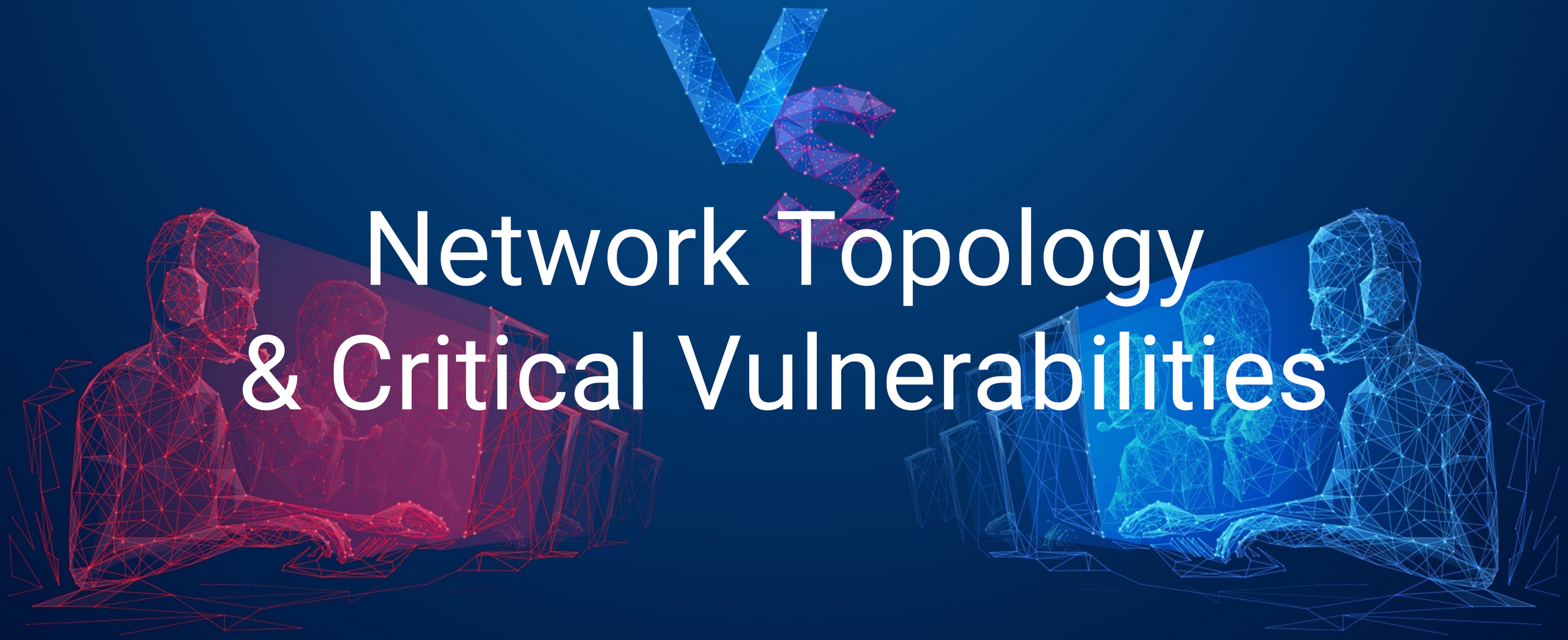
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

