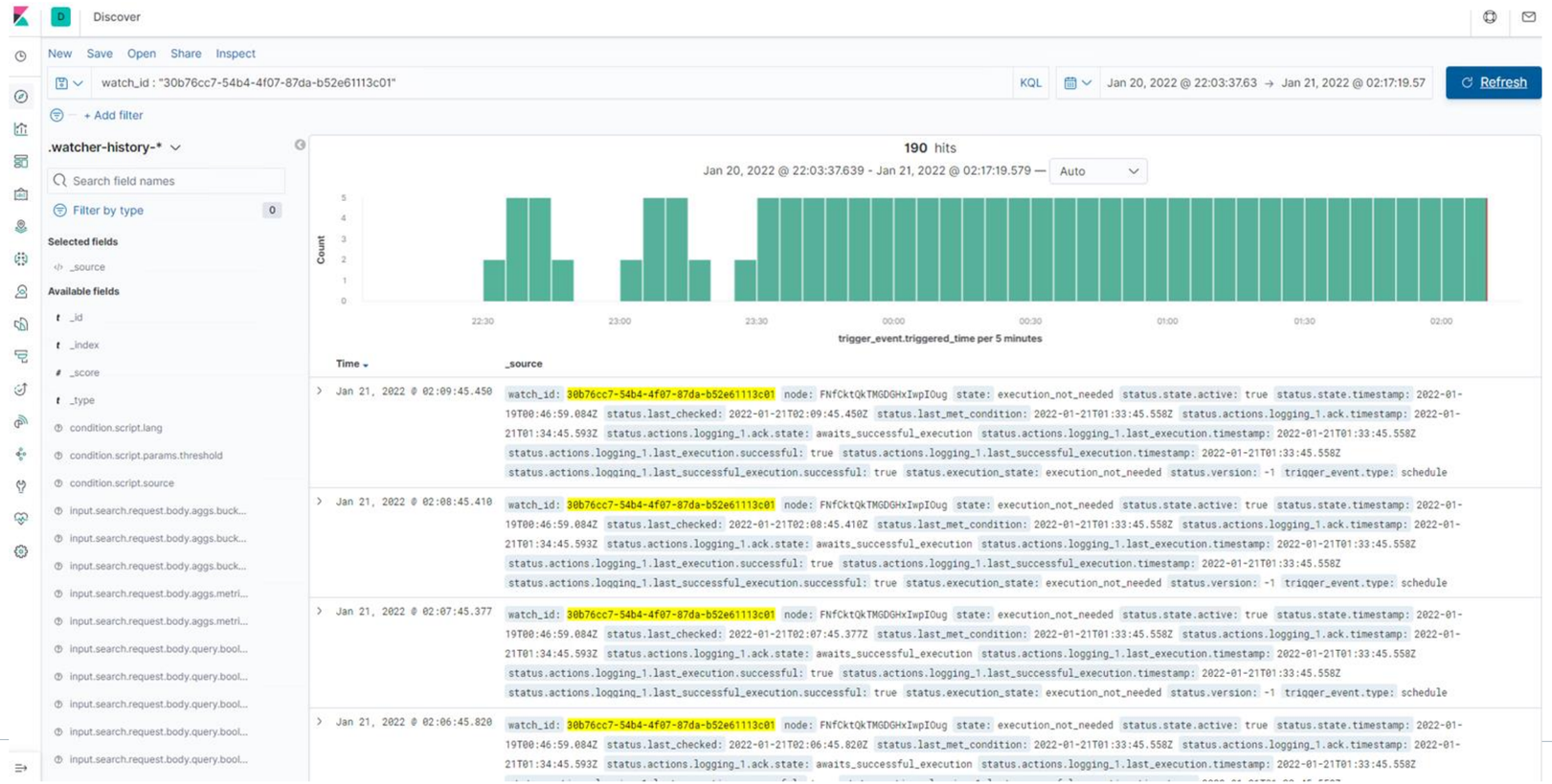
- Which **metric** does this alert monitor? “http.response.status_code”
- What is the **threshold** it fires at? “IS ABOVE 400 FOR THE LAST 5 minutes”
- Provide a screenshot of the alert in action.



