

ECE 4112 Internetwork Security
Lab NEW: Voice Over Internet Protocol (VoIP)

Group Number: _____

Member Names: _____

Date Assigned: TBD

Date Due: TBD

Last Edited: December 6, 2005

Lab Authored by: Luis Miguel Cortés Peña
Gédéon Kamga

Please read the entire lab and any extra materials carefully before starting. Be sure to start early enough so that you will have time to complete the lab. Answer ALL questions and be sure you turn in ALL materials listed in the **Turn-in Checklist** ON or BEFORE the **Date Due**.

Goal: The goal of this lab is to show you how VoIP works and demonstrate some of its vulnerabilities.

Summary: This lab will introduce you to VoIP, how to listen to VoIP conversations in both Windows and Linux. You will also get to perform ARP poisoning to obtain a VoIP conversation.

Background and Theory:

Introduction

VoIP (voice over IP - that is, voice delivered using the Internet Protocol) is a term used in IP telephony for a set of facilities for managing the delivery of voice information using the Internet Protocol (IP). Voice over IP uses Internet Protocol (IP) to carry voice as packets over a packet-switched data network. Voice information is then sent in digital form in discrete packets rather than in the traditional circuit-switched protocols of the public switched telephone network (PSTN). A major advantage of VoIP and Internet telephony is that it increases operating efficiency, avoiding expensive communication costs and reducing unnecessary expenses that occur with ordinary telephone service [1][2].

VoIP Security

VoIP uses the Internet for phone service, bypassing expensive long-distance communication providers, which results in significant savings. However, as with most technology advancements, if not set up and deployed correctly, a VoIP solution can expose an organization to security breaches (Figure 1). For instance, when VOIP is used externally, gateway technologies convert data packets from the IP network into voice before sending them over a public switched telephone network. When VOIP is used internally, the gateways basically route packetized voice data between the source and the destination. A potential issue is that VOIP gateways can be hacked into by malicious attackers in order to make free telephone calls. In addition, attackers can infiltrate phone conversations and steal confidential data in the same way they would hack an IT system. Spammers can also use denial of service attacks to render the phone system useless. To deploy a VoIP solution, one needs to assure that the solution is safe, secure and protected from outside threats.

Below is a list of typical attacks that a VoIP system might face [1][3].

Toll Fraud: The IP version of the classic attack by a person pretending to be an employee or Console Cracking (asking the operator for an outside trunk) to make long distance calls. However, the attacker impersonates a valid user and IP address by plugging in their phone or spoofing the MAC Ethernet address.

Eavesdropping: The attacker sniffs (taps into the LAN wireline or Wi-Fi connection) to intercept voice messages. Available tools such as VOMIT-Voice Over Misconfigured Internet Telephony allow performing this function.

Call Hijacking: Attacker spoofs a SIP Response redirecting the caller to a rogue SIP address and intercepts the call.

Resource Exhaustion: Also Known As DOS [Denial Of Service] attack. This attack reduces the number of available IP addresses, bandwidth, processor memory, and other router/server functions.

Message Integrity: MIM [Man-In-the-Middle] attack to intercept, alter, or redirect call.

Message Type Attacks: Attacker bombards (repetitive) SIP server with BYE or CANCEL messages or ICMP [Internet Message Control Protocol] "port unreachable" messages.

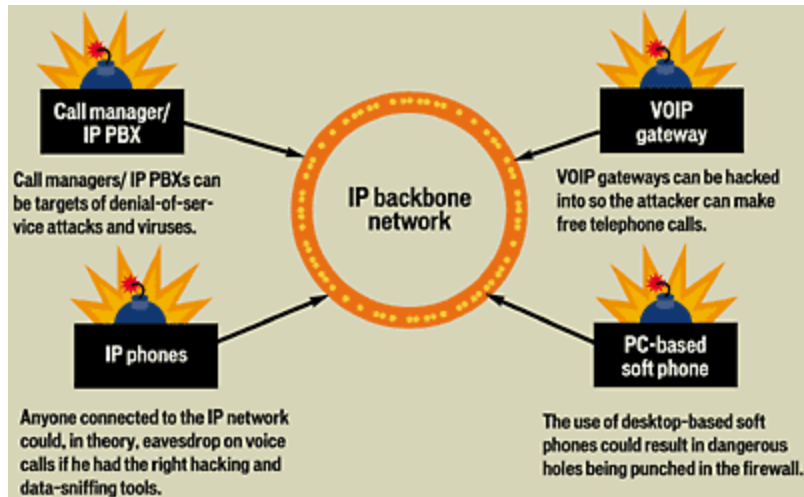


Figure 1. VoIP Vulnerabilities [1].

VoIP Session Initiation Routine

In VoIP, communication is established through a protocol called SIP. SIP (session initiation protocol), the protocol for VoIP is an application-layer control protocol for creating, modifying, and terminating sessions with one or more participants. When a user agent (UAC), a client wants to initiate a session with another user (UA), it sends an INVITE request to the SIP proxies sever, asking for a session creation. This server then forwards the request to the SIP proxy server (UAS) of the desired user agent. The UAS will in turn send an INVITE request to the user to determine if he wants to accept the invitation. If the callee accepts the invitation, it sends an ACK. The caller sends an ACK to indicate that the handshake is done and session is to be established. The SIP user agent – a combination of the UAC and the UAS – can also allow peer-to-peer calls to be made using a client-server protocol.

