Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

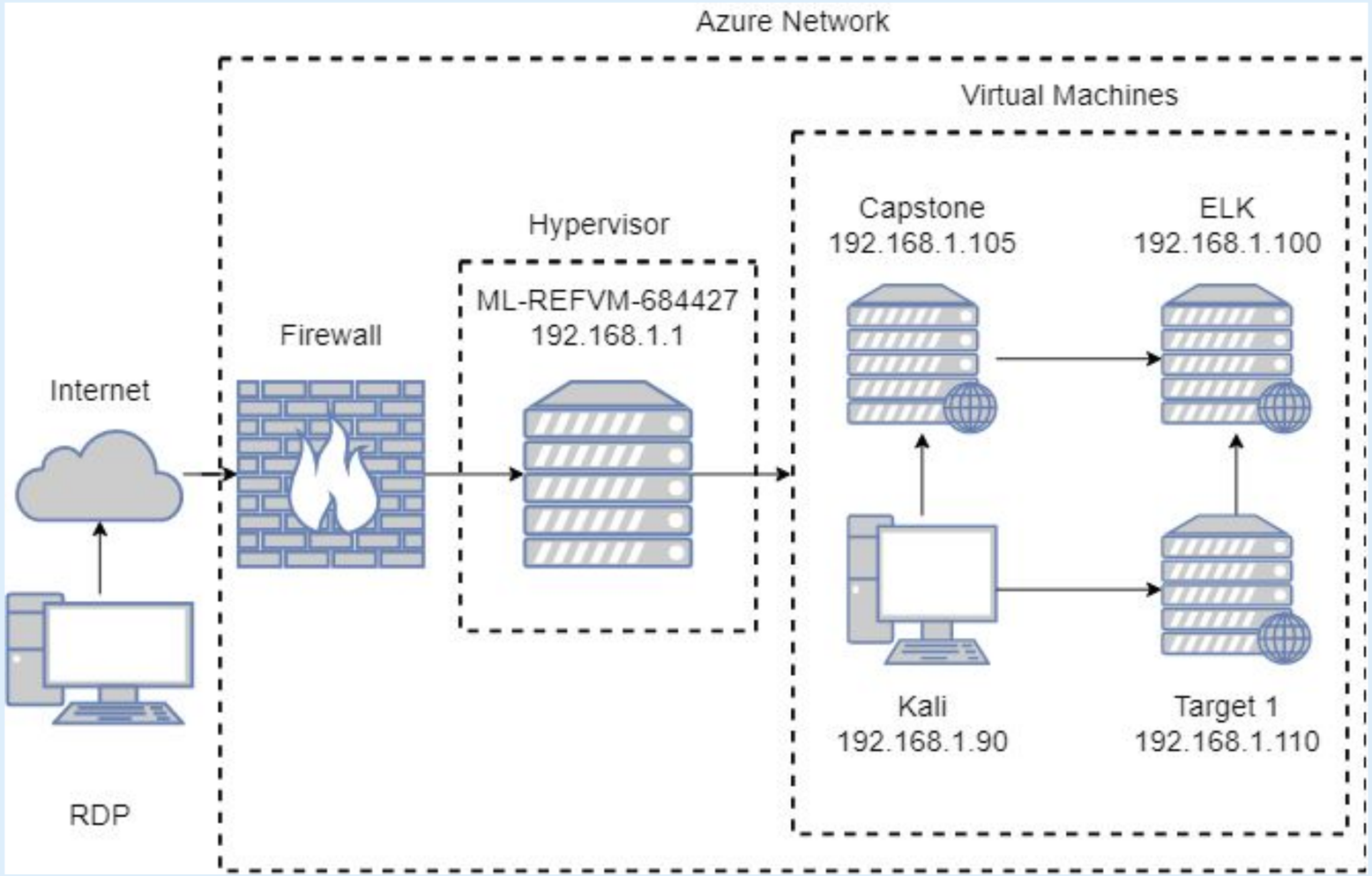
Alerts Implemented

Hardening

Implementing Patches

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.1/24
Netmask: 255.255.255.0
Gateway: 198.168.1.1

