4/13 Operation Lateral

After Action Report

Team Texas

Overview:

On March 31st, we began the initial attack on our targets systems. Out of 28 of the systems attacked, 21 were found to be infected with the vulnerability and allowed us to successfully exfiltrate sensitive data from their systems. On all infected systems, we were also able to install persistence we can use for future attacks, all of which will be discussed in this after actions report.

Initial Access:

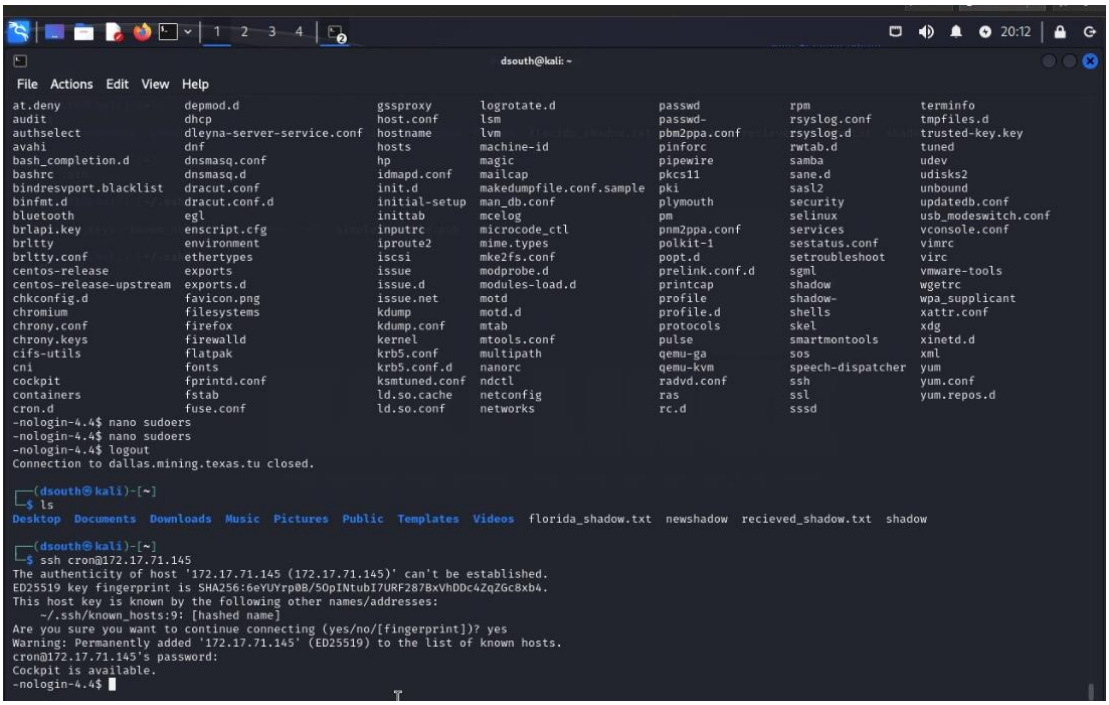
We began by attempting to log into the persistence accounts cron and webclient. The Table below shows all systems we were able to obtain access to:

Team	IP	Successful
California	172.17.71.56	Yes
California	172.17.71.59	Yes
California	172.17.71.93	Yes
California	172.17.71.114	Yes
California	172.17.71.145	Yes
California	172.17.159.7	Yes
California	172.17.159.197	Yes

Team	IP	Successful
Illinois	172.21.11.139	No
Illinois	172.21.43.153	No
Illinois	172.21.155.95	Yes
Illinois	172.21.155.134	Yes
Illinois	172.21.190.38	No
Illinois	172.21.190.51	Yes
Illinois	172.21.190.52	Yes
Illinois	172.21.190.144	Yes
Illinois	172.21.190.146	Yes
Illinois	172.21.190.206	No

Team	IP	Successful
Florida	172.19.229.17	Yes
Florida	172.19.229.79	Yes
Florida	172.19.229.162	Yes
Florida	172.19.229.229	Yes
Florida	172.19.229.248	Yes
Florida	172.19.203.122	No
Florida	172.19.203.34	Yes
Florida	172.19.114.138	No
Florida	172.19.114.13	No
Florida	172.19.101.29	Yes
Florida	172.19.114.4	No

During our attempts, we noticed many teams had patched the webclient account, but had not patched the cron account, making it our primary account for logging in. After successfully logging in with cron, we were greeted with a nologin shell, in which we can use to interact with the system.



