



Cisco VoIP Phones – A Hackers Perspective

By chap0

<http://www.seek-truth.net>

chap0x90 at gmail com

chap0 at corelan be

Table of Contents

1. Intro	3
2. VoIP Phones – What they advertise	4
3. Is this line secure?	13
4. Conclusion	20
5. References	21

Intro

In the world of VoIP phones, each person may look at them differently. For some, it is just an annoyance that sits on their desk. For others, it is simply a part of their job, either deploying them or as a help desk position taking phone calls all day. For the vast majority of people it is just a tool. They simply use them on a daily basis at home or in a lobby. But what about in a professional Penetration Tester's mind? What kind of simple yet sensitive information are we leaving out in the open for malicious users?

We see VoIP phones everywhere now days, for example Colleges, Local and Corporate businesses, even Government buildings. If we take a deeper look at these phones, what kind of information can we gather from just a phone on a desk? Immediately when I think of a VoIP phone, I don't think about making my daily phone calls or checking who left me voice mail. In the back of my mind I recall places where I have seen a VoIP system in place and the information I can gather from those places. For example, I was once at a local Library and took notice not only to the free WiFi sign but the study rooms with VoIP phones in them. Another place I recall is my local Bank with a private room just for customers to use with a VoIP phone inside. This is interesting enough already and this is without a laptop. Put that in the mix and it would be a field day!

As we know VoIP stands for Voice over IP, and usually companies move to VoIP when moving away from the PBX systems. One main good reason may be less cost and that you are able to use them on your local network side by side with your computers, servers and so on. So my main focus for the topic will be Cisco VoIP phones, mainly because the popularity of the vendor. This write up will also focus on having physical access to the phones. ***I will also like to state that this document is for educational purposes only! I cannot be held responsible for what any individual does with this information, and do not wish for anyone to use this information for illegal use.***

First let me just clarify that the models of Cisco VoIP phones that we will have a look at are the 7941, 7971 which are very similar (from non color display to a color display). The rest of the Cisco VoIP phones should be very similar in functions and navigation. Now let us begin the journey of unraveling what type of sensitive information we are placing on our network with these VoIP phones.

VoIP Phones – What they advertise

Now starting off with the VoIP phones, we can gather some useful information at our fingertips just by looking at it. We see the phone number, the extension the phone has and possibly a description name that has been applied to the phone.



On the Cisco 7941 and 7971 there are four major buttons, and these buttons all have different functions and settings for the phone:



Messages Button

Services Button

Directories Button

Settings Button

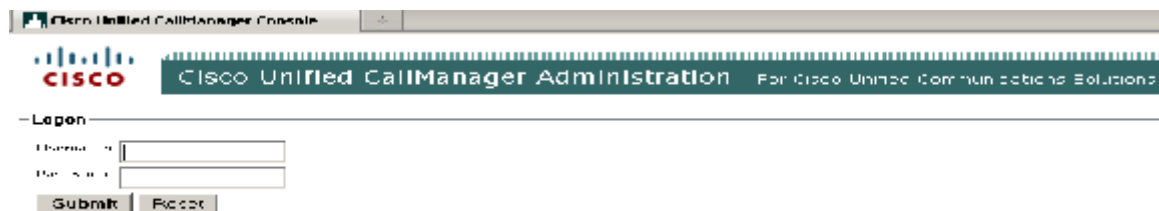
We will go through each button and learn about its features and functions. We will also be able to realize what information an attacker can gain about the phones, network and organization to help with their attack.

Messages Button

The Messages button is straight forward. It is where the users retrieve voicemails. Possibly from an attacker's side you could password guess the voicemail password or find it on a sticky note somewhere if you wanted to get into the voicemail to gather information. Or in some cases where the voicemail box is not set up yet, if you wanted to you could set it up and lock the voicemail box with a password of your choice, along with a custom message as a social engineer attack. This can be used by an attacker or pen tester, and if people happen to leave messages, it may help in information gathering. So if the attacker sets up the voicemail, they will be able to come back to that phone and retrieve those messages. It would probably take a while until the IT department figured out this was going on. Also in some cases there is no voicemail box for the phone period. However it still allows you either enter another extension to retrieve voicemails or browse the corporate directory by last and first name. With enough time on your hands you could sit at a phone all day, and try to guess passwords for other voicemail boxes. Also you could browse the corporate directory, which could be used for information gathering of users that can lead to usernames that may be used on the network.