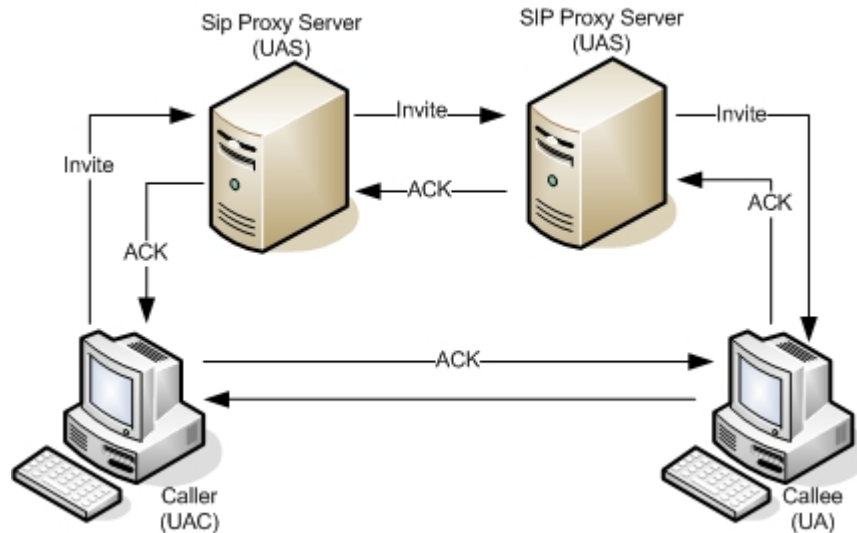


Figure 2: VoIP Session Initiation Routine.

Prelab Questions: None.

Lab Scenario: You will need your hard drive for this lab. You will be using two RedHat WS 4.0 machines running on your computer. Ensure that you pick a computer equipped with both an internal sound card and an Ensoniq AudioPCI installed by a TA. One of the RedHat 4.0 WS machine will be a virtual machine which you will download from NAS. This virtual machine will contain all the necessary software installed. The other can be a physical machine (your RedHat WS 4.0 host machine) or a copy of the virtual machine downloaded from NAS. If you decide to use your RedHat WS 4.0 host machine, you will have to install minisip (explained later in this lab manual). You will also need to use your Windows XP virtual machine to perform some of the attacks.

Section 1: Minisip

Minisip is a free SIP user agent. It features services such as Secure VoIP, SIP, MIKEY, RTP, SRTP, SDP, Video Telephony, Push-to-talk. This tool can be downloaded here:

<http://www.minisip.org>

The description from the website says:

“Minisip is a SIP User Agent ("Internet telephone") developed at KTH currently running on Linux. Keywords: Secure VoIP; SIP; MIKEY; RTP; SRTP; SDP; Video Telephony; Push-to-talk. You can download it for free from the [download page](#).

Minisip is developed by Ph.D and Master students at the Royal Institute of Technology, KTH, Stockholm, Sweden.

The source code is available as a number of libraries under the GNU Lesser General Public License (LGPL) and applications under the GNU General Public License (GPL).“

Section 1.1 Installing Minisip (Physical RedHat WS 4.0 Machine)

Installing minisip is a very tedious job because of all its dependencies and environmental variables needed. For this reason it is recommended that you use a script named minisipsintaller created for you. In order to do so, perform the following steps as root:

- Connect to nas4112
- Go to the lab directory
- Copy the directory named software to your home directory (/root/)
- Go to the directory software/minisip and run the script minisipinstaller:

```
# cd /root/software/minisip  
# ./minisipinstaller
```

NOTE: If the script does not finish successfully, you might want to run it again since there might be a dependency which is out of order and makes other dependencies fail.

This should take approximately 60-90 minutes. When this finishes, the script should have created a script named runminisip located in the root directory. Run this script which should open minisip without any problem.

```
# ~/runminisip
```

Now we will fix some dependencies so that vomit can be run:

```
# cd ~/software/lib/  
# tar xvfz libdnet-*.tar.gz  
# cd libdnet*  
# ./configure  
# make  
# make install  
# cd ..  
# tar xvfz libevent-*.tar.tar  
# cd libevent*  
# ./configure  
# make  
# make install
```

Next you need to install vomit:

```
# cd ~/software/vomit
# tar xvfz vomit*.tar.gz
# cd vomit*
# ./configure
# make
# make install
```

Finally you need to copy the file named waveplay-20010924.tar.tar to your root directory and make it:

```
# cp /root/software/waveplay/waveplay-20010924.tar.tar /root/waveplay-
20010924.tar.tar
# cd ~
# tar xvfz waveplay-20010924.tar.tar
# cd waveplay-20010924
# make
```

The computer is now ready. You should test that minisip is running.

Section 1.2 Configuring Minisip

Minisip will be configured to work as a P2P (peer-to-peer) VoIP service. This means that there will be no intermediary server that authenticates the user and tell each other IP address to accomplish the connection. Therefore previous knowledge of each other IP address is needed. The following diagram shows how the connection should be done:

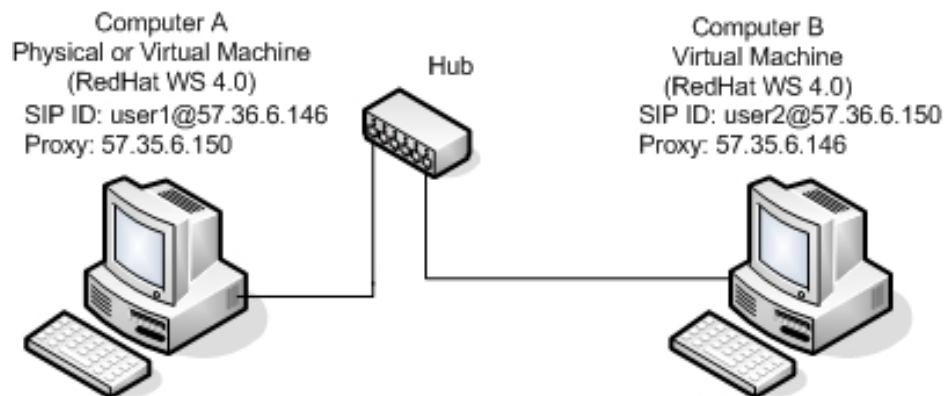


Figure 3. P2P connection diagram.

