GIAC
CERTIFICATIONS

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

# Security Considerations for Voice over Wi-Fi (VoWiFi) Systems

## GIAC (GCIA) Gold Certification and RES 5500

Author: Joel Chapman
Advisor: Bryan Simon

Accepted: 20190408

Abstract

As the world pivots from Public Switched Telephony Networks (PSTN) to Voice over Internet Protocol (VoIP)-based telephony architectures, users are employing VoIP-based solutions in more situations. Mobile devices have become a ubiquitous part of a person's identity in the developed world. In the United States in 2017, there were an estimated 224.3 million smartphone users, representing about 68% of the total population. The ability to route telephone call traffic over Wi-Fi networks will continue to expand the coverage area of mobile devices, especially into urban areas where high-density construction has previously caused high signal attenuation. Estimates show that by 2020, Wi-Fi-based calling will make up 53% of mobile IP voice service usage (roughly 9 trillion minutes per year) (Xie, 2018). In contrast to the more traditional VoIP solutions, however, the standards for carrier-based Voice over Wi-Fi (VoWiFi) are often proprietary and have not been well-publicized or vetted. This paper examines the vulnerabilities of VoWiFi calling, assesses what common and less well-known attacks are able to exploit those vulnerabilities, and then proposes technological or procedural security protocols to harden telephony systems against adversary exploitation.

© 2019 The SANS Institute, Author Retains Full Rights

© 2019 The SANS Institute Author retains full rights.

# 1. Introduction

Voice over Internet Protocol (VoIP) technology developed through a lengthy process that relied on several innovations in communications technology over the past centuries. A highly advanced, adaptable, and customizable solution, VoIP provides telephony services and other communications capabilities to users. Inherently a direct outgrowth of previously developed technologies and protocols for communication, VoIP is less an independent innovation than a combination of systems put together to enable voice communications in the modern age, despite the advanced nature of the protocol.

Since the development of the telephone by Alexander Graham Bell in 1876 (Bell, 1876), scientists and engineers have continually worked to increase the efficacy of voice transmissions for higher reliability and greater availability. One of the most critical developments for the eventual evolution of VoIP was the development of the Advanced Research Project Agency Network (ARPANET) in 1969 (Featherly, 2016). ARPANET was the world's first packet switching network, developed to provide high redundancy communications for military and governmental applications. By collecting data into packets for transmission rather than using a dedicated circuit, multiple distributed nodes could rapidly connect through central systems without the high infrastructure costs inherent to circuit-switched networks (Pepper, 2014). In 1988, the International Telecommunications Union (ICT) approved the G.722 audio codec, the first widely-accepted codec to provide quality approaching that was offered by legacy Public Switched Telephone Network (PSTN) systems while employing digital technology (Pepper, 2014). In 1991 NetFone, later known as Speak Freely, was the first software-based VoIP phone service (Pepper, 2014) released to the public domain. In 1996, the Session Initiation Protocol (SIP) was developed by Mark Handley, Henning Schulzrinne, Eve Schooler, and Jonathan Rosenberg and was eventually standardized in 1999 as RFC 2543 (Handley, Schulzrinne, Schooler, & Rosenberg, 1999). Highly scalable, SIP has been further developed and has been adopted by most mobile companies as the preferred protocol for VoIP applications (Pepper, 2014). In 2003, Skype first released its free peer-to-peer internet call system, with an option to pay for calls placed to the PSTN (Pepper, 2014). In 2004, the Federal Communications Commission (FCC) chair Michael Powell declared

© 2019 The SANS Institute Author retains full rights.

VoIP to be an information service rather than a telephony service for regulatory purposes (Hearn, 2004). This declaration had widespread implications, most importantly, taxing of any devices employing VoIP would be at a lower rate and that individual states would not be able to regulate VoIP systems. In 2005, the FCC further added that any VoIP device that connected to the PSTN must have the ability to complete calls to emergency services (FCC, 2005; Pepper, 2014). 2005 saw the development of the first dual-use Wi-Fi cell phone, developed by Calypso Wireless. This device was able to transition between employing the traditional cell tower and local Wi-Fi for internet connectivity, video conferences, and VoIP calls (Calypso Wireless, Inc, 2005; Pepper, 2014).

Today, VoIP supports a large proportion of all telephony traffic in numerous environments. Major carriers are moving away from employing legacy PSTN systems, and in 2014 AT&T petitioned the FCC to allow it to cease all support for its circuit switching infrastructure in favor of VoIP systems (Brodkin, 2014). However, as more traffic moves to VoIP, and as more public Wi-Fi environments become available, there is the danger that VoIP traffic, which traverses open Wi-Fi networks, may be susceptible to eavesdropping and manipulation.

## 1.1 Research

With the push towards more advanced carrier technologies such as 5G and VoLTE systems, service providers have begun offering Voice over Wi-Fi (VoWiFi) to compensate for areas with low or poor cell coverage. Wi-Fi calling technology utilizes the 3GPP IMS (IP Multimedia Subsystem) system to provide packet-switched voice service over Wi-Fi networks (Xie, 2018). This capability grants subscribers several inherent benefits, most obviously increased availability, especially into urban areas where high-density construction has previously caused high signal attenuation.

Within the developed world, mobile devices have become pervasive and are quickly becoming the primary communication and information platforms of users. In the United States in 2017, it was estimated that there were over 224.3 Million smartphone users ("Number of smartphone users in the U.S. 2010-2022", n.d.), representing about 68% of

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

the total population. Researchers estimate that Wi-Fi based calling will make up 53% of mobile IP voice service usage (roughly 9 trillion minutes per year) by 2020 (Xie, 2018). In contrast to the more traditional VoIP solutions, the standards for carrier-based VoWiFi are frequently proprietary and have not been well publicized or vetted.

The rapid rise in popularity of this new technology brings several security concerns. The preponderance of mobile device users are ill-informed on how their devices work, the backbone technologies, and infrastructures that facilitate calling. Though there is an inherent trust in the service provider that they will ensure the confidentiality and integrity of all their subscribers' calls, the employment of VoWiFi technologies introduces new elements that are not necessarily controlled by the service providers into call architecture. Users completing calls through publicly available Wi-Fi may be taking on risks that they do not understand. Additionally, though VoWiFi calls are employing the 3GPP standard and execute through an IPSec tunnel, they still utilize a basic SIP framework for their signaling, creating an opportunity for potential attackers to infer the types of data transmitted by the mobile device.

One of the simplest vectors for executing a man-in-the-middle attack involves establishing a fake access point. Fake access points, though easy to establish, are often highly successful in compromising an unaware user. Traditionally, creating fake access points required the use of a second wireless Network Interface Card (NIC) attached to a computer. Today, it is possible to employ a virtual wireless NIC (WNIC) to the same end using Microsoft Windows 10, negating any additional hardware requirement. The attacker establishes a new access point, frequently with an innocuous SSID or one that is similar to local free Wi-Fi, and then bridges the network connection on their computer to the 'real' network. By doing this, all the data of any user that connects to the 'fake' access point will traverse the attacker's machine, leaving that data open for follow-on attacks.

This research will assess the vulnerabilities of VoWiFi calling and what common or lesser known attacks can exploit those vulnerabilities, and then will propose technological or procedural security protocols to harden telephony systems against adversary exploitation. At present, there has not been an in-depth analysis of carrier-based VoWiFi technology. Though solutions providing similar capabilities to VoWiFi have been available through third-party applications such as Skype and Facebook Messenger for some time,

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

having the capability baked into the phone and directly available by the carrier is a much more recent development. As previously mentioned, these carrier-based VoWiFi systems have neither been researched nor vetted by the greater InfoSec industry. By conducting a deep dive into this technology across multiple platforms, this paper will provide both the consumer and the greater InfoSec community with as much information as possible regarding potential risk vectors and mitigations.

The rapid adoption of mobile devices over the past two decades has brought these potentially vulnerable protocols into daily use by the public. As more mobile devices begin to employ VoWiFi technologies, new vectors to exploitation open with potential ramifications not only for the users but also for the parent organizations. A user making a business call that traverses public Wi-Fi could potentially compromise his organization over lunch. While security researchers have studied traditional VoIP technologies, VoWiFi solutions offered by telephony carriers are a relatively new phenomenon. No detailed study of the potential risks incurred by users when using these newer technologies currently exists. By identifying the current vulnerabilities of Wi-Fi calling and proposing solutions, this paper will contribute to the overall security of current telephony services. The findings from this testing will highlight current flaws in VoWiFi implementation and will enable follow-on efforts to secure this developing pathway of communication.

## 2. Test Method

In order to test the susceptibility of currently-offered Wi-Fi enabled calling systems on the market, a lab environment was developed that employs commonly accessible off-the-shelf technology and systems. A laptop computer, Lenovo Legion Y720 running Windows 10 was used as the primary attack machine. This device was connected to the local network either by Cat5 cable or via a Wi-Fi connection. For man-in-the-middle attacks against the mobile devices, the organic Windows 10 Virtual NIC was employed when using the Windows client to broadcast the test Wi-Fi network over the wireless NIC (WNIC). The mobile device being tested was connected to the test Wi-Fi network and was the only device connected to that network during its testing timeline.

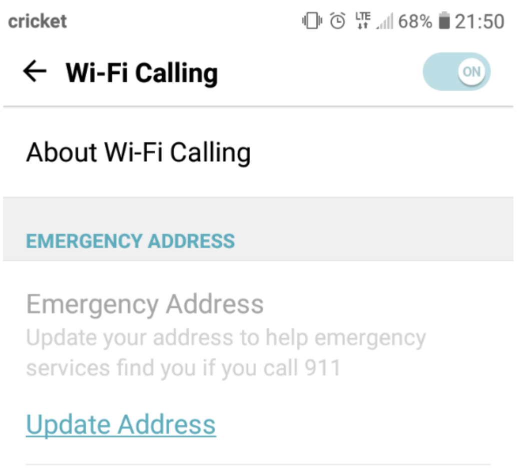
© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

© 2019 The SANS Institute, Author Retains Full Rights

Network and packet analysis was conducted using Wireshark and Tcpdump. Wireshark offers decryption support of many protocols to include IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2. It can further be used for deep dive analysis of captured packet data, employing a suite of tools to interpret collected protocols and assessing each packet at each header layer and the payload itself (Wireshark, n.d.). Tcpdump is another packet analyzer tool that, in contrast to Wireshark, is primarily run through the command line. It is a free software developed in 1988 by Van Jacobson, Sally Floyd, Vern Paxson, and Steven McCanne, researchers at the Lawrence Berkeley Laboratory Network Research Group (McCann, 2011). The primary benefit of Tcpdump compared to Wireshark is that it allows for the processing of significantly larger data sets than Wireshark can comfortably compute. Both tools used together complement each other to provide a thorough analysis of captured packet data.

Two mobile devices were employed to conduct the testing phase of this research. The first was an LG Fortune 2. This mobile device runs on an Android version 7.1.2 OS and is provided wireless service by Cricket Wireless. During initial testing, the mobile device settings were adjusted to deactivate data service to the cell tower, connect to the test Wi-Fi network, and employ VoWiFi capabilities (Figure 1). As annotated later, some issues arose during testing as different methods for forcing the flow of call traffic were tested. First, the mobile device was adjusted to avoid any 2G capability to attempt to force traffic flow across the test Wi-Fi network. Later, the mobile device was placed into Airplane Mode and the Wi-Fi manually re-enabled to prevent the device employing any non-Wi-Fi transport. To create a baseline of what traffic is generated by the mobile devices that allow follow-on isolation of the call traffic, the mobile devices were connected to the test Wi-Fi network, and a packet capture was run for approximately five minutes for each device. Unless specifically annotated, the traffic flow identified during this baselining was ignored during the follow-on testing. The second device employed was a Samsung Galaxy Amp Prime 3, again receiving its telephony services through Cricket Wireless. This device runs



cricket 68% 21:50

Wi-Fi Calling ON

About Wi-Fi Calling

EMERGENCY ADDRESS

Emergency Address
Update your address to help emergency services find you if you call 911

Update Address

*Figure 1 – VoWiFi enabled*

© 2019 The SANS Institute Author retains full rights.

on an Android version 8.0.0. This device had 2G capability disabled from the initial instance of testing.

Additionally, two popular applications for conducting VoIP calls from mobile devices were also tested in order to provide a comparison between the current baked-in solution offered by the service provider and other solutions on the market. The first was the Google Hangouts application. Though it has been discontinued, this does not alter the results that were collected and still provides insight into how large companies handle VoIP for the common consumer. The second was the popular Facebook Messenger application, which is widely employed as a VoIP solution and also as a chat application.

## 3. Lab Testing

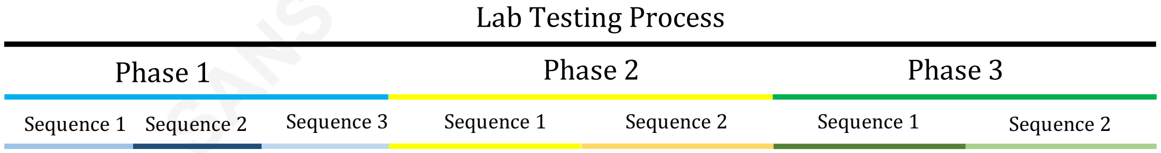


*Figure 2 – Testing Timeline*

Testing was conducted in three phases. Each phase had sequences of individual test calls. The sequences grouped together tests that employed the same or extremely similar variables to confirm consistency of results over multiple tests. The phases grouped together broad types of calls for clarity of discussion. A visual representation of the testing timeline can be seen in Figure 2. The first phase was conducted employing the LG Fortune 2 as the primary calling device, with the calls completing to another mobile device, a Samsung Galaxy Amp Prime 3, which operated with the same wireless service provider. This phase of testing was conducted in three sequences, each involving five test calls. The first five tests (Sequence 1) were conducted with the lab setup and mobile device in a public environment with relatively good cellular service, hereafter referred to as location A. Tests six through ten (Sequence 2) were conducted in location B, an early 20th century building in an area with high electromagnetic interference which limited cellular service. Sequence 3 was also conducted in location B but was unique in that the phone was placed in Airplane Mode prior to conducting the tests as a means to deactivate the phone's wireless

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

connections (LG, n.d.). While in Airplane Mode the LG Fortune 2's Wi-Fi link was enabled, and test calls completed in the same manner as the previous two sequences.