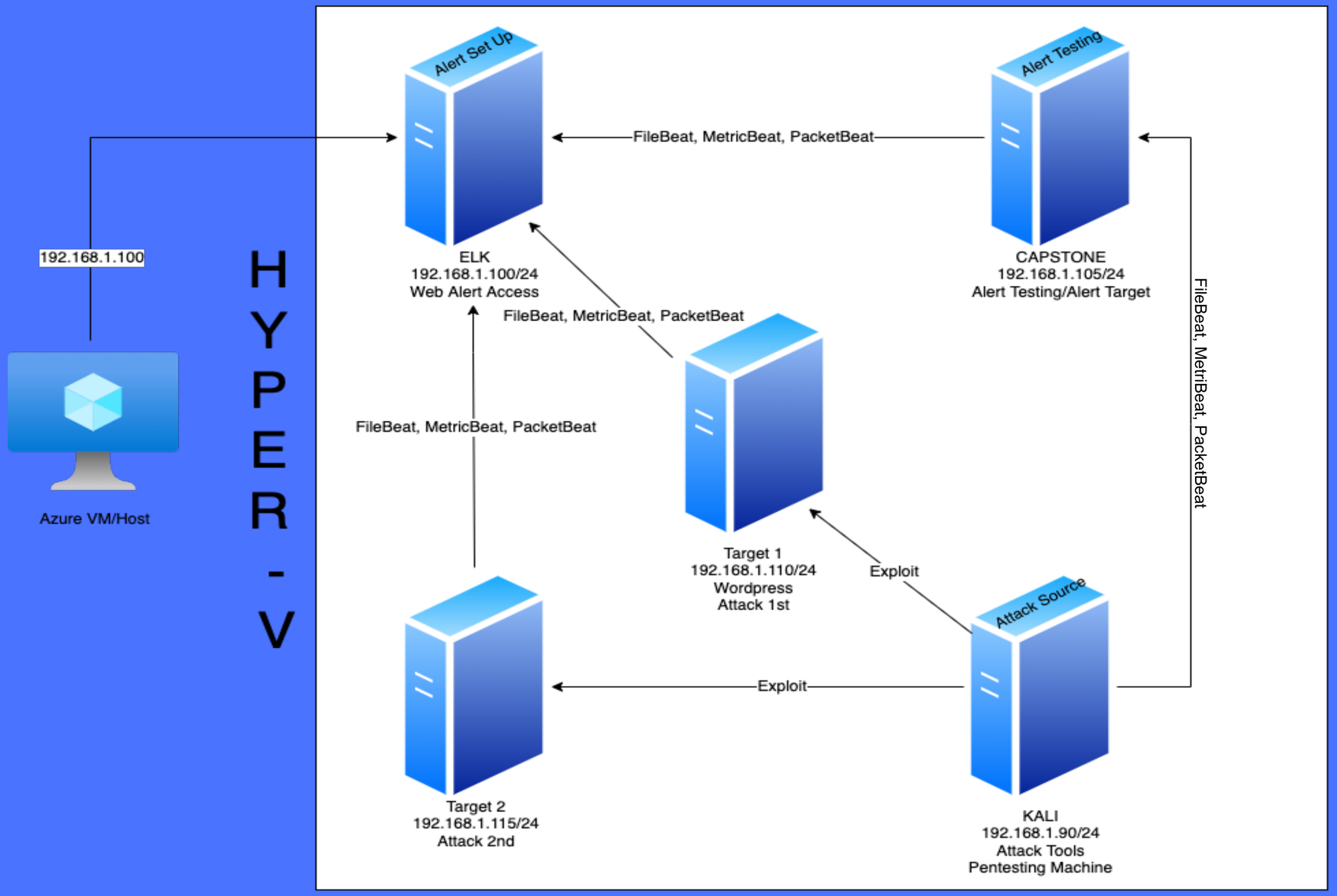
Table of Contents

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: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