To accomplish this, the first step is to open minisip in your RedHat physical or virtual machine A:

~/runminisip

If minisip fails to open, delete the file named .minisip-conf in the root's directory and try again:

rm ~/.minisip-conf

- Now click File > Preferences. You will see a screen which displays the sip account settings.
- Select the default one and click on Edit...
- Enter anything you want for the Account name (i.e. VoIP) and enter your SIP URI: as username@YourIPHere. Usually YourIPHere is replaced by the domain of the VoIP provider. You can find your IP by executing the command *ifconfig* in a shell.
- Now type the IP of the computer you will be connecting to (which is different from the host machine's IP as shown on Figure 3).

Your screen should now look similar to the following screen shot.

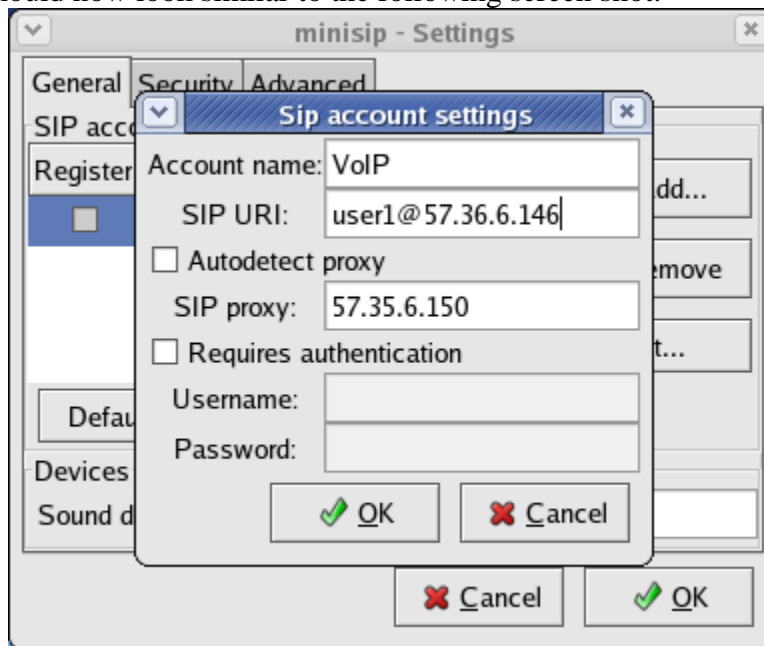


Figure 4: Minisip - Settings.

- Uncheck the box labeled "Requires authentication" since we are not connecting to an actual proxy server.
- Now hit OK on the "Sip Account settings"
- Ensure that Sound device is mapped to /dev/dsp
- Click OK on "minisip - Settings."
- Now we need to setup the contact information. Right click on the contact window and select "Add a Contact."
- Enter your group number in the Name, VoIP in Type and enter the username@IPAddress of the computer you are connecting to.

The contact information window should look like the following figure:

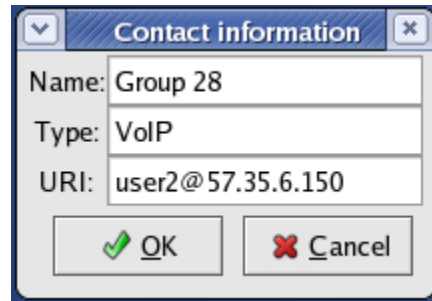


Figure 5: Contact Information.

- Hit OK when done.

Your minisip is now setup to connect to computer B. However, computer B needs to be setup also. Perform the same steps above on your RedHat WS 4.0 virtual machine computer B. Before powering on the machine:

- Select the Red Hat Enterprise Linux 4 tab.
- Power off the computer if its on
- Select VM > Settings > Sound Adapter
- Under connection, select dps1
- Make sure Connect at power on is checked
- Click OK
- Power on Machine

Section 1.2 Audio Setup

Now we need to setup the sound so that you can talk and listen to the conversation. To do this you need to mute the microphone (so that you do not hear your self talking) and enable your speakers.

- Go to Start > Sound & Video > Volume Control.