Phase 2 was conducted employing the Samsung Galaxy Amp Prime 3 device completing calls to the LG Fortune 2 and involved two sequences of five tests. This phase was conducted solely in location B. The first sequence was conducted in location B with the mobile device's VoWiFi enabled and the 2G capability disabled. The second sequence was conducted with the Samsung Galaxy Amp Prime 3 placed into Airplane Mode and the Wi-Fi enabled to preclude possible alternate data pathways.

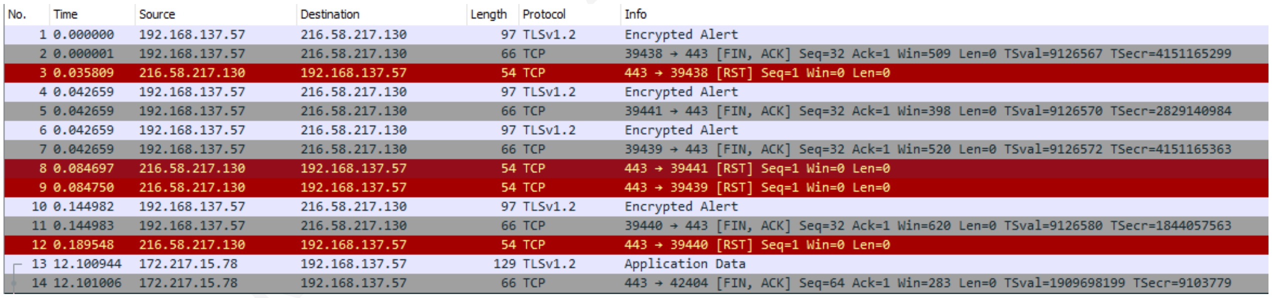
The third phase was conducted to assess popular software-based VoIP solutions and compare them to the results of the mobile device testing. This phase was also conducted in two sequences to address two separate software solutions. The first sequence tested the popular Facebook Messenger application. The second sequence was conducted employing the Google Hangouts application. Both systems are very similar in terms of user experience and features that are offered, and are also both tied to service offerings which provide a suite of other functions, including identity management.

### 3.1.1 Phase 1, Sequence 1

The first test was run using the LG Fortune 2 supported by the Cricket Wireless network. The mobile device was connected to a test Wi-Fi network broadcasting from the Lenovo Legion, and the computer was connected to the internet by a secondary connection. The packet sniffer was employed on the computer at the WNIC broadcasting the test network. The call was placed and lasted for nineteen seconds (19000ms), not counting the time required for call establishment. This sequence of tests was conducted in location A.

© 2019 The SANS Institute Author retains full rights.

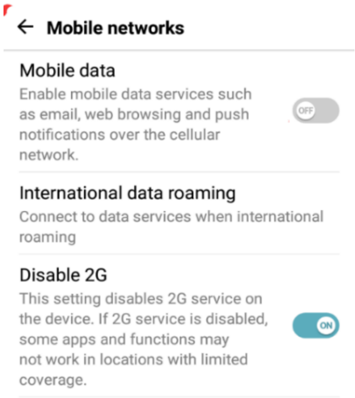
As can be seen in Figure 3, the preponderance of the data that was transmitted was sent encrypted from the mobile device to the server, in this case to an IP address owned by Google. For a call of 19000ms, we could reasonably expect between 600 and 950 packets to be generated to carry the necessary voice data ("Calculating Voice Bandwidth Requirements", 2007). This test call, however, only generated 20 packets. Given that insufficient data was transmitted across the network to account for the duration of the test call, we can extrapolate that alternative pathways for data transmission were identified by the mobile device. As there was no other intentionally active application with a data requirement for the mobile device at the time of the call, and that during baselining no packets similar to those observed in Figure 3 were captured, the assumption was originally made that the packets seen are managing the call setup while the voice traffic proper is employing legacy transmission techniques. The fact that the preponderance of the packets transmitted and received were done so prior to the commencement of the voice exchange appeared to give credence to this assumption. This issue was attempted to be corrected in follow-on testing in order to clarify whether the data observed was in fact related to the call.



```
No.  Time       Source          Destination     Length Protocol  Info
   1 0.000000   192.168.137.57  216.58.217.130      97 TLSv1.2   Encrypted Alert
   2 0.000001   192.168.137.57  216.58.217.130      66 TCP       39438 → 443 [FIN, ACK] Seq=32 Ack=1 Win=509 Len=0 TSval=9126567 TSecr=4151165299
   3 0.035809   216.58.217.130  192.168.137.57      54 TCP       443 → 39438 [RST] Seq=1 Win=0 Len=0
   4 0.042659   192.168.137.57  216.58.217.130      97 TLSv1.2   Encrypted Alert
   5 0.042659   192.168.137.57  216.58.217.130      66 TCP       39441 → 443 [FIN, ACK] Seq=32 Ack=1 Win=398 Len=0 TSval=9126570 TSecr=2829140984
   6 0.042659   192.168.137.57  216.58.217.130      97 TLSv1.2   Encrypted Alert
   7 0.042659   192.168.137.57  216.58.217.130      66 TCP       39439 → 443 [FIN, ACK] Seq=32 Ack=1 Win=520 Len=0 TSval=9126572 TSecr=4151165363
   8 0.084697   216.58.217.130  192.168.137.57      54 TCP       443 → 39441 [RST] Seq=1 Win=0 Len=0
   9 0.084750   216.58.217.130  192.168.137.57      54 TCP       443 → 39439 [RST] Seq=1 Win=0 Len=0
  10 0.144982   192.168.137.57  216.58.217.130      97 TLSv1.2   Encrypted Alert
  11 0.144983   192.168.137.57  216.58.217.130      66 TCP       39440 → 443 [FIN, ACK] Seq=32 Ack=1 Win=620 Len=0 TSval=9126580 TSecr=1844057563
  12 0.189548   216.58.217.130  192.168.137.57      54 TCP       443 → 39440 [RST] Seq=1 Win=0 Len=0
  13 12.100944  172.217.15.78   192.168.137.57     129 TLSv1.2   Application Data
  14 12.101006  172.217.15.78   192.168.137.57      66 TCP       443 → 42404 [FIN, ACK] Seq=64 Ack=1 Win=283 Len=0 TSval=1909698199 TSecr=9103779
```

*Figure 3 – Captured packet data from Phase 1, Sequence 1, Test 1*

In preparation for the second test, the ability of the mobile device to employ 2G was deactivated (Figure 4) in an attempt to force the traffic flow through the test Wi-Fi network. Following this, a second test call was made with all other conditions the same as for the first test. This test had significantly more traffic (Figure 5). Again, this traffic was encrypted. The first several packets were the setup of a follow-on tunnel for the transmission as can been seen in Figure 5. The distant end IP address, 72.21.207.87, is



← Mobile networks

Mobile data
Enable mobile data services such as email, web browsing and push notifications over the cellular network.
OFF

International data roaming
Connect to data services when international roaming

Disable 2G
This setting disables 2G service on the device. If 2G service is disabled, some apps and functions may not work in locations with limited coverage.
ON

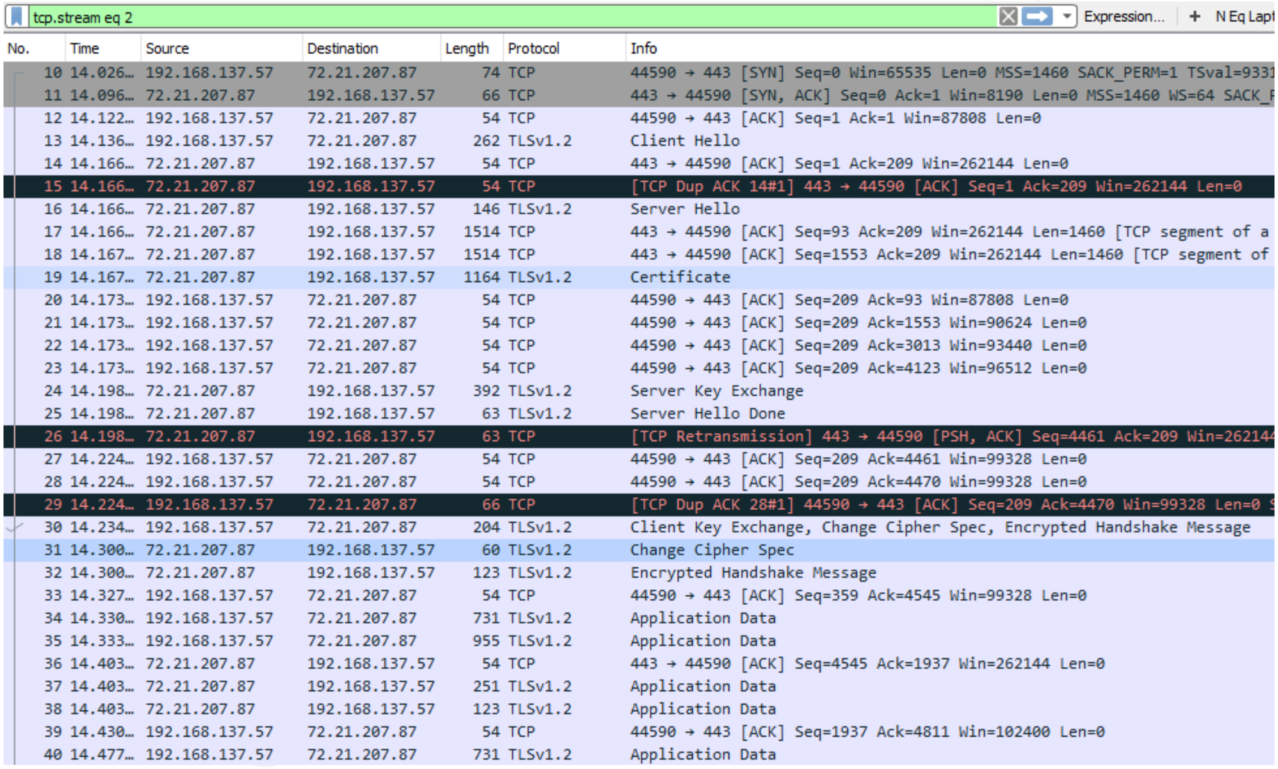
*Figure 4 – 2G deactivated*

© 2019 The SANS Institute Author Retains Full Rights

© 2019 The SANS Institute

Author retains full rights.

registered to Amazon.com, an unanticipated result considering the mobile device used Cricket Wireless as its service provider.



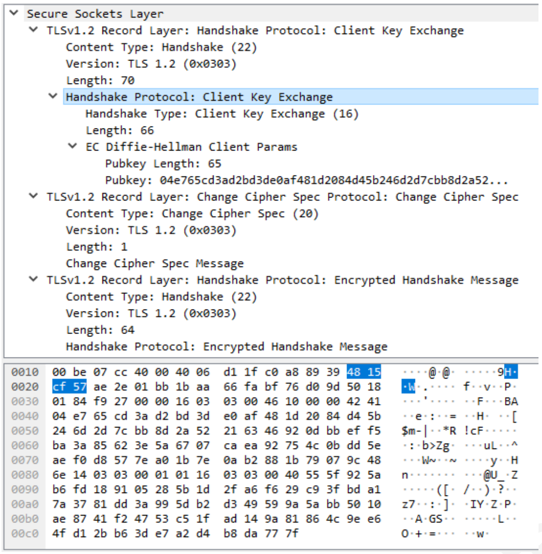
```
tcp.stream eq 2

No. Time      Source          Destination     Length Protocol  Info
 10 14.026… 192.168.137.57  72.21.207.87       74 TCP      44590 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=9331
 11 14.096… 72.21.207.87    192.168.137.57     66 TCP      443 → 44590 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1460 WS=64 SACK_P
 12 14.122… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=1 Ack=1 Win=87808 Len=0
 13 14.136… 192.168.137.57  72.21.207.87      262 TLSv1.2  Client Hello
 14 14.166… 72.21.207.87    192.168.137.57     54 TCP      443 → 44590 [ACK] Seq=1 Ack=209 Win=262144 Len=0
 15 14.166… 72.21.207.87    192.168.137.57     54 TCP      [TCP Dup ACK 14#1] 443 → 44590 [ACK] Seq=1 Ack=209 Win=262144 Len=0
 16 14.166… 72.21.207.87    192.168.137.57    146 TLSv1.2  Server Hello
 17 14.166… 72.21.207.87    192.168.137.57   1514 TCP      443 → 44590 [ACK] Seq=93 Ack=209 Win=262144 Len=1460 [TCP segment of a
 18 14.167… 72.21.207.87    192.168.137.57   1514 TCP      443 → 44590 [ACK] Seq=1553 Ack=209 Win=262144 Len=1460 [TCP segment of
 19 14.167… 72.21.207.87    192.168.137.57   1164 TLSv1.2  Certificate
 20 14.173… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=209 Ack=93 Win=87808 Len=0
 21 14.173… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=209 Ack=1553 Win=90624 Len=0
 22 14.173… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=209 Ack=3013 Win=93440 Len=0
 23 14.173… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=209 Ack=4123 Win=96512 Len=0
 24 14.198… 72.21.207.87    192.168.137.57    392 TLSv1.2  Server Key Exchange
 25 14.198… 72.21.207.87    192.168.137.57     63 TLSv1.2  Server Hello Done
 26 14.198… 72.21.207.87    192.168.137.57     63 TCP      [TCP Retransmission] 443 → 44590 [PSH, ACK] Seq=4461 Ack=209 Win=262144
 27 14.224… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=209 Ack=4461 Win=99328 Len=0
 28 14.224… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=209 Ack=4470 Win=99328 Len=0
 29 14.224… 192.168.137.57  72.21.207.87       66 TCP      [TCP Dup ACK 28#1] 44590 → 443 [ACK] Seq=209 Ack=4470 Win=99328 Len=0 S
 30 14.234… 192.168.137.57  72.21.207.87      204 TLSv1.2  Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
 31 14.300… 72.21.207.87    192.168.137.57     60 TLSv1.2  Change Cipher Spec
 32 14.300… 72.21.207.87    192.168.137.57    123 TLSv1.2  Encrypted Handshake Message
 33 14.327… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=359 Ack=4545 Win=99328 Len=0
 34 14.330… 192.168.137.57  72.21.207.87      731 TLSv1.2  Application Data
 35 14.333… 192.168.137.57  72.21.207.87      955 TLSv1.2  Application Data
 36 14.403… 72.21.207.87    192.168.137.57     54 TCP      443 → 44590 [ACK] Seq=4545 Ack=1937 Win=262144 Len=0
 37 14.403… 72.21.207.87    192.168.137.57    251 TLSv1.2  Application Data
 38 14.403… 72.21.207.87    192.168.137.57    123 TLSv1.2  Application Data
 39 14.430… 192.168.137.57  72.21.207.87       54 TCP      44590 → 443 [ACK] Seq=1937 Ack=4811 Win=102400 Len=0
 40 14.477… 192.168.137.57  72.21.207.87      731 TLSv1.2  Application Data
```

*Figure 5 – Captured packet data from Phase 1, Sequence 1, Test 2*

Public key exchange in TLS v1.2 is handled via the Diffie-Helman process. Diffie-Helman allows two end points, in this case the mobile device and the server, to generate a shared secret, the encryption key for follow on data transmission. The key exchange between the mobile device was captured and can be seen in packet 30, from TCP stream 2 (Figure 6). While the public key was successfully identified from this capture, it does not pose a security concern to the type of eavesdropping attacks employed here.