Machines
IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 198.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Insecure Password Management	Redteam obtained passwords using dictionary brute force against the webserver and used credentials to SSH - User Name: Micheal - Password: michael	This allows an attacker to gain access to protected web directories Note: Port 80 and & 22 were open
Privilege Escalation	We used Stevens sudo Python access to escalate from ‘Steven to root’	Allowed privilege escalation to root. We used sudo -1 to gain information needed to perform escalation then sudo Python access to escalate to root
Wordpress User Enumeration	We utilized enum4linux to gather user information for the web server via SSH Users: michael/ steven/vagrant	Allows attacker to extract data and gather usernames to gain access to the web server
Unprotected and Unsalted Hash	Used JohnTheRipper to brute force the hash located within the MySQL database	Password hashing add a layer of security. Hashing allows passwords to be stored in a format that can’t be reversed at any reasonable amount of time or cost for a hacker.

Alerts Implemented

Excessive HTTP Errors

Summarize the following:

- Which **metric** does this alert monitor?
  - WHEN count() GROUPED OVER top 5 ‘http_response_status_code’
- What is the **threshold** it fires at?
  - Is ABOVE 400
- Provide a screenshot of the alert in action.

Current status for 'Excessive HTTP Errors'

Execution history | Action statuses

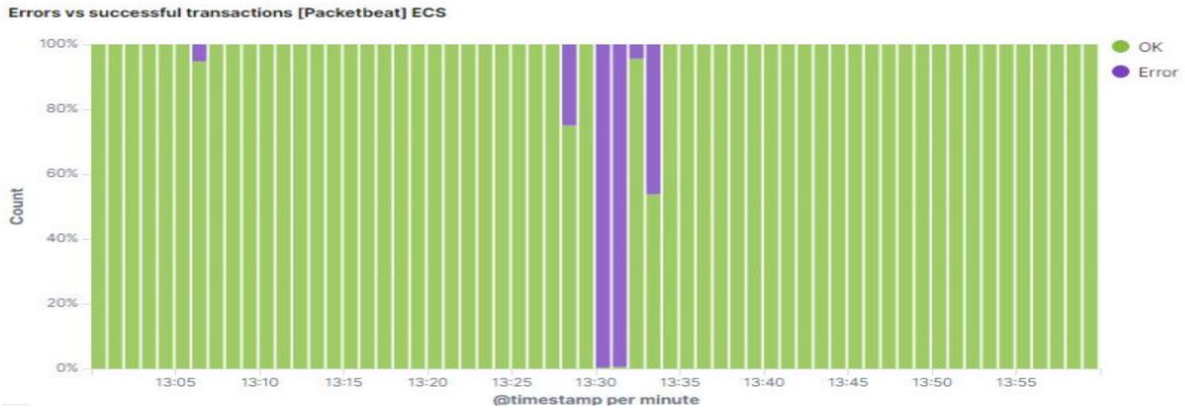
Last one hour

Trigger time	State	Comment
2021-08-11T21:57:39+00:00	✓ OK	
2021-08-11T21:56:39+00:00	✓ OK	
2021-08-11T21:55:38+00:00	✓ OK	
2021-08-11T10:16:48+00:00	✓ OK	
2021-08-11T10:15:48+00:00	✓ OK	
2021-08-11T10:14:48+00:00	✓ OK	
2021-08-11T10:13:49+00:00	✓ OK	
2021-08-11T10:12:49+00:00	✓ OK	

Excessive HTTP Errors

Summarize the following:

- This alert keeps track of any incoming HTTP error codes
- It is set to fire if more than 400 error codes are detected in the span of 5 minutes



HTTP Request Size Alert

- Metricbeat Indice: <WHEN sum( ) OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute>
- Metric alert: Monitoring HTTP Requests by size
- Threshold: Fires after any request that is over 3500 bytes in size.