Change your settings and your Volume Control window should look like the following:

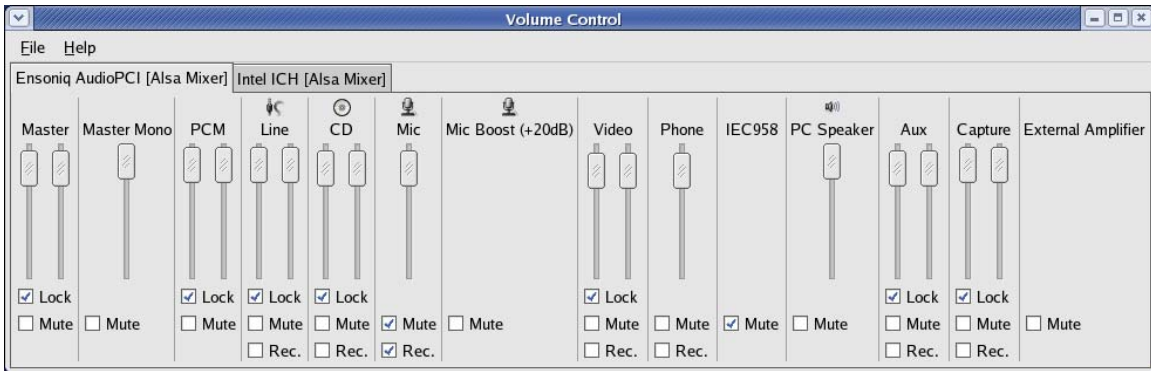


Figure 6: Volume Control Settings

Do this for both computers. When everything is setup, hit call on minisip on one of the computers and get a screen shot of minisip on the computer receiving the phone call.

Screenshot 1: Minisip receiving phone call.

If you encounter audio problems, it helps that you mute and un-mute both the microphones and the speakers.

Have your TA check you of for the VoIP conversation accomplished.

TA Check Off : Minisip correctly setup and working conversation.

Section 2: SIP URI Spoofing

Now you will change your settings such that the person receiving the phone call thinks the caller is administrator@vonage.com [hint: where did you input your SIP URI?]

Screenshot 2: SIP URI as administrator@vonage.com

This type of attack is a little more sophisticated in real life since a proxy server is involved for authentication.

Section 3: Vomit

Vomit, just in case you were wondering, stands for Voice Over Misconfigured Internet Telephones. Vomit converts a captured package into a wave file. The utility can be downloaded at:

<http://vomit.xtdnet.nl/>

The description from the web site says:

“The **vomit** utility converts a Cisco IP phone conversation into a wave file that can be

played with ordinary sound players. Vomit requires a tcpdump output file. Vomit is not a VoIP sniffer also it could be but the naming is probably related to H.323.”

- On either computer A or computer B, run Ethereal and begin capturing packets on eth0.
- Establish a VoIP connection on both computer A and computer B and have a conversation.
- Now stop capturing packets and save it to your home directory (/root) in a file named phone.dump.
- Get a screen shot of Ethereal displaying the connection Invite and ACK.

Screenshot 3: Ethereal displaying SIP Invite and Ack.

Open a shell and cd in to your home directory:

```
# cd ~
```

Now run vomit with the following command:

```
# vomit -r phone.dump | waveplay-20010924/waveplay -S8000 -B16 -C1
```

Listen to the output.

Question 1: Were vomit and waveplay able to playback the file?

Question 2: How is the quality of the playback compared to that of the actual conversation?

Section 4: Cain and Able

You have seen how linux plays back conversations, now you will use windows to both sniff and playback conversations. We use what people call window’s “Swiss Army knife of handy networking goodies.” This software can be downloaded from:

<http://www.oxid.it/cain.html>

Cain should be installed on your windows virtual machine from previews labs. If it is not, copy it from the nas lab directory named cainandable and install it.

- Open up VMware by typing *vmware* on the shell or finding the shortcut in Start > System Tools > VMware.
- Start your windows virtual machine.
- Now run Cain and Able whose shortcut is on the desktop (named Cain).
- Once Cain starts up, click Configure and then select the “Filters and ports” tab.
- Scroll all the way down and make sure that the SIP/RTP protocol is enabled.

- Click OK.
- Once that is done, click on the sniff button (🕵️).

Your network diagram should now look like the following:

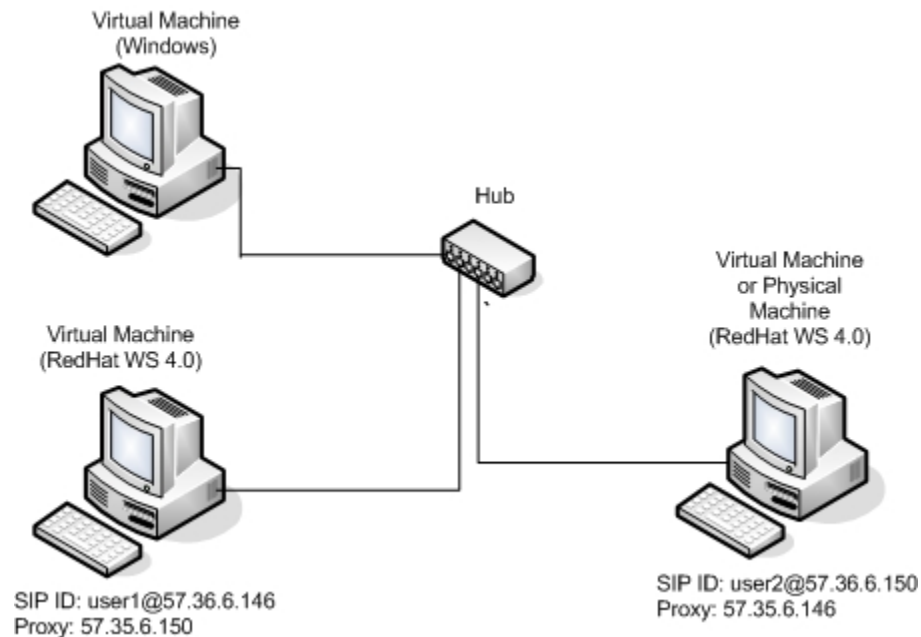


Figure 7: Network diagram with virtual machine.