Services Button

The Services button provides access to features that have been set on your phone related to web based features. Some features may include the local weather or stocks, or even sports scores. Also if no special web based features have been set on the specific phone and this button is pushed, it will reveal the direct link to the Cisco Call Manager login page. In most cases I have seen that these features are not set and the default is to give out the Call Manger URL instead. This could be very useful to an attacker. Even better, my testing shows with the default settings on the Cisco Call Manager, there is no threshold of how many attempts you can try to log into the Cisco Call Manager. This would be useful information for an attacker to start brute force attacks against the Cisco Call Manager. I should also mention if for some crazy reason the Administrator did not change the default username and password for login credentials, it will be either admin/admin or cisco/cisco. Most of the time the URL for the the admin page will be `nameoftheserver_or_IP/ccmadmin`



Another good point to make is once you have the URL for the Cisco Call Manager you may be able to login into the users' options webpage. If the defaults are also set, the phone number for the target phone is the username (ex: 5555555 would be the username). However, this will still leave us guessing a password depending on what setting the administrator or user has used. For example if the Cisco Call Manager URL is 192.168.0.1 to browse to the users page it would be 192.168.0.1/ccmuser. Some of the options users have inside this page are:

- Forward Calls
- View Users Guide
- Set Speed Dials
- Change Personal Address Options

In order to connect to the Cisco Call Manager Login page you must be on the network, but this is without doubt useful information.

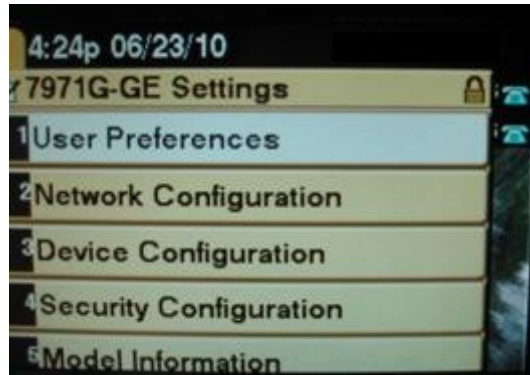
Directories Button

The Directories Button will give you several different directories and information. For example we can see listings of missed calls, received calls, and placed calls which can be useful to us, especially if it is an important person such as the CEO of the company. We can see who they have been talking to, who has called them etc. What comes next would be that much more important to us if we were trying to gather some good information about users, and usernames which can be used for social engineering and password guessing. The personal and corporate directory, if set up, will give you names of users and the phone numbers associated with their names. To add to this, all the information in this sub directory can be deleted or cleared.

Settings Button

To me this is where it starts to get real interesting. At first thought from a VoIP Administrator's point of view, the risks do not seem to be high. But to an attacker, this is just the foundation of the Lego blocks. What are we leaving wide open for anyone walking up to the VoIP phones? Settings, functions, network information, not only to view and take note of but to change also.

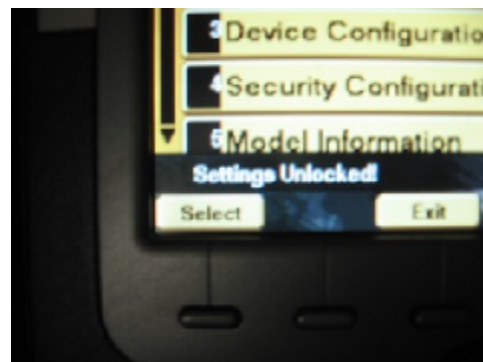
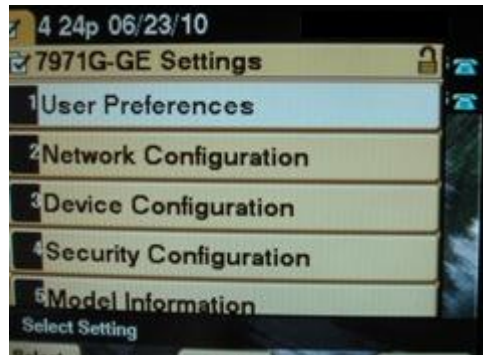
First off, when we push the settings button we can view all the directories, but take notice of the lock in the top right corner of phone display screen.



Well this lock indicates that the settings are, well locked! Never fear! Most likely the default unlock password is still in place. It took me one hard Google fu search of :

"cisco voip settings unlock password"

to discover the way to unlock these settings. Yes this is very difficult and brain straining I know. To unlock the settings so that some of the features can be changed or even erased, while in the Settings Directory on the touch pad simply push * + * + # and the lock will now show up unlocked. Also you will see an alert message indicating that the settings are unlocked.



Under the User Preferences directory, is where you will find the typical settings such as Ringtone, Background Image etc. Not to much valuable information in this directory.

