

# **Global Information Assurance Certification Paper**

# Copyright SANS Institute Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

# Interested in learning more?

Check out the list of upcoming events offering "Network Monitoring and Threat Detection In-Depth (Security 503)" at http://www.giac.org/registration/gcia

## Understanding Mobile Device Wi-Fi Traffic Analysis

GIAC (GCIA) Gold Certification

Author: Erik J. Choron, erik.j.choron.mil@mail.mil Advisor: Adam Kliarsky Accepted: March 19, 2018

**Template Version September 2017** 

#### Abstract

Mobile devices have become more than just a portable vehicle to place phone calls in locations previously deprived of traditional phone service. In addition to versatile phone service, mobile devices include the capability of utilizing the internet through the Mobile Internet Protocol (IP). This can cause a problem whenever a device is roaming through different points of the cellular network. The IP handoff that takes place during the transfer between cellular towers can result in a degraded performance which can possibly impede traffic analysis. A thorough understanding of Wi-Fi traffic and Mobile IP technology could benefit network and system administrators and defenders by heightening awareness in a field that is surpassing more commonly understood technology.

## 1. Introduction

According to the Pew Research Center, smart phone ownership has over doubled among adults in the United States from 35% in 2011 to 77% in 2018 (Mobile phone ownership, 2017). Cell phones present not only a means to place phone calls, but a method of connecting and completing tasks on the internet as well. Due to their convenience, more users are employing mobile devices to fulfill their needs while laptop and desktop ownership has remained stagnant over the last nine years (Anderson, 2015). These devices have been seen to create a sense of independence from wired connections and allow use in areas otherwise not possible on standard computers.

Mobile IP is a communications standard that was developed to allow for flexible mobility (Perkins, 2010). While a cellular device might be joined to a wireless network at a central location, current day devices allow a high level of freedom by not restricting the user's ability to travel almost limitlessly. This is made possible through the cellular signal on the device which allows Internet Protocol (IP) addressing to still occur and connections to be made.

An unnoticeable set back to the user is that Mobile IP can sometimes cause Outof-Sequence packets within the IP flow (Troubleshooting with WireShark: Locate the Source of Performance Problems, 2017). Often times, a change between networks or cellular towers whenever a mobile device is moving between networks, presents no performance issues. Connectivity is still maintained for the user, and applications will still load when called upon. Behind the scenes, there are some packets that are not in the right place.

Out-of-Sequence packets are the result of a TCP packet being transmitted at the wrong time, although it has a proper sequence number (Chappell, p. 174, 2014). While the other packets are transmitting normally, a change in cellular towers or Wi-Fi hotspot could cause the delay in transmitting a packet. Since TCP packets are sequentially numbered, a delay in connection or changing networks can result in a packet out of order.





Figure 1 illustrates a simple Out-of-Sequence packet example. An event such as driving down the road and changing cell phone towers while an email was being sent is an example where this could occur.

During a military cyber evaluation exercise that took place in March 2017, known attackers attempted a Denial of Service (DoS) attack using crafted packets within an unclassified Army network. These packets were recorded into a PCAP file using WireShark and showed to have Out-of-Sequence numbers being sent. While the attack was prevented, it was revealed after the exercise that this simulated a previously recorded attack through a mobile device accessing network resources via an undisclosed hotspot. This event highlighted the importance of implementing a narrower focus on Out-of-Sequence packets and better identifying attack vectors concerning crafted packets and their potential for malicious injection.

### 1.1. Security Implications

Out-of-Sequence packets have a variety of security implications and can be used as a vulnerability exploit against almost any system. For example, Juniper Networks explains in one of their vulnerability reports the risks of Out-of-Sequence packets (TCP Out-of-Sequence Denial of Service Vulnerability, 2013). Within their systems, Out-of-Sequence packets are stored in a buffer until all packets in the sequence have been

accounted for and can be delivered. However, depending on the size of the buffer, only a specific amount of data can be stored without causing a buffer overflow.

Another security issue involves spoofing traffic to trigger a DoS. Attackers can simulate Mobile IP traffic to replicate data and make it appear as if it is coming from a mobile device that is in the process of switching providers or locations. Through this, a potential DoS attack could take place (Conn, 2001). In order to protect services and safeguard against possible threats that could damage networks and services, firewalls could be put in place to prevent Out-of-Sequence packets. This can inadvertently cause trouble if a legitimate device is trying to access a resource. While a system administrator is taking the proper steps towards resource protection, genuine users could possibly be denied access while on mobile devices.

