DISCLAIMER

This is a little Disclaimer for if you havn't read the one on our site.
The tools and tutorials KD-Team develops and publishes are only ment for educational purpose only.WE DO NOT encourage the use of this tools and tutorials for mailicious purpose.We learned a lot during the development of them
so we hope you also learn and don't just use it without any brains.
We take completly NO responsability for any damage caused by them nor are we or our isp responsible for what you do with them.
Greetz: KD-Team
http://www.kd-team.com

ARP Poisoning In Practice

A Paper By:
DiabloHorn & Kimatrix

==========

This was a normal project just to bring the theory in practice.
The study was performed in a controlled environment with a ccnp instructor as supervisor.

Intro

Well here we are again DiabloHorn and Kimatrix this time with a finished CCNA semester. We have been busy with some school things like finishing the ccna lessons but it has brought us more things to play with like ARP. We digged up some info on arp and layer2 and started to read. After finishing ccna and done reading the papers we decided to put it all into practice in a controlled environment. So that we could test the things that where discussed in the papers. We decided to ask our teacher if we could borrow the lab and well he said yes :D

So we started to collect the needed things and thinking how we would setup the testing network but this and some other things will be discussed later on.

Hope you enjoy this paper. It isn't to technical on the arp part of how it exactly works cause there are dozens of papers about that. We will include a page with referenced of where we got our info on the end of this paper.

Enjoy this read

KD-Team

Tools Needed

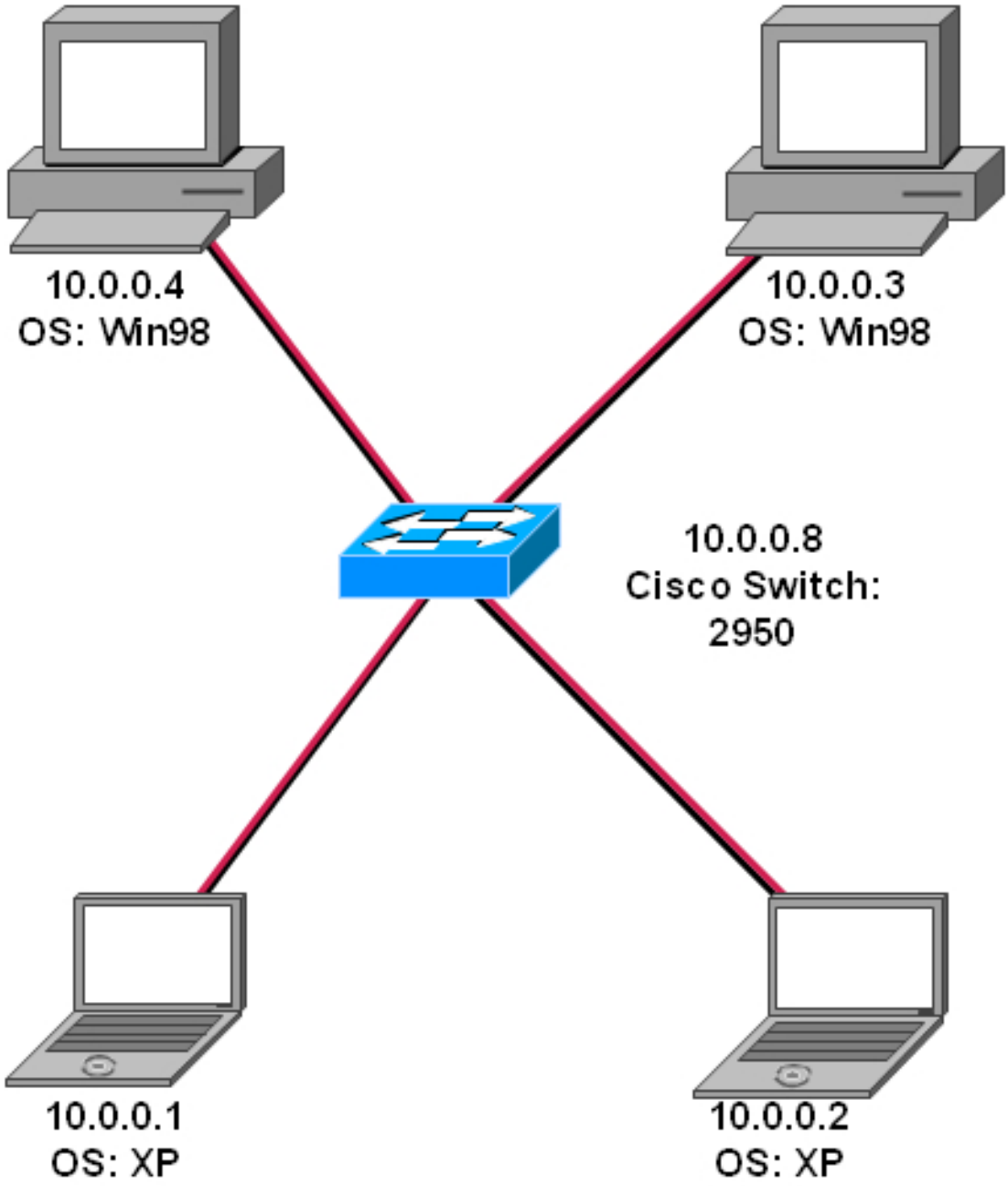
Hmm well this is a very short page:

- Brains
- A switch
- Clients
- Cain And Able

We used a CISCO switch to test this, you can try with some other switch it should work 100% the same. But hey you never know, if you test it with some other switch and you find some interesting things let us know so we can put that in here. That way every one learns :D

Links and credits will be mentioned at the end of the paper.

The Test Setup



Well this was our test setup as you can see we used a cisco switch to do the packet switching and 2 lab comp together with our notebooks.

Software running on clients:

2 lab comps : only internet explorer and command line ftp client.
Our Notebooks: xitami web server, trial version of serv-u ftp server and Ethereal sniffer

Purpose Of The Study

Like we said in the beginning it was all about testing the theory behind arp poisoning. Cause you all have to admit that even if you read things thousand times, you learn much better when you practice it.

We also wanted to make some things more clear for us like:

- Does the network encounter performance loss?
- Does it really work that easy?
- Can it be done without causing a DOS on the network?
- How big is the impact of this “bug”?

Those where the global questions we asked ourselves and what brought us to bring arp poisoning in practice.

Theory Of ARP Poisoning

Well here we are just going to explain how arp poisoning works but not in to much detail since there are enough papers on the net that explain this thing. In this paper we try to focus on the test we conducted so if any one has the need for a in depth explaining of ARP poisoning just let us know.

The theory behind arp poisoning is simple at least to understand it, bringing it into practice without good tools can be a pain in the ass.

The attacker: 10.0.0.1
MAC address: 00-AA-BB-CC-DD-00
The victims: 10.0.0.2
MAC address: 00-AA-BB-CC-DD-E1
10.0.0.3
MAC address: 00-AA-BB-CC-DD-E2

All the attacker actually does is sending a crafted packet to 10.0.0.2 with spoofed ip of 10.0.0.3 and his own MAC address and then it sends a crafted package to 10.0.0.3 with spoofed ip of 10.0.0.2 with his own ip. This means that both victims think they can find each other at the MAC address of the attacker.
The arp tables of both victims will look like this:

10.0.0.2 10.0.0.3

IP Address	MAC Address	IP Address	MAC Address
10.0.0.3	00-AA-BB-CC-DD-00	10.0.0.2	00-AA-BB-CC-DD-00

Like you can see each victims has a entry of the other but with the MAC address of the attacker.

