# ENPM685 – Security Tools for Information Security

## Section: 0101

## Incident Report

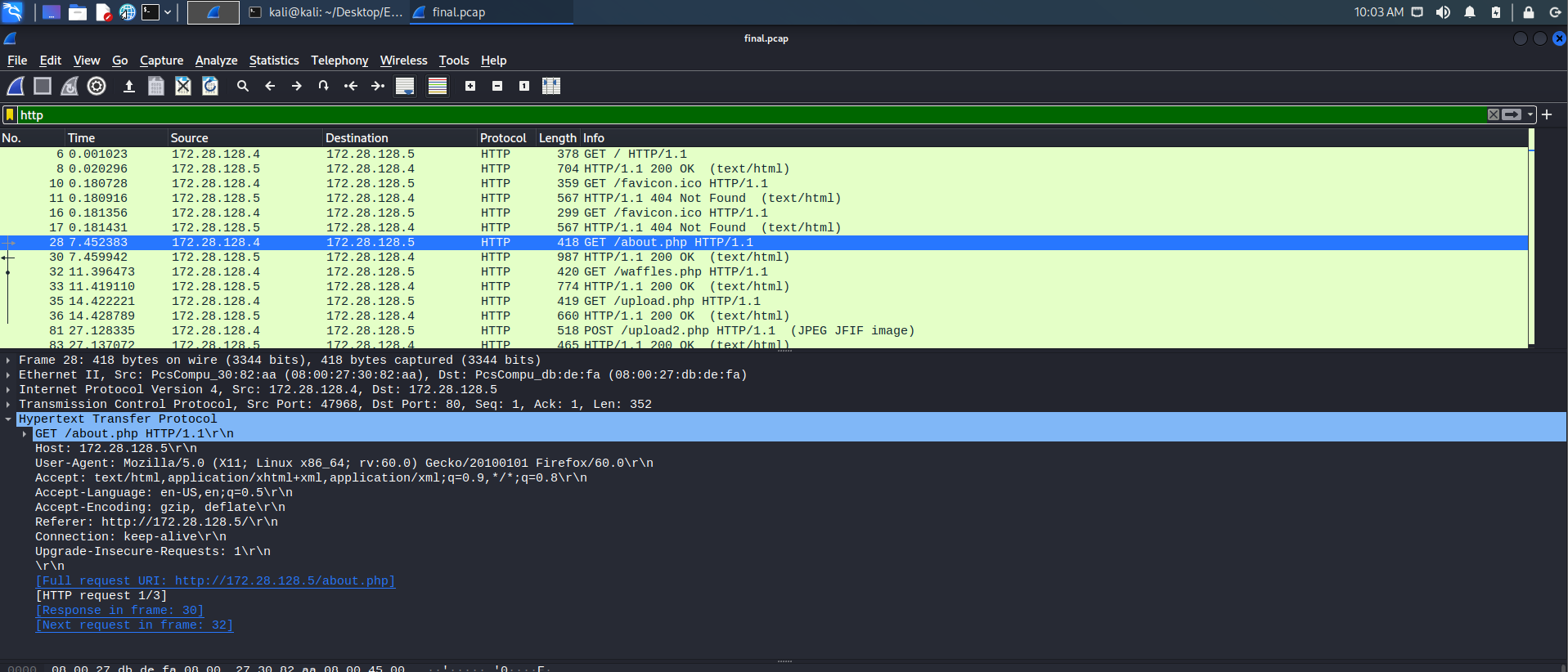
Name: Syed Mohammad Ibrahim

UID: *iamibi*

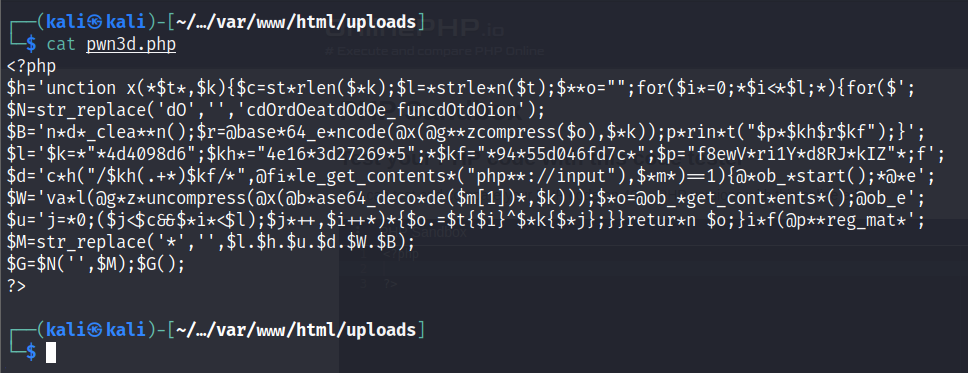
UID Number: 118428369

## Attack Narrative

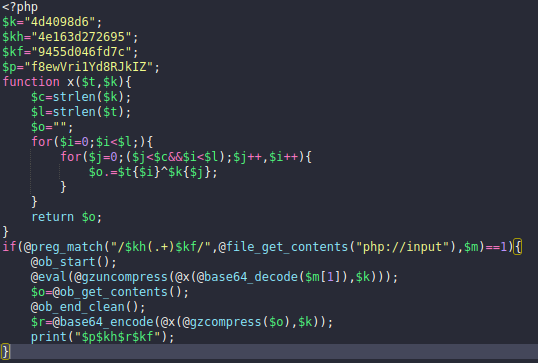
1. The IP address of the attacker is 172.28.128.4 and the compromised box IP address is 172.28.128.5. The incident took place on **February 14th between 16:41 to 16:47 UTC** time.
2. The attacker landed on the homepage of the ENPM685 Waffle Co. website. The **homepage** (index.php) contained a commented information which stated that the current website is a dev site with a work in progress. And Julia will be working on the DBA part. This was mentioned by Nathan – developer.
