Final Engagement

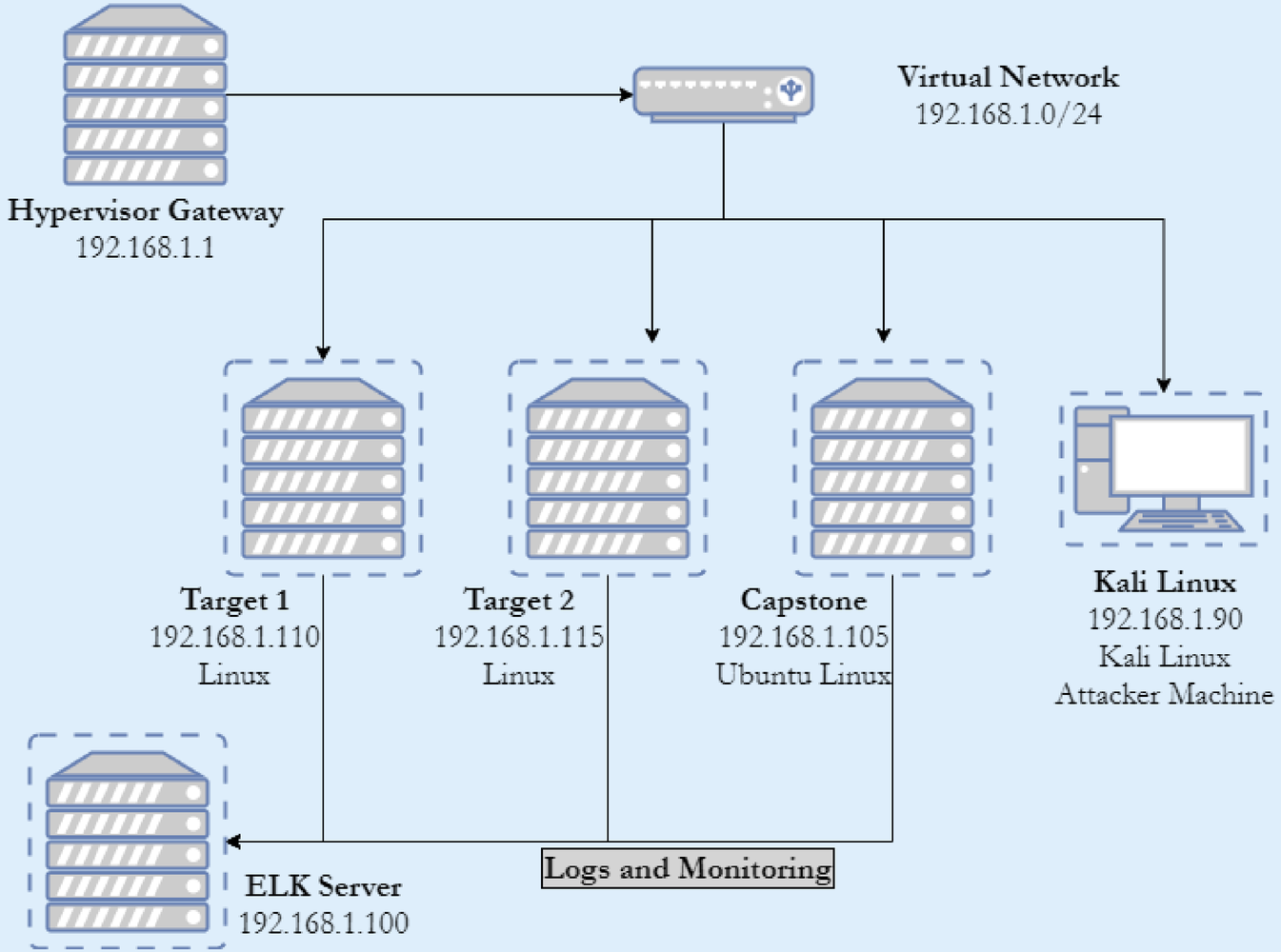
Attack of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & 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 Linux 18.04
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu Linux 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
Common web dir naming.	Having the wordpress software in a folder rather than the http root/vhost.	Allows wordpress to be more easily identified as the framework in use.
Incorrect dir permissions.	Enabling a user other than the owner/webserver to access dirs.	Allows users to view or modify files/folders containing sensitive information
SUID bit set for python.	Allows users to run python and therefore code as root	Privilege escalation to root.
Root database user.	Using the root user access for a web app rather than a specific credentials.	Malicious user/application access to entire database.

Exploits Used

Exploitation: Common web dir naming

- Tool: dirb
- Outcome: Located the /wordpress/ directory on the target, identifying the web framework in use.