HTTP Request Size Monitor

Summarize the following:

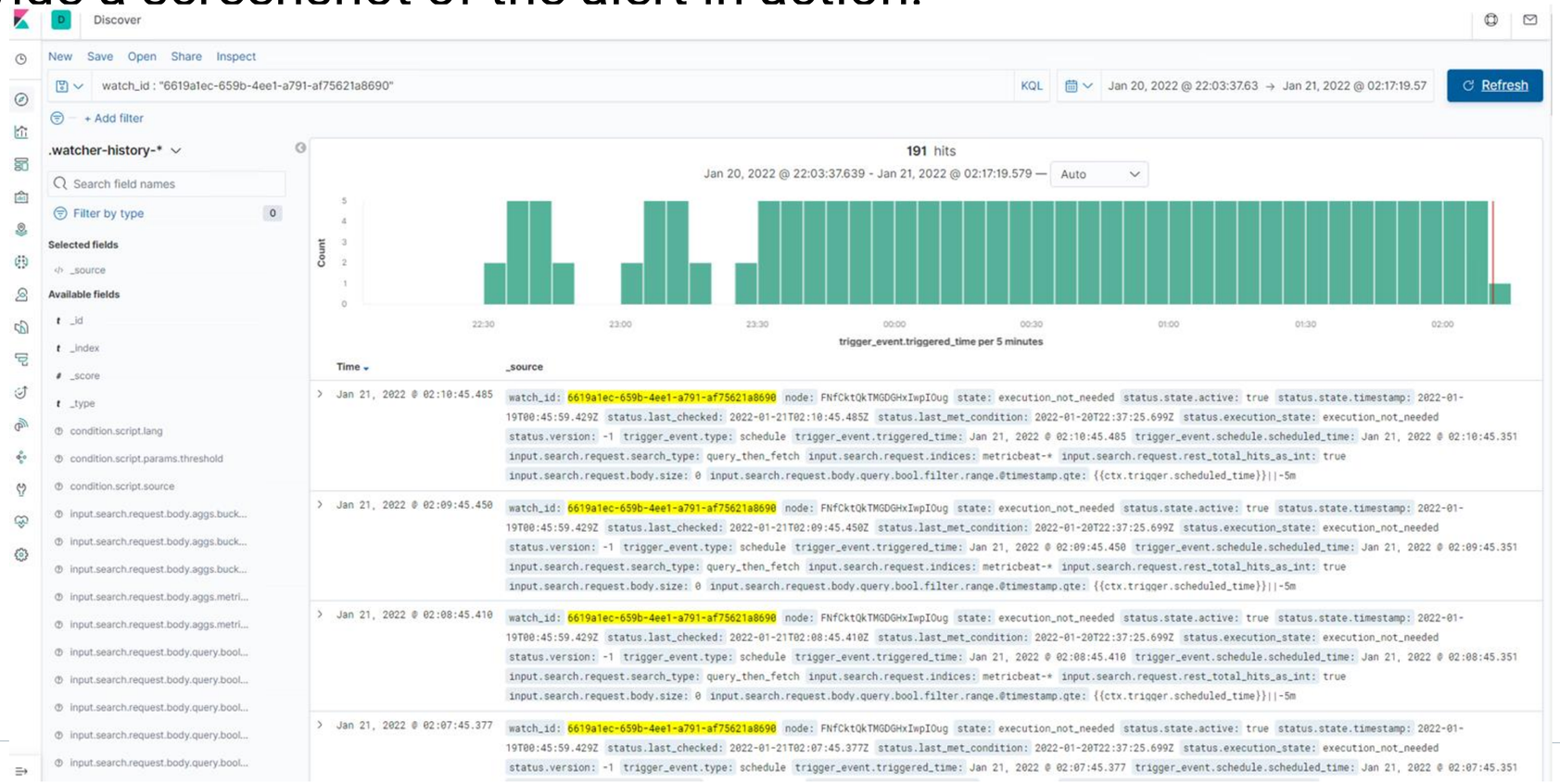
- Which **metric** does this alert monitor? “http.request.bytes”
- What is the **threshold** it fires at? “ABOVE 3500 FOR THE LAST 1 minute”
- Provide a screenshot of the alert in action.



CPU Usage Monitor

Summarize the following:

- Which **metric** does this alert monitor? “system.process.cpu.total.pct”
- What is the **threshold** it fires at? “ABOVE 0.5 FOR THE LAST 5 minutes”
- Provide a screenshot of the alert in action.



Hardening

Hardening Against Port Scanning on Target 1

Install a Firewall

- A properly configured firewall can detect and block a port scan. You can also blacklist an IP after detecting and analyzing the traffic.
- Using firewalld, the IP of the attacking machine can be blacklisted.
  - Commands:
    - $ sudo systemctl enable firewalld
    - $ sudo systemctl start firewalld
    - $ sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4' source address='192.168.1.90' reject"

Implementing an IPS can also help to actively detect and block malicious traffic.