3. The attacker then navigated to the **about** (about.php) page from the homepage. It did mention that there is a user with the name Julia in their development team whose position is DBA.



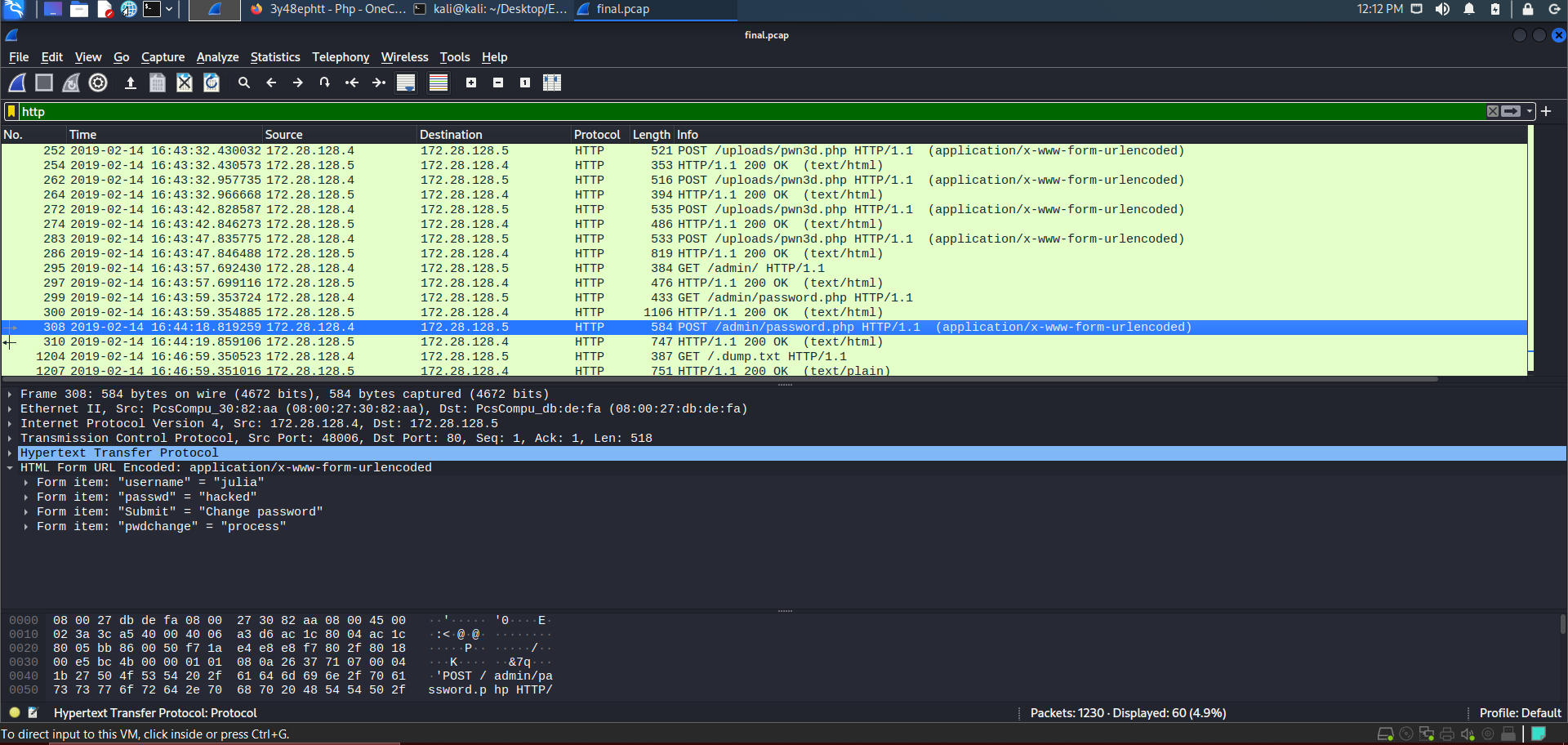
1. The attacker then proceeded to go to the waffles page and then to the competition’s page (upload.php) where anyone can upload an image file. The attacker tested the mechanism by uploading an image and then performing a GET request for that image.
2. After confirming this, the attacker uploaded a malicious php file **pwn3.php** which contained the obfuscated code for the shell commands. The server, after uploading the file ran it, which printed the output of it to standard output and was returned as part of the response.