© 2019 The SANS Institute Author retains full rights.

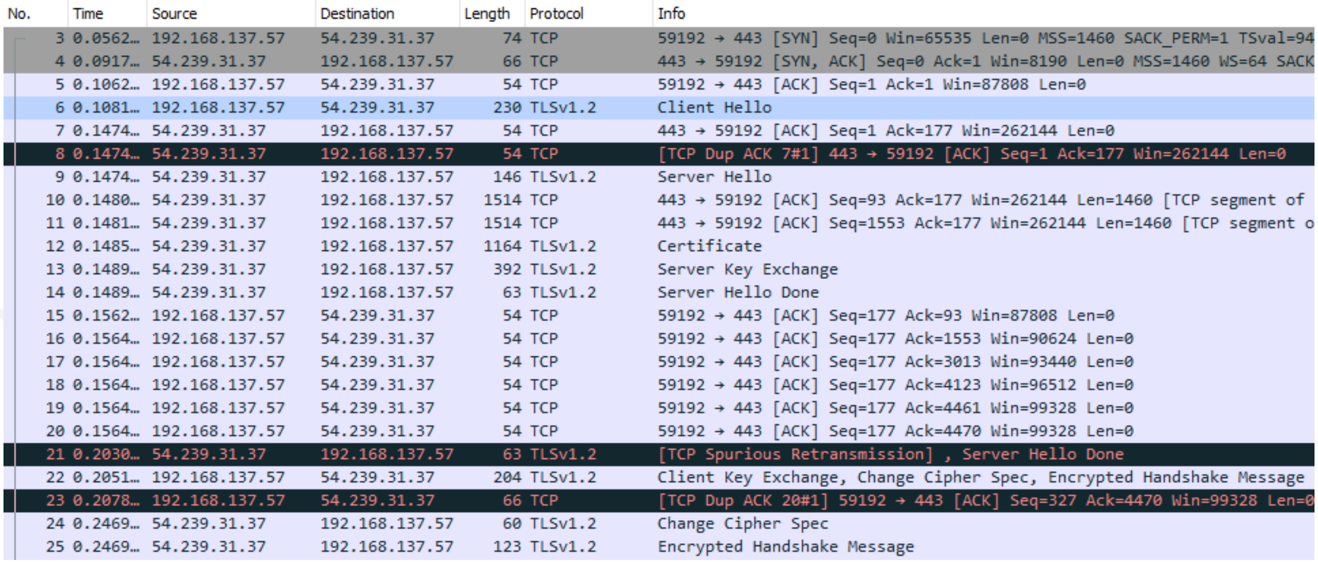


```
Secure Sockets Layer
  TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 70
    Handshake Protocol: Client Key Exchange
      Handshake Type: Client Key Exchange (16)
      Length: 66
      EC Diffie-Hellman Client Params
        Pubkey Length: 65
        Pubkey: 04e765cd3ad2bd3de0af481d2084d45b246d2d7cbb8d2a52...
  TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.2 (0x0303)
    Length: 1
    Change Cipher Spec Message
  TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 64
    Handshake Protocol: Encrypted Handshake Message

0010  00 be 07 cc 40 00 40 06  d1 1f c0 a8 89 39 48 15   ····@·@· ·····9H·
0020  cf 57 ae 2e 01 bb 1b aa  66 fa bf 76 d0 9d 50 18   ·W.····· f··v··P·
0030  01 84 f9 27 00 00 16 03  03 00 46 10 00 00 42 41   ···'···· ··F···BA
0040  04 e7 65 cd 3a d2 bd 3d  e0 af 48 1d 20 84 d4 5b   ··e·:··= ··H· ··[
0050  24 6d 2d 7c bb 8d 2a 52  21 63 46 92 0d bb ef f5   $m-|··*R !cF·····
0060  ba 3a 85 62 3e 5a 67 07  ca ea 92 75 4c 0b dd 5e   ·:·b>Zg· ···uL··^
0070  ae f0 d8 57 7e a0 1b 7e  0a b2 88 1b 79 07 9c 48   ···W~··~ ····y··H
0080  6e 14 03 03 00 01 01 16  03 03 00 40 55 5f 92 5a   n······· ···@U_·Z
0090  b6 fd 18 91 05 28 5b 1d  2f a6 f6 29 c9 3f bd a1   ·····([· /··)·?··
00a0  7a 37 81 dd 3a 99 5d b2  d3 49 59 9a 5a bb 50 10   z7··:·]· ·IY·Z·P·
00b0  ae 87 41 f2 47 53 c5 1f  ad 14 9a 81 86 4c 9e e6   ··A·GS·· ·····L··
00c0  4f d1 2b b6 3d e7 a2 d4  b8 da 77 7f                O·+·=··· ··w·
```

*Figure 6 –key exchange*

The third test was conducted using the same parameters as the second test and is observable in Figure 7. For this test, however, instead of allowing the call to complete, the call was allowed to go to the voicemail service of the Samsung Galaxy Amp Prime 3. The results of this test were much the same as before. Again, the IP address that the mobile device was communicating to, 54.239.31.37, was registered to Amazon. These three tests demonstrate that, given the specific circumstances constructed for these tests, the data that is traversing the Wi-Fi network is secured through encryption and not immediately compromisable by an attacker who is intercepting the traffic.



| No. | Time | Source | Destination | Length | Protocol | Info |
|---|---|---|---|---|---|---|
| 3 | 0.0562… | 192.168.137.57 | 54.239.31.37 | 74 | TCP | 59192 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=94 |
| 4 | 0.0917… | 54.239.31.37 | 192.168.137.57 | 66 | TCP | 443 → 59192 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1460 WS=64 SACK |
| 5 | 0.1062… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=1 Ack=1 Win=87808 Len=0 |
| 6 | 0.1081… | 192.168.137.57 | 54.239.31.37 | 230 | TLSv1.2 | Client Hello |
| 7 | 0.1474… | 54.239.31.37 | 192.168.137.57 | 54 | TCP | 443 → 59192 [ACK] Seq=1 Ack=177 Win=262144 Len=0 |
| 8 | 0.1474… | 54.239.31.37 | 192.168.137.57 | 54 | TCP | [TCP Dup ACK 7#1] 443 → 59192 [ACK] Seq=1 Ack=177 Win=262144 Len=0 |
| 9 | 0.1474… | 54.239.31.37 | 192.168.137.57 | 146 | TLSv1.2 | Server Hello |
| 10 | 0.1480… | 54.239.31.37 | 192.168.137.57 | 1514 | TCP | 443 → 59192 [ACK] Seq=93 Ack=177 Win=262144 Len=1460 [TCP segment of |
| 11 | 0.1481… | 54.239.31.37 | 192.168.137.57 | 1514 | TCP | 443 → 59192 [ACK] Seq=1553 Ack=177 Win=262144 Len=1460 [TCP segment o |
| 12 | 0.1485… | 54.239.31.37 | 192.168.137.57 | 1164 | TLSv1.2 | Certificate |
| 13 | 0.1489… | 54.239.31.37 | 192.168.137.57 | 392 | TLSv1.2 | Server Key Exchange |
| 14 | 0.1489… | 54.239.31.37 | 192.168.137.57 | 63 | TLSv1.2 | Server Hello Done |
| 15 | 0.1562… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=177 Ack=93 Win=87808 Len=0 |
| 16 | 0.1564… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=177 Ack=1553 Win=90624 Len=0 |
| 17 | 0.1564… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=177 Ack=3013 Win=93440 Len=0 |
| 18 | 0.1564… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=177 Ack=4123 Win=96512 Len=0 |
| 19 | 0.1564… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=177 Ack=4461 Win=99328 Len=0 |
| 20 | 0.1564… | 192.168.137.57 | 54.239.31.37 | 54 | TCP | 59192 → 443 [ACK] Seq=177 Ack=4470 Win=99328 Len=0 |
| 21 | 0.2030… | 54.239.31.37 | 192.168.137.57 | 63 | TLSv1.2 | [TCP Spurious Retransmission] , Server Hello Done |
| 22 | 0.2051… | 192.168.137.57 | 54.239.31.37 | 204 | TLSv1.2 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 23 | 0.2078… | 192.168.137.57 | 54.239.31.37 | 66 | TCP | [TCP Dup ACK 20#1] 59192 → 443 [ACK] Seq=327 Ack=4470 Win=99328 Len=0 |
| 24 | 0.2469… | 54.239.31.37 | 192.168.137.57 | 60 | TLSv1.2 | Change Cipher Spec |
| 25 | 0.2469… | 54.239.31.37 | 192.168.137.57 | 123 | TLSv1.2 | Encrypted Handshake Message |

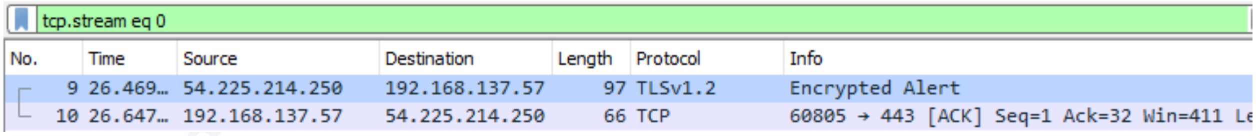
*Figure 7 – Captured packet data from Phase 1, Sequence 1, Test 3*

A fourth and fifth test were conducted employing the same conditions as the second. An anomaly was observed at this time; despite no settings changing and the lab environment remaining the same, extremely limited packet traffic was detected and captured by Wireshark during the conduct of these two tests. During the fourth test, only

© 2019 The SANS Institute Author retains full rights.

two packets that were attributable to the call were caught (Figure 8). The fifth test intercepted no packets traversing the test Wi-Fi network.

Ultimately, it became increasingly clear that most, if not all the call-related data traffic was not traversing the Wi-Fi test network and was employing the mobile device's RF-based transmission systems in order to complete calls to the receiving mobile device. Due to this probability, additional sequences of tests were conducted that attempted various means of blocking the RF transmission capability of the mobile device in order to force the call traffic across the test Wi-Fi network and provide a conclusive look at the type of data packets being transmitted. While this sequence of testing did not provide the anticipated results of call-related packets for follow-on analysis, it did reveal that despite the user enabling the mobile device to place calls over Wi-Fi, the mobile device chose pathways based upon a decision-making system not visible to the user. This lack of capability on the part of the user to force certain functions on the part of the mobile device is addressed in the analysis section of this paper.



```
tcp.stream eq 0
No.   Time      Source          Destination     Length  Protocol  Info
   9  26.469…   54.225.214.250  192.168.137.57      97  TLSv1.2   Encrypted Alert
  10  26.647…   192.168.137.57  54.225.214.250      66  TCP       60805 → 443 [ACK] Seq=1 Ack=32 Win=411 Le
```

*Figure 8 – Captured packet data from Phase 1, Sequence 1, Test 4*

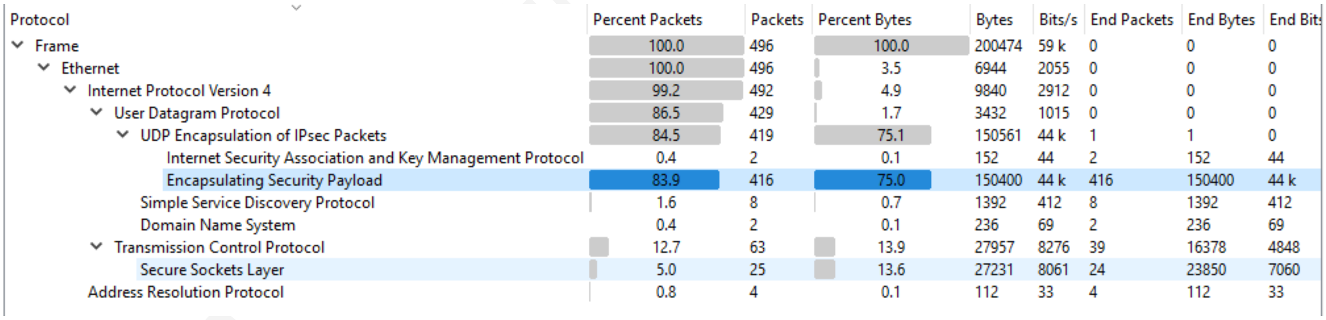
## 3.1.2 Phase 1, Sequence 2

The second sequence of five more test calls was conducted using approximately the same conditions as in the second test, with the call completing from the LG Fortune 2 to the Samsung Galaxy Amp Prime 3 across the test Wi-Fi network. However, the physical location of the testing was moved to location B. One significant change in the testing results, apparently elicited by the change in location, was that the data seen traversing the network was not encrypted in TLS 1.2 but rather in Encapsulated Security Protocol traffic (ESP) ("ESP, Encapsulated Security Protocol", n.d.). ESP traffic is most commonly associated with IPSec Tunnels, which is consistent with the 3GPP standard. These tests all generated data in quantities more aligned with that anticipated for the traffic type and