The possibility also exists that delayed transmission of a packet due to default Mobile IP functionality could result in the same TCP packet being retransmitted until a proper ACK has been received. While this would not be an attack, the characteristics are present, and an improper firewall denial could take place.

## 2. Transferred Traffic Analysis

In order to properly identify and establish baseline traffic characteristics, a method of analysis would need to be established. The most assured way to verify traffic is to create a testing environment that is mostly controlled and monitored (Serral-Garcia, Jakab, and Domingo-Pascual, 2006). First, a network administrator would need to set up a wire analysis tool capable of reading and capturing inbound traffic. This could be accomplished by setting up Wireshark on a connection to a test server that is accessible to public IP addresses. Along with this step, a user would need to be online with a mobile device and moving between cellular and Wi-Fi access points while sending traffic to the test site.

Second, after a predetermined amount of time and traffic, review of the captured traffic would need to be completed in order to build a proper baseline around the occasional TCP packet retransmission. Configuration changes, if needed, would likely be required at the firewall level of security on the network first. Depending on the amount

of buffer usage on the test server, configuration changes would be needed in case the buffer storage is limited and not capable of handling multiple retransmitted packets.

## 2.1. Testing Environment

For this analysis, three different scenarios were established using the predetermined equipment.<sup>1</sup> For the packet analysis, a virtual machine was setup hosting an FTP server with Wireshark running in promiscuous mode. The FTP server received the files being transmitted while Wireshark collected the traffic for analysis. The host machine was used as an FTP client to test in one of the scenarios. A laptop was used as the mobile delivery with an iPhone tethered for internet connection only using cellular data. A 250MB compressed file comprised of office documents containing multiple lines of the number zero was transmitted.

#### 2.1.1. Scenario 1

In this scenario, the virtual machine ran the FTP server and WireShark, while the host machine connected via an FTP client to send the compressed file.

C Transfer-PhysicalToVirtual.pcap [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]	
Ele Edit View Go Capture Analyze Statistics Telephony Iools Internals Help	
Filter: tcp.analysis.out_of_order	
No. Time Source Destination Protocol Length Info	
78241 98.2254850 192.168.0.1 192.168.0.8 TCP 60 [TCP out-of-order] 1900-57214 [FIN, ACK] seq=39 Ack=226 win=16384 Len=	0
Hrame 78241: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)	
Ethernet II, Src: f8:da:0c:f5:eb:91 (f8:da:0c:f5:eb:91), Dst: 00:23:54:94:1a:85 (00:23:54:94:1a:85)	
Internet Protocol Version 4, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.8 (192.168.0.8)	
⊞ Transmission Control Protocol, Src Port: 1900 (1900), Dst Port: 57214 (57214), Seq: 39, Ack: 226, Len: 0	
0000       00 23 54 94 1a 85 f8 da 0c f5 eb 91 08 00 45 00       .#TE.         0010       00 28 69 12 00 00 40 06       90 64 c0 a8 00 01 c0 a8       .(i€d         0020       00 80 07 6c df 7e 98 0c c4 90 43 44 4c c1 50 11 <e.< td="">         0030       40 00 1a ec 00 00 00 00 00 00       00 00 00      </e.<>	
🔗 🕅 File: "C:\Documents and Settings\Administrator\   Packets: 78280 - Disolayed: 1 (0.0%) - Load time: 0:01.281	

Figure 2 Wireshark displaying the out-of-sequence packet in Scenario 1

1. All scenarios were captured using Wireshark. The link to the folder containing the PCAP output is: https://drive.google.com/open?id=1a\_hpjjRTJMibqZsprwb0sSOslA\_-jtqF

The test was to set up a baseline of understanding and to verify packet collection. During the least successful transfer from physical to virtual machine, only one packet of was outof-sequence and three packets were retransmitted.

#### 2.1.2. Scenario 2

With the virtual machine still running and Wireshark reset for another capture, the same compressed file was sent from the laptop that was tethered from the iPhone using only cellular data.