1. After analyzing the **pwn3.php**, it was inferred that the malicious file was performing Remote Code Execution because of a server weakness, specifically **CWE-434**[1]. Because of this weakness, the attacker was able to upload a dangerous file and the server never restricted or validated it.



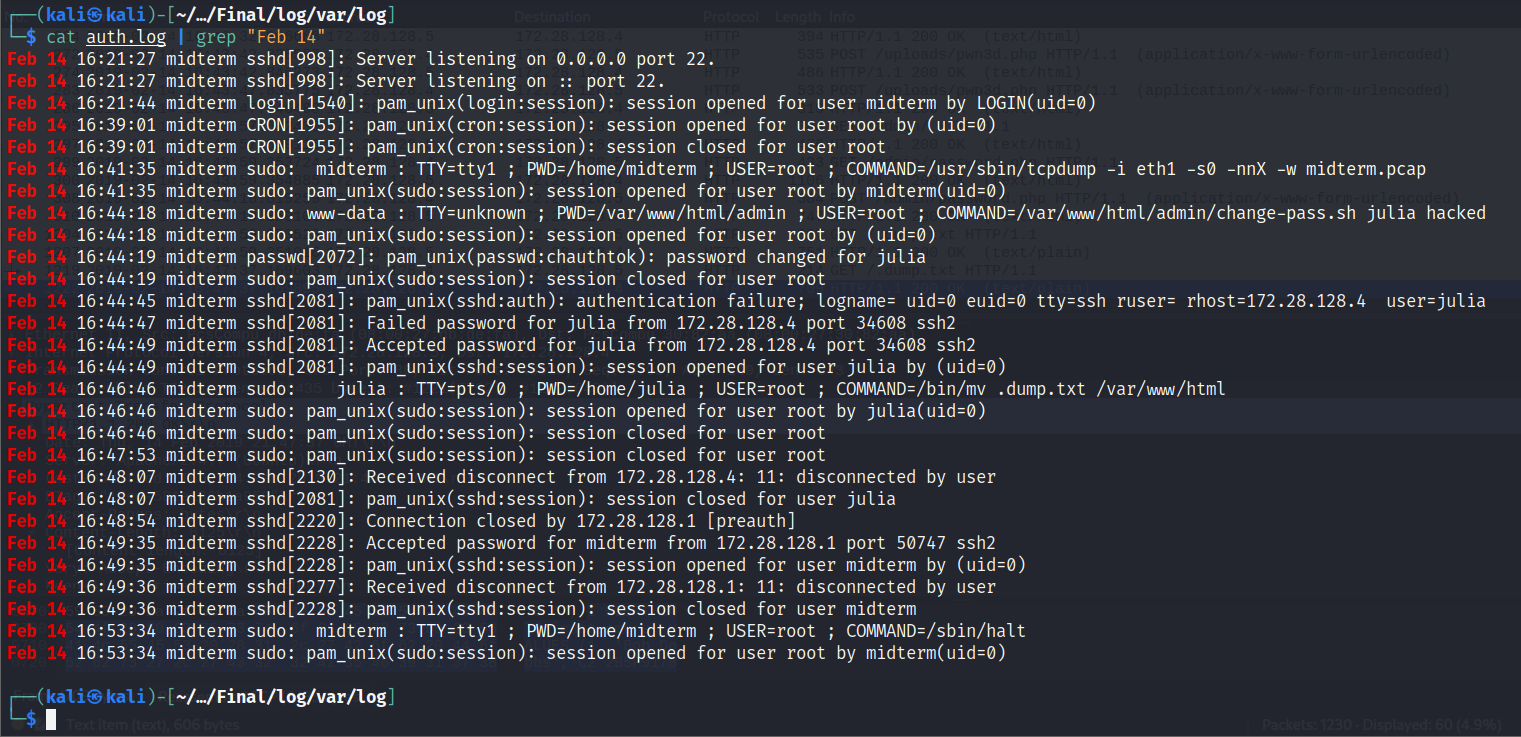
1. The network log then shows that the request was made multiple times by the attacker, passing different data with every request, and using **pwn3.php** code as base to perform command execution. Few commands involved are echo, chdir, pwd, get\_hostname, etc.
2. The attacker was able to traverse the whole system box and was able to figure out the website structure.
3. They found the admin path, which was accessible via the website, with the change password functionality which doesn’t perform any additional validation on the data that was entered or by whom it was entered (**CWE-20**[2]).