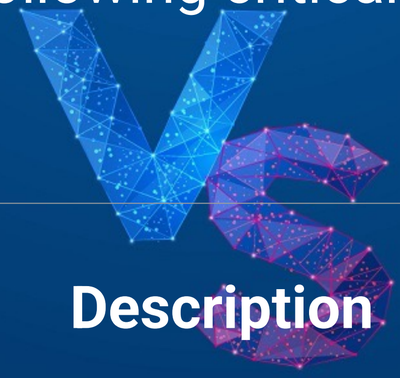
Critical Vulnerabilities: Target 1

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

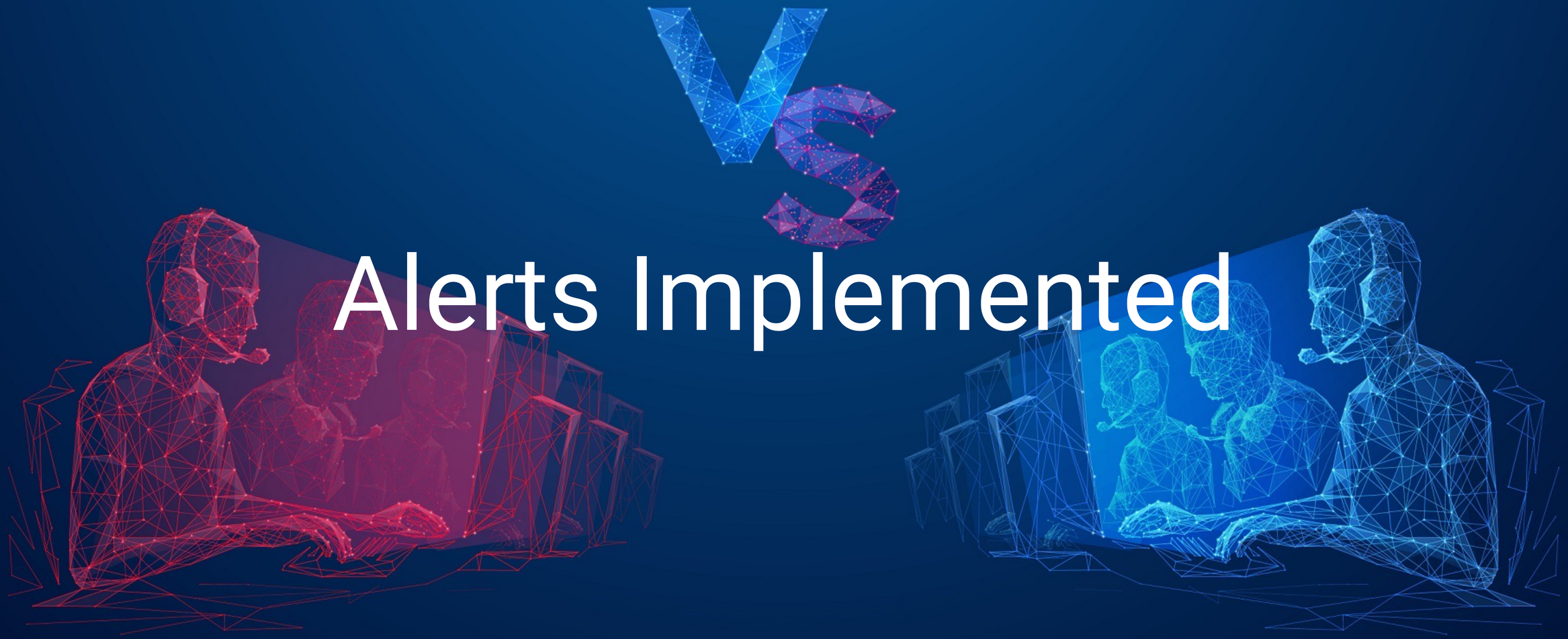
Vulnerability	Description	Impact
Ability to enumerate Wordpress usernames	Using wpscan against url to enumerate usernames.	Easily accessed usernames
Easily guessed passwords	Once usernames were enumerated, passwords were easy to guess and brute force due to simplicity of passwords	Ability to move within system
Mitre.org: Abuse Elevation Control Mechanism: Sudo & Sudo Caching	Adversaries may perform sudo caching and/or use the suoders file to elevate privileges. Adversaries may do this to execute commands as other users or spawn processes with higher privileges.	Ability to escalate privileges to root