Now moving onward to the Network Configuration directory, in this area we do retrieve some useful and valuable information that we could put to good use. Here is a list that we will focus on as the “major settings”.

Network Configuration Directory:

- DHCP Server
- BOOTP Server
- MAC Address
- Host Name
- Domain Name
- IP Address
- Subnet Mask
- TFTP Server 1
- TFTP Server 2
- Default Router 1 – 5 (Separate entry for each)
- DNS Server 1 – 5 (Separate entry for each)
- Operation VLAN ID
- Admin. VLAN ID
- DHCP Enabled
- DHCP Address Released
- Alternate TFTP
- SW Port Configuration (Auto, 10 Full/Half, 100 Full/Half)
- PC Port Configuration (Auto, 10 Full/Half, 100 Full/Half)
- PC VLAN

For starters we see a whole bunch of information that we are familiar with. Let us consider the Domain name. If this option is configured on the VoIP Phone itself, it’s a very good possibility that this is the domain name not only for the VoIP phones that reside on the network, but also for all the machines that are on the network. Consider the list here of these settings:

- IP Address
- Subnet Mask
- Default Router
- DNS Server 1

You can gather some valuable information from these settings that are set on the VoIP phone. You could possibly even jump on the network if DHCP is not set to dynamic and set a static ip address on your laptop. Now taking a look at the rest of the settings, usually DHCP is set to “yes” (set to on) for the phone. Since I am on the topic, this one button stood out to me while looking at some of the other setting that seem to be configurable, but the edit button was not functional. Every time I would try to “edit” a setting, the

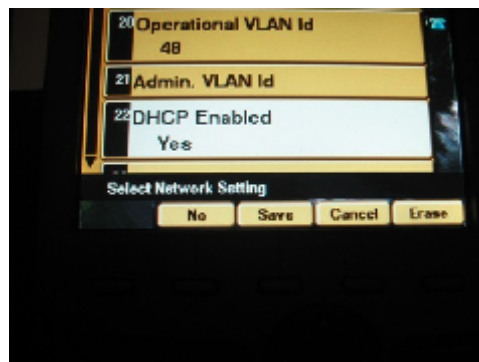
phone would scream back, “this is not active here.” Now as a pen tester or attacker you would not give up that easily.



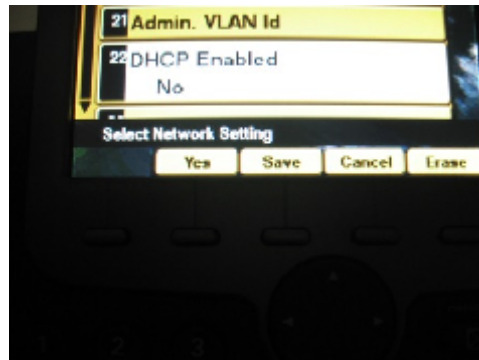
If you take a look at the picture, the edit button is sort of what I would call grayed out. Possibly if it wasn't grayed out then we would be able to do something with it, just as it shows here on the IP Address setting. So as the gears in my head started turning again to figure this out, I played around with different settings and this is where the DHCP setting comes into play. When DHCP is configured to “yes” on the VoIP phone, these settings cannot be edited. But when DHCP is set to “no” then they are free to change to whatever we want them to be. Since we unlocked the Settings, we are able to change the DHCP Setting, allowing us to also configure the rest of the settings. Consider this list of settings affected by this:

- Domain Name
- IP Address
- Subnet Mask
- TFTP Server 1
- TFTP Server 2
- Default Router 1 – 5
- DNS Server 1 – 5
- SW Port Configuration (Auto, 10 Full/Half, 100 Full/Half)
- PC Port Configuration (Auto, 10 Full/Half, 100 Full/Half)

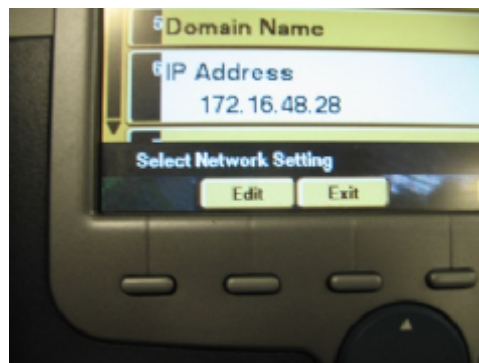
DHCP set to YES:



Change DHCP set to NO:



Compared to previous IP Address settings picture from above, when DHCP is set to No, this setting along with the list of others can now be changed once the DHCP setting is saved.