HTTP Request Size Monitor

Summarize the following:

- This alert monitors the size of all incoming HTTP requests
- It is set up to fire upon receiving any HTTP request that is over 3500 bytes in size

Source IP	Destination IP	Source Bytes	Destination Bytes
192.168.1.90	192.168.1.110	5.1MB	18.5MB
192.168.1.90	192.168.1.115	89MB	177.9MB

Request Entity Too Large

The requested resource
/wp-admin/update.php
does not allow request data with POST requests, or the amount of data provided in the request exceeds the capacity limit.

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

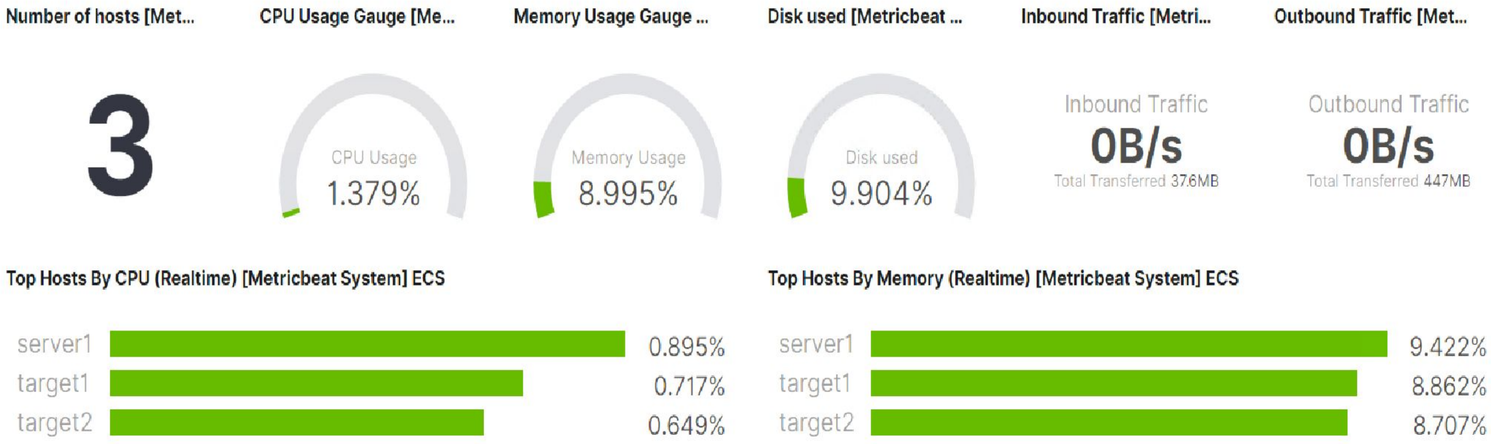
CPU Usage Monitor

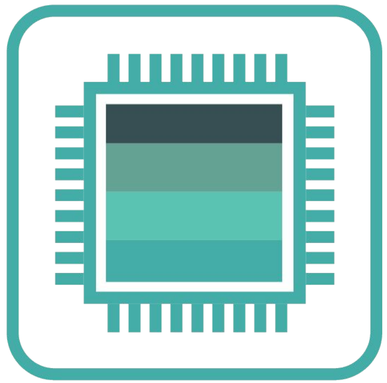
metricbeat indice:

WHEN max() OF system.process.cpu.total.pct OVER all documents

IS ABOVE 0.5 FOR THE LAST 5 minutes

- **Metric**: Notify when the total of CPU processes
- **Threshold**: are at or are above 50% usage ------ in the last 5 minutes **(ALERT)**
- **Vulnerability Mitigated**: Malicious activity or malware in the midst of an attack. A spike in CPU usage can determine if someone is performing *a brute force attack*. This alert can work in combination of other metrics to determine illicit mining activity.
- **Reliability**: It's very reliable although it can create some potential false positives from multiple services running. However, that can help recognize and maintain background processes to improve CPU usage moving forward.