Critical Vulnerabilities: Target 1

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

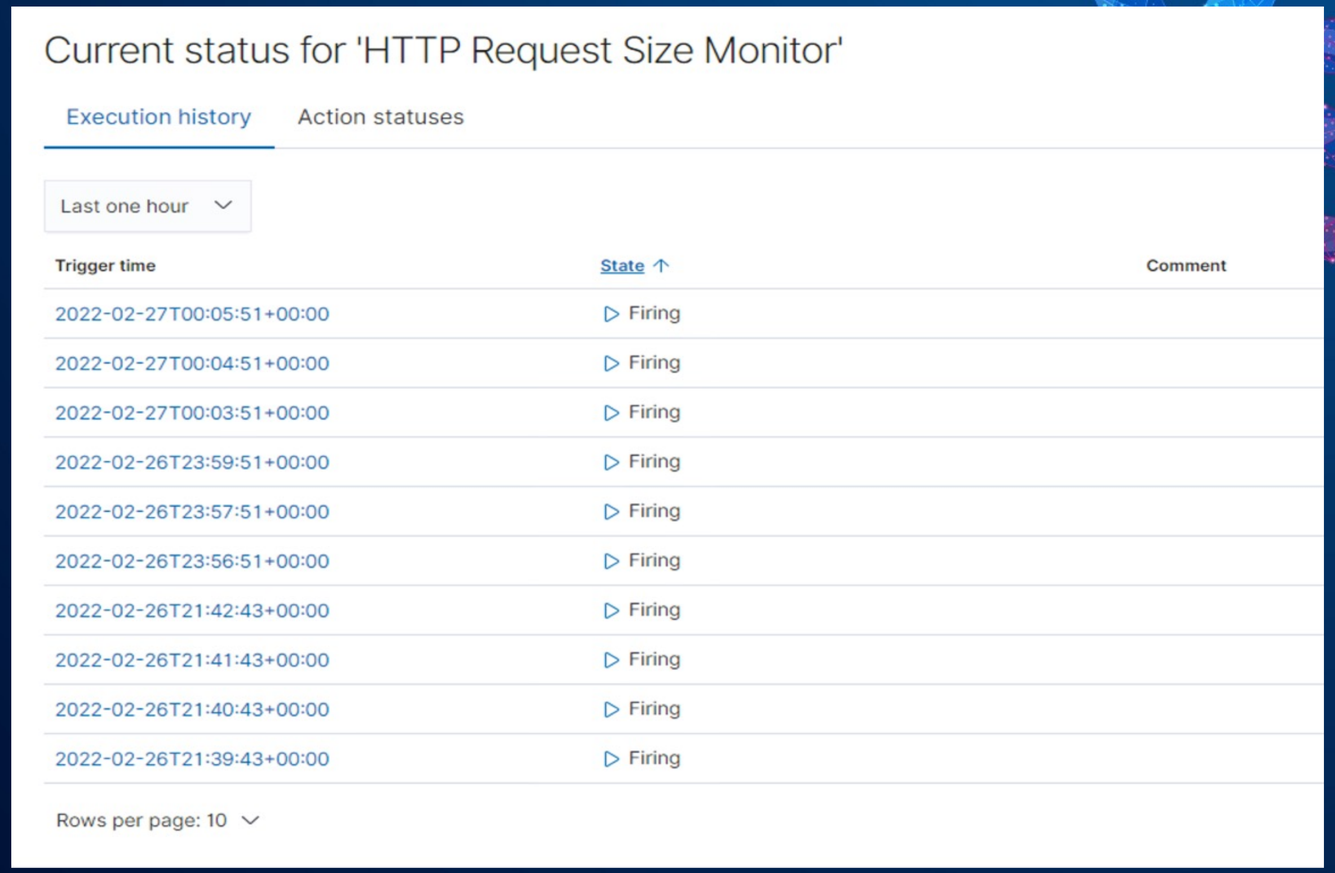


Vulnerability	Description	Impact
Open ports with unrestricted access(22)	Nmap scan discovered two open and unrestricted ports leading to users being vulnerable to malicious services	Allows attackers to exploit programs and access private files.
Open access to SQL database with with accessible login information	Backup file is saved in a openly accessible directory on the server	Attacker is able to gain unrestricted access to system information