Take note of the Edit button. It is no longer “grayed out”. There are still some other settings that are configurable in the Network directory. These settings are not affected by the DHCP setting but are changeable:

- Alternate TFTP yes/no
- DHCP Address Released yes/no
- Admin VLAN ID may be edited

In the same directory, the Settings directory, if you go to the main view you can see Device Configuration. In here is a sub directory, the Call Manager Configuration directory. This will allow you to know the IP address of the call manager, if you could not find it from the Services Button, and also in some cases multiple call managers are set up on the network. In that directory you can view the IP addresses of them. Again this may be useful information if an attacker wished to brute force the Cisco Call Manager. The next valuable sub directory I found is the Security Configuration sub directory and

the settings of the PC port. Let me explain. On the back of most Cisco VoIP phones, notice two ethernet ports. They are built with the ability to “daisy chain” a laptop, desktop, printer or almost whatever you want that will run on CAT5 to the back of the phone, and hop onto the network. This includes wireless routers!



First glance at the image should look some what familiar. These can be used as NICs from the phone. One is labeled PC for the “daisy chained” device the other is labeled SW for the CAT5 that will be plugged into the network. Now the light bulb turns on when we recall that study room in the library with no one around acting like we are studying. If this was the only way to get onto the network, no WiFi, no other data drops, this would be the best solution. In fact since it is a study room, if you placed a wireless router and connected it to the phone, chances are no one will ever say a word. People studying would assume it was put in place by the IT staff. Better yet take a long enough CAT5 cable so that you can daisy chain it from the phone onto the floor or in the ceiling. Thanks Cisco! In most cases the PC Port feature is turned on by default and left alone. This is left wide open just waiting for a device to be plugged into it.

One last setting that is functional, in the Settings Directory, is the Status sub Directory. You can clear all status messages and network statistics messages from the phone, and no one will ever know.

Lastly, if a person wants to be flat out malicious, there is a huge risk of VoIP phones just waiting to be reset on the network. This could be like a VoIP Phone frenzy or something and could cause hours of work for many VoIP Administrators. These VoIP phone have a “reset to factory setting” functionality on them that are not complicated to follow. As well as taking little to no time, phones can be reset everywhere. Now I can understand why Cisco would do this for their Routers and Switches, these devices are suppose to be locked up in a closet with some super duper James Bond type security. But for devices that are going to be out in the open that people use on a daily basis? To some it may not matter, but I know if I was the one having to do the resets, well I better have the coffee and meds to back it up.

These are the steps to do so:

Step 1 – Unplug the CAT5 cable from the phone, the cable that connects to the switch. Once the phone is powered off reconnect the CAT5 cable again. This will begin the power up cycle and will start to turn on the phone.

Step 2 – Immediately press and hold the # key after you have plugged in the CAT5 cable. While the Headset, Mute, and Speaker buttons begin to flash in sequence, release #. The line buttons flash in sequence in order to indicate that the phone waits for you to enter the key sequence for the reset. (They will flash orange)

Step 3 – While the line buttons are flashing orange Press 123456789*0# within 60 seconds after the Headset, Mute, and Speaker buttons begin to flash.

If you do not complete this key sequence or do not press any keys, after 60 seconds the Headset, Mute, and Speaker buttons no longer flash, and the phone continues with its normal startup process. The phone does not reset.

If you enter an invalid key sequence, the buttons no longer flash, and the phone continues with its normal startup process. The phone does not reset. If you enter this key sequence correctly, the phone will restart and displays this prompt:

upgrading

VoIP Phone waiting for the key sequence for reset:



Now that we have gathered some information from the phone physically, what if we went ahead and added a laptop in the mix? So the big picture now is, you are pen-testing for a company, you are in a hole in the wall room, with a VoIP phone a laptop running UCSniff, and no one around. . . Let us see what more information we can gather.

Is This Line Secure?

Well you have heard it plenty of times especially in the movies “*Is this line secure?*” If you have heard it first hand in real life, wow what are you into! Anyway this is actually how UCSniff makes me feel, every time I pick up a VoIP phone. This program is easy to run and use, and is flat out wicked. Let’s look into some background first, and then we will go through some simple steps and commands that will help us in owning VoIP on the network.

UCSniff

UCSniff is an Open Source VoIP and IP Video capturing tool written in C. It is able to run on both Windows and Linux. UCSniff was created for security professionals, to test for the possible threat of eavesdropping on a VoIP network. When you run UCSniff it works like a Man in the Middle attack. It uses arp poisoning against the VoIP phones on the network “fooling” the VoIP phones on the network, allowing all the VoIP traffic to be sent through your laptop/Desktop. So if anything, at this point this tool gives people the power to be like Will Smith and Gene Hackman (excellent name) in the movie Enemy of the State. But instead of taping phone lines, you can be on the network and listen in on VoIP conversations, with a little more information being gathered. UCSniff is also capable of vlan discover and vlan hopping, which places you on the same vlan of the VoIP phones. No worries if the Voice vlan is different then the data vlan.

