Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

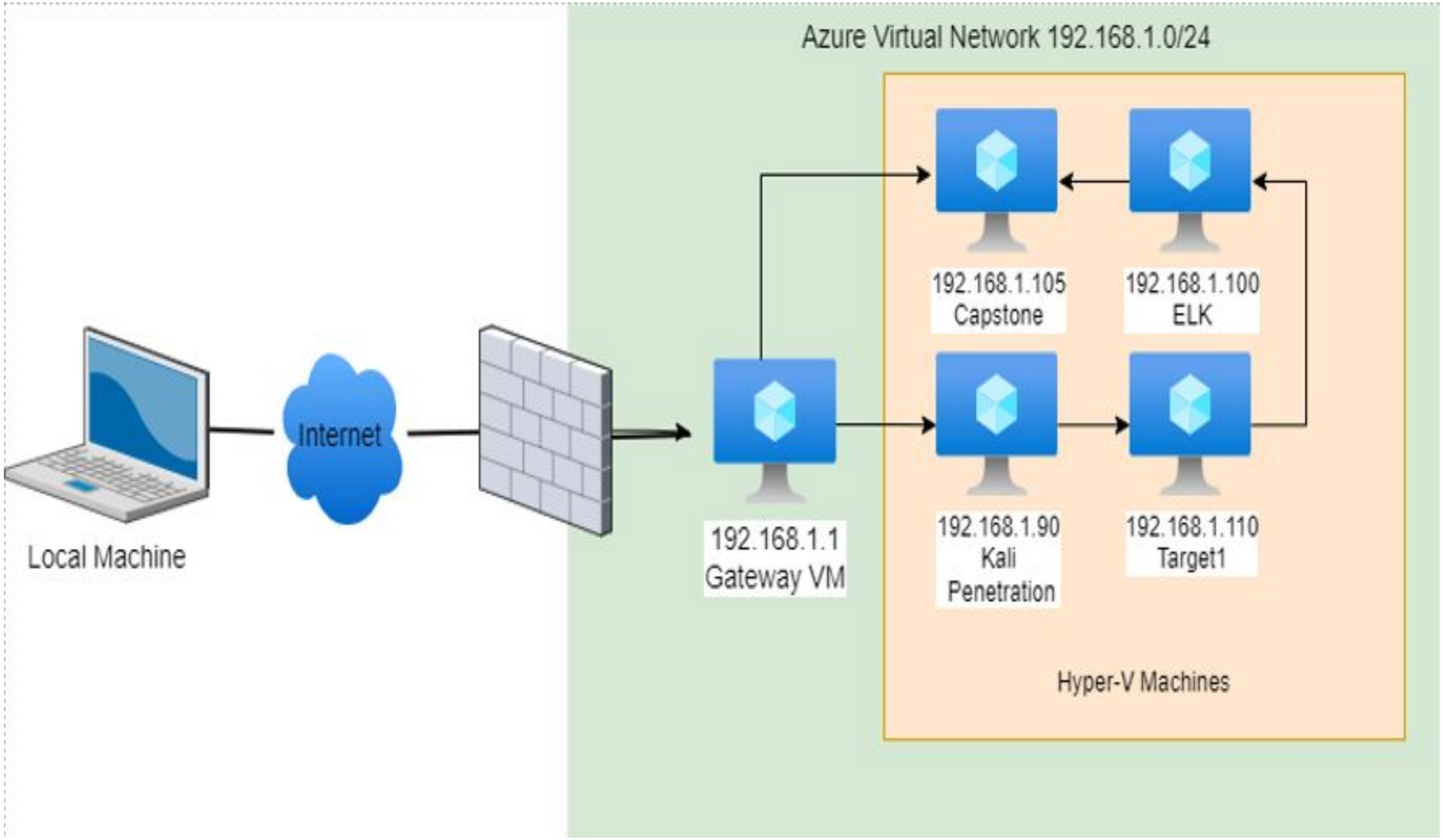
Iris Carrell, Jacob Starks, Braden Welsh, Crystal Hamilton,and Carolina Hernandez

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: Kali Linux 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.105
OS: Linux Ubuntu
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

Vulnerability	**Description**	**Impact**
Ports 22 and 80 are vulnerable	Direct access to machine via SSH scans and a direct access to the Target 1 machine	All integrity and confidentiality because of direct access to machine and ability to gain more details about users/visitors
Weak/Insecure Passwords	User ‘Michael’ had an easy password which was cracked using brute force	All integrity and confidentiality due to the easy ability to breach the machine and gain more information about users/operations
Enumerate WordPress Site	Users were identifiable via WPScan	All confidentiality is impacted through the disclosure of usernames and other details