Alerts Implemented

HTTP REQUEST SIZE MONITOR



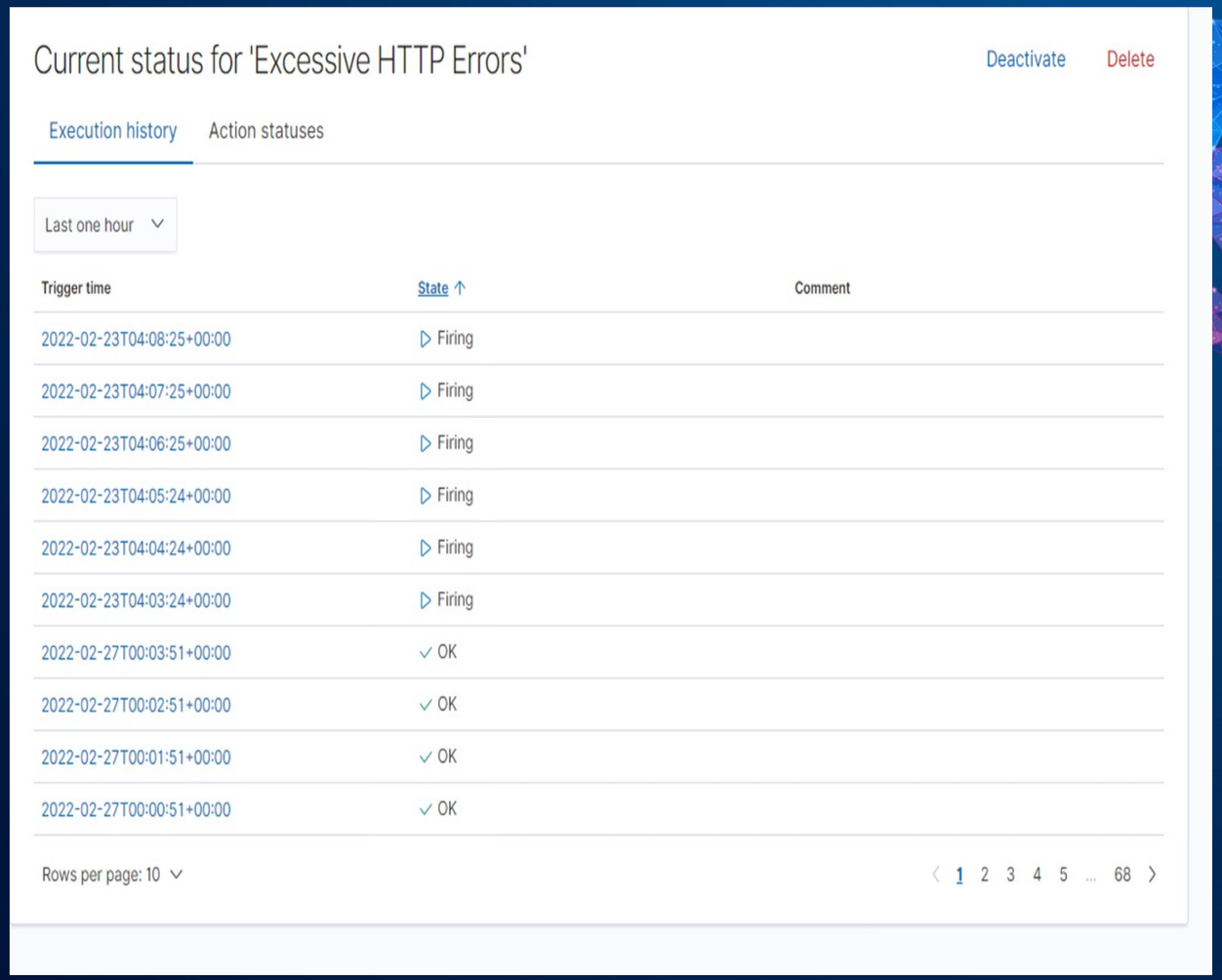
Metric: http.request.bytes

Threshold: Alert triggers when the sum of HTTP request bytes reaches higher than 3500 bytes per minute. Queries PacketBeat to monitor HTTP data requests

Vulnerability Mitigated: Denial of Service Attack

Reliability: High reliability.