Transfer1-NoMovement.pcan [Wireshark	1.12.0 (v1.12.0-0-e4fa	h41a from master-1	12)]			
File Edit View Go Capture Analyze Statistics	s Telephony Tools Internal	s Help				
	4 4 5 T 4 1		9 F7   24 F7 🐔	\$2   <b>178</b>		
				974°   808		
Filter:	~	Expression Clear	Apply Save			
Vo. Time Source	Destination	Protocol Length	Info	Con 1 Act 1221021 Line	64340 1 00 0	
1419 19.8326510 192.168.0.13	100.137.104.124	TCP 54	TCP Spurious Pet	Seq=1 ACK=1321921 Win	=64240 Len=0	
1421 19.9099810 192.168.0.13	166,137,104,124	TCP 54	TCP DUD ACK 1419	11 1105+64230 ACK	Sep=1 Ack=1321921 win=64240 Len=0	
1422 19.9101540 166.137.104.124	192.168.0.13	FTP-DAT 1424	[TCP Spurious Ret	ransmission] FTP Data	: 1370 bytes	
1423 19.9101810 192.168.0.13	166.137.104.124	TCP 54	[TCP Dup ACK 1419	#2] 1105+64230 [ACK]	5eq=1 Ack=1321921 win=64240 Len=0	
1424 19.9102260 166.137.104.124	192.168.0.13	FTP-DAT 1424	[TCP Spurious Ret	ransmission] FTP Data	: 1370 bytes	
1425 19.9102520 192.168.0.13	166.137.104.124	TCP 54	[TCP Dup ACK 1419	#3] 1105+64230 [ACK] :	5eq=1 Ack=1321921 win=64240 Len=0	
1426 19.9102950 166.137.104.124	192.168.0.13	FTP-DAT 1424	[TCP Spurious Ret	ransmission] FTP Data	: 1370 bytes	
1427 19.9103200 192.168.0.13	166.137.104.124	TCP 54	TCP DUD ACK 1419	4] 1105-64230 [ACK]	5eq=1 Ack=1321921 Win=64240 Len=0	
1428 19.9104430 166.137.104.124	192.168.0.13	FTP-DAT 1424	TCP Spurious Ret	ransmission] FTP Data	: 1370 bytes	
1429 19,9104090 192,108,0.13	100.137.104.124	TCP 04	TCP DUD ACK 1419	#5] 1103+04230 [ACK]	5ed=1 Ack=1321921 W1H=04240 LeH=0	
1421 10 0108720103 168 0 12	166 127 104 104	TCD 54	TCP Spur Tous Ret	Farismissioni FIP Data	: 1370 Dytes	
1432 19 9109430 166 137 104 124	192 168 0 13	ETP_041 1424	FTCP Sourious Per	ransmission] ETP Data	1370 bytes	
1433 19 9109720 192 168 0 13	166 137 104 124	T/TP 54	TCP DUD ACK 1410	27 1105-64230 [ACK]	Seg=1 Ack=1321921 Win=64240 Len=0	
1434 19,9110160 166,137,104,124	192,168,0,13	ETP-DAT 1424	TCP Sourious Ret	ransmission] FTP Data	: 1370 bytes	
1435 19,9110450 192,168,0,13	166,137,104,124	TCP 54	TCP DUD ACK 1419	#8] 1105+64230 [ACK]	5eg=1 Ack=1321921 win=64240 Len=0	
1436 19.9110880 166.137.104.124	192.168.0.13	FTP-DAT 1424	[TCP Spurious Ret	ransmission] FTP Data	: 1370 bytes	
orgene porticer . v	107 107 104 104	255 54	Faters Aver 1841-1415			
[SEQ/ACK analysis]						
[iRTT: 0.070696000 seconds]						
[Bytes in flight: 1370]						
[TCP Analysis Flags]				2		
[Expert Info (Note/Sequenc	e): This frame is a	(suspected) spu	rious retransmiss	10n]		
[This frame is a (suspec	ted) spurious retran	smissionj				
[Severity level: Note]						
Expert Info (Note/Sequence)	a). This frame is a	(suspected) ret	ransmission]			
This frame is a (suspec	ted) retransmission]	(suspecced) rec	and an			
[Severity level: Note]						
[Group: Sequence]						
FTP Data (1370 bytes data)						
0000 00 0c 29 23 9d e3 f8 da 0c -	F5 eb 91 08 00 45 18		- Marine			
0010 05 82 70 c3 00 00 72 06 02	e0 a6 89 68 7c c0 a8	pr.	.h]			
0020 00 0d fa e6 04 51 45 f2 d9 1	9c 33 73 b1 40 50 10	QE3	s.@P.		/	
0030 40 00 35 45 00 00 allec 30 . 0040 1c 3e b3 7a 8a ad cc 0f 73 .	23 76 f0 8a 0f 19 d6 09 03 dc 33 3a ff 66	9.5E 0#{			🤳 Found New Hardware	×
0050 2c 05 09 20 30 do so f2 77	6 7F 99 4d 17 57 6d		M		Your new hardware is installed and ready to us	e.
🍠 🎦 Frame (frame), 1424 bytes	Packets: 253327 • Displayed	: 253327 (100.0%) · Dro	opped: 0 (0.0%) · Load time:	0:00.000		1