Hardening Against Weak Passwords on Target 1

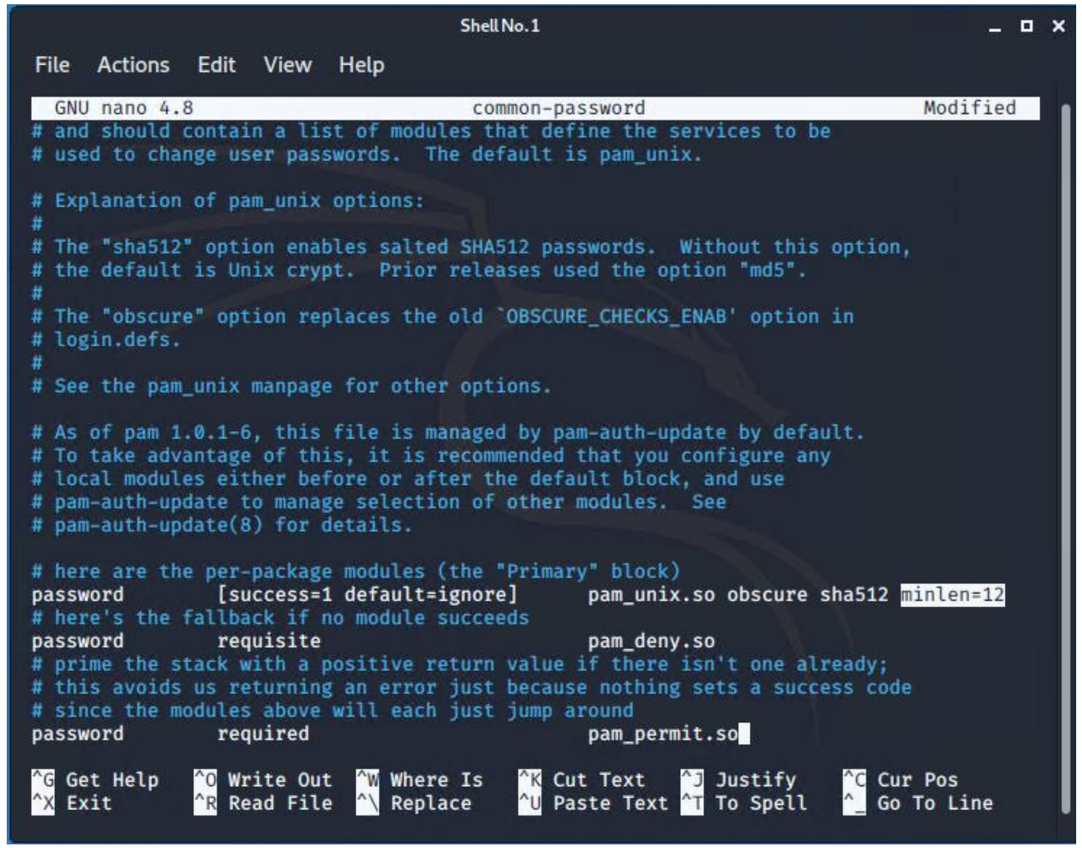
Implement a Strong Password Policy

- A strong password policy can strengthen authentication to the network.
- Password Policy:
  - Set a minimum password length.
  - Require the use of upper and lower case letters, numbers, and special characters
  - Enforce a password history policy
  - Require multi-factor authentication (MFA)
  - Limit bad login attempts
  - Use of a CAPTCHA
  - Require the use of SSH keys to remotely access a server

Commands to set a minimum password length:

- sudo nano /etc/pam.d/common-password

Edit the common-password file with: minlen=(length of password)