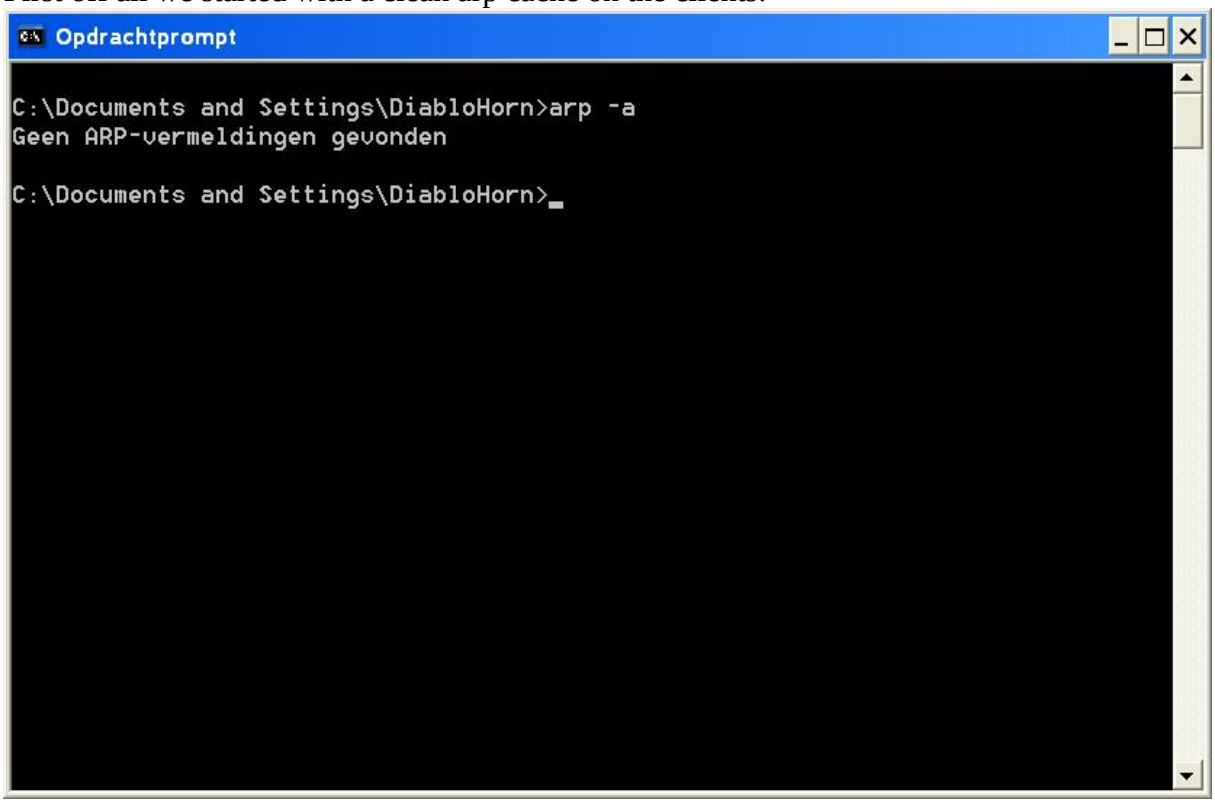
Now all the traffic between those 2 hosts will go through the attacker first. So this means that the attack will need to reroute the packets to the real destination else you get a DOS on the network and there will be no traffic possible. Also remember that the arp tables get updated so if during a long period of time there is no arp poisoning the entries will be deleted and you won't be able to sniff until you start poisoning again.

Hope this explains a little bit how arp poisoning works.

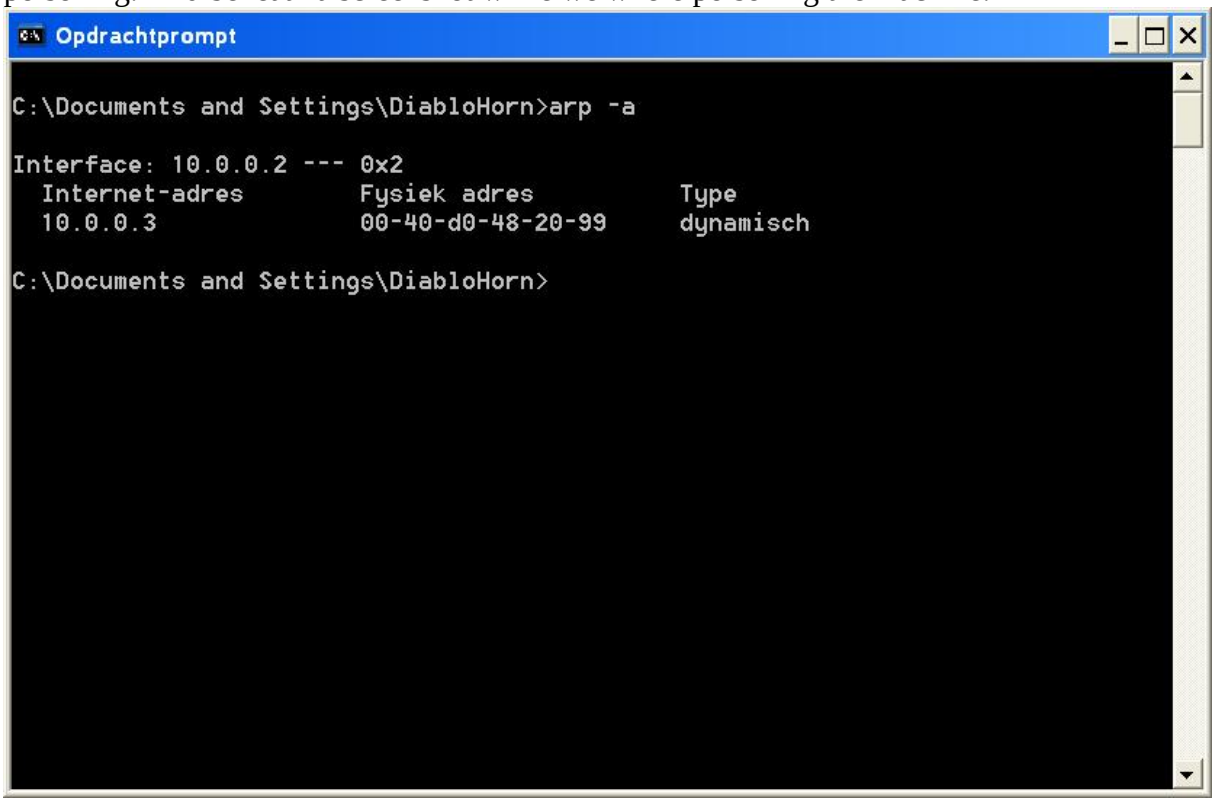
This still remains a difficult subject specially when trying to sniff data on complicated networks. So if you are interested in the subject just read the last page of this papers it contains some helpful things. Just remember google is still your best friend to find most info on any kind of subject.