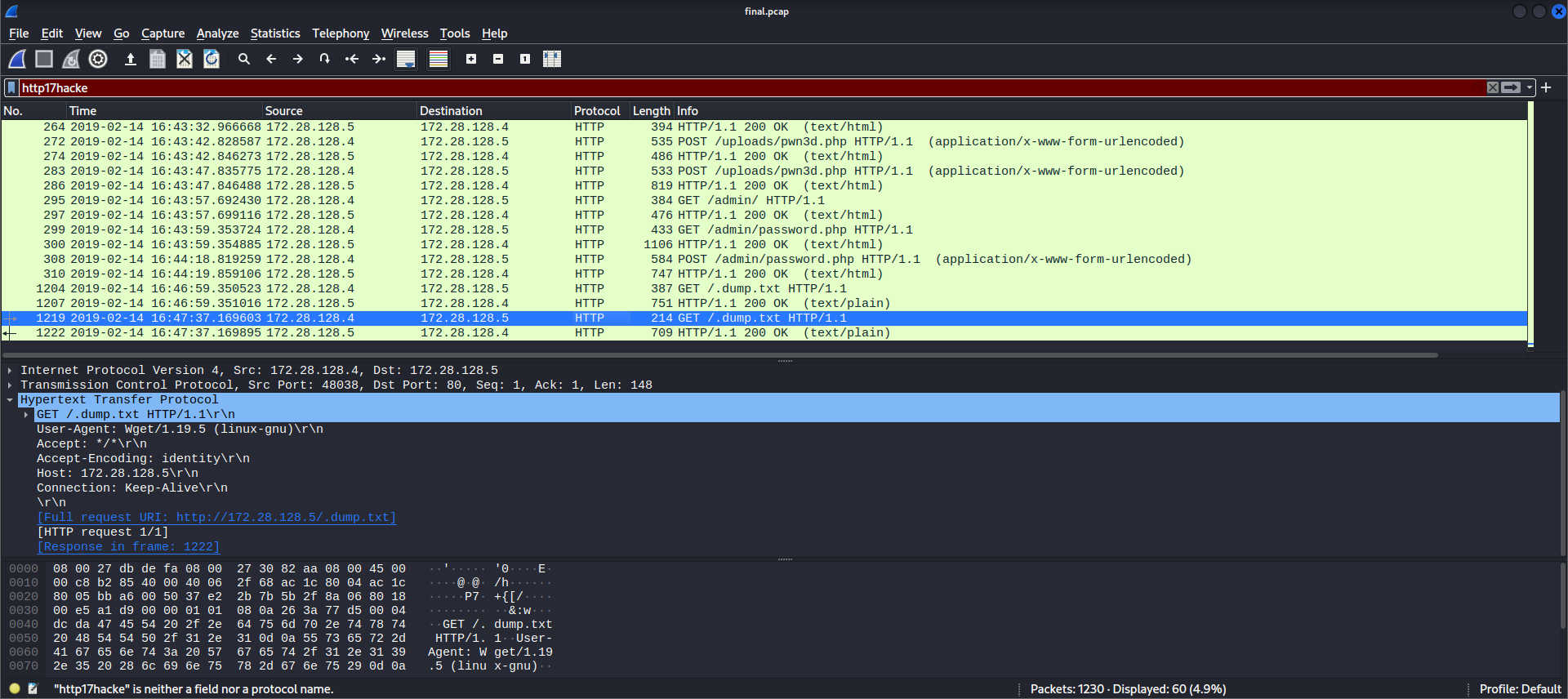
1. Attacker then entered the username as **julia** and password as **hacked** which made the user julia’s password to be changed.
2. This was done with the intention to gain access to their account and execute commands locally.
3. The attacker the opened a **Secure Shell** (SSH) in the compromised host box and went through the contents of the said compromised account. They found a **.dump.txt** file which was a MySQL database file containing various commands that are mandatory for setting up the **ENPM685 Waffle Co.** website’s database.
4. The file also contained information about the schema of the database and information that was supposed to be inserted into those respective databases. One of the database insertions involved few users’ information to be inserted which was PII (Personally Identifiable Information) with only password and their credit card number’s being hashed (**CWE-200**[3]). Other information like Name, Contact Number and Credit Card Expiry date were present in plaintext (**CWE-311**[4]).