Hardening Against MySQL Access Control on Target 1

Secure Access to MySQL Server

- Properly configuring the access controls to a MySQL Database is an important step in protecting against this vulnerability.
- Command
  - sudo mysql_secure_installation
    - Options are made available to update password plugin, select level of password validation, change the root password, remove anonymous user, disallow remote root login, and remove the test database.

```
Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

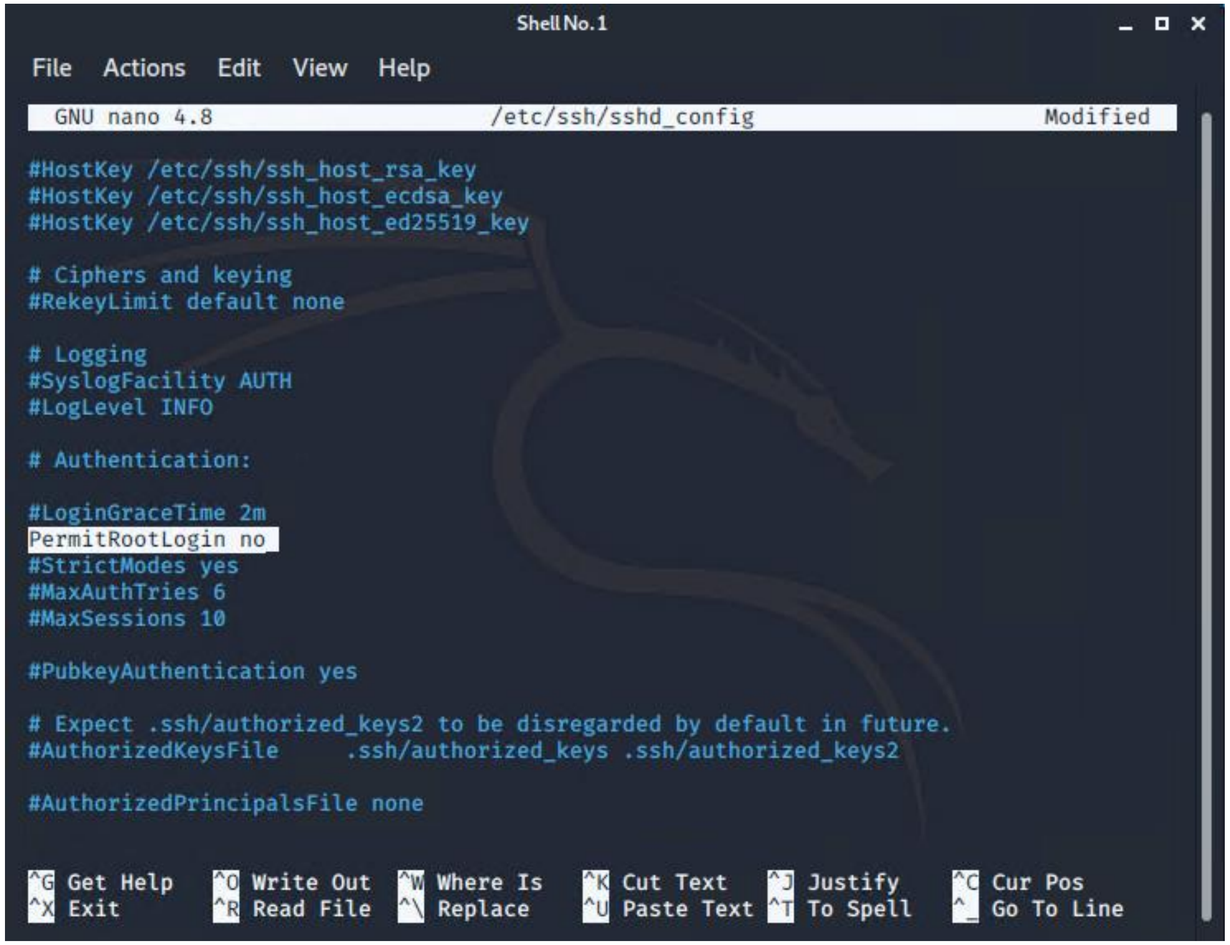
https://yallalabs.com/linux/securing-mysql-server-mariadb-with-mysql_secure_installation/

Reference: https://dev.mysql.com/doc/refman/5.7/en/mysql-secure-installation.html

Hardening Against Privelege Escalation on Target 1

Disable Remote Root Access

- This patch works because it will keep remote users from accessing the root (which is the ultimate goal in a privilege escalation attack).
- Commands:
  - sudo nano /etc/ssh/sshd_config
    - Edit the sshd_config file to read “PermitRootLogin no”



Implementing Patches

Implementing Patches with Ansible

Playbook Overview

Vulnerabilities: Port Scanning and Privilege Escalation

```yaml
---

- hosts: all
  tasks:
    - name: Disable root login over SSH
      hosts: all:!tiny:!proxmox
      gather_facts: false
      user: root

    - name: Add hardened SSH config
      copy:
        dest: /etc/ssh/sshd_config
        src: etc/ssh/sshd_config
        owner: root
        group: root
        mode: 0600
         permitrootlogin: no
      notify: Reload SSH