With the help of UCSniff you can capture conversations and keypad digits that are pressed on the phone. For example when someone is checking voicemail you can obtain the password this way. Not only does UCSniff *sniff* the traffic, it also has the capabilities to download the whole corporate phone directory into a nice text file for you. As a matter of fact, UCSniff saves multiple files for you as you are sniffing/capturing VoIP data on the network. Here is a list of files along with a brief description of what UCSniff is capable of gathering for you while sniffing/capturing data.

List of Files UCSniff saves for you:

.wav files – The voice conversations captured

.pcap files – To view in wireshark, take a closer look at the packets

calldetail.log – A log with the call details, phone numbers (internal, external), ip address of phones, what time the call was made and ended and also the duration of the calls.

call.log – This file is very similar to the calldetail.log

targets.txt – A file that shows the phone numbers and ip address UCSniff targeted while sniffing/capturing data.

arpsaver.txt – This file is gold in a way. It contains all the mac addresses of the phones along with the IP addresses. It is perfect for picking your targets. We will get into that in a minute.

directory-users.txt – This file name is obvious, it saves the VoIP corporate directory in this file for you, and it associates name of person to extension. This can be very useful also.

UCSniff also saves a sip.log, skinny.log, and xml file from the phone that was your target.

As we see, these files would be interesting enough. The true question is, how do we reach our goal to obtaining these files? Simple! After all this time messing with recon on physical settings of the VoIP phone, we know that our laptop can plug into the phone because the PC port is set to default which allows people to get onto the network. We also know information about the actual network which will be useful to us, such as VLAN Id for the VoIP VLAN and UCSniff allowing us to VLAN hop.

Now that we have some background of the actual program, let's look into the different modes and commands and the possibilities of what we can capture. As an attacker this would be their main goal to capture data. So it is useful to learn, know and reproduce as security professionals what a malicious person can do on your network and what information they can obtain.

UCSniff has two modes:

- Monitor mode
- Man in the middle (MitM) mode.

I am going to go over the MitM mode, only because our main goal is to actually capture sensitive data, and there is actually a way to use MitM mode that will act very similar to monitor mode. When running MitM mode, this is when the Arp poisoning begins, and with the understanding of Arp poisoning, while running UCSniff all the VoIP traffic will start to flow through your laptop. This is unnoticeable to the users that are making and receiving calls because UCSniff is just forwarding traffic between phones. While running in MitM mode there are two modes for MitM mode.

MitM modes:

- Learning mode
- Target mode

Running UCSniff MitM mode with Learning mode is the way you would reproduce running UCSniff in Monitor mode. While UCSniff MitM is in learning mode, users are mapped out and this is when the targets.txt file is created for you. So once you have the targets.txt file as stated above, this file will contain phone numbers and ip address and this is the beginning of mapping out your targets. With Monitor Mode, UCSniff only passively listens for traffic. To run UCSniff MitM mode with learning mode the commands you would supply are:

```
ucsniff -i eth0 // //
```

The “-i eth0” is pretty obvious states the interface you are working off of, and the “// //” is the command to use to allow UCSniff to enable MitM learning mode. While running UCSniff and performing MitM also gives us the capabilities to:

- VLAN hop
- Download the targets corporate phone directory

On a flat network, no dedicated VLANS for DATA or VoIP, it can be quite simple to obtain the corporate phone directory by issuing this command at the Konsole:

```
ucsniff -i eth0 -a -m 00:00:00:00:00:00 // //
```

We already know some of the switches shown and what they do. The -a is the switch in UCSniff for a utility called ACE. ACE is a stand alone utility but it is also included in UCSniff. Let me give you a little background on ACE. ACE (Automated Corporate Enumerator) basically imitates the behavior of a VoIP phone using DHCP, TFTP, and HTTP to pull down the users, the corporate directory, and in the end this information that is gathered is placed in the directry-users.txt file as stated earlier. This is a major reason why we use the -m with a valid VoIP phone mac address. You can think of this as proof of imitation or sort of tricking the TFTP server into thinking that our laptop is a valid VoIP phone on the network. It also should not be a problem to obtain a mac address since we already gathered our information from a physical phone early on.

