Global Information Assurance Certification Paper

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 Out-of-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.

Erik J. Choron, erik.j.choron.mil@mail.mil

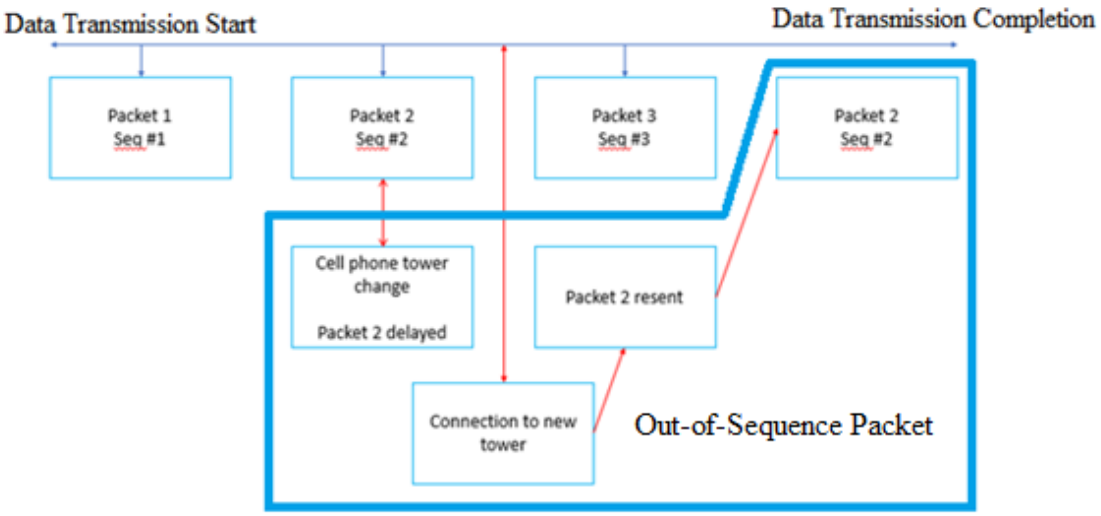

Figure 1 Out-of-Sequence Packet example

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.[1] 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.

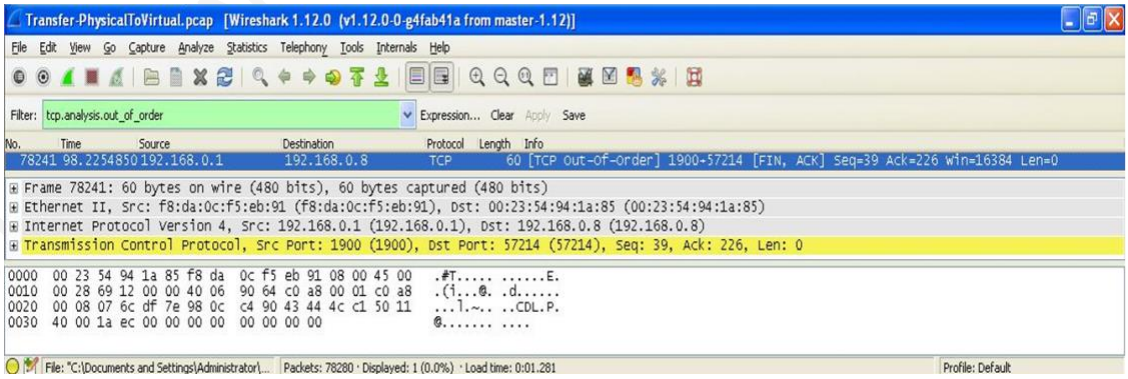

Figure 2 Wireshark displaying the out-of-sequence packet in Scenario 1

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 out-of-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.