© 2019 The SANS Institute, Author Retains Full Rights

© 2019 The SANS Institute Author retains full rights.

duration. Test 6 generated 884 packets, of which 878 were ESP carrying 350856 Bytes of data. Test 7 generated 416 ESP packets carrying 150400 Bytes. Test 8 generated 731 ESP packets carrying 275468 Bytes. Test 9 generated 549 ESP packets carrying 224612 Bytes, and finally Test 10 generated 592 ESP packets carrying 247856 Bytes of data. The breakdown of the protocol hierarchy for the traffic that was captured during Test 7 can be seen in Figure 9. The traffic rate of 44kBits/s represents an approximate average of the traffic rate observed during all five tests in this sequence. This traffic rate is more consistent with the bandwidth utilization that would be expected from a VoIP call being placed across the network. These results seem to confirm the concerns raised during the first sequence of testing that the results gained were not actually test call traffic at all and that the mobile device was still using its RF capabilities to complete the call.



| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bit |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 496 | 100.0 | 200474 | 59 k | 0 | 0 | 0 |
| Ethernet | 100.0 | 496 | 3.5 | 6944 | 2055 | 0 | 0 | 0 |
| Internet Protocol Version 4 | 99.2 | 492 | 4.9 | 9840 | 2912 | 0 | 0 | 0 |
| User Datagram Protocol | 86.5 | 429 | 1.7 | 3432 | 1015 | 0 | 0 | 0 |
| UDP Encapsulation of IPsec Packets | 84.5 | 419 | 75.1 | 150561 | 44 k | 1 | 1 | 0 |
| Internet Security Association and Key Management Protocol | 0.4 | 2 | 0.1 | 152 | 44 | 2 | 152 | 44 |
| Encapsulating Security Payload | 83.9 | 416 | 75.0 | 150400 | 44 k | 416 | 150400 | 44 k |
| Simple Service Discovery Protocol | 1.6 | 8 | 0.7 | 1392 | 412 | 8 | 1392 | 412 |
| Domain Name System | 0.4 | 2 | 0.1 | 236 | 69 | 2 | 236 | 69 |
| Transmission Control Protocol | 12.7 | 63 | 13.9 | 27957 | 8276 | 39 | 16378 | 4848 |
| Secure Sockets Layer | 5.0 | 25 | 13.6 | 27231 | 8061 | 24 | 23850 | 7060 |
| Address Resolution Protocol | 0.8 | 4 | 0.1 | 112 | 33 | 4 | 112 | 33 |

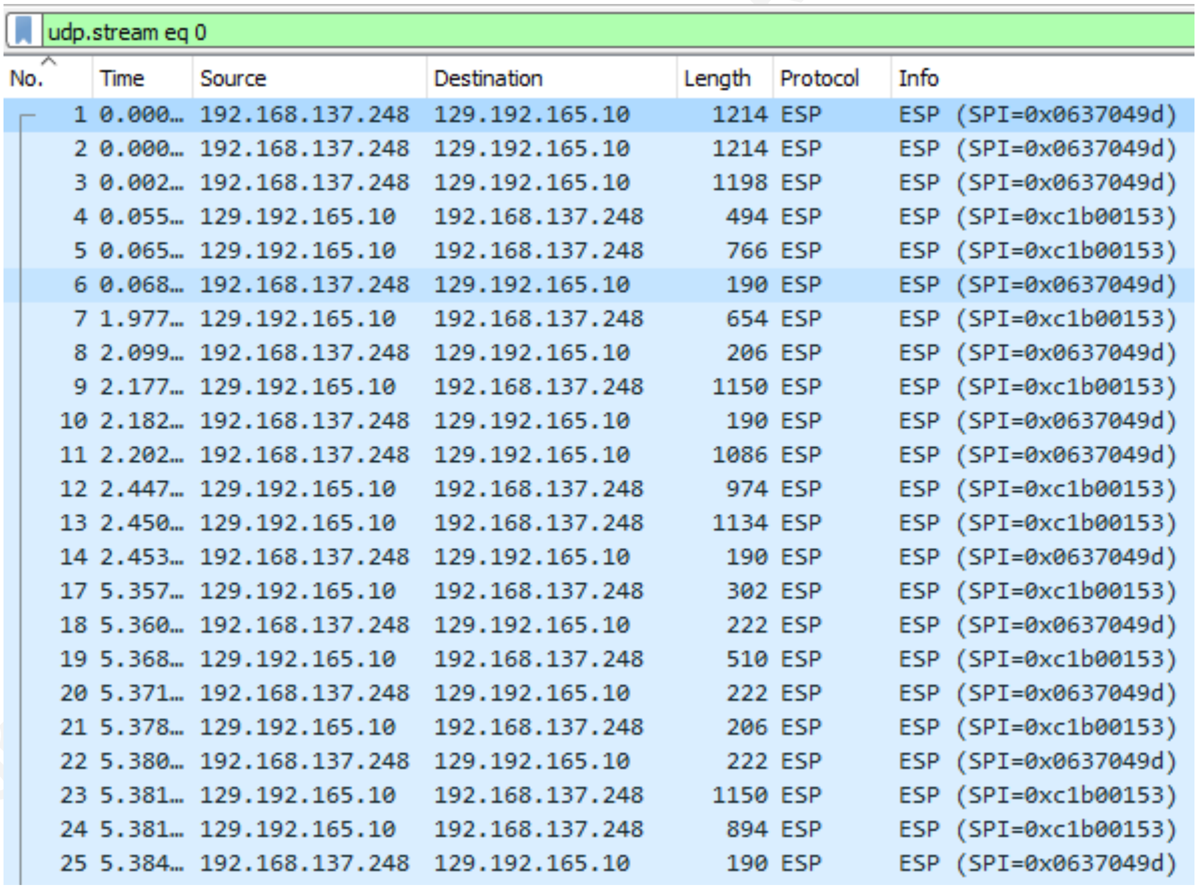
*Figure 9 – Protocol hierarchy for Phase 1, Sequence 2, Test 2*

The first twenty-five packets captured during Test 10 can be observed in Figure 10. The first three packets are larger with the first two having consistent sizing, while following packets are more randomized and smaller. From these observations, it is possible to extrapolate that the first few packets transmitted represented the call setup between the LG Fortune 2 mobile device and the carrier's server, while the follow-on packets represent the actual voice traffic. This same pattern, with the initial three packets of the ESP traffic flow having a size of 1214 and 1198 bytes, was consistent across all five test calls completed during this sequence.

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

An interesting finding of the packet analysis was that the packets were being routed to a server registered not by Cricket Wireless, who is was the service provider for the mobile device, but rather to Ericsson North America Managed Services. If call data was traversing third-party networks that were not telecommunications providers, this information was not readily available to a user of the Cricket Wireless service, creating privacy concerns that are further discussed in the analysis portion of this paper.



```
udp.stream eq 0

No.  Time      Source           Destination      Length Protocol Info
   1 0.000…    192.168.137.248  129.192.165.10     1214 ESP      ESP (SPI=0x0637049d)
   2 0.000…    192.168.137.248  129.192.165.10     1214 ESP      ESP (SPI=0x0637049d)
   3 0.002…    192.168.137.248  129.192.165.10     1198 ESP      ESP (SPI=0x0637049d)
   4 0.055…    129.192.165.10   192.168.137.248     494 ESP      ESP (SPI=0xc1b00153)
   5 0.065…    129.192.165.10   192.168.137.248     766 ESP      ESP (SPI=0xc1b00153)
   6 0.068…    192.168.137.248  129.192.165.10      190 ESP      ESP (SPI=0x0637049d)
   7 1.977…    129.192.165.10   192.168.137.248     654 ESP      ESP (SPI=0xc1b00153)
   8 2.099…    192.168.137.248  129.192.165.10      206 ESP      ESP (SPI=0x0637049d)
   9 2.177…    129.192.165.10   192.168.137.248    1150 ESP      ESP (SPI=0xc1b00153)
  10 2.182…    192.168.137.248  129.192.165.10      190 ESP      ESP (SPI=0x0637049d)
  11 2.202…    192.168.137.248  129.192.165.10     1086 ESP      ESP (SPI=0x0637049d)
  12 2.447…    129.192.165.10   192.168.137.248     974 ESP      ESP (SPI=0xc1b00153)
  13 2.450…    129.192.165.10   192.168.137.248    1134 ESP      ESP (SPI=0xc1b00153)
  14 2.453…    192.168.137.248  129.192.165.10      190 ESP      ESP (SPI=0x0637049d)
  17 5.357…    129.192.165.10   192.168.137.248     302 ESP      ESP (SPI=0xc1b00153)
  18 5.360…    192.168.137.248  129.192.165.10      222 ESP      ESP (SPI=0x0637049d)
  19 5.368…    129.192.165.10   192.168.137.248     510 ESP      ESP (SPI=0xc1b00153)
  20 5.371…    192.168.137.248  129.192.165.10      222 ESP      ESP (SPI=0x0637049d)
  21 5.378…    129.192.165.10   192.168.137.248     206 ESP      ESP (SPI=0xc1b00153)
  22 5.380…    192.168.137.248  129.192.165.10      222 ESP      ESP (SPI=0x0637049d)
  23 5.381…    129.192.165.10   192.168.137.248    1150 ESP      ESP (SPI=0xc1b00153)
  24 5.381…    129.192.165.10   192.168.137.248     894 ESP      ESP (SPI=0xc1b00153)
  25 5.384…    192.168.137.248  129.192.165.10      190 ESP      ESP (SPI=0x0637049d)
```

*Figure 10 – Captured packet data from Phase 1, Sequence 2, Test 5*

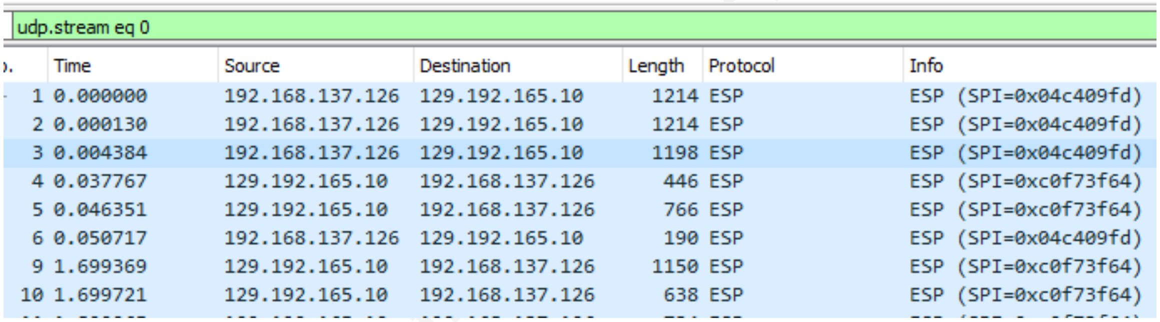
## 3.1.3 Phase 1, Sequence 3

The third sequence of tests was conducted with the mobile device placed into Airplane Mode and the Wi-Fi capability manually enabled. It was expected that this would ultimately defeat any alternate data transport paths and force all traffic through the packet

© 2019 The SANS Institute Author retains full rights.

capture. As in Sequences 1 and 2 of this phase of testing, five test calls were completed between the LG Fortune 2 and the Samsung Galaxy Amp Prime 3.

The packets captured during this sequence were largely homogenous across all five tests. As can be observed in Figure 11, the output from each individual call also largely mirrored that which was observed in Sequence 2. Again, the same pattern of the first three ESP packets exchanged with a packet length of 1214 bytes and 1198 bytes was seen.



udp.stream eq 0

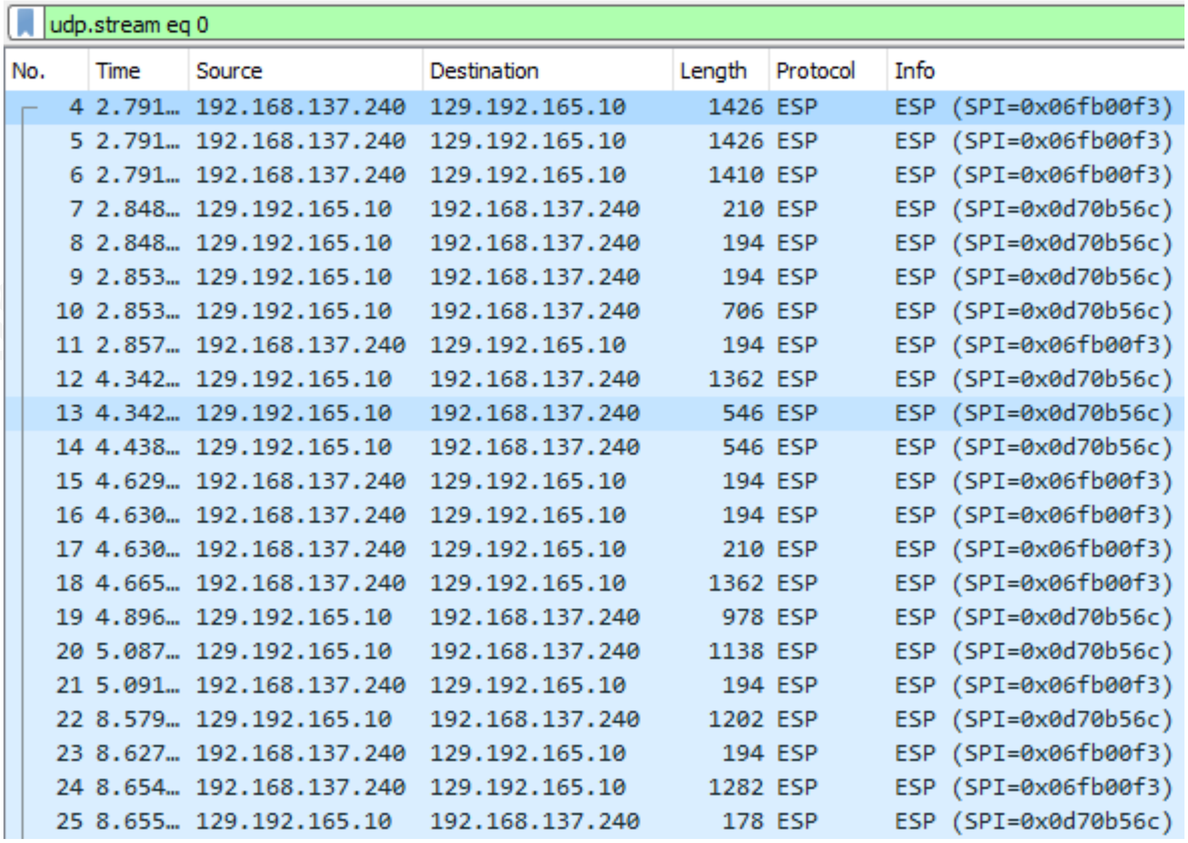
| . | Time | Source | Destination | Length | Protocol | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.137.126 | 129.192.165.10 | 1214 | ESP | ESP (SPI=0x04c409fd) |
| 2 | 0.000130 | 192.168.137.126 | 129.192.165.10 | 1214 | ESP | ESP (SPI=0x04c409fd) |
| 3 | 0.004384 | 192.168.137.126 | 129.192.165.10 | 1198 | ESP | ESP (SPI=0x04c409fd) |
| 4 | 0.037767 | 129.192.165.10 | 192.168.137.126 | 446 | ESP | ESP (SPI=0xc0f73f64) |
| 5 | 0.046351 | 129.192.165.10 | 192.168.137.126 | 766 | ESP | ESP (SPI=0xc0f73f64) |
| 6 | 0.050717 | 192.168.137.126 | 129.192.165.10 | 190 | ESP | ESP (SPI=0x04c409fd) |
| 9 | 1.699369 | 129.192.165.10 | 192.168.137.126 | 1150 | ESP | ESP (SPI=0xc0f73f64) |
| 10 | 1.699721 | 129.192.165.10 | 192.168.137.126 | 638 | ESP | ESP (SPI=0xc0f73f64) |

*Figure 11 – Captured packet data fromPhase 1, Sequence 3, Test 1*

## 3.2.1 Phase 2, Sequence 1

© 2019 The SANS Institute Author retains full rights.