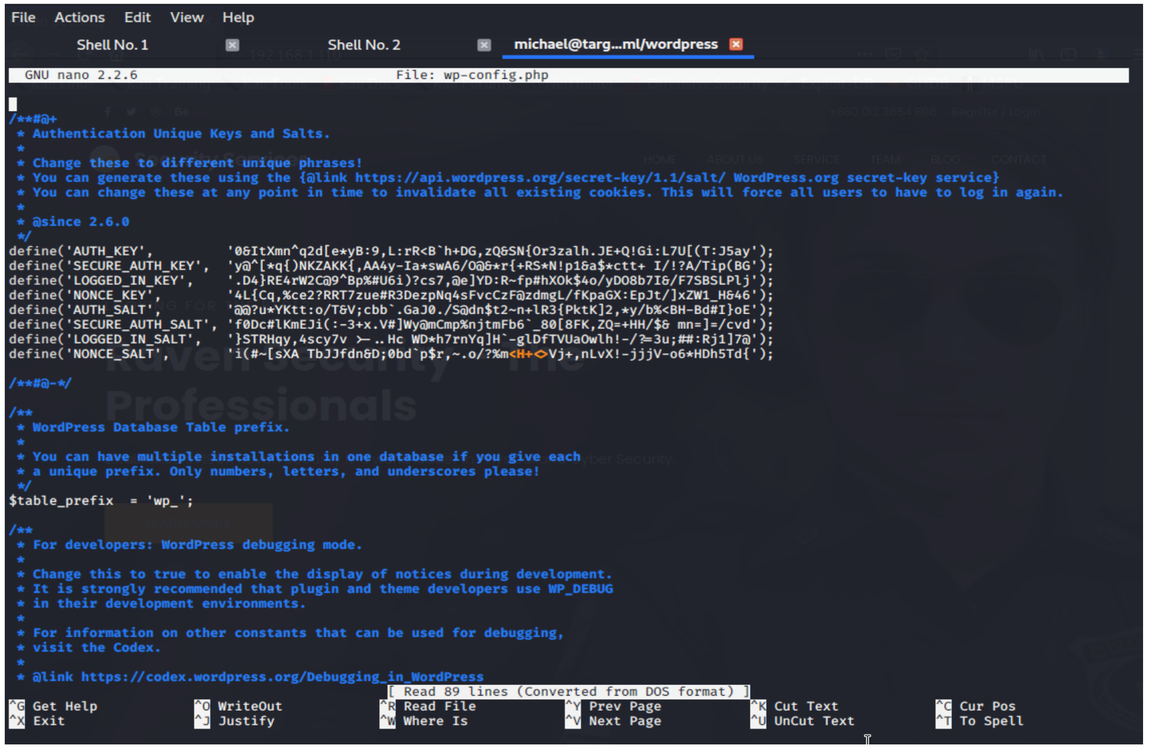
```
--> Testing: http://192.168.1.110/word

--> Testing: http://192.168.1.110/wordpress

==> DIRECTORY: http://192.168.1.110/wordpress/
```

Exploitation: Incorrect dir permissions

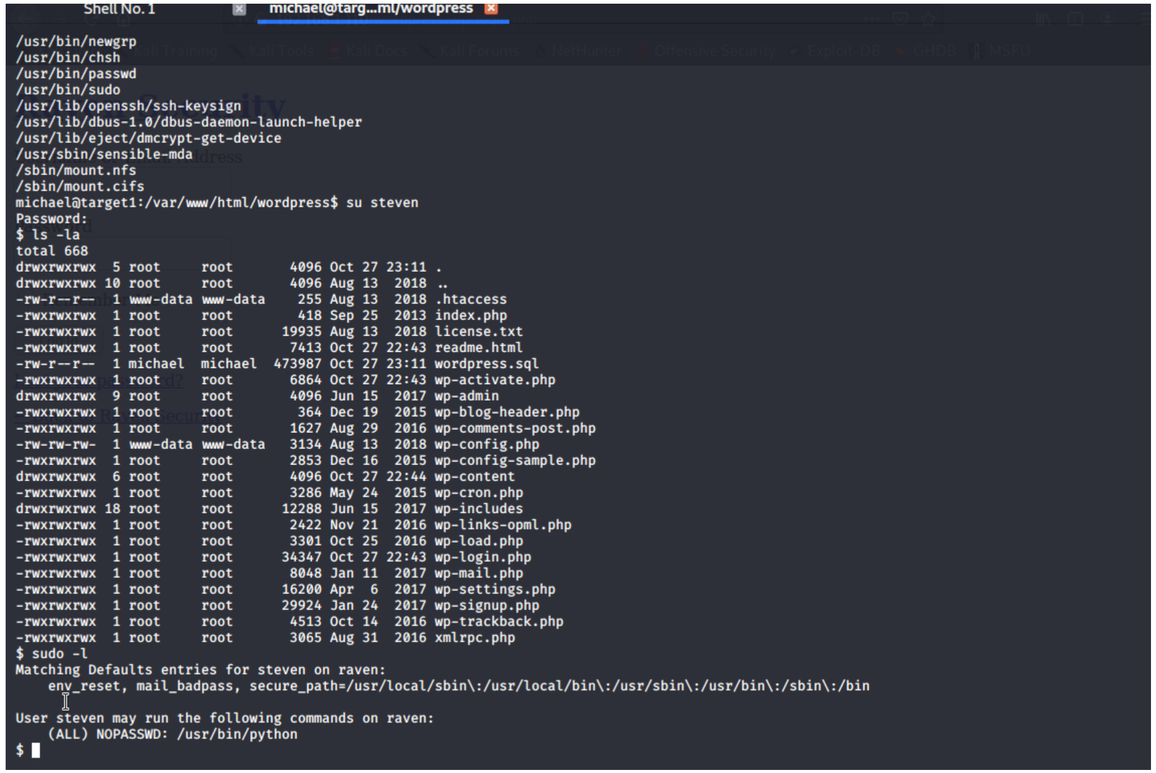
- Tool: ls -la
- Outcome: The wordpress directory permissions were incorrectly set allowing all users to read the configuration file for the wordpress installation which contains sensitive information including database credentials in plantext .