The Practice Itself

Well like you all have read above about what arp poisoning is we will just try to give here a global impression of how it all went in the lab.
First off all we started with a clean arp cache on the clients:

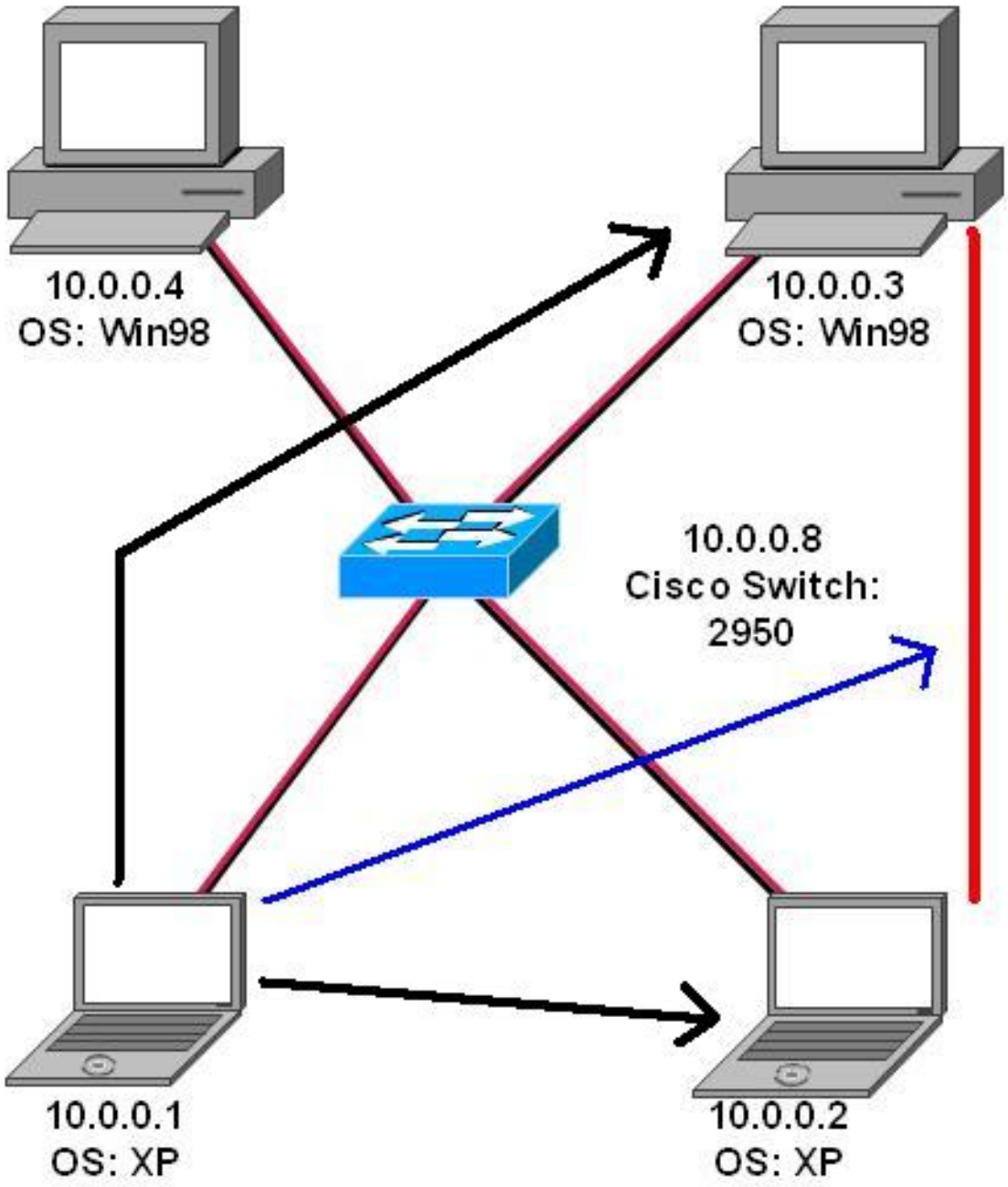