The next series of tests conducted employed a different mobile device. This time the Samsung Galaxy Amp Prime 3 was employed rather than the LG Fortune 2. The same test Wi-Fi network was employed with the settings remaining the same for the network. The testing occurred in location B. The initial test returned results similar to those returned for the LG in Phase 1, Sequence 2. The first three packets that were transmitted had the same protocol, distant end IP address, and size as the first three packets in the previous test runs, as is observable in Figure 12. Test 2 returned the same initial three packet sequence (Figure 13). The consistency with which these first three packets appear continues to lend credence to the idea that this is a specific indication of SIP traffic traversing the network. Once the call was connected during the third test, no attempt was made to pass voice traffic along the circuit. During this test, it was observed within the follow-on data stream that all the ESP packets being passed between the nodes were 178 bytes long consistently. In contrast, varying loud voice traffic was passed along the circuit during Test 4. However, the result



udp.stream eq 0

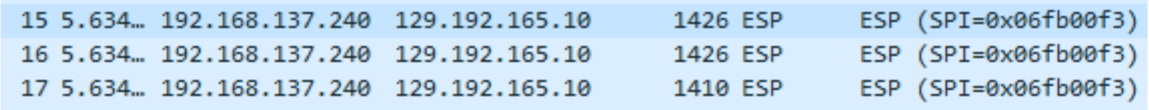
| No. | Time | Source | Destination | Length | Protocol | Info |
|---|---|---|---|---|---|---|
| 4 | 2.791… | 192.168.137.240 | 129.192.165.10 | 1426 | ESP | ESP (SPI=0x06fb00f3) |
| 5 | 2.791… | 192.168.137.240 | 129.192.165.10 | 1426 | ESP | ESP (SPI=0x06fb00f3) |
| 6 | 2.791… | 192.168.137.240 | 129.192.165.10 | 1410 | ESP | ESP (SPI=0x06fb00f3) |
| 7 | 2.848… | 129.192.165.10 | 192.168.137.240 | 210 | ESP | ESP (SPI=0x0d70b56c) |
| 8 | 2.848… | 129.192.165.10 | 192.168.137.240 | 194 | ESP | ESP (SPI=0x0d70b56c) |
| 9 | 2.853… | 129.192.165.10 | 192.168.137.240 | 194 | ESP | ESP (SPI=0x0d70b56c) |
| 10 | 2.853… | 129.192.165.10 | 192.168.137.240 | 706 | ESP | ESP (SPI=0x0d70b56c) |
| 11 | 2.857… | 192.168.137.240 | 129.192.165.10 | 194 | ESP | ESP (SPI=0x06fb00f3) |
| 12 | 4.342… | 129.192.165.10 | 192.168.137.240 | 1362 | ESP | ESP (SPI=0x0d70b56c) |
| 13 | 4.342… | 129.192.165.10 | 192.168.137.240 | 546 | ESP | ESP (SPI=0x0d70b56c) |
| 14 | 4.438… | 129.192.165.10 | 192.168.137.240 | 546 | ESP | ESP (SPI=0x0d70b56c) |
| 15 | 4.629… | 192.168.137.240 | 129.192.165.10 | 194 | ESP | ESP (SPI=0x06fb00f3) |
| 16 | 4.630… | 192.168.137.240 | 129.192.165.10 | 194 | ESP | ESP (SPI=0x06fb00f3) |
| 17 | 4.630… | 192.168.137.240 | 129.192.165.10 | 210 | ESP | ESP (SPI=0x06fb00f3) |
| 18 | 4.665… | 192.168.137.240 | 129.192.165.10 | 1362 | ESP | ESP (SPI=0x06fb00f3) |
| 19 | 4.896… | 129.192.165.10 | 192.168.137.240 | 978 | ESP | ESP (SPI=0x0d70b56c) |
| 20 | 5.087… | 129.192.165.10 | 192.168.137.240 | 1138 | ESP | ESP (SPI=0x0d70b56c) |
| 21 | 5.091… | 192.168.137.240 | 129.192.165.10 | 194 | ESP | ESP (SPI=0x06fb00f3) |
| 22 | 8.579… | 129.192.165.10 | 192.168.137.240 | 1202 | ESP | ESP (SPI=0x0d70b56c) |
| 23 | 8.627… | 192.168.137.240 | 129.192.165.10 | 194 | ESP | ESP (SPI=0x06fb00f3) |
| 24 | 8.654… | 192.168.137.240 | 129.192.165.10 | 1282 | ESP | ESP (SPI=0x06fb00f3) |
| 25 | 8.655… | 129.192.165.10 | 192.168.137.240 | 178 | ESP | ESP (SPI=0x0d70b56c) |

*Figure 12 – Captured packet data from Phase 2, Sequence 1, Test 1*

© 2019 The SANS Institute

Author retains full rights.

© 2019 SANS Institute, Author Retains Full Rights



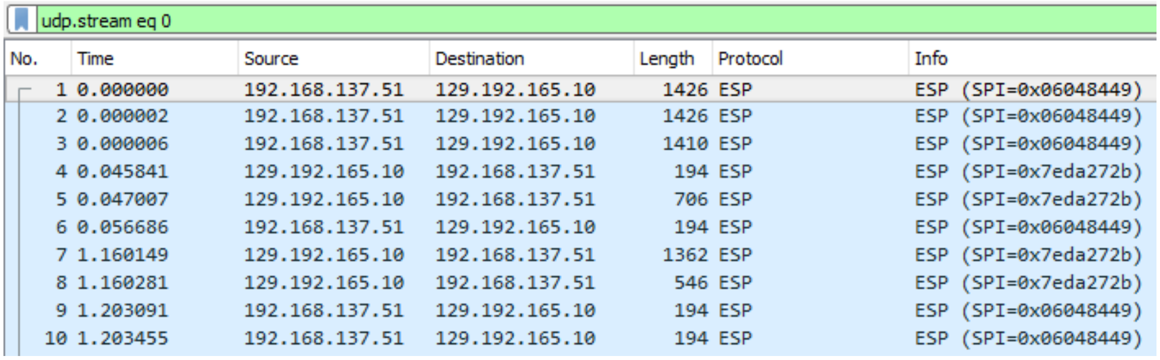
```
15 5.634… 192.168.137.240  129.192.165.10     1426 ESP     ESP (SPI=0x06fb00f3)
16 5.634… 192.168.137.240  129.192.165.10     1426 ESP     ESP (SPI=0x06fb00f3)
17 5.634… 192.168.137.240  129.192.165.10     1410 ESP     ESP (SPI=0x06fb00f3)
```

*Figure 13 – Captured packet data from Phase 2, Sequence 1, Test 2*

yielded the same consistent 178-byte length packets for the vast majority of the traffic that was passing between the nodes. Test 5 reflected the same patterns that were observed in the previous four tests during this phase of testing.

## 3.2.2 Phase 2, Sequence 2

The second sequence of tests was conducted with the mobile device placed into Airplane Mode and the Wi-Fi capability manually enabled. As in the previous sequence, the test calls were completed from the Samsung Galaxy Amp Prime 3 to the LG Fortune 2. As can be observed in Figure 14, the packet capture from these tests validated the capture of the previous sequence. Across all ten tests, consistent results showed that the call traffic was routed via an IPSec tunnel from the mobile device to the service provider. The initial three packets had a size of 1426 bytes, 1426 bytes, and 1410 bytes respectively, matching the first three packets of the ESP streams identified previously. The cause of the variance in packet size between the calls completing from the LG to the Samsung and the calls completing from the Samsung to the LG was not able to be identified through the course of these tests. Finally, the distant end server that the packets were being routed to was again observed to be 129.192.165.10. This IP address, registered to Ericsson North America Managed Services, is the same distant end server to which  all the test calls from both devices were routed.



```
udp.stream eq 0

No.  Time       Source          Destination     Length Protocol  Info
   1 0.000000   192.168.137.51  129.192.165.10    1426 ESP       ESP (SPI=0x06048449)
   2 0.000002   192.168.137.51  129.192.165.10    1426 ESP       ESP (SPI=0x06048449)
   3 0.000006   192.168.137.51  129.192.165.10    1410 ESP       ESP (SPI=0x06048449)
   4 0.045841   129.192.165.10  192.168.137.51     194 ESP       ESP (SPI=0x7eda272b)
   5 0.047007   129.192.165.10  192.168.137.51     706 ESP       ESP (SPI=0x7eda272b)
   6 0.056686   192.168.137.51  129.192.165.10     194 ESP       ESP (SPI=0x06048449)
   7 1.160149   129.192.165.10  192.168.137.51    1362 ESP       ESP (SPI=0x7eda272b)
   8 1.160281   129.192.165.10  192.168.137.51     546 ESP       ESP (SPI=0x7eda272b)
   9 1.203091   192.168.137.51  129.192.165.10     194 ESP       ESP (SPI=0x06048449)
  10 1.203455   192.168.137.51  129.192.165.10     194 ESP       ESP (SPI=0x06048449)
```

*Figure 14 – Captured packet data from Phase 2, Sequence 2, Test 1*

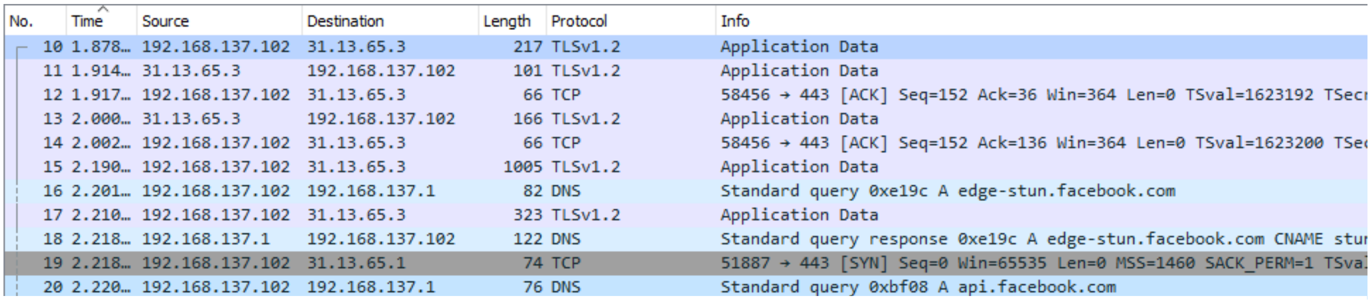
© 2019 The SANS Institute Author Retains Full Rights

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

### 3.3.1 Phase 3, Sequence 1

The third phase of testing was conducted in two sequences to provide a comparison between the VoWiFi capability resident within the test mobile devices and popular software-based VoIP solutions. The first VoIP solution to be tested in Sequence 1 of this phase was the Facebook Messenger application. This application is available either integrated into the Facebook webpage or as a stand-alone application for employment on mobile devices. Facebook Messenger was launched in 2008. Facebook has continued to optimize and improve the application, including a range of features from location sharing to integrating Short Message Service (SMS) support. In 2014, Facebook Messenger was evaluated by the Electronic Frontier Foundation and was given a security rating of two out of seven ("Secure Messaging Scorecard", 2018), receiving positive credit only for providing encryption in transit and having had an independent security audit. For these tests, the Facebook Messenger application was loaded onto the LG Fortune 2. The phone was connected to the test Wi-Fi network. The test calls were completed to the Facebook messenger application loaded onto the Samsung device. The first test call was allowed to run for approximately ten seconds from completion of the connection to disconnection. The traffic was routed out of the mobile device to an IP address associated with Facebook Ireland. The first pattern of traffic observed was primarily TLS 1.2 packets, but Domain Name Service (DNS) queries were generated from the mobile device looking for edge-stun.facebook.com and api.facebook.com (Figure 15).



```
No.  Time     Source           Destination      Length Protocol  Info
  10 1.878…  192.168.137.102  31.13.65.3          217 TLSv1.2   Application Data
  11 1.914…  31.13.65.3       192.168.137.102     101 TLSv1.2   Application Data
  12 1.917…  192.168.137.102  31.13.65.3           66 TCP       58456 → 443 [ACK] Seq=152 Ack=36 Win=364 Len=0 TSval=1623192 TSecr
  13 2.000…  31.13.65.3       192.168.137.102     166 TLSv1.2   Application Data
  14 2.002…  192.168.137.102  31.13.65.3           66 TCP       58456 → 443 [ACK] Seq=152 Ack=136 Win=364 Len=0 TSval=1623200 TSec
  15 2.190…  192.168.137.102  31.13.65.3         1005 TLSv1.2   Application Data
  16 2.201…  192.168.137.102  192.168.137.1        82 DNS       Standard query 0xe19c A edge-stun.facebook.com
  17 2.210…  192.168.137.102  31.13.65.3          323 TLSv1.2   Application Data
  18 2.218…  192.168.137.1    192.168.137.102     122 DNS       Standard query response 0xe19c A edge-stun.facebook.com CNAME stur
  19 2.218…  192.168.137.102  31.13.65.1           74 TCP       51887 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval
  20 2.220…  192.168.137.102  192.168.137.1        76 DNS       Standard query 0xbf08 A api.facebook.com
```