Figure 3 Wireshark displaying the out-of-sequence packet in Scenario 2

This scenario tested the connection and reliability of using cellular data to transfer a file across the internet and packet capturing for analysis. The laptop remained stationary during the scenario and produced more out-of-sequence and retransmitted packets than the first scenario. Two packets were out-of-sequence, and fifty-seven packets were retransmitted.

### 2.1.3. Scenario 3

For the last scenario, the laptop was still tethered to the iPhone but was introduced to being mobile. After starting the file transfer, a total of fifteen miles were driven to ensure that the iPhone had to change cellular towers.

## Understanding Mobile Device Wi-Fi Traffic Analysis | 7

Transfer2-KempnerAndBack.pcapng [Wire	eshark 1.12.0 (v1.12.0-0-g4fab41a	from master-1.12)]	
File Edit View Go Capture Analyze Statistics	Telephony Tools Internals Help		
Filter: tcp.analysis.retransmission	Expression	Clear Apply Save	
No. Time Source	Destination Protocol L	ength Info	14
970 25.1562300 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Retransmission] FTP Data: 1370 bytes	
1609 25.9561380 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Retransmission] FTP Data: 1370 bytes	
8565 38.5479020 192.168.0.1	192.168.0.8 TCP	60 [TCP Spurious Retransmission] 80-56955 [FIN, ACK] Seq=546 Ack	=631 W1n=16384 Len=0
8583 38.5600150 192.168.0.1	192.168.0.8 TCP	bU [TCP Spurious Retransmission] 80-56956 [FIN, ACK] Seq=561 ACK	=639 W1n=16384 Len=0
14051 45 7604160 166 137 106 65	102.168.0.13 FTP-DAT	1424 [ICP RETENTISSION] FIP Data: 1570 Dytes	
20720 55 0915020 166 127 106 65	102 169 0 12 FTP-DAT	1424 [TCP Retrainstitution] FIP Data: 1570 bytes	
20722 55 0885960 166 137 106 65	192.108.0.13 FTP-DAT	1424 [ICF opurious Recrammission] FIP Data: 1570 bytes	
21506 56 3710540 166 137 106 65	192 168 0 13 ETP-DAT	1424 [TCP Sourious Retransmission] ETP Data: 1370 bytes	
21508 56 3712280 166 137 106 65	192.168.0.13 ETP-DAT	1424 [TCP Sourious Retransmission] FTP Data: 1370 bytes	
21510 56, 3790750 166, 137, 106, 65	192.168.0.13 FTP-DAT	1337 [TCP Spurious Retransmission] FTP Data: 1283 bytes	
21512 56.3792990 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
21514 56.3797620166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
21516 56.3800670 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
21518 56.3809770 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
21520 56.3815690 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
21522 56.3816510 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
21524 56.3819650 166.137.106.65	192.168.0.13 FTP-DAT	1424 [TCP Spurious Retransmission] FTP Data: 1370 bytes	
⊞ Frame 970: 1424 bytes on wire (11	392 bits), 1424 bytes capture	ed (11392 bits) on interface 0	
Ethernet II, Src: f8:da:Oc:f5:eb:	91 (f8:da:Oc:f5:eb:91), Dst:	00:0c:29:23:9d:e3 (00:0c:29:23:9d:e3)	
Internet Protocol Version 4, Src:	166.137.106.65 (166.137.106	.65), Dst: 192.168.0.13 (192.168.0.13)	
Transmission Control Protocol, Sr	c Port: 10038 (10038), Dst P	ort: 1117 (1117), Seq: 827951, Ack: 1, Len: 1370	
Source Port: 10038 (10038)			
Destination Port: 1117 (1117)			
[Stream index: 6]			_
[TCP Segment Len: 13/0]			
Sequence number: 82/951 (rei	ative sequence number)		
LNext sequence number: 829321	(relative sequence number)		
Acknowledgment number: 1 (re	factive ack number)		
E 0000 0001 0000 - Elage: 0x	010 (ACK)		
Window size value: 16384	OTO (ACK)		×
0000 00 0c 29 23 9d e3 f8 da 0c f	5 eb 91 08 00 45 18)#	E.	
0010 05 82 41 b2 00 00 72 06 30 2	c a6 89 6a 41 c0 a8	. 0,jA	
0020 00 0d 27 36 04 5d 61 44 32 4	6 c2 41 4d ab 50 10'6.]	D 2F.AM.P.	
0040 29 0e 58 59 39 c7 97 5f 55 3	R e2 4h 76 72 ac 2f ).xy9	18. Kur. /	
0050 24 02 20 11 20 05 08 00 74 0	he 70 df 20 hs no 0	- 1	M
💛 🌁 File: "C:\Documents and Settings\Administrator\	Packets: 265944 · Displayed: 165 (0.1%) ·	Dropped: 0 (0.0%) · Load time: 0:04.843	Profile: Default