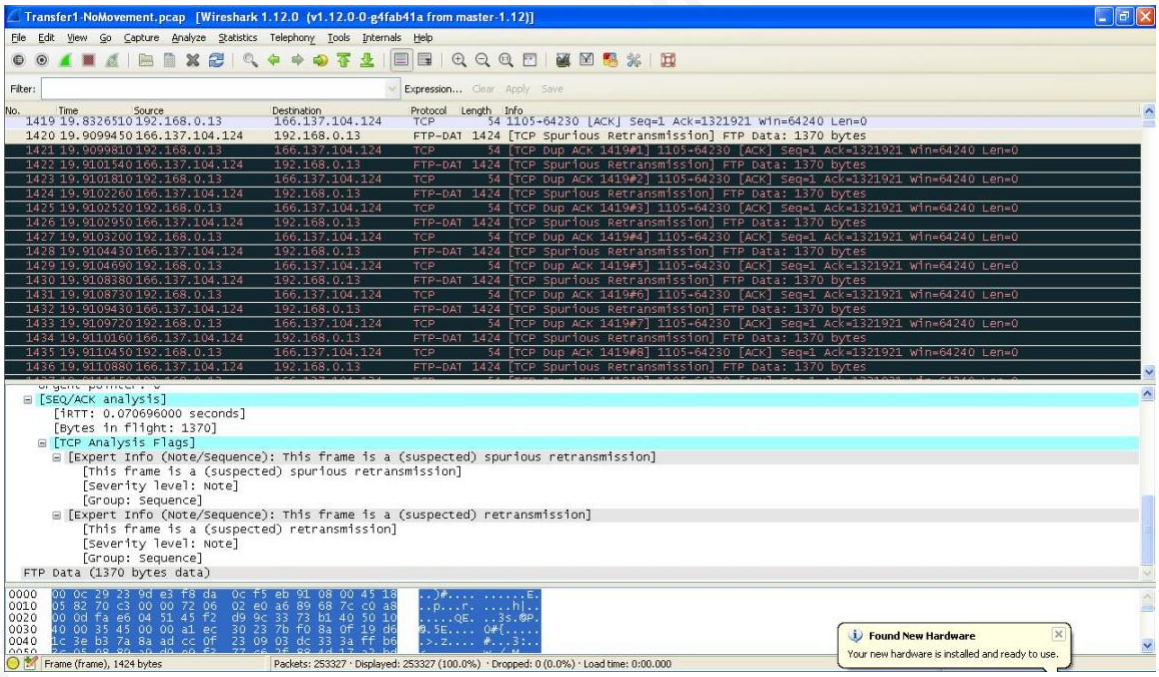

Figure 3 Wireshark displaying the out-of-sequence packet in Scenario 2

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.