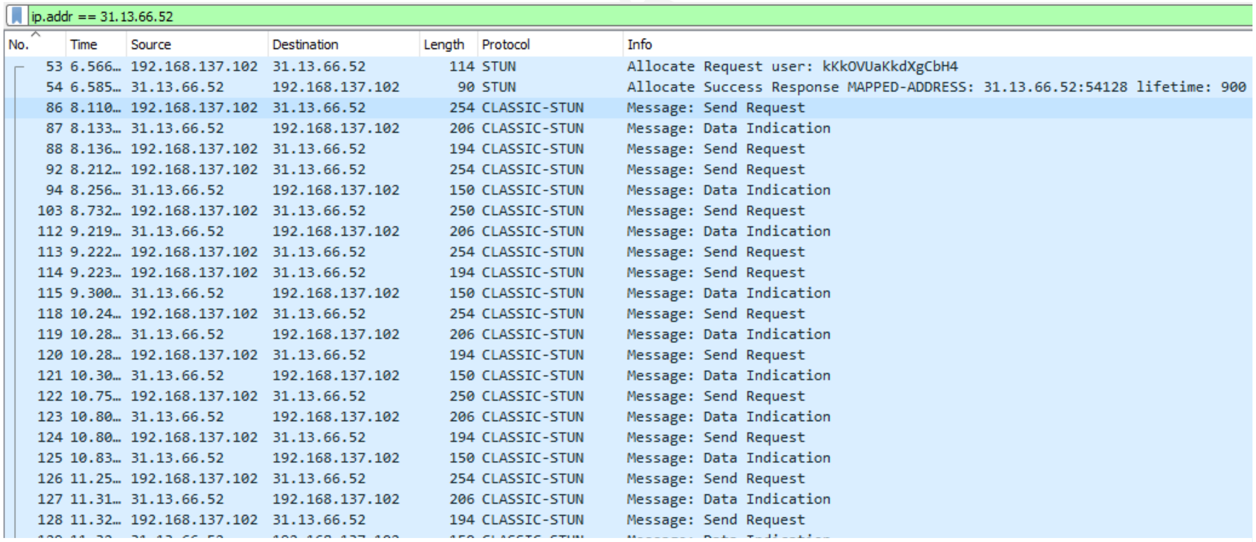
*Figure 15 – Captured packet data from Phase 3, Sequence 1, Test 1*

These DNS requests were transmitted in plain text and were easily observable to any packet capture operating on the network. The next step in the process appears to have been

© 2019 The SANS Institute Author retains full rights.

the call setup process. The preponderance of packets were either UDP datagrams (66572 Bytes) or TLS 1.2 packets (35140 Bytes). There were also several packets exchanged in Simple Traversal of UDP through NAT (STUN) protocol. The STUN protocol is a client-server protocol that was created to solve some of the issues involved in traversing a Network Address Translator (NAT) for VoIP implementations. STUN works by discovering the presence of a NAT, the type of NAT, and the IP address/port mappings assigned by the NAT (VOCAL Technologies, LTD., 2017).

The second test call completed demonstrated very similar results. Again, a DNS query was sent by the mobile device requesting the edge-stun.facebook.com server address. Also, a pattern emerged; while there were still TLS 1.2 data packets exchanged between the mobile device and Facebook, there were two seemingly connected, yet separate conversations occurring. The first appears to be the TLS traffic between the Mobile Device and the Facebook server responsible for the authentication and operation of the messenger application. There were multiple IP addresses communicating with the mobile device either in TLS v1.2 or TLS v1.3, though all were within the same Class B network as the others. The second conversation was predominantly STUN and UDP traffic and was directed between the mobile device and another Facebook server within the same Class B. This conversation can be observed in Figure 16. From this, it is determined that the UDP traffic was the voice traffic while the call maintenance was handled by the TLS traffic. Follow-on test calls on the Facebook Messenger application reflect the same traffic pattern and seem to confirm this analysis.



```
ip.addr == 31.13.66.52
No.   Time     Source           Destination      Length Protocol      Info
   53 6.566… 192.168.137.102 31.13.66.52        114 STUN          Allocate Request user: kKkOVUaKkdXgCbH4
   54 6.585… 31.13.66.52     192.168.137.102     90 STUN          Allocate Success Response MAPPED-ADDRESS: 31.13.66.52:54128 lifetime: 900
   86 8.110… 192.168.137.102 31.13.66.52        254 CLASSIC-STUN  Message: Send Request
   87 8.133… 31.13.66.52     192.168.137.102    206 CLASSIC-STUN  Message: Data Indication
   88 8.136… 192.168.137.102 31.13.66.52        194 CLASSIC-STUN  Message: Send Request
   92 8.212… 192.168.137.102 31.13.66.52        254 CLASSIC-STUN  Message: Send Request
   94 8.256… 31.13.66.52     192.168.137.102    150 CLASSIC-STUN  Message: Data Indication
  103 8.732… 192.168.137.102 31.13.66.52        250 CLASSIC-STUN  Message: Send Request
  112 9.219… 31.13.66.52     192.168.137.102    206 CLASSIC-STUN  Message: Data Indication
  113 9.222… 192.168.137.102 31.13.66.52        254 CLASSIC-STUN  Message: Send Request
  114 9.223… 192.168.137.102 31.13.66.52        194 CLASSIC-STUN  Message: Send Request
  115 9.300… 31.13.66.52     192.168.137.102    150 CLASSIC-STUN  Message: Data Indication
  118 10.24… 192.168.137.102 31.13.66.52        254 CLASSIC-STUN  Message: Send Request
  119 10.28… 31.13.66.52     192.168.137.102    206 CLASSIC-STUN  Message: Data Indication
  120 10.28… 192.168.137.102 31.13.66.52        194 CLASSIC-STUN  Message: Send Request
  121 10.30… 31.13.66.52     192.168.137.102    150 CLASSIC-STUN  Message: Data Indication
  122 10.75… 192.168.137.102 31.13.66.52        250 CLASSIC-STUN  Message: Send Request
  123 10.80… 31.13.66.52     192.168.137.102    206 CLASSIC-STUN  Message: Data Indication
  124 10.80… 192.168.137.102 31.13.66.52        194 CLASSIC-STUN  Message: Send Request
  125 10.83… 31.13.66.52     192.168.137.102    150 CLASSIC-STUN  Message: Data Indication
  126 11.25… 192.168.137.102 31.13.66.52        254 CLASSIC-STUN  Message: Send Request
  127 11.31… 31.13.66.52     192.168.137.102    206 CLASSIC-STUN  Message: Data Indication
  128 11.32… 192.168.137.102 31.13.66.52        194 CLASSIC-STUN  Message: Send Request
```

*Figure 16 – Captured packet data from Phase 3, Sequence 1, Test 2*

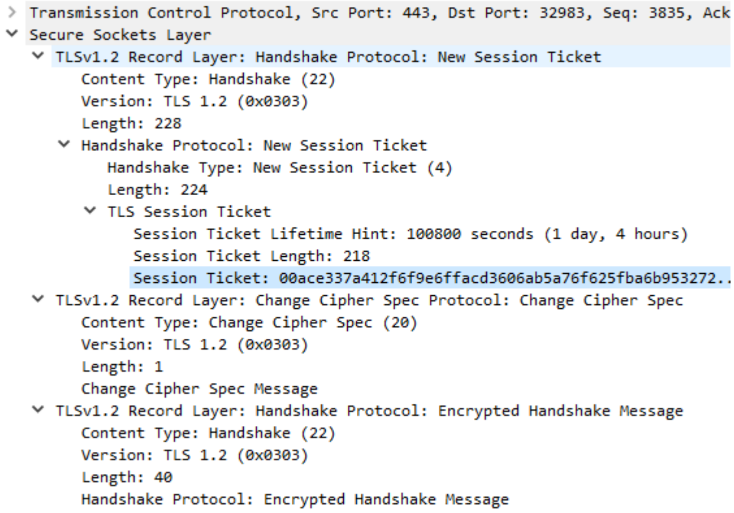
© 2019 The SANS Institute

Author retains full rights.

Author Retains Full Rights

## 3.3.2 Phase 3, Sequence 2

The second sequence of testing evaluated the security of the telephony service in Google's Hangouts application. This test was conducted using the same methodology as the previous testing for the Facebook Messenger application. The application was loaded onto the two mobile devices, and the LG Fortune 2 was used as the primary testing device. The LG was then connected to the test Wi-Fi network and a packet capture was run during the establishment of the call and the call duration.



```
> Transmission Control Protocol, Src Port: 443, Dst Port: 32983, Seq: 3835, Ack
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 228
    v Handshake Protocol: New Session Ticket
        Handshake Type: New Session Ticket (4)
        Length: 224
      v TLS Session Ticket
          Session Ticket Lifetime Hint: 100800 seconds (1 day, 4 hours)
          Session Ticket Length: 218
          Session Ticket: 00ace337a412f6f9e6ffacd3606ab5a76f625fba6b953272..
  v TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  v TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 40
      Handshake Protocol: Encrypted Handshake Message
```

*Figure 17 – TLS session ticket*

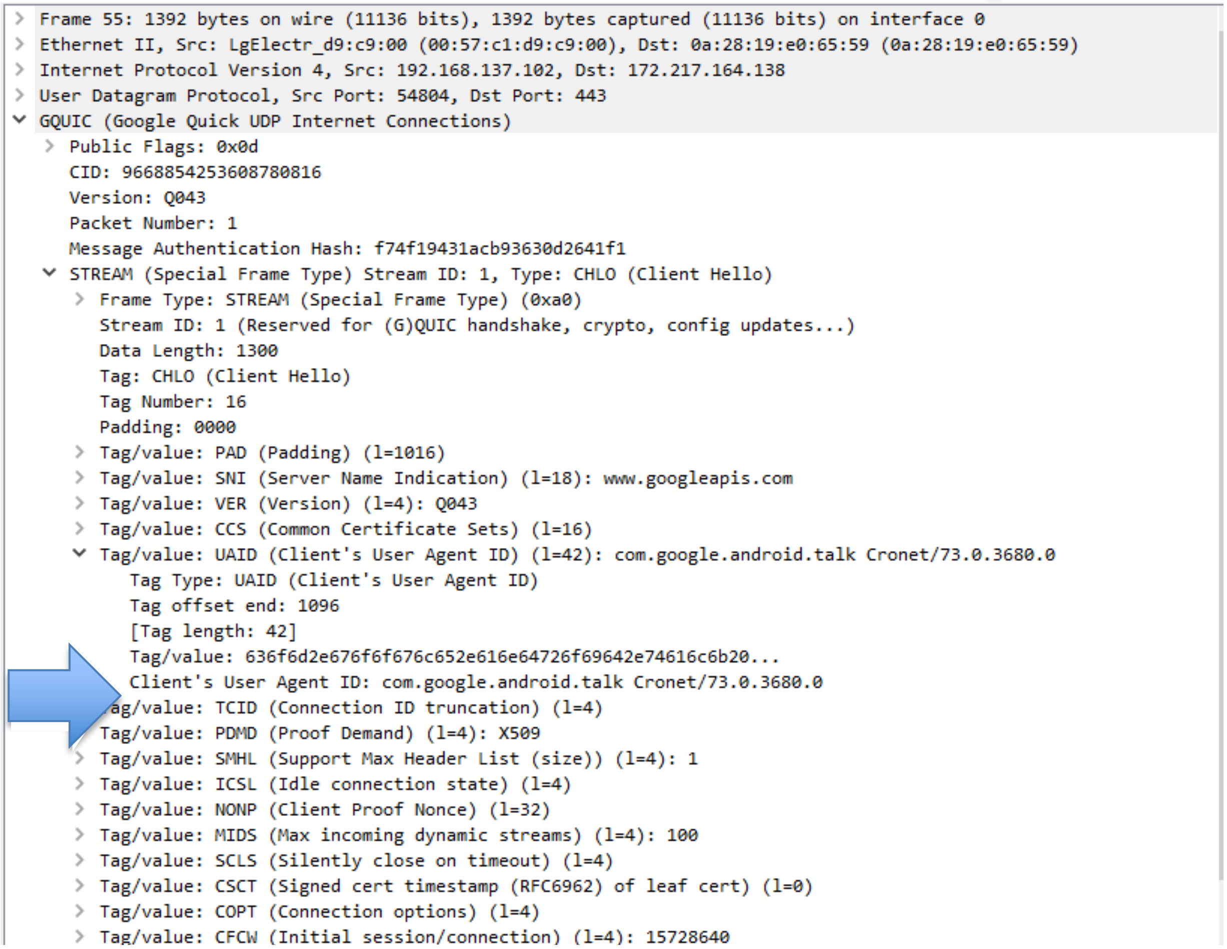
The traffic from the Hangouts application mirrored that which was originally seen in the Messenger application. Upon the initial startup of the application, a significant amount of TLS traffic traversed the network, establishing the connection between the application and the Google servers. Once the call was started, STUN and UDP packets began to be exchanged. During the traffic between the application and the server, the TLS session ticket was transferred across the compromised network (Figure 17).