Current status for 'CPU Usage Monitor'

Execution history | Action statuses

Last one hour

Trigger time	State
2021-08-11T23:17:49+00:00	✓ OK
2021-08-11T23:16:49+00:00	✓ OK
2021-08-11T23:15:49+00:00	✓ OK
2021-08-11T23:14:49+00:00	✓ OK
2021-08-11T23:13:49+00:00	✓ OK
2021-08-11T23:12:49+00:00	✓ OK
2021-08-11T23:11:49+00:00	✓ OK

Hardening

Hardening Against Brute Force Attacks

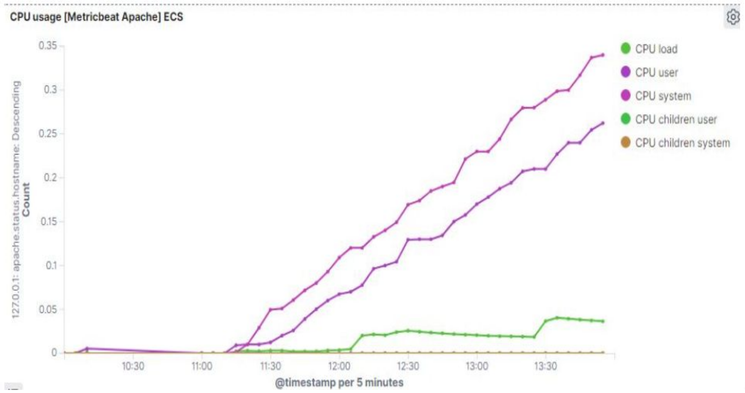
Creating a stronger password policy is a solid foundation of a hardening process. The policy will shield from brute force attacks. To implement this policy, we would access and edit the group policy, adding rules to the policy. We also will suggest the following actions:

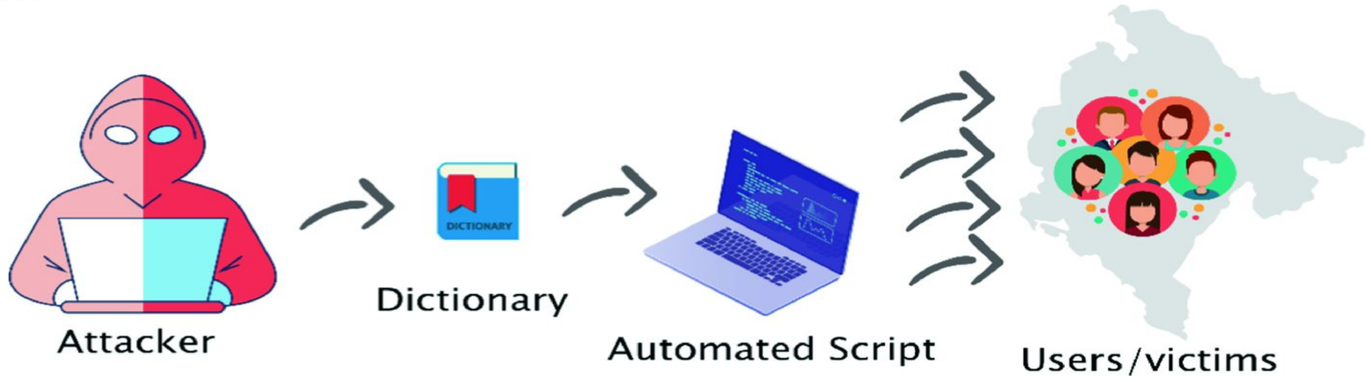
- Account Lockout &/or Inject random pauses when checking a password to delay an attack
- Use CAPTCHA
- Ask users to answer a security response upon multiple failed login attempts.
- Having a strong, strict, password.
- Locking out IP addresses with multiple failed login attempts.
- Two factor authentication

CPU Usage Monitor

Summarize the following:

- Monitors percentage of total CPU usage since last event.
- Threshold fires if 50% is used within the last 5 minutes.