Now if we didn't have access to the phone and we needed to learn what vlan the VoIP phones are on we would run this command at the Konsole:

```
ucsniff -i eth0 -c 1 // //
```

The -c is new to us. This is for using UCSniff in CDP spoof mode. This will allow us to learn the vlan of the VoIP phones and then allow us to “hop” onto that vlan. So this command would be very useful to us if we did not have the physical phone to look at. If the network is running CDP, and most of the time they are, then the CDP spoof mode will be perfect to discover what the VoIP vlan is. Of course there is a way to place the vlan ID in the command if you already know what it is:

```
ucsniff -i eth0 -v # // //
```

This command will “hop” to the specified vlan, and start running in MiTM mode.

So where does this leave us? We now know some background of UCSniff and what commands can be ran with it. Interesting enough we know that if we can gather some useful information before hand by using the VoIP phone, and then running UCSniff with specified commands we can gather even more information. To me this is great, I mean really great. We just went in, gathered information from the VoIP phone, and then used UCSniff to gather the corporate directory of our target. But I am not satisfied and you shouldn't be as well.

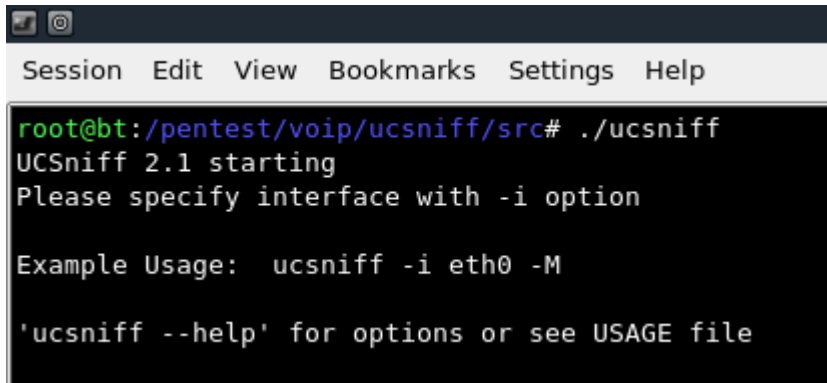
Example case study ~ Painting the Picture

We work for a Pen-testing company and the target is a bank. Our main goal is to succeed at gathering sensitive information, specific to conversations on the wire about bank records, accounts etc. This will be shown as proof of concept that the VoIP network in the organization is insecure. At this point you have scoped out the bank, gathered some useful information, vlan ID of VoIP vlan, if PC port is enabled on the physical VoIP phone, the mac address of the phone. Also we have taken some other notes, specifically, the IP scheme the organization is using just in case they do not have Dynamic DHCP set. We are using a netbook, only because it is small and compact and lightweight. Also we cannot forget our CAT5 cable. Remember we need this in order to plug into the network, and my weapon of choice, Back Track running off a USB drive which includes UCSniff. (I recommend Back Track on a USB drive just in case you need to run out and toss or smash the thing on the floor. What evidence officer?)

Our target is a small Bank branch office. The bank tellers' window aligns with the room in which you will be working out of, our targeted area. This is good for us because they will not have line of sight, and have a visual of what we are doing. As we walk into the bank, no hesitations, poker face on, ready to social engineer if necessary, and we explain to one of the bank tellers how we need to use the phone for calls about our credit, bank accounts, anything really that will give us access to the room. (I know when I tried to use the phone at my local Bank, all it took was “Excuse me, May I use this phone to make a call?” Was I rejected? No. Did I have any reason to be? No.) These people do not wish to be rude. Also, it's a bank. They do not want any upset customers. That only leads to bank robbery. Also remember they are human too. It is not like you are walking through airport security. These people are just here to do their jobs. They have a “Who cares who you are, make a call!” type of attitude.

So we do not waste too much time, let us say we had the laptop turned on already with Back Track booted up, and ready to log in and go. So we log in, plug in onto the network through the phone, and make sure we are alive on the network. Starting up UCSniff is not

difficult at all, in Back Track you can either browse through command line or use the menu. (I prefer command line):



```
root@bt:/pentest/voip/ucsniif/src# ./ucsniif
UCSniff 2.1 starting
Please specify interface with -i option

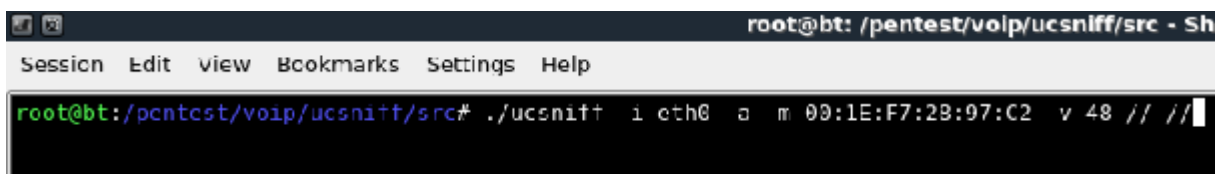
Example Usage: ucsniif -i eth0 -M

'ucsniif --help' for options or see USAGE file
```