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

    - name: Remove undesirable packages
      package:
        name: "{{ unnecessary_software }}"
        state: absent

    - name: Stop and disable unnecessary services
      service:
        name: "{{ item }}"
        state: stopped
        enabled: no
      with_items: "{{ unnecessary_services }}"
      ignore_errors: yes

    - name: Block specific IP
      ansible.builtin.iptables:
        chain: INPUT
        source: 192.168.1.90
        jump: DROP
      become: yes

  vars:
    allowed_ssh_networks:S
      - 192.168.1.0/24
    unnecessary_services:
      - postfix
      - telnet
    unnecessary_software:
      - tcpdump
      - nmap
      - wpa_supplicant
```

Implementing Patches with Ansible

Playbook Overview

Vulnerabilities: Weak Passwords

```
---
- name: Set Enforce password history to 24 or more passwords
| cis enforce-password-history 1.1.1
  win_command: net accounts /uniquepw:24
  register: passHistory
  args:
    creates: C:\passHistory.lock

- name: Create passHistory.lock if password history is enforced
  win_copy:
    dest: C:\passHistory.lock
    content: ""
    force: no
  when: passHistory

- name: Set Maximum password age to 60 or more days | cis
maximum-password-age 1.1.2
  win_command: net accounts /maxpwage:60
  register: maxPassAge
  args:
    creates: C:\maxPassAge.lock

- name: Create maxPassAge.lock if password history is
enforced
  win_copy:
    dest: C:\maxPassAge.lock
    content: ""
    force: no
  when: maxPassAge
```

```
- name: Set Minimum password age to 1 or more days | cis
minimum-password-age 1.1.3
  win_command: net accounts /minpwage:1
  register: minPassAge
  args:
    creates: C:\minPassAge.lock

- name: Create minPassAge.lock if password history is
enforced
  win_copy:
    dest: C:\minPassAge.lock
    content: ""
    force: no
  when: minPassAge

- name: Set Minimum password length to 14 or more
characters | cis minimum-password-length 1.1.4
  win_command: net accounts /minpwlen:14
  register: minPassLength
  args:
    creates: C:\minPassLength.lock

- name: Create minPassLength.lock if password history is
enforced
  win_copy:
    dest: C:\minPassLength.lock
    content: ""
    force: no
  when: minPassLength
```

Implementing Patches with Ansible

Playbook Overview

Vulnerabilities: MySQL Access

```
---
- name: Set logging
  community.general.ufw:
    rule: limit
    port: ssh
    proto: tcp


- name: Deny all access to port 3306(mysql)
  community.general.ufw:
    rule: deny
    port: '3306'


- name: Deny all access to tcp port 80
  community.general.ufw:
    rule: deny
    port: '80'
    proto: tcp
```

Utilized Resources

5 ways to harden a new system with Ansible: https://www.redhat.com/sysadmin/harden-new-system-ansible

Ansible Windows Hardening: https://github.com/dev-sec/ansible-windows-hardening/blob/master/tasks/password_policy.yml

Best Practices for Preventig HTTP Flood Attacks: https://www.alibabacloud.com/help/en/doc-detail/100694.htm

Blocking Malware and Prohibit Files: https://www.cisco.com/c/en/us/td/docs/security/firesight/541/firepower-module-user-guide/asa-firepower-module-user-guide-v541/AMP-Config.pdf

Configure MySQL Database Reference: https://stackoverflow.com/questions/51251225/install-and-configure-mysql-using-ansible

Disable Remote Root Access Reference: https://stackoverflow.com/questions/68404640/use-ansible-playbook-to-enable-and-disable-root-login

Fifty Shades of Malware Hashing: https://medium.com/malware-buddy/fifty-shades-of-malware-hashing-3783d98df59c#:~:text=Hashing%20is%20a%20commonly%20used%20method%20to%20uniquely%20identify%20malware.&text=The%20malware%20is%20run%20through,be%20used%20to%20classify%20malware

Hardening SSH with Ansible: https://www.nathancurry.com/blog/23-hardening-ssh-with-ansible/

Manage Detection Alerts: https://www.elastic.co/guide/en/security/current/alerts-ui-manage.html

Manage firewall with UFW: https://docs.ansible.com/ansible/latest/collections/community/general/ufw_module.html

MetaFlows Network Antivirus: Content MD5 Hash: https://research.metaflows.com/stats/content_md5_hash