Credential Stuffing   
(Green)

Graphical user interface, text, application

Description automatically generated

Summary:

Target Address: 192.168.168.161 Port: 9030

Target Account: Brailee.ogden@windomain.local

System Target: PHP, Piwigo

Techniques: Hydra, SSH reverse shell, PHP Backdoor, Base64.

MITTRE ATTACK FRAME Explains:

Large numbers of username and password pairs are dumped online when a website or service is compromised and the user account credentials accessed. The information may be useful to an adversary attempting to compromise accounts by taking advantage of the tendency for users to use the same passwords across personal and business accounts.

However \*Credential stuffing is a risky option because it could cause numerous authentication failures and account lockouts, depending on the organization's login failure policies.

Ssh reverse shall to pivot our connections to access to the server, as well as to hide our connections.

Text

Description automatically generated

By modifying the firefox’s profile setting, we can have separate profiles for our proxy to hide and access to the right port.

Graphical user interface, text, application

Description automatically generated

We are attacking the webserver 192.168.168.161. We can right click to see the web code and analyze the code login username and password for properly align with the payload.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

We are using the credential list that we created from previous attack as payload.

Text

Description automatically generated

We are using Hydra tool to implement the list to see the matches of USER as username, and PASS as password.

As we can see, we found a match of [brailee.odgen@windomain](mailto:brailee.odgen@windomain) and its password.

Text

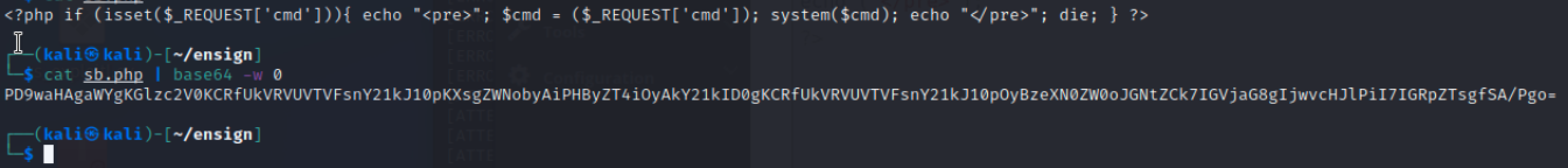
Description automatically generated

After successfully login, we can see Piwigo. Piwigo has vulnerability and it can active plugins in order to put our backdoor.

Graphical user interface, application, website

Description automatically generated

We created a sb.php as website backdoor and encode it as base64 to hide the code.



We copy the base64 encode backdoor as credential to put it in the Piwigo localFiles Editor and decode with base64 to establish backdoor.

Graphical user interface, application, Word

Description automatically generated

As we have the use the backdoor of the server using “sb.php” and it can execute the command after ?cmd=

In this case, we execute “ls” command and it executes successfully

Graphical user interface, text, application

Description automatically generated

In this case, we execute “whoami” command and it executes successfully

And the backdoor is officially established.

Graphical user interface, application, website

Description automatically generated