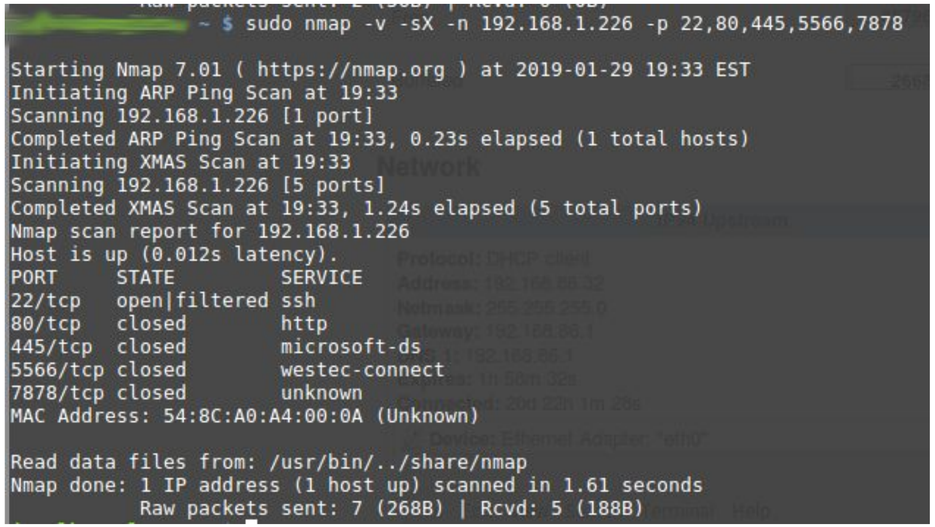




Hardening Against Port Scans

Firewalls and IP tables are effective mitigation techniques against port scans.. Splunk and Kibana offer immediate alerting of the port scans which prompts a response from the security team. **We found that Port 80 and & 22 were open!**

To harden, the following steps are implemented.



Mitigation of Port Scans

>Block/forward/delay port scans.

Create ipset lists:
<ipset create port_scanners hash:ip family inet hashsize 327 maxelem 65536 timeout 600>
<ipset create scanned_ports hash:ip, port family inet hashsize 327 maxelem 65536 timeout 60>

Create Rules:
< iptables -A INPUT -m state --state INVALID -j DROP
iptables -A INPUT -m state --state NEW -m set ! --match-set scanned_ports src, dst -m hashlimit --hashlimit-above-1/hour --hashlimit-burst 5 --hashlimit-mode srcip --hashlimit-name portscan --hashlimit-htable-expire 1000 -j SET --add-set port_scanners src --exist
iptables -A INPUT -m state --state NEW -m set --match-set port_scanners src -j DROP
iptables -A INPUT -m state --state NEW -j SET --add-set scanned_ports src, dst >
Firewall: Block all incoming and outgoing ports except for those needed
< iptables -A INPUT -p tcp -m multiport ! -dports 80, 443 -j DROP >

> Iptables and ip6tables are used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

These rules assume any incoming connection that reaches the rule must be a result of a port scan, probably because all legitimate traffic has already been accepted. If there's more than 1 scanned missed port per hour from a source ip (the hashlimit matcher), the source ip is added to a cool down penalty box - the "port_scanners" ipset.

Hardening Against Wordpress User Enumeration

WordPress is a free, open-source website platform that is vulnerable to user enumeration by default.
An attacker could utilize a variety of methods

- **Error messages** could display if an existing user login is valid during a login attempt

- **REST API** provides data & content accessible to the public

- **Permalinks** are enabled. This allows the attacker to gain usernames by browsing through the author archives.
  - `http://wordpress1.com/author/`**`johnmcafee`**`/`

- **Tools** to successfully scan, extract data & obtain user names.
  - **`enum4linux - tool for enumerating information`**
  - *`enum4linux -a 192.168.1.110`*
  - **`wpscan - checks vulnerabilities & can enumerate users`**
  - *`wpscan -url http://192.168.1.110/wordpress - ue`*

Wordpress User Enumeration Prevention

Web application firewall (WAF)

- This should be done by default.

Remove Detailed Error Messages that display valid usernames

Disable WordPress REST APIs (it's enabled by default) so that it can be inaccessible to the public.