Once we are logged in and UCSniff is ready to go, our goal is to be in and out as fast as possible so that we do not get caught. We reference our notes, the VoIP vlan is 48, and mac address of the phone is 00:1E:F7:28:97:C2. We know by now that this bank is not running a flat network and has data and VoIP data separated by vlans on the network, so the usually command will not work for us.

```
ucsniif -i eth0 -a -m 00:00:00:00:00:00 // //
```

Thinking fast, and since we have spent time with UCSniff, and gathered information before our initial attack, we know of a way that will do exactly what we want it to and quickly.



```
root@bt:/pentest/voip/ucsniif/src - Sh
root@bt:/pentest/voip/ucsniif/src# ./ucsniif -i eth0 -a -m 00:1E:F7:28:97:C2 -v 48 // //
```

By issuing this command at the Konsole, we are starting our MITM attack on the VoIP network. Actually most of this should look familiar by now.

- -i eth0 Our interface on the network
- -a For the Ace utility
- -m Our mac address of our target VoIP phone
- -v This is new to us but obvious, it is for the VoIP vlan
- // // Which is to start MITM attack

Yes at first we are attacking the phone we are sitting by, and this will get us no where. We are only doing this to download the corporate directory of our target. From there we

will gain the rest of the VoIP phones mac addresses and choose our targets at will. Let's take a closer look at what the command we issued is doing.

```
UCSniff 3.07 starting
File targets.txt can't be opened for reading in working directory
dhcpd: MAC address = 00:1c:25:98:d8:b0
dhcpd: your IP address = 172.16.48.36
TFTP request for file SEP001EF72897C2.cnf.xml sent
Successfully received file via TFTP, beginning to parse [file name: SEP001EF72897C2.cnf.xml]
Sending HTTP request for corporate directory - likely UCM 5.x or 6.x server
641 directory users written to file: directory-users.txt
Listening on eth0.48... (Ethernet)

eth0.48 ->      00:1c:25:98:d8:b0      172.16.48.36      255.255.240.0

Randomizing 4095 hosts for scanning...
* |=====| 100.00 %

60 hosts added to the hosts list...
60 hosts saved to arpsaver.txt

ARP poisoning victims:

GROUP 1 : ANY (all the hosts in the list)

GROUP 2 : ANY (all the hosts in the list)

Starting Unified sniffing...
```

Now we see the bright light! We are successfully running UCSniff, and it is automatically creating the files for us. Looking at the output, UCSniff successfully created the .xml file along with retrieving the Directory users and placing them in a file named directory-users.txt for us. Taking even a closer look, UCSniff has saved a file called arpsaver.txt.

```
Randomizing 4095 hosts for scanning...
* |=====| 100.00 %

60 hosts added to the hosts list...
60 hosts saved to arpsaver.txt
```

If we recall from earlier, I mentioned that this file is like gold to us. Basically we can use the information in this file for our next victims. The arpsaver.txt file contains all the mac addresses for the VoIP phones on the network. Also very important to remember: when we are running UCSniff, and when we finish gathering our information and conversations, we must quit properly.

```
Warning: Please ensure that you hit 'q' when you are finished with this program.
Warning: 'q' re-ARPs the victims. Failure to do so before program exit will result in a DoS.
```

Note that through testing, if you do not properly quit UCSniff will cause a DOS (denial of service) on the VoIP network and all phones will be unusable. This will bring the kind of attention that we do not want.

Moving on, we have successfully captured targets and we are ready to capture conversations on the wire. Simple. All that needs to be done to the command we issued before is to change the mac address to a different VoIP phone on the network and just wait for the calls to come in. Of course we get this mac address of our new target from the arpsaver.txt file. Once this is done you will be capturing bank account numbers in no time, through conversations of course.