Section 4.1 Cain Sniffing.

- Re-establish a VoIP conversation and again have a conversation with your partner.
- Meanwhile go to the Sniffer tab on Cain and then select the VoIP tab on the bottom.
- Get a screenshot of Cain sniffing and recording the conversation.

Screenshot 4: Cain recording a VoIP conversation.

- Stop Cain from sniffing by pressing the sniff button again and go to C:\Program Files\Cain\VoIP where your conversation was saved.
- Play the audio and listen to the conversation.

Question 3: Was Cain able to save the file as a wave file?

Question 4: How is the quality of the playback compared to that of the actual conversation?

Question 5: Comparing Cain with vomit, which one did a better job?

Section 4.2: Cain Man in the Middle

Usually attacks of this kind are rare since attackers encounter switches more often

than hubs. In order for this kind of attack to work on a switch, it must be combined with other methods such as ARP poisoning. Cain, is capable of both sniffing VoIP conversations and performing ARP poisoning. Finding software that does everything is not the only way of performing multiple attacks like this one. Individual software can be used so that one performs the ARP poisoning attack while another listens and records. The network diagram now expands to the following:

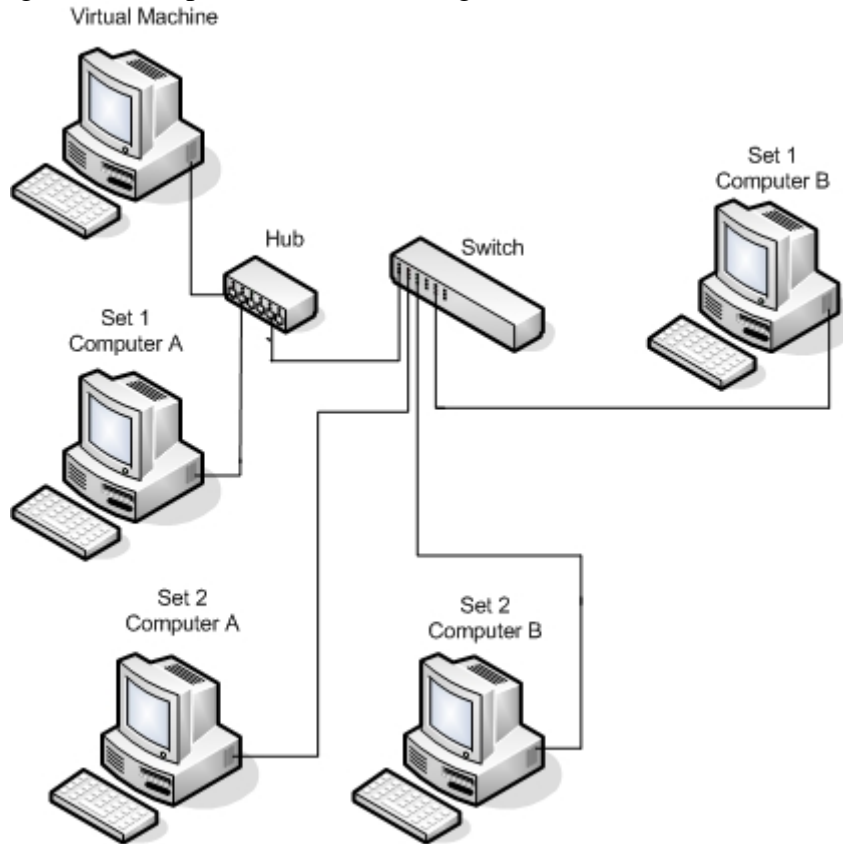


Figure 8: Network diagram with second set of computers.

The attack we are about to perform requires that either another group is doing the lab and carrying out VoIP conversations or you setup a VoIP conversation on another computer and perform the attack from your Windows virtual machine.

- Open Cain in your virtual machine if you do not already have it opened.
- Click on the sniff button and go to the Sniffer tab.
- Select the Hosts tab on the bottom. Click on the plus button (+) to add hosts.
- Now click OK on the scan options to begin scanning for available computers.
- Select the ARP tab on the bottom.
- Click the plus button again to add a set of computers to perform ARP from.
- In the popup window, select one of the computers in one column and the other in the second column.
- Click OK.

- Now click on the ARP Poison button (👹) to begin poisoning the computers.

If another team is doing their lab, wait until they establish a VoIP connection. If there is no other team working, make the connection your self and then come back to Cain. Get a screen shot of Cain performing ARP poisoning and recording a VoIP conversation.

Screenshot 5: Cain performing ARP poisoning to record a VoIP conversation.

Now if you have access to speakers, replay their conversation and let them hear it.

Question 6: How can you protect yourself from this type of attack?

Question 7: How can this be performed on Linux?

Now that you have finished the lab, you need to delete your .minisip.conf file in your home directory so that other classmates can do the lab:

```
# rm ~/.minisip.conf
```

Question 8: How long did it take you to complete this lab? Was it an appropriate length lab?

Question 9: What corrections and or improvements do you suggest for this lab? Please be very specific and if you add new material give the exact wording and instructions you would give to future students in the new lab handout. You may cross out and edit the text of the lab on previous pages to make corrections/suggestions. Note that part of your lab grade is what improvements you make to this lab. You may want to search the world wide web for other Buffer Overflow examples. What tools can we add to this lab that teach something else new? You need to be very specific and provide details. You need to actually do the suggested additions in the lab and provide solutions to your suggested additions. Caution as usual: only extract and use the tools you downloaded in the safe and approved environment of the network security laboratory.