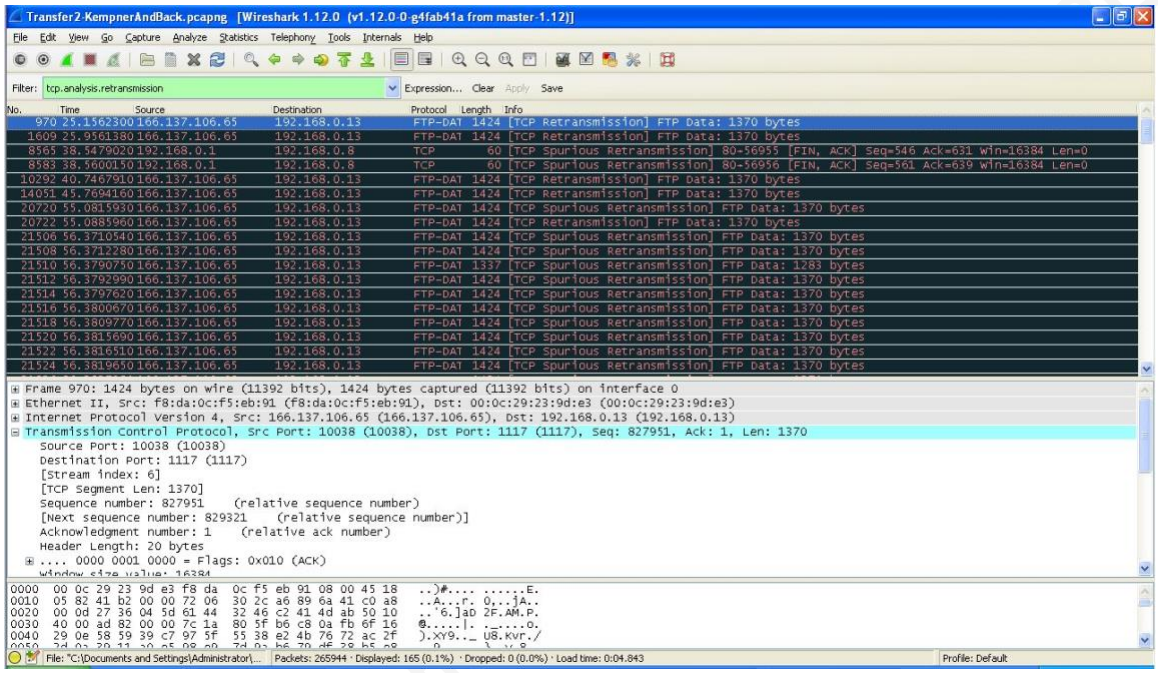

Figure 4 Wireshark displaying the out-of-sequence packet in Scenario 3

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

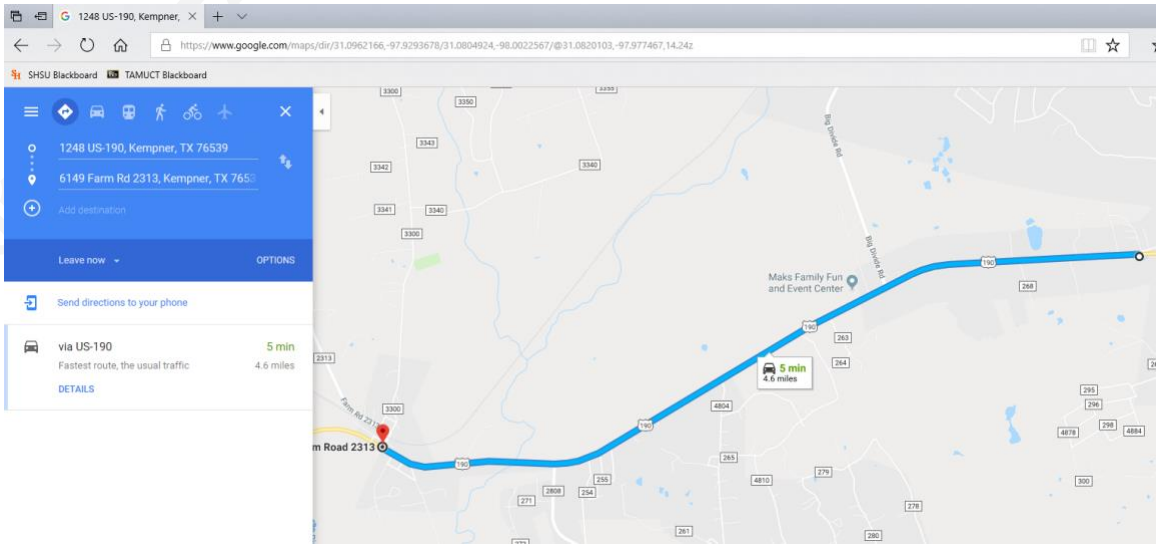

Figure 5 Route traveled for Scenario 3

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.

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 [redacted]
  4   202 ms    30 ms    32 ms xe-0-0-0.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.

Erik J. Choron, erik.j.choron.mil@mail.mil

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.

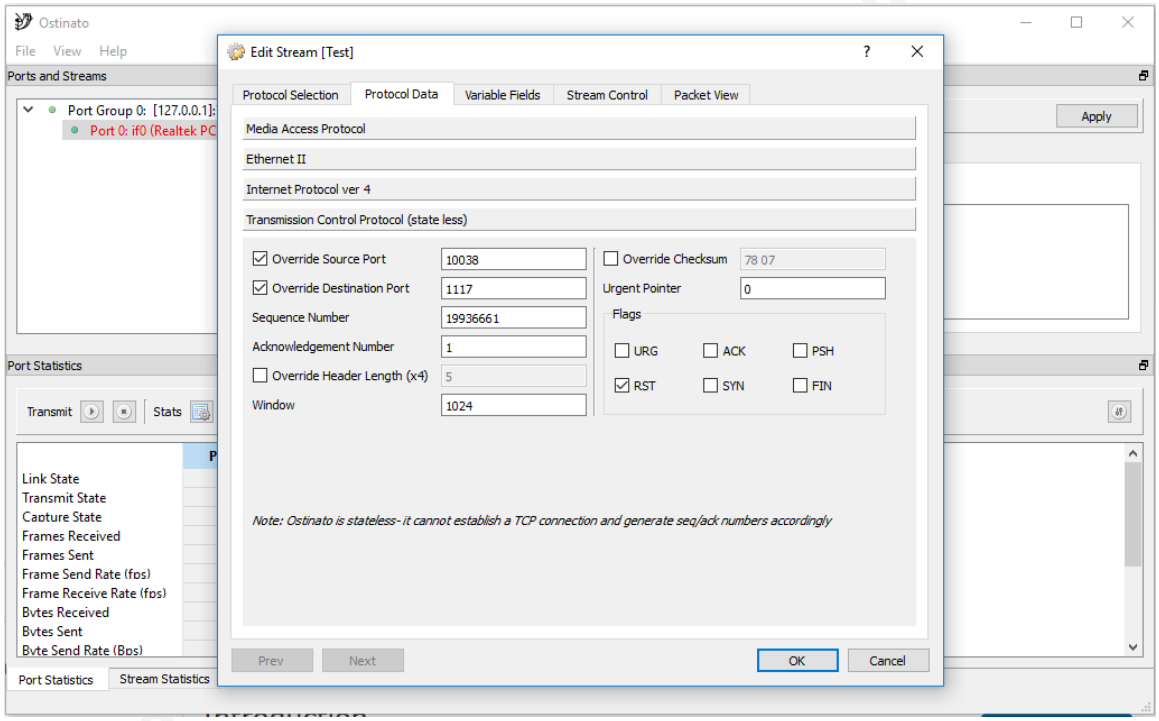

Figure 9 Screenshot of Ostinato

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 consideration the method of possible attacks, system defense analysts can craft procedures to prevent data loss or inaccessibility.

References

Anderson, M. (2015, October 29). Technology Device Ownership: 2015. Retrieved March 20, 2018, from http://www.pewinternet.org/2015/10/29/technology-device-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-mobile-ip-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_inference.pdf

Erik J. Choron, erik.j.choron.mil@mail.mil

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/978-1-893939-97-4ooo174.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.