Figure 4 Wireshark displaying the out-of-sequence packet in Scenario 3

Overall, sixty-two packets were out-of-sequence and 165 packets were retransmitted, providing the worst results out of all three scenarios.



Figure 5 Route traveled for Scenario 3

During travel of the route, the iPhone made three cellular tower changes while maintaining data connections to the internet over Long-Term Evolution (LTE). This

allowed for the compressed file as a whole to be transmitted with no interruptions visible to the user on either end. But the packet analysis reflects that several packets had difficulties in arriving at the destination.

## 3. Applicable Use of Identifying Out-of-Sequence Packets

All methods of tracing cell phone usage have pros and cons. On the plus side, IP traffic can be measured through traceroute, based on Time to Live (TTL) values, with decent reliability. The last hop can be verified against known IP addresses of cell phone towers. Although cell phone towers primarily offer communication for Global System for Mobile Communications (GSM) signals, they also have to be able to route TCP traffic since newer models of cell phones can access internet-related resources. Global Positioning System (GPS) signaling is dependent on having a decently clear path to satellites and can often times be inaccurate due to signal search (Moore, 2016). To compensate for the lack of tracking, TCP tracing can fill the gap. However, in order to effectively rely on this method, the investigator will need certain key points of information in order to verify the findings. Each cell phone tower has a layer 3 router as part of its basic components (Anthony, 2013). Since each cell phone tower has a routable IP address to relay traffic to the rest of the internet, the IP, if seen in the last hop of a traceroute, can be used as a verifiable location. The downside of this method is that the investigator will need the IPs of the routers and what tower they reside in. This information will have to be provided by the cell carrier.

With complex internet-based attacks developing rapidly, this information would benefit law enforcement agencies with a way to fill in the gaps of tracking effectively. Consider the following example.

a brank der de hotes an wire 1732 Brezh, We hotes saminted 2732 Brezh	
- BOD_LD replie Milanapitar	
- more \$53.12 how Bairs, #lags:	
- Gapting of the supercept	
a program dividual and an an a second s	
- Department of the second of the April 19814 (19814), but April 1982 (1981), one of the of	
inserve dants, balant filling	
ment supplicant dent ( 1000 ( 1000))	
Annual Solution (St.	
This impact that II	
instants makes i it instants commits instants	
and the second second of the second	
and the second sec	
a second s	
and the second sec	
and a second statement of the second statement statement statement and	
and the second sec	
the set of	
Contraction and The point and the	
and body and - Address address and and	
ALL ALL BLO T PROP ME AND ALL ALL	
A DATE OF ALL AND A DESCRIPTION OF A DES	
	and the second
· Dispert 200+ 204/ Separation prover the weight the report 2040/ never port 19402	
Decomposition and all the composition of the second	
[aeverity tevet) (thet]	
(arms) (esumo))	
<ul> <li>Solar solar solar + File well and</li> </ul>	
uninger size verse data	
Deforfaces whose sizes with the	
a chapter based (reflection dramfed)	
ingent pointer: #	
o personal test management and a second part of the second person and a second person of the second person person	the second second second
The set and set all	
They is defined a second (	
The subscript Plant	
- Transfer and a labor transfer with the set of a laborational and of some research	
a sum of the state	