A second item of interest was the employment of a new protocol that was not observed in any of the previous tests, the Google Quick UDP Internet Connection (GQUIC). GQUIC was developed by Google to serve as a secure lightweight replacement to TCP. It employs stream identifiers, or CIDs, that align the packets to the correct session exchange. GQUIC is supposed to be fully encrypted and offer security equivalent to modern implementations of SSL/TLS (Geniar, 2016; Walding, 2018). The preponderance of the GQUIC traffic was indeed encrypted. However, as can be observed in Figure 18, some relevant information

© 2019 The SANS Institute

Author retains full rights.

can still be seen in plain text. In the initial packet exchanged in a GQUIC session, the Client User Agent ID was identified as Android Talk. From this information, an attacker would



```
> Frame 55: 1392 bytes on wire (11136 bits), 1392 bytes captured (11136 bits) on interface 0
> Ethernet II, Src: LgElectr_d9:c9:00 (00:57:c1:d9:c9:00), Dst: 0a:28:19:e0:65:59 (0a:28:19:e0:65:59)
> Internet Protocol Version 4, Src: 192.168.137.102, Dst: 172.217.164.138
> User Datagram Protocol, Src Port: 54804, Dst Port: 443
⌄ GQUIC (Google Quick UDP Internet Connections)
  > Public Flags: 0x0d
    CID: 9668854253608780816
    Version: Q043
    Packet Number: 1
    Message Authentication Hash: f74f19431acb93630d2641f1
  ⌄ STREAM (Special Frame Type) Stream ID: 1, Type: CHLO (Client Hello)
    > Frame Type: STREAM (Special Frame Type) (0xa0)
      Stream ID: 1 (Reserved for (G)QUIC handshake, crypto, config updates...)
      Data Length: 1300
      Tag: CHLO (Client Hello)
      Tag Number: 16
      Padding: 0000
    > Tag/value: PAD (Padding) (l=1016)
    > Tag/value: SNI (Server Name Indication) (l=18): www.googleapis.com
    > Tag/value: VER (Version) (l=4): Q043
    > Tag/value: CCS (Common Certificate Sets) (l=16)
    ⌄ Tag/value: UAID (Client's User Agent ID) (l=42): com.google.android.talk Cronet/73.0.3680.0
        Tag Type: UAID (Client's User Agent ID)
        Tag offset end: 1096
        [Tag length: 42]
        Tag/value: 636f6d2e676f6f676c652e616e64726f69642e74616c6b20...
        Client's User Agent ID: com.google.android.talk Cronet/73.0.3680.0
     ag/value: TCID (Connection ID truncation) (l=4)
      Tag/value: PDMD (Proof Demand) (l=4): X509
    > Tag/value: SMHL (Support Max Header List (size)) (l=4): 1
    > Tag/value: ICSL (Idle connection state) (l=4)
    > Tag/value: NONP (Client Proof Nonce) (l=32)
    > Tag/value: MIDS (Max incoming dynamic streams) (l=4): 100
    > Tag/value: SCLS (Silently close on timeout) (l=4)
    > Tag/value: CSCT (Signed cert timestamp (RFC6962) of leaf cert) (l=0)
    > Tag/value: COPT (Connection options) (l=4)
    > Tag/value: CFCW (Initial session/connection) (l=4): 15728640
```

*Figure 18 – Observed plaintext information in GQUIC packet*

be able to confidently infer that the follow-on traffic sharing the same CID was launched from that application and the type of information contained in that packet flow.

## 4. Findings and Discussion

The results of the testing conducted disproved the initial hypothesis that the calls made from the mobile devices employing VoWiFi would be vulnerable to eavesdropping. The use of encryption by the mobile devices prevented attempts to decipher the content of the traffic that was traversing the network. As the private keys that the encryption relied upon were not exposed during the communication of the mobile device and the service provider,

© 2019 The SANS Institute Author retains full rights.

the traffic was not easily exploitable from a man-in-the-middle attack. Likewise, software-based VoIP solutions were also employing encryption. Prima facie, this presents a highly secure communications pathway and would be sufficient to ensure user privacy.

Despite the overall high level of inherent security that was identified, there were some security flaws which, while relatively minor, could eventually pose a security concern for the user if industry best practices are not being upheld elsewhere. First, there was extremely limited user control of the routing methodology employed by the mobile device. A second potential risk was the employment of third-party service providers to support the call infrastructure that were not identified to the users. Third, the clear text transmission of certain data packets and the consistency of certain datagrams during call setup could allow a listening device to begin to ascertain the type of traffic that was transmitted.

## 4.1 User Control

One of the most interesting initial findings was the difference in captured traffic affected by the change in location. When configuring the mobile devices for testing, there was no way to force the phone to employ the VoWiFi capability. The VoWiFi could either be enabled or disabled, but there was no user ability to designate a specific route as preferred. Settings were adjusted during the conduct of the testing phases in order to attempt to influence the packet flow across the network, including deactivating mobile data, deactivating 2G functionality, moving geographic locations to limit cellular reception to the mobile devices, and eventually disabling the RF radio on the phone by activating Airplane Mode and then manually enabling the W-FI interface. While these different efforts eventually had the desired effect for the purpose of allowing interception of the call traffic, they were excessive and required normalization of the mobile device before resuming normal operations. The presence of a work-around to a problem is not the same as having a solution to a problem. Based on the difference in traffic observed in the two environments, it is reasonable to assume there was a quality assurance capability at work which assessed the possible routes of traffic, either traditional RF to the tower or across the local Wi-Fi network and selected the route that provided the better option. Providing a more robust control framework for the VoWiFi that would allow a user to specify only

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

certain networks or networks with certain security thresholds for VoWiFi, force all calling to employ VoWiFi, and add personal authentication and encryption to calling for certain instances, would be beneficial to users. This level of user control could be explained as an inverse of the traditional 'comply to connect' concept, where users could specify the level of risk they are willing to assume in connecting to and employing Wi-Fi networks for their mobile devices.

## 4.2 Privacy Concerns

One of the findings of the research not initially predicted was the fact that the VoWiFi traffic originating from the mobile devices supported by Cricket Wireless was being routed not to Cricket or AT&T, its parent company (Welch, 2014), but rather to IP space owned by Ericsson. Ericsson Managed Services is known to provide cloud-based telephony support services to other companies such as EE and has developed specialized analytics engines to provide "Customer Experience Management" that includes support for VoWiFi. Given that AT&T and Ericsson have collaborated on the implementation of 5G technology in other sectors, it is not unlikely that AT&T also employs Ericsson solutions to support aspects of its voice network (AT&T, 2016; Ericsson, 2018; You, n.d.). Such a relationship between the companies is not inherently malicious. Most telephony services already make use of competitors' infrastructure within their networks, and this will likely continue with the rise of cloud-based services. However, it does serve to mask from the user who has access to their data once it departs their mobile device.

Over the past century, law and jurisprudence has cemented a right to privacy when employing telephony systems in the United States. While privacy is defined in different ways at different times, for this paper it is understood to be the ability of two parties engaged in a conversation to be secure from a third party intercepting or eavesdropping on the conversation. In 1967, Justice John Harlan wrote a concurring opinion for the Supreme Court decision in Katz v. United States. In this opinion, Justice Harlan outlined two requirements that must be met for protection of privacy. First, a person had to demonstrate an actual expectation of privacy, and secondly, society had to be willing to accept that expectation as reasonable (McInnis, 2011). Writing the majority opinion, Chief Justice Earl

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

Warren outlined that this right to privacy is inherent in individuals, and therefore is carried into their conversations and data traffic when it meets the previously described thresholds. Title 18 US Code, Chapter 119, Section 2511, further specifies that "any person who intentionally intercepts, endeavors to intercept, or procures any other person to intercept or endeavor to intercept, any wire, oral, or electronic communication…shall be punished" (Cornell Law School, n.d.).

Carriers must meet specific legal requirements and employ safeguards in order to licitly capture and analyze voice traffic traversing the PSTN, as detailed in the Communications Assistance for Law Enforcement Act and the Foreign Intelligence Surveillance Act (Federal Communications Commission, 2017). Other nations, especially those in the European Union, also have legislation of a similar nature which provides for the confidentiality of telephone calls (Subsentio, n.d.). These requirements have built within the public a general trust in their phone systems which has carried over to phone-like systems such as mobile devices employing VoWiFi. To the user, there is practically no difference between their cell phone and their landline other than a number and mobility. However, as VoWiFi systems are classified as information services rather than telephony services (Hearn, 2004), the fact that this traffic is being passed through an undisclosed third-party opens several potential privacy concerns.

Further, while the issues of privacy within the information domain are coming to the public attention through legislation such as the General Data Protection Regulation (GDPR) in the EU and the recent Facebook hearings in the US Congress, no worldwide or national scope consensus has been reached to protect user privacy to the same level of assurance as that of the PSTN network. Indeed, in 2014, Cricket Wireless was accused of actively preventing the transmission of encrypted email across its mobile network (Hoffman-Andrews, 2014; Scola & Soltani, 2014). In the same year, Verizon was found to be tracking its customers' web surfing habits and disclosing the data to third-parties (Hoffman-Andrews, 2014). It is well documented (Castillo, 2018) that companies that trade in information are employing metadata collected on users to sell targeted advertising capabilities. With the rise of privacy concerns industry-wide, the employment of third-parties by service providers that users are not aware of creates an area of risk for users and service providers.

© 2019 The SANS Institute Author retains full rights.

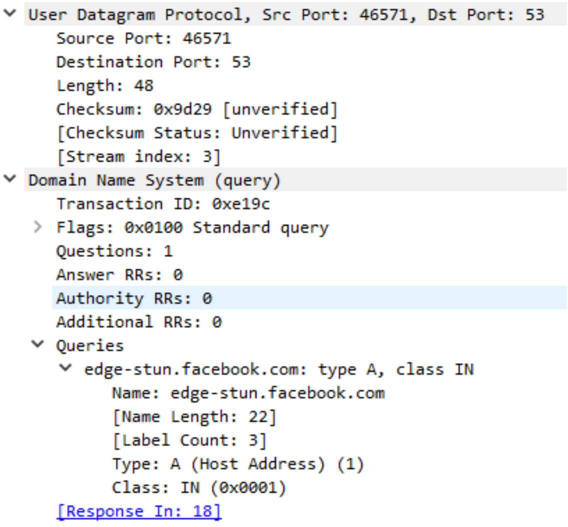
© 2019 The SANS Institute, Author Retains Full Rights

© 2019 The SANS Institute, Author Retains Full Rights

Of greater concern, the fact that Cricket Wireless felt enabled to employ third-party services to power their VoWiFi capabilities without alerting their users in a rapidly-understandable manner opens the possibility that other carriers could employ less reputable or less secure third-party providers for their own services. These third-party organizations may not have the technical knowledge or resources to fully secure their infrastructure, leaving open the potential for malicious actors to cull user information from their domains. Additionally, with no user-oversight to where their calls are being routed, it is not easily discernable where the services supporting a user's call traffic are hosted. Services hosted in nations with fewer consumer protections could open avenues for nation-states to seize the VoWiFi traffic for exploitation. This risk is understood for users who frequently travel abroad, but the employment of third-party cloud providers creates the potential for traffic data on a call placed between two persons in one nation to be routed through or stored in a separate nation, one which does not respect the same level of privacy as the nation in which the calls were actually placed.

## 4.3 Clear Text Data Packets

A second concern identified through this research was the transmission of clear text data packets which could allow an attacker to guess the type of traffic that was traversing the network. During the third phase of testing when Facebook Messenger was assessed, it was observed that DNS requests were transmitted by the mobile device to determine the destination for follow-on data transmission. As can be seen in Figure 19, the DNS request asked for the STUN server at the Facebook domain. As previously described in this paper, STUN protocol is used exclusively for addressing issues arising from non-publicly routable IPs seeking to connect VoIP calls. Therefore, anyone



```
˅ User Datagram Protocol, Src Port: 46571, Dst Port: 53
    Source Port: 46571
    Destination Port: 53
    Length: 48
    Checksum: 0x9d29 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
˅ Domain Name System (query)
    Transaction ID: 0xe19c
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ˅ Queries
    ˅ edge-stun.facebook.com: type A, class IN
        Name: edge-stun.facebook.com
        [Name Length: 22]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 18]
```

*Figure 19 – DNS request*

who is listening to the traffic traversing that network would be able to guess that a VoIP call is being placed. Given the Facebook domain, it is not a difficult deduction to assess from that one packet that the device is using the Facebook messenger application in order to complete a call. Google Hangouts also transmitted packets with clear text information. The initial 'Client Hello' packet, roughly the GQUIC equivalent of a TCP SYN packet, transmitted from the mobile device upon attempting to establish the call included within it information identifying the application that was transmitting the packet. While given the presence of GQUIC packets it would not be difficult to deduce that the originating application was a Google product, the specification of the specific product identified in this packet makes it simple for an attacker to deduce the type of traffic.

Even in the organic VoWiFi calls placed from the mobile devices had certain indicators that could lead the informed attacker to deduce the VoIP traffic was being passed. During testing the initial three packets of each call followed a consistent pattern of packet length. Additionally, the encapsulated packets traversing the network at the high rates of bytes per second that were observed are consistent with well-established VoIP protocols and bandwidth requirements. From this it would be fairly trivial for an attacker to piece together this information and determine the length of a phone call and the service provider of a user.

## 4.4 TLS Vulnerabilities

A final concern involved the implementation of TLS v1.2 by the software VoIP implementations and VoWiFi calling. TLS v1.2 was developed to address certain issues found in TLS v1.1 and allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery (Dierks & Rescorla, 2008). TLS v1.2 was adopted as the standard for PCI-compliant transactions in 2008 (Goodspeed, 2015) and is the minimum standard for most internet applications requiring security. TLS v1.2 employs the so-called "Perfect Forward Security", designed to defeat the potential for advanced actors to intercept and decrypt traffic that is passed along major transport lines (McCullagh, 2013). However, to increase connection speed and further to prevent the loss of a session when a client is connecting to multiple load balanced servers, TLS v1.2 employs a session ticket. The server takes the cryptographic specifications of the

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

session that it established to the client and uses that to generate an encrypted ticket that is transmitted to the client. The client is not able to decrypt or modify the ticket but holds it in storage for the next time it needs to connect to the server. When the client initiates a new connection, it transmits the session ticket back which the server is then able to verify and decrypt to provide the necessary session information without the requirement to perform a full Diffie-Helman key exchange (Rutishauser, 2017). While admittedly decreasing latency for the server connection, this system presents a potential vulnerability and is one of the weakest points of TLS v1.2. TLS v1.2 always encrypts session tickets with AES-128-CBC and executes integrity protection using HMAC-SHA-256 (Rutishauser, 2017). If a malicious actor was able to compromise the service provider's server and exfiltrate the key for the session ticket, that actor could decrypt the TLS session from the client.