Exploitation: SUID bit set for python

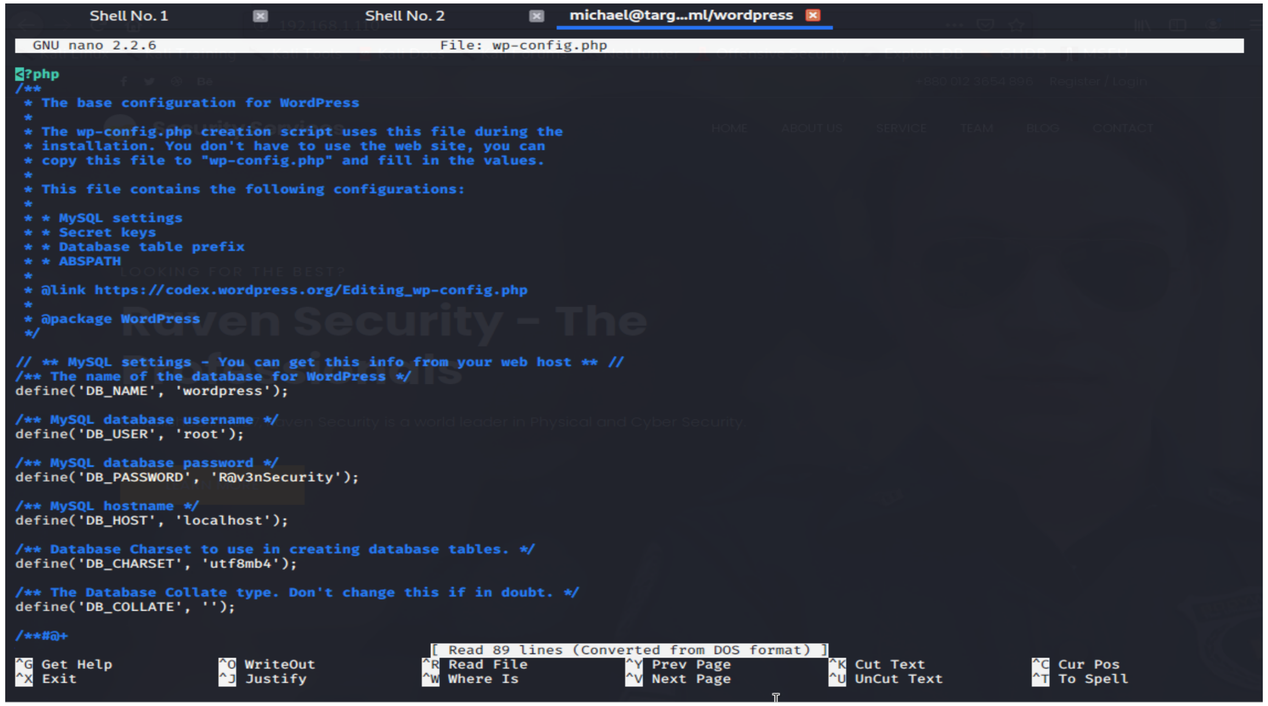
Summarize the following:

- Tool: sudo -l
- Outcome: Displays applications the user is able to run under sudo privileges, in this case python



Exploitation: Root database user

- Tool: mysql
- Outcome: By exposing the root user credentials, we were able to dump all the records from the database including user credentials for other areas.



Exploitation Mitigations

Exploitation of Common web directory naming

Mitigation

- Set monitoring alerts for above normal numbers 404 errors.
- Set monitoring alerts for above normal file requests/minute.
- Consider a virtual host or installing to the web root.

Exploitation of Incorrect directory permissions

Mitigation

- Utilise the least permissions practice.
- Create a user or group for specific access.
- If possible, do not store plain text credentials. Nb in the case of wordpress and many PHP webapps this is just not possible.

Exploitation of SUID bit set for python

Mitigation

- Utilise the least permissions practice.
- Harden filesystem access preventing SUID use.

Exploitation of Root database user

Mitigation

- Ensure use of a highly secure root password.
- Give each webapp it’s own database credentials and grant access only to it’s individual database or tables.
- If possible do not store credentials in plain text.