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Apache 2.4.10 CVE-2016-4975	Apache Server can be vulnerable for CRLF Injection	Integrity impact as it allows the attacker to set fake cookies, steal CSRF tokens, disclose user information by injecting a script (XSS) and perform a variety of other attacks. It also allows attackers to deactivate & bypass security measures like XSS filters & Same Origin Policy (SOP) (See more at (CRLF Injection Attack - (https://www.geeksforgeeks.org/crlfinjection-attack/))
Python Privilege Escalation	The user Steven can circumvent lower privileges by using python scripting allowed for sudo	Integrity and Confidentiality by gaining root access to the machine

Alerts Implemented

HTTP Request Size Monitor

- This monitoring rule watches the http.request.bytes from metricbeat. It will fire when it exceeds a sum of 3500 for the last minute
- WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

Current status for 'HTTP Request Size Monitor '

Deactivate Delete

Execution history | Action statuses

Last one hour

Trigger time	State	Comment
2022-05-04T02:50:13+00:00	OK	
2022-05-04T02:49:13+00:00	OK	
2022-05-04T02:48:13+00:00	OK	
2022-05-04T02:47:13+00:00	OK	
2022-05-04T02:46:13+00:00	OK	
2022-05-04T02:45:13+00:00	OK	
2022-05-04T02:44:13+00:00	OK	
2022-05-04T02:43:13+00:00	OK	
2022-05-04T02:42:13+00:00	OK	
2022-05-04T02:41:13+00:00	OK	

Rows per page: 10

< 1 2 3 4 5 ... 43 >

Excessive HTTP Errors

- This monitoring rule watches the http.response.status_code from metricbeat. It will fire when it reaches above a count of 400 for the last 5 minutes
- WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes

Current status for 'Excessive HTTP Errors'

Deactivate Delete

Execution history | Action statuses

Last one hour

Trigger time	State	Comment
2022-05-04T02:50:13+00:00	OK	
2022-05-04T02:49:13+00:00	OK	
2022-05-04T02:48:13+00:00	OK	
2022-05-04T02:47:13+00:00	OK	
2022-05-04T02:46:13+00:00	OK	
2022-05-04T02:45:13+00:00	OK	
2022-05-04T02:44:13+00:00	OK	
2022-05-04T02:43:13+00:00	OK	
2022-05-04T02:42:13+00:00	OK	
2022-05-04T02:41:13+00:00	OK	

Rows per page: 10

< 1 2 3 4 5 ... 43 >

CPU Usage Monitor

- This monitoring rule watches the system.process.cpu.total.pct from metricbeat. It will fire when its max value remains above 0.5 over all processes for the last 5 minutes The condition syntax is WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes

Current status for 'CPU Usage Monitor' Deactivate Delete

Execution history Action statuses

Last one hour

Trigger time	State	Comment
2022-05-04T02:51:13+00:00	OK	
2022-05-04T02:50:13+00:00	OK	
2022-05-04T02:49:13+00:00	OK	
2022-05-04T02:48:13+00:00	OK	
2022-05-04T02:47:13+00:00	OK	
2022-05-04T02:46:13+00:00	OK	
2022-05-04T02:45:13+00:00	OK	
2022-05-04T02:44:13+00:00	OK	
2022-05-04T02:43:13+00:00	OK	
2022-05-04T02:42:13+00:00	OK	

Rows per page: 10 < 1 2 3 4 5 ... 43 >

Hardening

Hardening Against Vulnerable Ports 22 and 80 on Target 1

- Close port 22 and use port 443 with https instead of 80.
- Port 22 will prevent open ssh access to the machine. Using port 443 will
- provide a layer of security using ssl instead of the open port.
- Port 80 and 22 can be shut down with:
  - sudo ufw deny PORT 80
  - sudo ufw deny PORT 22
  - sudo ufw allow PORT 443
- Each command should be run one at a time and checked status with:
  - sudo ufw status verbose

Hardening Against Weak/Insecure Passwords on Target 1

- Users are required to update their passwords, involving at least 16 characters, numbers, and symbols. 1-hour lockouts should be implemented after 5 unsuccessful attempts within 15 minutes. Multi-factor authentication should also be used.