1. The original user, Julia, may have put it there with the intention of setting up the database or may have forgotten to clean it up after the database was setup.
2. The bash history of the user doesn’t display much information apart from the passwd command that was executed to reset the password initially. The attacker may have cleared this information from the history. However, there was a move (**/bin/mv**) command that was executed by the attacker which moved the **.dump.txt** file from the home directory to **/var/www/html** directory – which is the root to the website. This was recorded as part of the **auth.log** logs.



1. Once the file was in the website’s home directory, the attacker used their browser to verify whether it was accessible or not. After which, they downloaded the file using WGET command on their system, thus causing a leak.



## Questions

* How did the attacker get in?
  + The attacker gained control of the weakness of the website that it didn’t validate or restricted any unintended file. Through which, the attacker was able to reset the password of a user named julia and was successful in getting inside the machine using SSH.
* What did the attacker do once they were on the system?
  + The attacker copied the MySQL dump file which contained sensitive information from the user’s home folder to the website’s root folder. This was done to fetch the file from their system using a GET command.
* Was sensitive data accessed? How can you tell if it was/was not accessed?
  + The sensitive data was contained in .dump.txt file. It had a few PII data and hashed passwords and credit card numbers (which were hashed using SHA-1 and can be un-hashed easily). The file was moved from the home directory of the compromised user to the website’s root folder and then it was fetched using the WGET by the attacker on their system. The network entry confirms that.
* Were you able to learn anything about the attacker? (What were their attack tools, tactics, techniques, and procedures?)
  + The attacker has used OSINT to gather information about the website and the people developing it.
  + They knew how to generate an obfuscated PHP code to execute shell commands.
  + They used WGET command line utility to fetch the dump file from the server.
  + They were aware of Linux operating system structure and the website structure once they were inside the system.
  + They knew how to SSH into the remote system.
  + They were aware of the bash\_history and thus, cleared it up.
  + They may know what type of hashes were stored on the dump.txt file and can potentially use tools like Hashcat to break them.

## References

[1] CWE – 434: <https://cwe.mitre.org/data/definitions/434.html>

[2] CWE – 20: <https://cwe.mitre.org/data/definitions/20.html>

[3] CWE – 200: <https://cwe.mitre.org/data/definitions/200.html>

[4] CWE – 311: <https://cwe.mitre.org/data/definitions/311.html>