Figure 6 Out-of-Sequence Packet Wireshark capture

Figure 6 displays an Out-of-Sequence event captured by Wireshark. Packets were crafted on an iPhone and sent to a virtual test server hosted locally. Using the information in the Wireshark capture, an investigator can trace back to the IP address the packet came from.

```
Tracing route to gar27.dlstx.ip.att.net [12.123.16.117] over a maximum of 30 hops:
```

1	4	ms	4	ms	4	ms	Linksys01116 [192.168.1.1]
2	5	ms	6	ms	5	ms	192.168.0.1
3	19	ms	39	ms	16	ms	
4	202	ms	30	ms	32	ms	xe-0-0.cpcvtx1601h.texas.rr.com [24.26.192.21]
5	22	ms	18	ms	18	ms	be21.wacotxjb01r.texas.rr.com [24.175.62.84]
6	21	ms	24	ms	26	ms	agg24.dllatx1301r.texas.rr.com [24.175.62.234]
7	32	ms	27	ms	25	ms	bu-ether14.dllstx976iw.bcr00.tbone.rr.com [66.109.9.88]
8	21	ms	21	ms	24	ms	unk-426d0579.adelphiacom.net [66.109.5.121]
9	16	ms	15	ms	29	ms	dls-b21-link.telia.net [62.115.156.208]
10	26	ms	245	ms	278	ms	gar27.dlstx.ip.att.net [12.123.16.117]

#### Figure 7 Windows tracert

Figure 7 shows the results of a tracert command given in Windows in response to the Out-of-Sequence packet. The IPs following hop number ten were left out in order to protect personal and testing equipment information. In order to verify the results were accurate, a tracert command was issued from the iPhone to the Google's public DNS server.

### Understanding Mobile Device Wi-Fi Traffic Analysis | 10

14 AT&T LTE 22:42	L D C 30% ED
(	•
Traceroute - 8.8.8.	8
001 172.26.96.161 (014.29 mg	
002 172 10 22 106 (26 85 mg	
003 7	
004 12.83 179 194 (79.97 ms	
005 gar27.distx.ip.att.net	
006 ?	
007 7	
008 108 170.236 222 (87 55 ms)	
009 216-239.57.9 (48.96 mg	
010 google-public-dns-a.google.co	om 🛄

#### Figure 8 iPhone traceroute

The results of an iPhone trace route in Figure 8 show routes to the public DNS server hosted by Google. This was the same iPhone that sent the crafted Out-of-Sequence packet to the virtual test server. In this trace route, hop five is the public IP address of the local cell phone tower the iPhone is routing through.

#### 3.1. Attack Vector

From an attacker's perspective, any number of methods could be used to get an unsuspecting user to accept an incoming transmission that is malicious or a pathway to something malicious. Regardless of how the TCP data stream is established from the attacker to the user, a sequence number-based attack can open up individual users, and make entire networks vulnerable. One of the first known attacks against utilizing sequence numbers was in 1999 (Qian & Mao, 2012). During Qian and Mao's report, they documented an attack called Sequence Number Interference, where the third part of the TCP handshake either does not complete or is delayed. This could allow for an attacker to utilize a sequence number to infiltrate the data stream.

An attacker using either of the scenarios tested in this research could also utilize Wireshark on the sending end to identify sequence numbers that did not transmit correctly and craft packets similar to the original Out-of-Sequence or retransmitted packet. Robbie Myers states that an attack can be crafted using the RST flag on a TCP packet causing a premature end to a connection (Myers). The quickest method an attacker can utilize is a Denial of Service (DoS) method. Making this attack more potent is the

possibility that this DoS can be just as effective as a data flood and overburden a network device. Simply crafting packets with the RST flag set could trigger the DoS if crafted properly. There are many tools such as Hping and coding languages like Python that can achieve this.