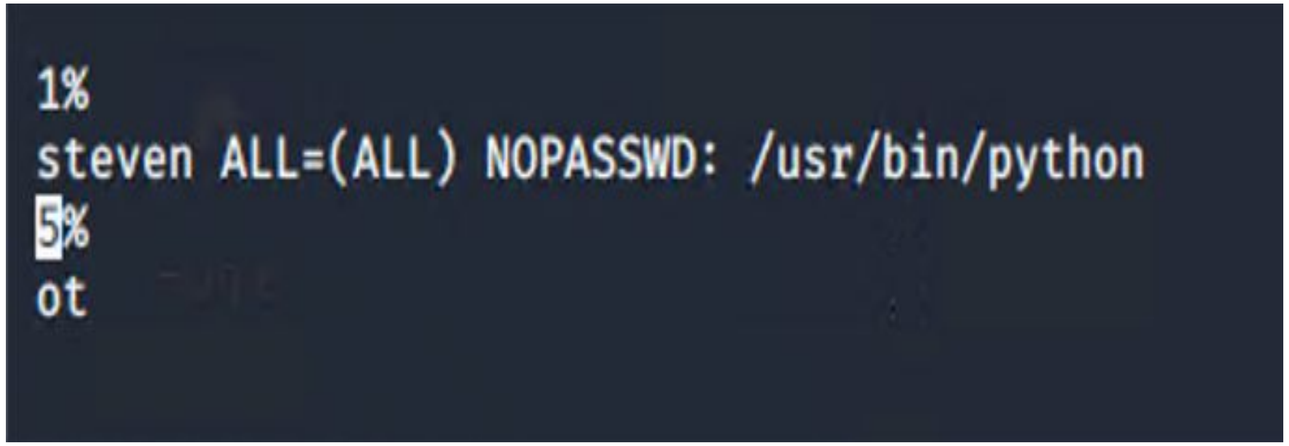
- Difficult passwords are the hardest to crack with brute force and lockouts will prevent multiple attempts. Notification alerts could be generated to further protect the accounts

- The bottom link can install the following processes at:
https://ostechnix.com/how-to-set-password-policies-in-linux/

Hardening Against Python Privilege Escalation on Target 1

- Python privileges should be removed for users vulnerable to ssh as well as users who are not authorized for root privileges.
    - Removing the python sudo privileges will eliminate the potential for circumventing access restrictions
        - vi /etc/sudoers

- Delete this line: steven ALL=(ALL) NOPASSWD: /usr/bin/python



Hardening Against Enumerate Wordpress Site on Target 1

- Deploy an Ansible-Playbook that updates the WordPress site to a patched version with Stop User Enumeration plug-in and adjust firewall to block similar behaviors of enumerating traffic
- Normally, the updated versions Wordpress won't allow enumeration with appropriate plugins
    - Run the ansible playbook discussed in the concluding slide and make sure to apply the Stop-User-Enumeration plug-in is installed and enabled
        - https://wordpress.org/plugins/stop-user-enumeration/
        - sudo ansible-playbook -v WPandApache.yml

Hardening Against Apache 2.4.10 CVE-2016-4975 on Target 1

- Regularly update Apache server to latest stable version:
  - Apache tends to have significant vulnerabilities with every version. In order to stay ahead of these future threats, it is vital to maintain a consistent approach when planning on upgrading the versions
  - The Playbook that needs to run its course will be shown in the concluding slide

Implementing Patches

Implementing Patches with Ansible

- **Playbook Overview**

1. **Lines 7-55 update the wordpress html files and check the website.**
2. **Lines 56-75 update the Apache Serve**
3. **On the final slide, there will be pictures of the Ansible Playbook for viewing**

Implementing Patches with Ansible (Overview)

```yaml
---
- name: WPandApacheUpdate
  hosts: 192.168.1.110
  become_user: root
  become: true
  tasks:
  - name: stop httpd
      systemd:
        name: httpd
        state: stopped
      become: true

    - name: backup html files
      archive:
        path: /var/www/html
        dest: "/home/michael/backups/wordpress-bck-{{ansible_date_time.iso8601_basic_short}}.tgz"
        format: gz
      become: true

    - name: backup wordpress database
      command: /etc/backup-wpdb.sh
      become: true

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

    - name: Wait until wordpress has been downloaded
      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

    - name: start httpd
      systemd:
        name: httpd
        state: started
        daemon_reload: yes
      become: true

    - name: simple check website
      uri:
        url: http://192.168.1.110
    - name: Apache latest version installation
    dnf:
      name: httpd
      state: latest
  - name: Enable service to start on boot up
    service:
      name: httpd
      state: started
  - name: Create firewall rule for apache service
    firewalld:
      service: http
      zone: public
      permanent: yes
      immediate: yes
      state: enabled
  handlers:
  - name: Restart apache service
    service:
      name: httpd
      state: restarted
```