Group Number: _____

Member Names: _____

Minisip correctly setup and working conversation.

TA Check Off : _____

Question 1: Were vomit and waveplay able to playback the file?..... 10

Question 2: How is the quality of the playback compared to that of the actual conversation? 10

Question 3: Was Cain able to save the file as a wave file?..... 11

Question 4: How is the quality of the playback compared to that of the actual conversation? 11

Question 5: Comparing Cain with vomit, which one did a better job?..... 11

Question 6: How can you protect yourself from this type of attack? 13

Question 7: How can this be performed on Linux? 13

Question 8: How long did it take you to complete this lab? Was it an appropriate length lab? 13

Question 9: What corrections and or improvements do you suggest for this lab? Please be very specific and if you add new material give the exact wording and instructions you would give to future students in the new lab handout. You may cross out and edit the text of the lab on previous pages to make corrections/suggestions. Note that part of your lab grade is what improvements you make to this lab. You may want to search the world wide web for other Buffer Overflow examples. What tools can we add to this lab that teach something else new? You need to be very specific and provide details. You need to actually do the suggested additions in the lab and provide solutions to your suggested additions. Caution as usual: only extract and use the tools you downloaded in the safe and approved environment of the network security laboratory..... 13

Turn-in Checklist

- Answer Sheet
- Screenshot 1: Minisip receiving phone call..... 9
- Screenshot 2: SIP URI as administrator@vonage.com 9
- Screenshot 3: Ethereal displaying SIP Invite and Ack. 10
- Screenshot 4: Cain recording a VoIP conversation. 11
- Screenshot 5: Cain performing ARP poisoning to record a VoIP conversation... 13

References:

- [1] <http://www.computerworld.com/securitytopics/security/story/0,10801,74840,00.html>
- [2] http://searchenterprisevoice.techtarget.com/sDefinition/0,sid66_gci214148,00.html
- [3] http://www.macvoip.com/resources/voip_record_calls.php

Future Enhancements

Future enhancements include:

Implement a SIP Proxy Server

 Perform Call Hijacking

 DoS

 Kill VoIP Connection

 Invite Flooding

Packet insertion

Conversation Encryption

Connection Certification

Answer Sheet:

Question 1: Were vomit and waveplay able to playback the file?..... 10

YES

Question 2: How is the quality of the playback compared to that of the actual conversation? 10

Very similar if not the same

Question 3: Was Cain able to save the file as a wave file?..... 11

YES

Question 4: How is the quality of the playback compared to that of the actual conversation? 11

Very similar if not the same

Question 5: Comparing Cain with vomit, which one did a better job?..... 11

Both work same. Cain has the advantage that does not depend on a sniffer and it can be used to save the conversation as an actual wave file instead of having the conversation saved as a packet dump and then have to be played by Vomit.

Question 6: How can you protect yourself from this type of attack? 13

Protecting against ARP poisoning in general is difficult but you can use encryption so that the third party sniffs garbage.

Question 7: How can this be performed on Linux? 13

You can use any software that performs ARP poisoning such as ethercap used in previews labs.

Question 8: How long did it take you to complete this lab? Was it an appropriate length lab? 13

Should take approximately 6 hours

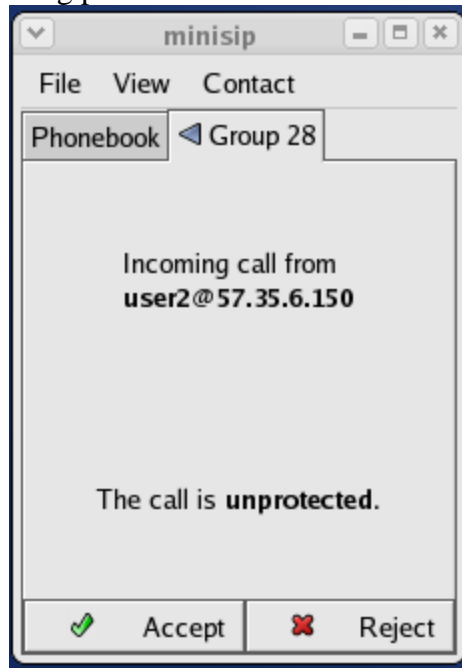
Question 9: What corrections and or improvements do you suggest for this lab? Please be very specific and if you add new material give the exact wording and instructions you would give to future students in the new lab handout. You may cross out and edit the text of the lab on previous pages to make corrections/suggestions. Note that part of your lab

grade is what improvements you make to this lab. You may want to search the world wide web for other Buffer Overflow examples. What tools can we add to this lab that teach something else new? You need to be very specific and provide details. You need to actually do the suggested additions in the lab and provide solutions to your suggested additions. Caution as usual: only extract and use the tools you downloaded in the safe and approved environment of the network security laboratory. 13

Possible Additions

- Implement a SIP Proxy Server
 - Perform Call Hijacking
 - DoS
 - Kill VoIP Connection
 - Invite Flooding
- Packet insertion
- Conversation Encryption
- Connection Certification