Escalation:

After successfully logging in, our priority was giving the account cron full administrative privileges. While the account cannot use sudo, the account had misconfigurations to allow it to update any file on the system with nano. Abusing this, we were able to write to the file sudoers to give the account "cron" the same administrative privileges as root. Because cron was only located in shadow, sometimes sudo would be unable to reference its password (or it pointed to the account ftp for a password for sudo privileges), requiring us to add "NOPASSWD" to the access as shown below:

```
dsouth@kali: ~
File  Actions  Edit  View  Help
  GNU nano 2.9.8                    /etc/sudoers                    Modified

Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults   env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
cron    ALL=(ALL)       ALL
ftp     ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
#%wheel ALL=(ALL)       ALL

## Same thing without a password
%wheel  ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the

^G Get Help   ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text   M-] To Bracket
^X Exit       ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text   M-W WhereIs Next
```

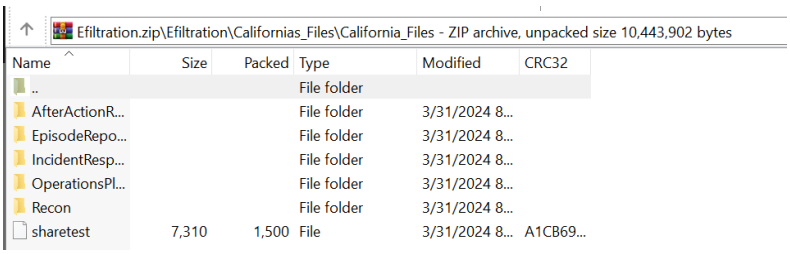
After this change, we now had full administrative privileges on the targeted system.

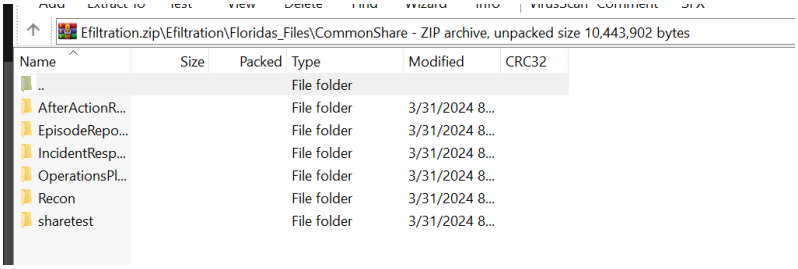
```
-nologin-4.4$ ls
24562493-1de6-4190-b5dd-ff603768303f  943d287c-328d-4cf1-914e-5bb7079dc8d0  Desktop    Downloads         Music     Public     Videos
48b5635b-13e5-4b66-a79d-bb0e62008641  a7a03641-b552-479e-bf06-6f5f6aa6a3b2  Documents  examples.desktop  Pictures  Templates
-nologin-4.4$ sudo nano /root/which.sh
-nologin-4.4$ ls /root
ls: cannot open directory '/root': Permission denied
-nologin-4.4$ sudo ls /root
find-malwarex  which.sh
-nologin-4.4$
```

Execution:

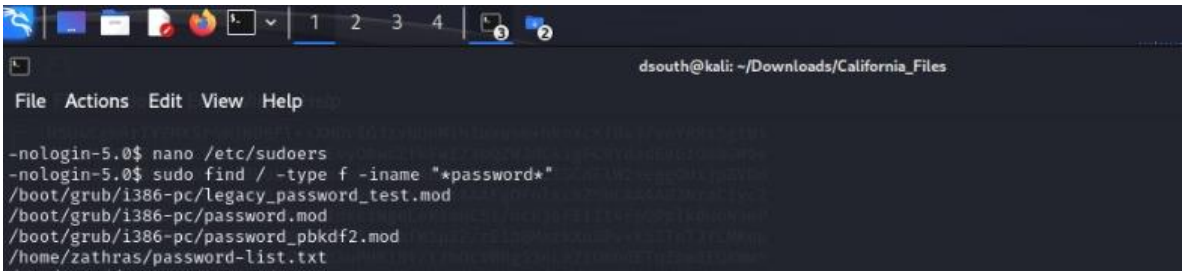
The first systems we targeted were their Samba systems, as it was a target we were sure, if successfully accessed, we could exfiltrate sensitive data from each team. We were successfully able to target each team's CommonShare File and exfiltrate all contents inside with SFTP, giving us access to each team's offense reports, incident response reports, episode reports, and any updated reconnaissance performed by each team.

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
AfterActionR...			File folder	3/31/2024 9...	
EpisodeRepo...			File folder	3/31/2024 9...	
IncidentResp...			File folder	3/31/2024 9...	
OperationsPl...			File folder	3/31/2024 9...	
Recon			File folder	3/31/2024 9...	
removeDirAn...	281	194	Shell Script	3/31/2024 9...	6B86DF...
sharetest	50	50	File	3/31/2024 9...	709D6...
sshInstructio...	311	172	Text Document	3/31/2024 9...	A86B34...