As best practice, it is necessary to frequently rotate the key employed by the session ticket. Twitter rotates keys for their session tickets every 12 hours. Cloudflare does so every hour (Rutishauser, 2017). However, as was observed in Figure 13, the session ticket for Google Hangouts was accepted for 28 hours. The longer time to live increases the potential window of exposure during which an attacker can intercept and collect call traffic that is decryptable with a single key. While not providing a real-time vulnerability, any sensitive information conveyed through the network is still vulnerable to exploitation. An attacker who compromised the server that the mobile device is communicating to and exfiltrated stored session ticket information could decrypt the captured call traffic.

## 5. Conclusion

This research project was initially undertaken to assess the vulnerabilities of current implementations of VoWiFi that are organic to mobile devices and to then propose solutions to help mitigate the exposures identified. The original hypothesis that VoWiFi was highly vulnerable and would require additional education and technical control to decrease risk was demonstrated to be unfounded. The mobile devices encrypted all call setup and voice traffic and did not exchange significant packet data in plain text. However, several security concerns were still identified based on the analysis of the data collected which should inform users of the inherent risk they accept every day when employing

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

mobile devices and the way they expose themselves when they assume technology is safe without first verifying the method of operation of that technology. Overall, service providers should seek to improve the amount of user control that is offered on their mobile devices to allow the user to decide what level of risk is acceptable to themselves.

© 2019 The SANS Institute, Author Retains Full Rights

© 2019 The SANS Institute Author retains full rights.

## Resources

Apple. (2015, September). IOS Security. Retrieved from https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf

Arora, M. (2012, May 07). How secure is AES against brute force attacks? Retrieved from https://www.eetimes.com/document.asp?doc_id=1279619#

AT&T. (2016, December 05). AT&T Launches First 5G Business Customer Trial with Intel and Ericsson. Retrieved from https://about.att.com/story/att_launches_first_5g_business_customer_trial_with_intel_and_ericsson.html

Bell, A. G. (1876). *U.S. Patent No. 174465*. Washington, DC: U.S. Patent and Trademark Office.

Bellamy, J. (1991). *Digital Telephony* (2d ed.). New York: John Wiley & Sons.

Brodkin, J. (2014, January 30). AT&T plan to shut off Public Switched Telephone Network moves ahead at FCC. Retrieved January 28, 2019, from https://arstechnica.com/tech-policy/2014/01/att-plan-to-shut-off-public-switched-telephone-network-moves-ahead-at-fcc/

Calculating Voice Bandwidth Requirements. (2007, July). Retrieved from http://www.vertical.com/media/support/an-xip07010_calculating-voice-bandwidth-requirements.pdf

Calypso Wireless, Inc. (2005, July 22). Calypso Wireless' Dual Mode WiFi/GSM-GPRS VoIP Cellular Phones Available for Demonstration. Retrieved January 28, 2019, from https://globenewswire.com/news-release/2005/07/22/330554/82551/en/Calypso-Wireless-Dual-Mode-WiFi-GSM-GPRS-VoIP-Cellular-Phones-Available-for-Demonstration.html

Castillo, M. (2018, March 19). Here's how Facebook ad tracking and targeting works. *CNBC*. Retrieved from https://www.cnbc.com/2018/03/19/how-facebook-ad-tracking-and-targeting-works.html

Clark, B. (2013). *RTFM: Read Team Field Manual.* Middletown DE.

Collier, M., & Endler, D. (2014). *Hacking Exposed: Unified Communications and VoIP security secrets & solutions* (2d ed.). McGraw Hill.

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

Cornell. (n.d.). 18 U.S. Code § 2511 - Interception and disclosure of wire, oral, or electronic communications prohibited. Retrieved from https://www.law.cornell.edu/uscode/text/18/2511

Delgado, C. (2017, March 25). How to perform a Man-in-the-middle (MITM) attack with Kali Linux. Retrieved from https://ourcodeworld.com/articles/read/422/how-to-perform-a-man-in-the-middle-mitm-attack-with-kali-linux

Dierks, T., & Rescorla, E. (2008, August). The Transport Layer Security (TLS) Protocol Version 1.2. Retrieved from https://tools.ietf.org/html/rfc5246

Ericsson. (2018, May 21). Ericsson Expert Analytics selected by EE to improve customer experience. Retrieved from https://www.ericsson.com/en/press-releases/2018/5/ericsson-expert-analytics-selected-by-ee-to-improve-customer-experience

ESP, Encapsulated Security Protocol. (n.d.). Retrieved from http://www.networksorcery.com/enp/protocol/esp.htm

Featherly, K. (2016, November 28). ARPANET. Retrieved January 28, 2019, from https://www.britannica.com/topic/ARPANET

Federal Communications Commission. (2005, May 19). *Commission Requires Interconnected VoIP Providers to Provide Enhanced 911 Service* [Press release]. Retrieved January 28, 2019, from https://www.fcc.gov/document/commission-requires-interconnected-voip-providers-provide-enhanced-911

Federal Communications Commission. (2017, October 06). Communications Assistance for Law Enforcement Act. Retrieved from https://www.fcc.gov/public-safety-and-homeland-security/policy-and-licensing-division/general/communications-assistance

Geniar, M. (2016, December 04). Google's QUIC protocol: Moving the web from TCP to UDP. Retrieved from https://ma.ttias.be/googles-quic-protocol-moving-web-tcp-udp/

Goodspeed, L. (2015, December 18). PCI SECURITY STANDARDS COUNCIL REVISES DATE FOR MIGRATING OFF VULNERABLE SSL AND EARLY TLS ENCRYPTION. Retrieved from https://www.pcisecuritystandards.org/pdfs/15_12_18_SSL_Webinar_Press_Release_FINAL_(002).pdf

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

Handley, M., Schulzrinne, H., Schooler, E., & Rosenberg, J. (1999, March). SIP: Session Initiation Protocol. Retrieved from https://www.ietf.org/rfc/rfc2543.txt

Hearn, T. (2004, October 19). Powell: States Can't Regulate VoIP. Retrieved from https://www.multichannel.com/news/powell-states-can-t-regulate-voip-270919

Hoffman-Andrews, J. (2014, November 03). Verizon Injecting Perma-Cookies to Track Mobile Customers, Bypassing Privacy Controls. Retrieved from https://www.eff.org/deeplinks/2014/11/verizon-x-uidh

Hoffman-Andrews, J. (2014, November 11). ISPs Removing Their Customers' Email Encryption. Retrieved from https://www.eff.org/deeplinks/2014/11/starttls-downgrade-attacks

How to Decrypt SSL and TLS Traffic Using Wireshark. (2014, August 5). Retrieved from https://support.citrix.com/article/CTX116557

Huculak, M. (2016, March 17). How to set up and manage a Network Bridge connection on Windows 10. Retrieved from https://www.windowscentral.com/how-set-and-manage-network-bridge-connection-windows-10

Iveson, S. (2014, November 05). Using ssldump to Decode/Decrypt SSL/TLS Packets. Retrieved from https://packetpushers.net/using-ssldump-decode-ssltls-packets/

Johnston, A. B. (2016). *SIP: Understanding the Session Initiation Protocol* (4th ed.). Norwood, MA: Artech House.

Kravets, D. (2017, September 22). NSA Leak Vindicates AT&T Whistleblower. Retrieved from https://www.wired.com/2013/06/nsa-whistleblower-klein/

Langley, A. (2013, June 27). ImperialViolet. Retrieved from https://www.imperialviolet.org/2013/06/27/botchingpfs.html

La Porta, L. (2018, May 23). 4 ways hackers are infiltrating phones with malware on Android phones. Retrieved from https://www.wandera.com/mobile-security/mobile-malware/malware-on-android/

LG. (n.d.). Airplane Mode. Retrieved from https://www.lg.com/us/mobile-phones/VS985/Userguide/156.html

Lin, Z. (2018, August 27). TLS Session Resumption: Full-speed and Secure. Retrieved from https://blog.cloudflare.com/tls-session-resumption-full-speed-and-secure/

McCann, S. (n.d.). Libpcap: An Architecture and Op2miza2on Methodology for Packet Capture. Reading presented at Sharkfest '11.

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute Author Retains Full Rights

McCullagh, D. (2013, June 26). Data, meet spies: The unfinished state of Web crypto. Retrieved from https://www.cnet.com/news/data-meet-spies-the-unfinished-state-of-web-crypto/

McInnis, T. (2011). The Changing Definition of Search or Seizure. *American Bar Association*. Retrieved from https://www.americanbar.org/content/dam/aba/images/public_education/presentations/ChangingDefinitionsofSearch.pdf.

Nortel Networks. (1999). *Voice Fundamentals.* Nortel Networks.

Northcutt, S., & Novak, J. (2003). *Network Intrusion Detection* (3d ed.). Indianapolis: New Riders.

Number of smartphone users in the U.S. 2010-2022. (n.d.). Retrieved December 4, 2018, from https://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/

Pepper, R. (2014, January 27). The History of VoIP and Internet Telephones. Retrieved January 28, 2019, from https://getvoip.com/blog/2014/01/27/history-of-voip-and-internet-telephones/

Protecting data for the long term with forward secrecy. (2011, November 22). Retrieved from https://security.googleblog.com/2011/11/protecting-data-for-long-term-with.html

Rutishauser, D. (2017, June 29). About TLS Perfect Forward Secrecy and Session Resumption. Retrieved from https://blog.compass-security.com/2017/06/about-tls-perfect-forward-secrecy-and-session-resumption/

Sanders, C. (2017). *Practical Packet Analysis* (3d ed.). San Francisco: No Starch Press.

Shannon, C. E., & Weaver, W. (1949). *The Mathematical Theory of Communication.* Urbana: University of Illinois Press.

Scola, N., & Soltani, A. (2014, October 28). Mobile ISP Cricket was thwarting encrypted emails, researchers find. Retrieved from https://www.washingtonpost.com/news/the-switch/wp/2014/10/28/mobile-isp-thwarted-customers-attempts-to-send-encrypted-e-mails-research-finds/?noredirect=on&utm_term=.fb9ff7fc0bc9

Secure Messaging Scorecard. (2018, March 13). Retrieved from https://www.eff.org/node/82654

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

Singh, H. (2019, January 19). Evil Twin Attack [A Step by Step Guide] (Updated 2018). Retrieved from https://rootsh3ll.com/evil-twin-attack/

Subsentio. (n.d.). International Lawful Surveillance. Retrieved from http://www.subsentio.com/international/international-lawful-surveillance/

Transport Layer Security (TLS) Session Resumption without Server-Side State. (n.d.). Retrieved from https://tools.ietf.org/html/rfc5077

Valsorda, F. (2017, September 28). We need to talk about Session Tickets. Retrieved from https://blog.filippo.io/we-need-to-talk-about-session-tickets/

Valsorda, F. (2016). Ticketbleed (CVE-2016-9244). Retrieved from https://filippo.io/Ticketbleed/

VOCAL Technologies, Ltd. (2017). Classic STUN: Simple Traversal of UDP Through NAT Retrieved from https://www.vocal.com/networking/classic-stun-simple-traversal-of-udp-through-nat/

Vmware, Inc. (2019, February 27). VMware Empowers Communication Service Providers with 5G-Ready Telco Cloud Infrastructure. Retrieved from https://www.globenewswire.com/news-release/2019/02/27/1743132/0/en/VMware-Empowers-Communication-Service-Providers-with-5G-Ready-Telco-Cloud-Infrastructure.html

Walding, A. (2018, March 08). Is there a lot of QUIC/GQUIC in your Packet Captures? Retrieved from https://www.cellstream.com/reference-reading/tipsandtricks/381-is-there-a-lot-of-quic-in-your-packet-captures

Walding, A. (2018, March 08). Using Wireshark to Analyze QUIC/GQUIC Traffic. Retrieved from https://www.cellstream.com/reference-reading/tipsandtricks/382-using-wireshark-to-analyze-quic-traffic

Welch, C. (2014, March 13). FCC approves AT&Ts purchase of Leap Wireless, says its in the public interest. Retrieved from https://www.theverge.com/2014/3/13/5505798/fcc-approves-att-purchase-of-leap-wireless

What is TLS & How Does it Work? | ISOC Internet Society. (n.d.). Retrieved from https://www.internetsociety.org/deploy360/tls/basics/

Win32 Disk Imager. (n.d.). Retrieved from https://sourceforge.net/projects/win32diskimager/

© 2019 The SANS Institute Author retains full rights.

© 2019 The SANS Institute, Author Retains Full Rights

Wireshark. (n.d.). About. Retrieved January 28, 2019, from https://www.wireshark.org/

Wireshark. (n.d.). Help to set up a "pass through bridge" sniffer. Retrieved from https://ask.wireshark.org/question/1073/help-to-set-up-a-pass-through-bridge-sniffer/

Making a Kali Bootable USB Drive. (n.d.). Retrieved from https://docs.kali.org/downloading/kali-linux-live-usb-install

Xie, T., Tu, G., Yin, B., Li, C., Peng, C., Zhang, M., . . . Liu, X. (2018, November 29). The Untold Secrets of Operational Wi-Fi Calling Services: Vulnerabilities, Attacks, and Countermeasures. Retrieved November 30, 2018, from https://arxiv.org/pdf/1811.11274.pdf

You, J. (n.d.). Ericsson Expert Analytics Solution Brief. Retrieved from https://www.ericsson.com/assets/local/digital-services/offerings/network-automation/solution-brief_expert-analytics.pdf

Zetter, K. (2016, March 28). The FBI Drops Its Case Against Apple After Finding a Way Into That iPhone. Retrieved from https://www.wired.com/2016/03/fbi-drops-case-apple-finding-way-iphone/

© 2019 The SANS Institute, Author Retains Full Rights

© 2019 The SANS Institute Author retains full rights.