Screenshot 1: Minisip receiving phone call. 9



Screenshot 2: SIP URI as administrator@vonage.com 9



Screenshot 3: Ethereal displaying SIP Invite and Ack. 10

The screenshot shows the Ethereal (Wireshark) interface with a packet capture of SIP messages. The main display area shows a list of 18 packets, with the first packet selected. The packet list pane shows the following details:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_b3:bc:8c	Cisco_b3:bc:8c	LOOP	Reply
2	1.790433	57.35.6.136	57.35.6.137	SIP/SD	Request: INVITE sip:57.35.6.137, with session description
3	1.792017	57.35.6.136	57.35.6.137	SIP/SD	Request: INVITE sip:57.35.6.137, with session description
4	1.792026	57.35.6.136	57.35.6.137	SIP/SD	Request: INVITE sip:57.35.6.137, with session description
5	1.792768	57.35.6.137	57.35.6.136	SIP	Status: 100 Trying
6	1.792843	57.35.6.137	57.35.6.136	SIP	Status: 101 Dialog Establishment
7	1.793509	57.35.6.137	57.35.6.136	SIP	Status: 100 Trying
8	1.793515	57.35.6.137	57.35.6.136	SIP	Status: 100 Trying
9	1.793757	57.35.6.137	57.35.6.136	SIP	Status: 101 Dialog Establishment
10	1.793762	57.35.6.137	57.35.6.136	SIP	Status: 101 Dialog Establishment
11	2.063210	57.35.6.137	57.35.6.136	SIP	Status: 180 Ringing
12	2.064196	57.35.6.137	57.35.6.136	SIP	Status: 180 Ringing
13	2.064204	57.35.6.137	57.35.6.136	SIP	Status: 180 Ringing
14	3.280644	57.35.6.137	57.35.6.136	SIP/SD	Status: 200 OK, with session description
15	3.281727	57.35.6.137	57.35.6.136	SIP/SD	Status: 200 OK, with session description
16	3.281736	57.35.6.137	57.35.6.136	SIP/SD	Status: 200 OK, with session description
17	3.286735	57.35.6.136	57.35.6.137	SIP	Request: ACK sip:57.35.6.137:5060
18	3.287660	57.35.6.136	57.35.6.137	SIP	Request: ACK sip:57.35.6.137:5060

The packet bytes pane shows the following details for the selected packet:

```

    > Frame 1 (60 bytes on wire (60 bytes captured)
    > Ethernet II, Src: 00:13:19:b3:bc:8c, Dst: 00:13:19:b3:bc:8c
    > Configuration Test Protocol (loopback)
    Data (40 bytes)
  
```

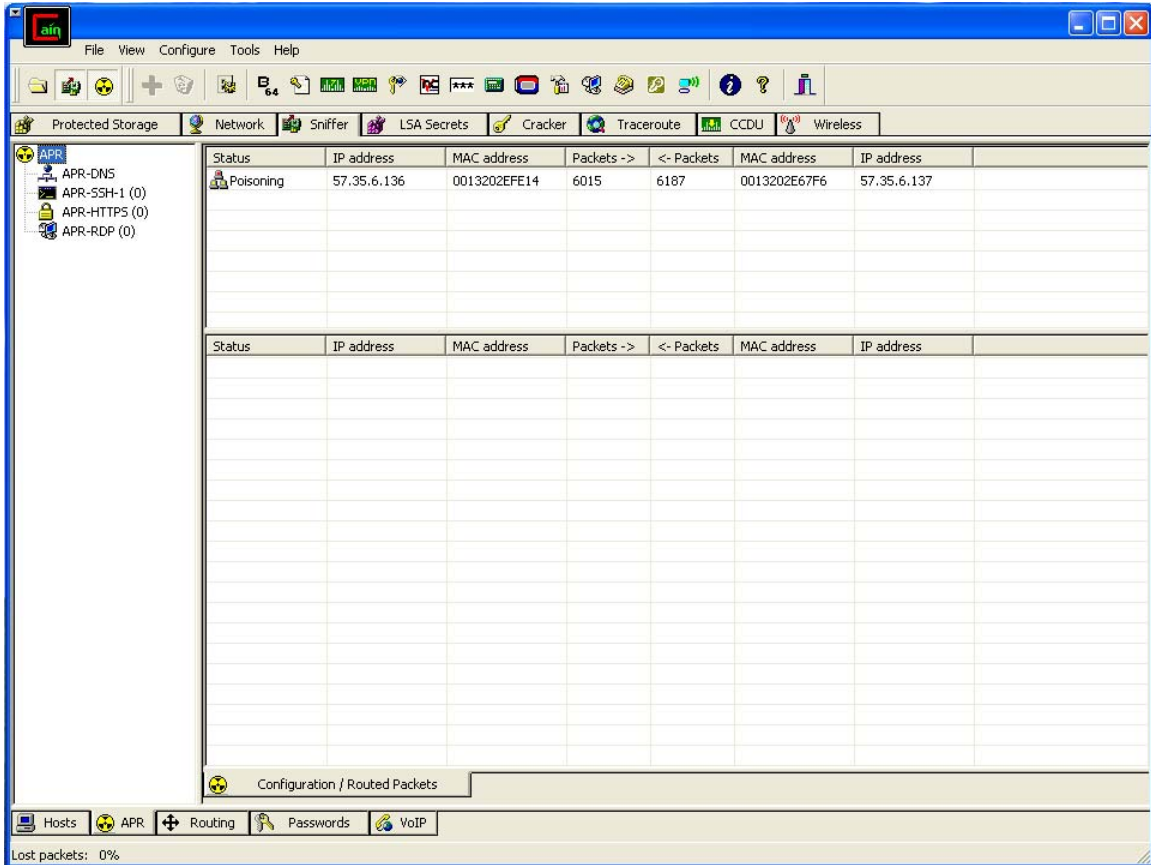
The packet bytes pane also shows the raw data in hexadecimal and ASCII format:

```

0000  00 13 19 b3 bc 8c 00 13 19 b3 bc 8c 90 00 00 00  .....
0010  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00  .....
  