EXCESSIVE HTTP ERRORS



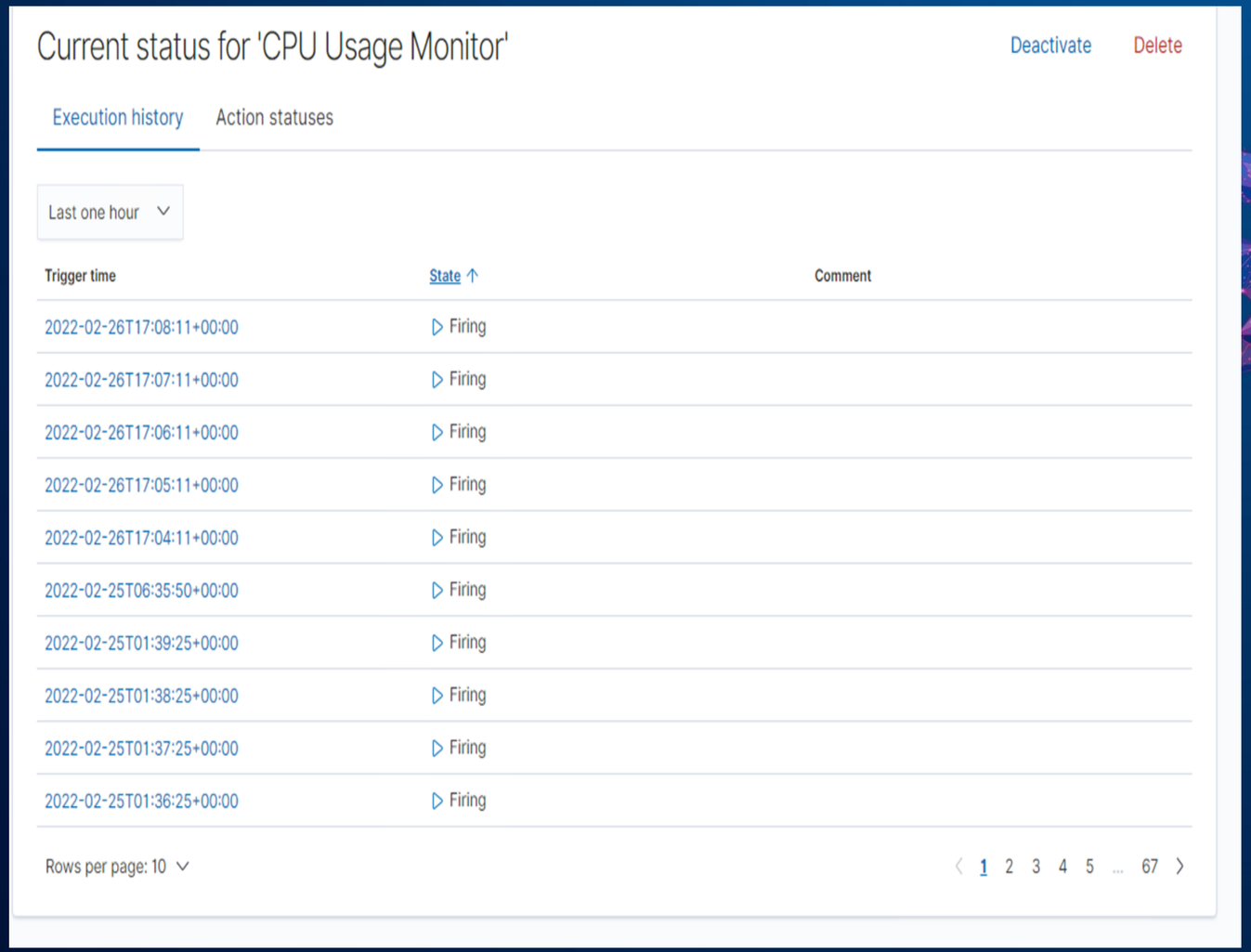
Metric: HTTP errors

Threshold: Measures the error codes above 400, while filtering out normal activity and successful responses. Error codes that occur at a high rate are indicators of attacks(brute force) and should be cause for concern. Queries PacketBeat for HTTP status code responses.

Vulnerability Mitigated: Ability to allocate resources and Brute Force Attacks

Reliability: High Reliability

CPU USAGE MONITOR



Metric: system.process.cpu.total.pct

Threshold: Alert triggers when CPU activity exceeds 50% (or 0.5) for the last 5 minutes. Queries MetricBeat for system processes as a percentage of CPU activity.

Vulnerability Mitigated: malicious programs that are stealing resources

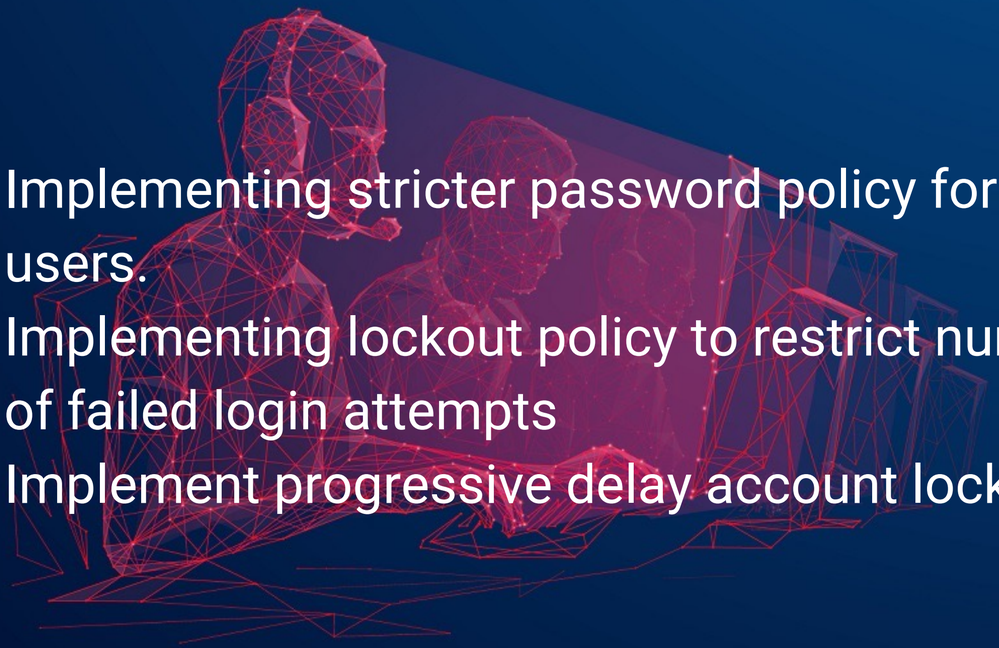
Reliability: High Reliability

Hardening

Hardening Against Weak Passwords and Open Ports on Target 1

Recommendations:

Weak Passwords



- Implementing stricter password policy for users.
- Implementing lockout policy to restrict number of failed login attempts
- Implement progressive delay account lockout

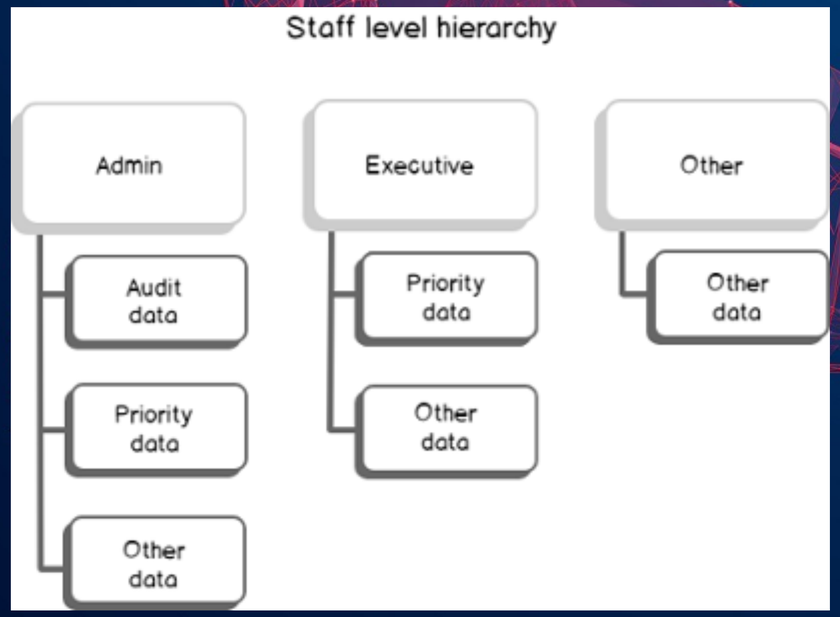
Open Ports(22 and 80)

- Disallow access via Port22(OpenSSH) to disable any connection requests to the server, thereby preventing attacker access.

Hardening Against Available SQL Database on Target 1

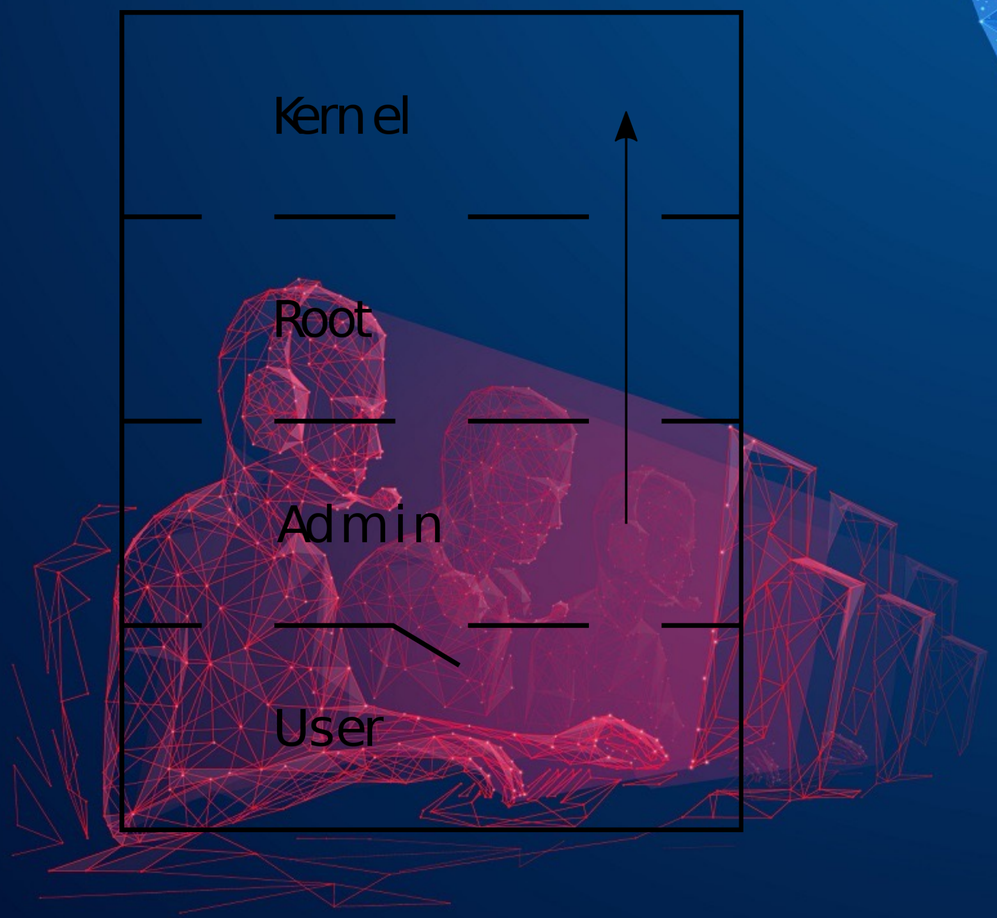
Recommendations:

- Implement database access employee hierarchy(groups & roles) to prevent open access to all databases and tables.
- Configure and hash the wp-config.php file to prevent access to login credentials




```
USE master
GO

CREATE LOGIN [exampleExecutiveStaff] WITH PASSWORD = 'this1passwordisONLYanexample!'
CREATE USER [exampleExecutiveStaff] FROM LOGIN [exampleExecutiveStaff]

CREATE LOGIN [exampleOtherStaff] WITH PASSWORD = 'this2passwordisONLYanexample!'
CREATE USER [exampleOtherStaff] FROM LOGIN [exampleOtherStaff]


---- Example database
USE GenExaAll
GO

CREATE USER [exampleExecutiveStaff] FROM LOGIN [exampleExecutiveStaff]
CREATE USER [exampleOtherStaff] FROM LOGIN [exampleOtherStaff]
CREATE ROLE [executive]
CREATE ROLE [other]
ALTER ROLE [executive] ADD MEMBER [exampleExecutiveStaff]
ALTER ROLE [other] ADD MEMBER [exampleOtherStaff]
```

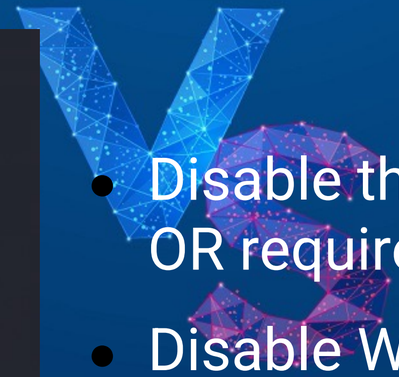
Hardening Against Privilege Escalation on Target 1



- Effective management of accounts(especially privileged)
- Monitor and log user behavior
- Effective file permission management of all user accounts.
- Limiting SUDO right permissions using principle of least privilege

Hardening Against WordPress Enumeration on Target 1

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress/ --enumerateu

Scan Aborted: invalid option: --enumerateu
root@Kali:~# wpscan --url http://192.168.1.110/wordpress/ --enumerate u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Thu Feb 24 19:03:51 2022

Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access
```



- Disable the Wordpress REST API when not in use OR require authentication for all requests.
- Disable Wordpress XML-RPC when not in use.
  - add_filter(‘xmlrpc_enabled’,’ ‘_return_false’
- Configure webserver to deny requests to /?author=<number>.
- Obscure /wp-admin and wp-login.php directly to the public internet.

Implementing Patches

Implementing Patches with Ansible

- **The Ansible playbook should ensure all systems and packages are up to date and patched.**

```
  - name: update the system
    yum:
      name: "*"
      state: latest
```

```
---
- name: OS update
  hosts: dev
  gather_facts: yes
  tasks:
    - name: OS update - all packages or security fixes only
      include_role:
        name: os_update
...
```

```
tasks:

  - name:  Task 1 - verify web/database processes are not running
    shell: if ps -eaf | egrep 'apache|http|nginx|mysql|postgresql|
    ignore_errors: true
    register: app_process_check


  - name:  Task 2 - decision point to start patching
    fail: msg="{{ inventory_hostname }} have running Application.
    when: app_process_check.stdout == "process_running"


  - name:  Task 3 - upgrade kernel package on RHEL/CentOS server
    yum:
     name="kernel"
     state=latest
    when: app_process_check.stdout == "process_not_running" and an
    register: yum_update

  - name:  Task 4 - upgrade kernel package on Ubuntu server
    apt:
      update_cache: yes
      force_apt_get: yes
      cache_valid_time: 3600
      name: linux-image-generic
      state: latest
    when: app_process_check.stdout == "process_not_running" and an
    register: apt_update
```


The End