This screenshot is of the machine with the ip: 10.0.0.2 Before we started with our arp poisoning. And beneath a screenshot while we where poisoning the machine.



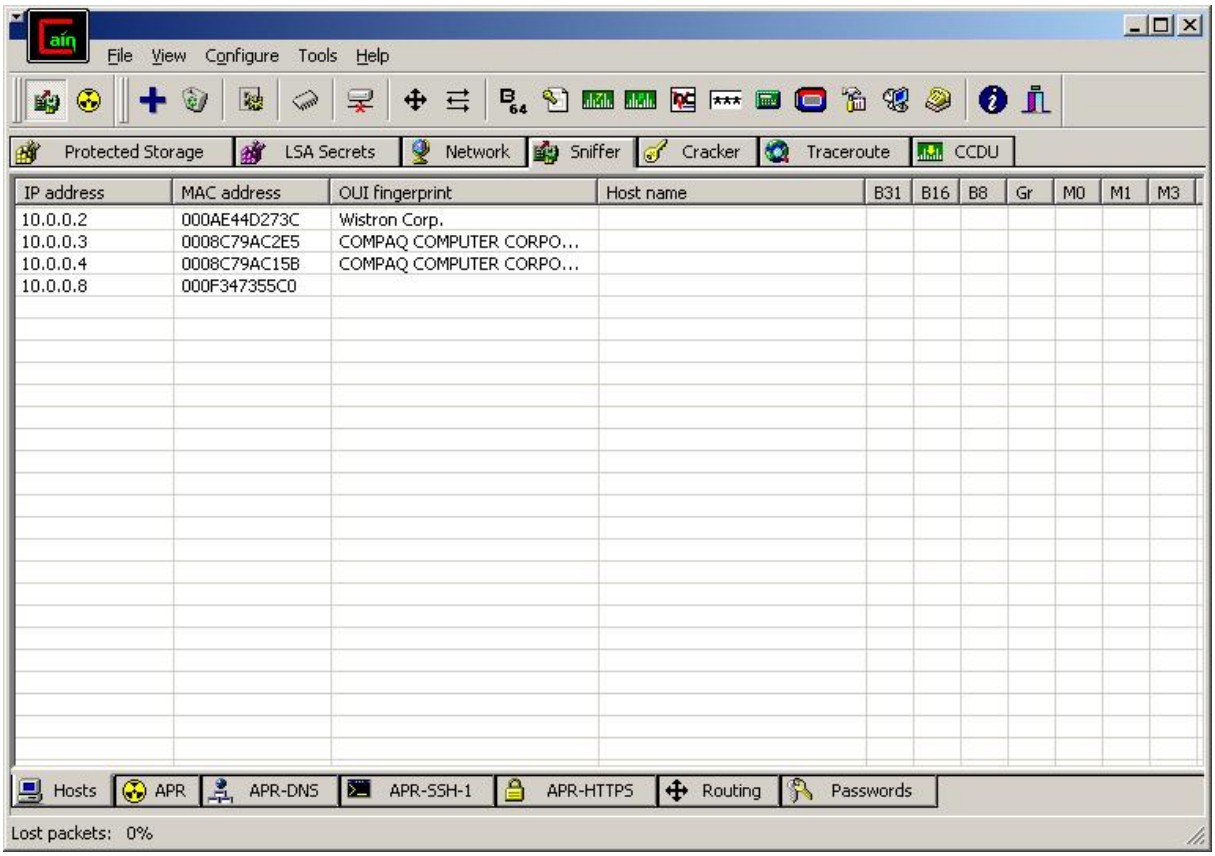
ARP Poisoning In Practice

The idea was to sniff the traffic between machine 10.0.0.2 and 10.0.0.3 with machine 10.0.0.1.