🕑 Ostinato		- 🗆 X
File View Help	🌼 Edit Stream [Test] ? 🗙	
Ports and Streams		8
M. B. Det Crews 0: [137.0.0.1]	Protocol Selection Protocol Data Variable Fields Stream Control Packet View	
<ul> <li>Port Group 0: [127.0.0.1];</li> <li>Port 0: if0 (Realtek PC</li> </ul>	Media Access Protocol	Apply
	Ethernet II	
	Internet Protocol ver 4	
	Transmission Control Protocol (state less)	
	Override Source Port     10038     Override Checksum     78 07       Override Destination Port     1117     Urgent Pointer     0	
	Sequence Number 19936661	
Port Statistics	Acknowledgement Number	8
1011010100	Override Header Length (x4) 5	
Transmit 💽 💿 Stats 🔜	Window 1024	(8)
P Link State Transmit State Caoture State Frames Received Frame Sent Atate (fos) Frame Receive Rate (fos) Bytes Received Bytes Sent Bytes Sent Atate (Bos)	Note: Ostinato is stateless- it cannot establish a TCP connection and generate sec/ack numbers accordingly	
Port Statistics Stream Statistics	Prev Next OK Cancel	

Figure 9 Screenshot of Ostinato

Figure 9 references a recently developed software (Ostinato) to craft packets. The ability to craft specialized packets, regardless of purpose, is becoming easier and more accessible to malicious perpetrators.

## 4. Conclusion

In a test environment, tracing IP addresses can be easily accomplished. Normally, users will have firewalls while more complex networks will have intrusion protection of some kind. Having a good defense strategy can be effective against a formidable advisory. It should be understood though that critical pieces of information need to be obtained and verified before relying on logs and trace routing. One of the most critical pieces needed are the IP addresses of the cellular towers. Although normal users might not be able to obtain this, law enforcement agencies would be able to through programs

such as the Law Enforcement Information Sharing Program Exchange Specification (Law Enforcement Information Sharing, 2017).

This is an area of digital forensics that is still in its infancy and continues to be developed. With cellular and mobile devices relying more on IP communication, the possibly of attacks will increase from more diverse platforms. But, taking into .ck consideration the method of possible attacks, system defense analysts can craft

## References

- Anderson, M. (2015, October 29). Technology Device Ownership: 2015. Retrieved March 20, 2018, from http://www.pewinternet.org/2015/10/29/technologydevice-ownership-2015/
- Anthony, S., "A rare look inside an LTE cell site, operated by Sprint in San Francisco," 12-Jun-2013. [Online]. Available: https://www.extremetech.com.
- Chappell, L. A. (2014). Troubleshooting with wireshark: Locate the source of performance problems. San Jose, CA.: Protocol Analysis Institute, Chappell University.
- Conn, D., "Security Aspects of Mobile IP," SANS Institute, 2001. [Online]. Available: https://www.sans.org/readingroom/whitepapers/wireless/security-aspects-mobileip-160.
- "Law Enforcement Information Sharing," 2017. [Online]. Available: https://www.ise.gov.
- Mobile phone ownership. (2017, January 11). Retrieved March 20, 2018, from http://www.pewinternet.org/chart/mobile-phone-ownership/
- Moore, E., "Most Common GPS Tracking Problems," *Trackimo*, 01-Aug2016. [Online]. Available: https://trackimo.com.
- Myers, R. (n.d.). Attacks on TCP/IP Protocols. Retrieved from https://www.utc.edu/center-information-security-assurance/pdfs/course-paper-5620-attacktcpip.pdf
- Perkins, C., "IP Mobility Support for IPv4, Revised," RFC 5944 IP Mobility Support for IPv4, Revised, Nov-2010. [Online]. Available: https://tools.ietf.org/html/rfc5944.
- Qian, Z., & Mao, M. (2012, May 23). Off-PathTCPSequenceNumberInferenceAttack How Firewall Middleboxes Reduce Security. Retrieved from https://web.eecs.umich.edu/~zmao/Papers/oakland12\_TCP\_sequence\_number\_inf erence.pdf

Serral-Garcia, R., Jakab, L., and Domingo-Pascual, J., "Out of order packets analysis on a real network environment," 2006 2nd Conference on Next Generation Internet Design and Engineering, 2006. NGI '06., Apr. 2006.

"Troubleshooting with WireShark: Locate the Source of Performance Problems," *Troubleshooting with WireShark*, 2017. [Online]. Available: http://WireSharkbook.com/tr\_samplepages/978-1-893939-97-4000174.pdf.

"TCP Out-of-Sequence Denial of Service Vulnerability (FreeBSD Security Advisory SA-04:04.tcp)," *TCP Out-of-Sequence Denial of Service Vulnerability*, 09-May-2013. [Online]. Available:

https://kb.juniper.net/InfoCenter/index?page=content&id=JSA10321.