Sample Output of conversations being captured:

```
Call 1 (SCCP) in progress at 22-39-8. (Number 9289051, 172.16.48.10) calling 'DO Ruben Alejandro' (Number 7936, 172.16.48.28).
Mapped new target entry: (IP: 172.16.48.10) --> extension 9289051
Mapped new target entry: (IP: 172.16.48.28) --> extension 7936 and name: DO Ruben Alejandro
Saving audio conversation to file, '9289051 Calling DO Ruben Alejandro_22-39-8_both.wav'
Call 1 (SCCP) ended at 22-39-13. Call duration is 5 seconds.

Call 2 (SCCP) in progress at 22-39-31. (Number 9289051, 172.16.48.10) calling 'DO Ruben Alejandro' (Number 7936, 172.16.48.28).
Saving audio conversation to file, '9289051 Calling DO Ruben Alejandro-22-39-31 both.wav'
Call 2 (SCCP) ended at 22-40-43. Call duration is 72 seconds.
```

Of course as it says when capturing the call is complete and both sides have disconnected from the phone call, UCSniff will save the calls to .wav files for you. All this is great but why stay limited to capturing one phone call at a time and only pointed at one target?

Instead, we use this command:

```
Ucsniff -i eth0 -a -m 00:00:00:00:00:00 -v 48 -garpdb // //
```

In short, the new switch `-garpdb` is GARP Disablement Bypass. Since Cisco phones by default have GARP Disabled in the IP Phone settings, this command will help in getting around that. Basically you will be able to capture more than just your targeted VoIP phone conversation. Any other call that is in progress will be captured as well. **Success! We have captured conversations on the wire and the job is done.** As always make sure you save your evidence. Don't forget to go up to the counter with a smile ask to close your bank account, and when they ask why simple tell them,

“I just had this weird feeling that my money wasn't safe here anymore.” (Sneakers 1992)

Possible Preventions

So now we have learned how to attack, but what about how to defend? Well it is always a game of cat and mouse, capture the flag, good vs evil. What ever you may call it! Here is

a list of some possible preventive steps we can take, making it harder to leave your network for dead, and having VoIP calls being collected off your network. I will not be going into too much detail. I enjoy the attacking side a lot more, with good ethics of course.

- VoIP Encryption
- Dynamic Arp Inspection & DHCP Snooping (They sort of go hand in hand)
- Mac address access list (Yes, tedious, but will work)
- Enable Port Security
- Disable Gratuitous Arp
- Change Defaults on Phone
- Lock Down Physical Security
- Don't put phones tucked away. Have them in the open for people to use.

Conclusion

VoIP is the newer technology for phone systems. As in anything “new” at one time or another, flaws are going to happen and arise and sometimes those flaws will never vanish. In VoIP technology many experts may take a look at this paper and feel that there is nothing they need to worry about. Plenty of times I have seen even Voice Engineers and Professionals plan, set up and configure VoIP networks and these flaws are in place. This is not entirely on them. The organization that is having VoIP installed should also have some kind of awareness. Hopefully this will open some eyes, and these organizations will realize what a security risk VoIP phones in small tucked away rooms can be. As I stated before, Lego blocks being used as foundation for an attack is something I do not want to be handing out or helping with. We all know and understand that we cannot lock down every single phone. That would kill the ease of use for users. But perhaps we can come up with solution on how to mitigate security risk. I do not necessarily feel at ease giving away the IP scheme, subnet mask or my default gateway to anyone walking up to my VoIP phones. Also, we need to keep in mind that rarely used rooms containing a VoIP phone, are just waiting for a Wireless Router to be plugged into them. At this point I believe it is as they say, you have to make your organization look less attractive to attackers so they will move along. Remember best practices. Disable these functions, change default passwords etc.

Greetz

I would like to say thanks to the Corelan Crew, a special thanks to breadcrumbs for the support. I would also like to thank the community. I hope this paper helps it as much as the community has helped me, and last but not least God the creator for the talent.

References

<https://supportforums.cisco.com/thread/284746?tstart=0>

http://www.cisco.com/en/US/products/hw/phones/ps379/products_tech_note09186a00800941bb.shtml

<http://ucsniff.sourceforge.net/usage.html>

<http://www.viperlab.net/Demo.php>

http://en.wikipedia.org/wiki/Man-in-the-middle_attack

http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/vlnwp_wp.pdf

http://en.wikipedia.org/wiki/DHCP_snooping

<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/19ew/configuration/guide/dynarp.html>

<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/12ew/configuration/guide/dhcp.pdf>

<http://ucsniff.sourceforge.net/ace.html>

<http://www.imdb.com/title/tt0105435/>

<http://www.imdb.com/title/tt0120660/>

http://www.cisco.com/en/US/docs/voice_ip_comm/cuipph/7902g/4_0_2/english/user/guide/7902gopt.html

<http://www.phenoelit-us.org/dpl/dpl.html>