```

The status bar at the bottom of the window shows: File: creditcard.dump 1583 K... P: 7031 D: 7031 M: 0

Screenshot 5: Cain performing ARP poisoning to record a VoIP conversation.



TA-SETUP

For this lab, you will have to install an Ensoniq AudioPCI sound card provided by Dr. Henry Owen. This is necessary because the students will have a virtual machine using one sound card and another using the other. Another thing that will be necessary is two headphones or speakers and two microphones for each team.

You will provide a RedHat WS 4.0 virtual machine in the nas lab directory. For this you will either have to create a virtual machine and install all the software necessary or use one if it was already done.

As root, copy the folder software to the root directory of the RedHat WS 4.0 host machine:

```
# mount /media/cdrom  
# cp -r /media/cdrom/software /root/
```

Installing minisip is a very tedious job because of all its dependencies and environmental variables needed. For this reason it is recommended that you use a script named minisipsintaller created for you. Go the directory software/minisip and run the script minisipinstaller:

```
# cd /root/software/minisip  
# ./minisipinstaller
```

NOTE: If the script does not finish successfully, you might want to run it again since there might be a dependency which is out of order and makes other dependencies fail.

This should take approximately 60-90 minutes. When this finishes, the script should have created a script named runminisip located in the root directory. Run this script which should open minisip without any problem.

```
# ~/runminisip
```

Next you need to install vomit:

```
# cd ~/software/vomit  
# tar xvfz vomit*.tar.gz  
# cd vomit*  
# ./configure  
# make  
# make install
```

Finally you need to copy the file named waveplay-20010924.tar.tar to your root directory and make it:

```
# cp /root/software/waveplay/waveplay-20010924.tar.tar /root/waveplay-20010924.tar.tar
```

```
# cd ~  
# tar xvfz waveplay-20010924.tar.tar  
# cd waveplay-20010924  
# make
```

The computer is now ready. You should test that minisip is running.

The following is a printout of the script written to easily install minisip:

```
#!/bin/sh

echo "Exporting some system variables needed"
export LD_LIBRARY_PATH=/usr/local/lib/
export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/local/lib/pkgconfig

echo "Installing Initial Dependencies"
cd dependencies
echo "Installing cairo"
tar xvfz cairo*.tar.gz
cd cairo*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH CAIRO"

echo "Installing jpeg"
tar xvfz jpegsrc*.tar.gz
cd jpeg-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH JPEG"

echo "Installing tiff"
tar xvfz tiff-*.tar.gz
cd tiff-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH TIFF"

echo "Installing pkg-config"
tar xvfz pkg-config*.tar.gz
cd pkg-config*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH PKG-CONFIG"

echo "Installing pango"
tar xvfz pango*.tar.gz
cd pango*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH PANGO"
```

```
echo "Installing atk"
tar xvfz atk*.tar.gz
cd atk*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH ATK"

echo "Installing glib-"
tar xvfz glib-*.tar.gz
cd glib-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH GLIB"

echo "Installing libsigc"
tar xvfz libsigc*.tar.gz
cd libsigc*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH libsigc"

echo "Installing gtk+"
tar xvfz gtk+*.tar.gz
cd gtk+*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH gtk+"

echo "Installing glibmm-"
tar xvfz glibmm-*.tar.gz
cd glibmm-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH glibmm-"

echo "Installing gtkmm-"
tar xvfz gtkmm-*.tar.gz
cd gtkmm-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH gtkmm-"

echo "Installing libglademmm"
```

```
tar xvfz libglademmm-*.tar.gz
cd libglademmm-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH libglademmm"

echo "DONE WITH DEPENDENCIES"

echo "Installing minisip libraries"
cd ..
cd minisip

echo "Installing libmutil-"
tar xvfz libmutil-*.tar.tar
cd libmutil-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH libmutil-"

echo "Installing libmnetutil-"
tar xvfz libmnetutil-*.tar.tar
cd libmnetutil-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH libmnetutil-"

echo "Installing libmikey-"
tar xvfz libmikey-*.tar.tar
cd libmikey-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH libmikey-"

echo "Installing libmsip-"
tar xvfz libmsip-*.tar.tar
cd libmsip-*
./configure --prefix=/usr
make
make install
cd ..
echo "DONE WITH libmsip-"

echo "Done installing minisip libraries"

echo "Installing minisip"
tar xvfz minisip*.tar.tar
cd minisip*
./configure --prefix=/usr
```

```
make
make install
cd ..
echo "DONE WITH minisip"

echo '/usr/local/lib' >> /etc/ld.so.conf
echo "export LD_PRELOAD=\"/lib/libc.so.6 /lib/libpthread-0.10.so\"" >
~/runminisip
echo 'minisip' >> ~/runminisip
chmod 755 ~/runminisip
echo "DONE"
```

**Georgia Institute of Technology
Atlanta, GA 30332**

**ECE4112: Internetwork Security
Final Project:
Voice Over Internet Protocol (VoIP) Security**

By: Luis Miguel Cortés-Peña
Gedeon Kamga

Group: 28

To: Dr. Henry Owen

Date: December 6, 2005