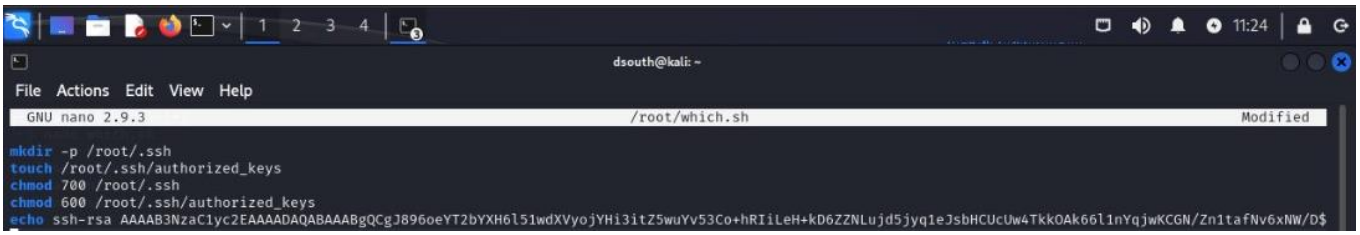


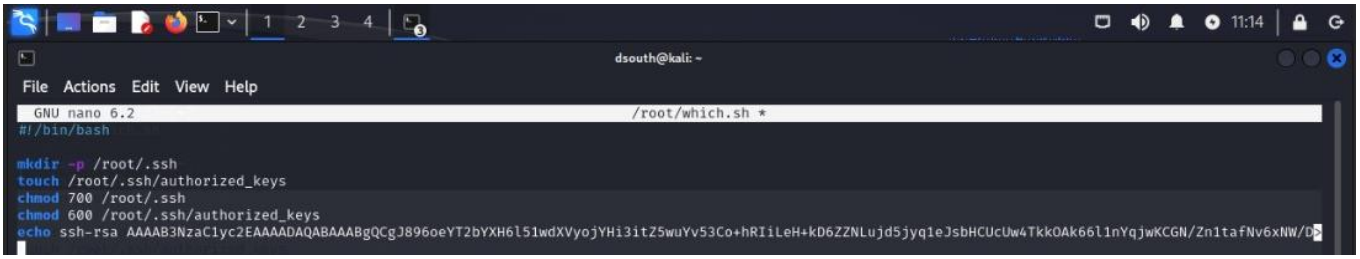
We then used search on each system for any files labeled “password”, “nagios”, and “exercise” using the find command, in the attempt of finding any sensitive files containing usernames and passwords. Doing so, we were able to exfiltrate data believed to be all the usernames and passwords in plain text for all their system users.



Persistence:

Persistence was kept simple due to the number of systems we attacked. A list of randomly selected administrative users were found, recorded, and public keys were placed inside their authorized_keys accounts. Public keys were also planted inside of root, along with a bash script called in .bashrc to replace the public keys upon login.





All public keys associated with user accounts have been stored in our exfiltration file labeled “important.txt”, that correlates which users contain our public keys for which systems

Evasion:

After performing the attack on each system, log files were deleted locally for the appropriate system (syslog, messages, audit, etc.). From our research, cron itself does not contain any files that will keep logs of commands run on the system. We had also checked the logs for the root user and discovered they also do not appear there when running sudo. While we understand that this data will be sent to graylog, due to the nature of the attack, the goal was to overwhelm and compromise as many systems as possible, needing the victims to remove every single trace of the attack across over 30 systems.

After Events/Future Plans:

Overall, we believe the attack to be a major success. This attack relied heavily on the persistence of a back door left by Red Team, without any knowledge that the attack will succeed on each system. To our surprise, every system we attacked was found to be vulnerable, allowing us to exfiltrate data and install persistence on a wide range of systems.

The biggest success from this attack would be Californias plain text usernames and passwords. Since the core password is shared between every system, we simply need to crack the prefix for each system (which is only 3 character) and we potentially have access to every account on all their systems (unless they change the passwords). With this information, we can potentially install keyloggers on their user specific accounts on every system and determine their Nagios credentials.

Another noteworthy file would be Floridas offense report, in which they discuss using a keylogger to attack Californias system. This information would save us the trouble of developing our own key logger and allows us to piggy back off of their attacks.

We also exfiltrated a very well documented updated reconnaissance report from team California. We compared their reconnaissance to our own file to determine authenticity and plan to update

sections that contained any extra details onto our own (ie. Any missed systems or missed information about the systems).

A secondary goal of this attack was to also get insight on which teams were attacking which systems. Unfortunately, at the time off the attack, only team Florida had submitted an operations plan. While we plan to attempt another exfiltration in the future to get insight on attacks against us, if this attack was performed again, we would make the change of establishing a server to constantly exfiltrate sensitive data from their systems, including (but not limited to), their Samba Common Share files.