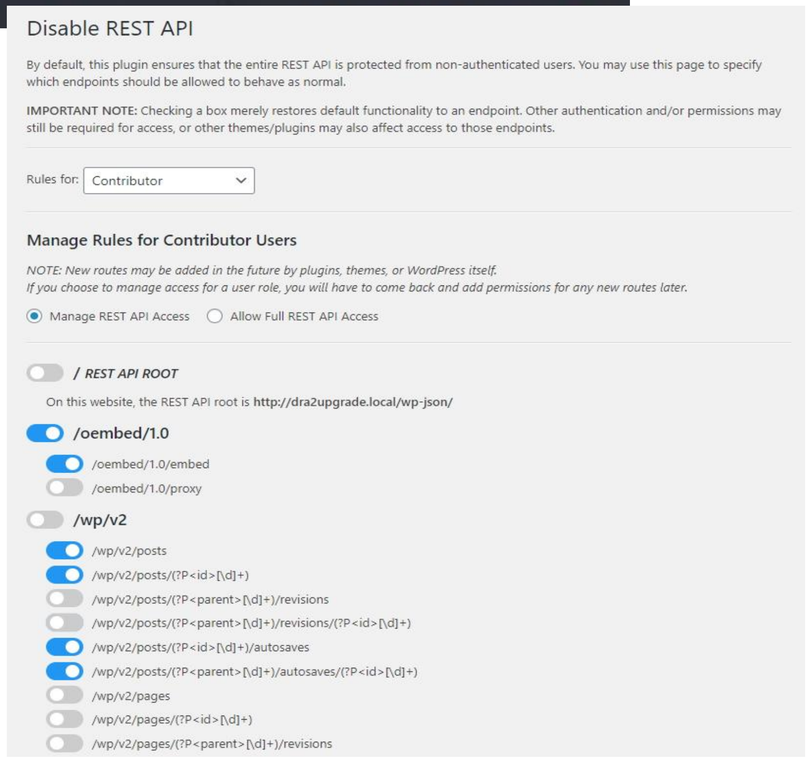
Author/user pages can be linked by iD rather than name or role ($userid);

Mask user_nicenames by changing the value to something **random**

- This can be done by configuring the database. The random value could confuse the attacker once it's displayed to them
  - 'user_nicename' => 'charliebutterfly'

Additional security methods would be to use a WordPress security plugin as many are readily available.

- https://wordpress.org/plugins/stop-user-enumeration/
- https://wordpress.org/plugins/disable-json-api/
- https://wordpress.org/plugins/edit-author-slug/
- customize permalinks by changing /author/ portion of the URL
  - default: http://example.com/author/username/
  - custom: http://example.com/cereal/chihuahua/



Implementing Patches

Implementing Patches with Ansible

Ansible for Security Infrastructure

Ansible is a configuration management tool that can be used to automate and patch systems. One can use Ansible to respond to threats and automate security solutions with modules, rules and playbooks.

Playbooks are written in YAML which allows for Human readable automation

Use Cases

- Infrastructure Provisioning
- Deploying security updates
- Installing updates on a test environment
- Identifying unknown issues
- Creating playbooks for configuration changes

```
- name: Update root password
  hosts: all
  become: yes

  vars:
     root_password: "{{ root_password }}"
     encrypt: "sha512_crypt"
     confirm: yes
     salt_size: 7

  tasks:
     - name: Update root password
       user:
          name: root
          password: "{{root_password | password_hash('sha512') }}"
          update_password: always
```

Can be scheduled to update root password every 60 days

Stay up to date and automate!

Resources:

OWASP - https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks

Common Vulnerabilities and Exposure (CVE) - https://cve.mitre.org/cve/search_cve_list.html

Palo Alto Networks - https://docs.paloaltonetworks.com/

Ansible - https://www.ansible.com/use-cases/security-automation

WordPress.org - https://wordpress.org/plugins/

Trustwave - https://www.trustwave.com/en-us/resources/security-resources/security-advisories/

Linux IP Tables -https://www.man7.org/linux/man-pages/man8/iptables.8.html

The End