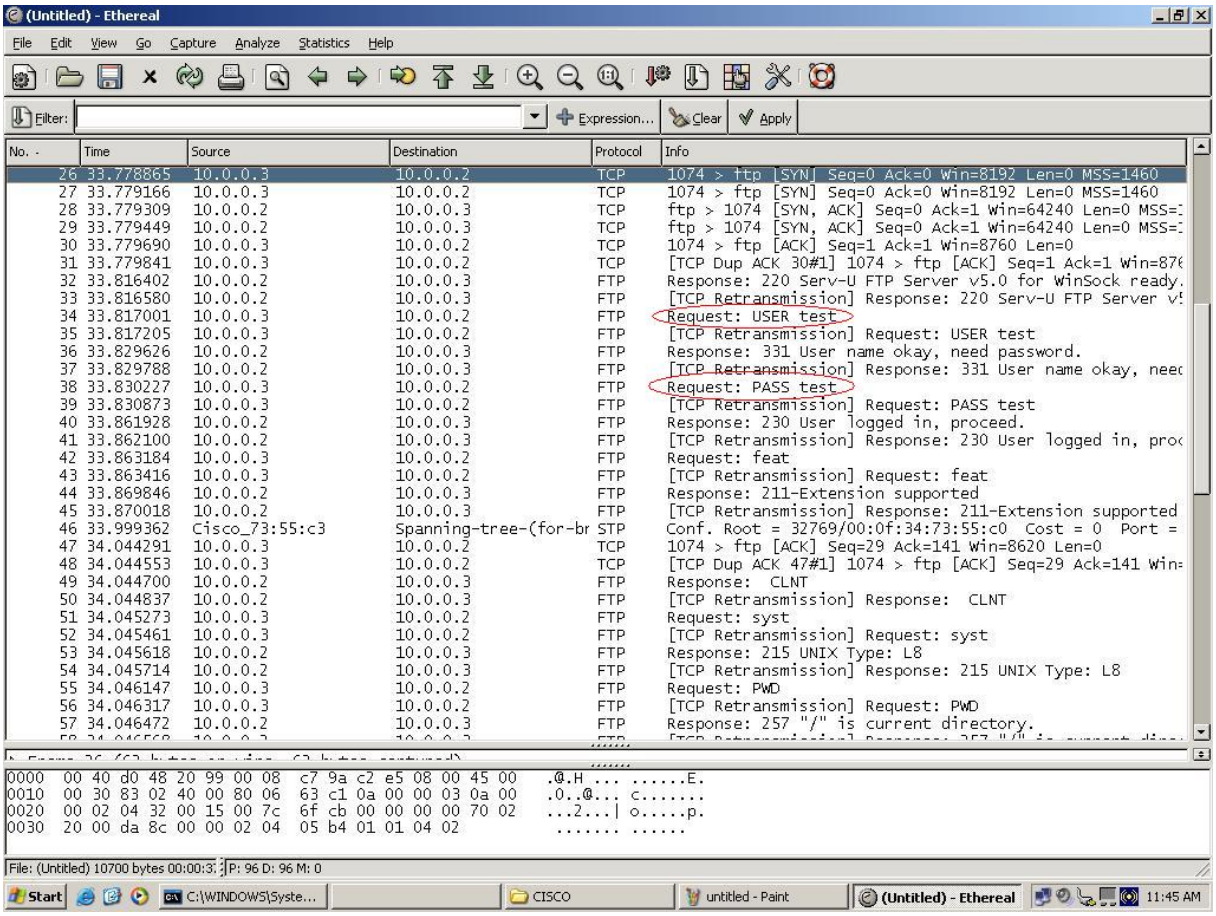
The blue arrow indicates what we want to sniff. The black arrows indicate what we are going to poison. So we fired C&A* up and started to scan for the MAC addresses.

* From now on we will be referring to Cain and Able as C&A



Like you can see the scan succeeded and displayed all hosts on the network. Including their MAC addresses.
With this info you can choose which hosts you want to poison and what kind of source MAC address you want to use. This come in handy when there are very serious restrictions and you need be a specific client. This can also be useful when the DHCP server is MAC address based and you need to impersonate a other host. So that you can access some parts of the network or other machines.

Then we started to poison the 2 hosts we selected 10.0.0.2 and 10.0.0.3 so that 10.0.0.1 could sniff the traffic. We sniffed during a ftp login here is the result:



Like you can see within the red circles the host 10.0.0.1 captured the traffic between the host 10.0.0.2 and host 10.0.0.3.

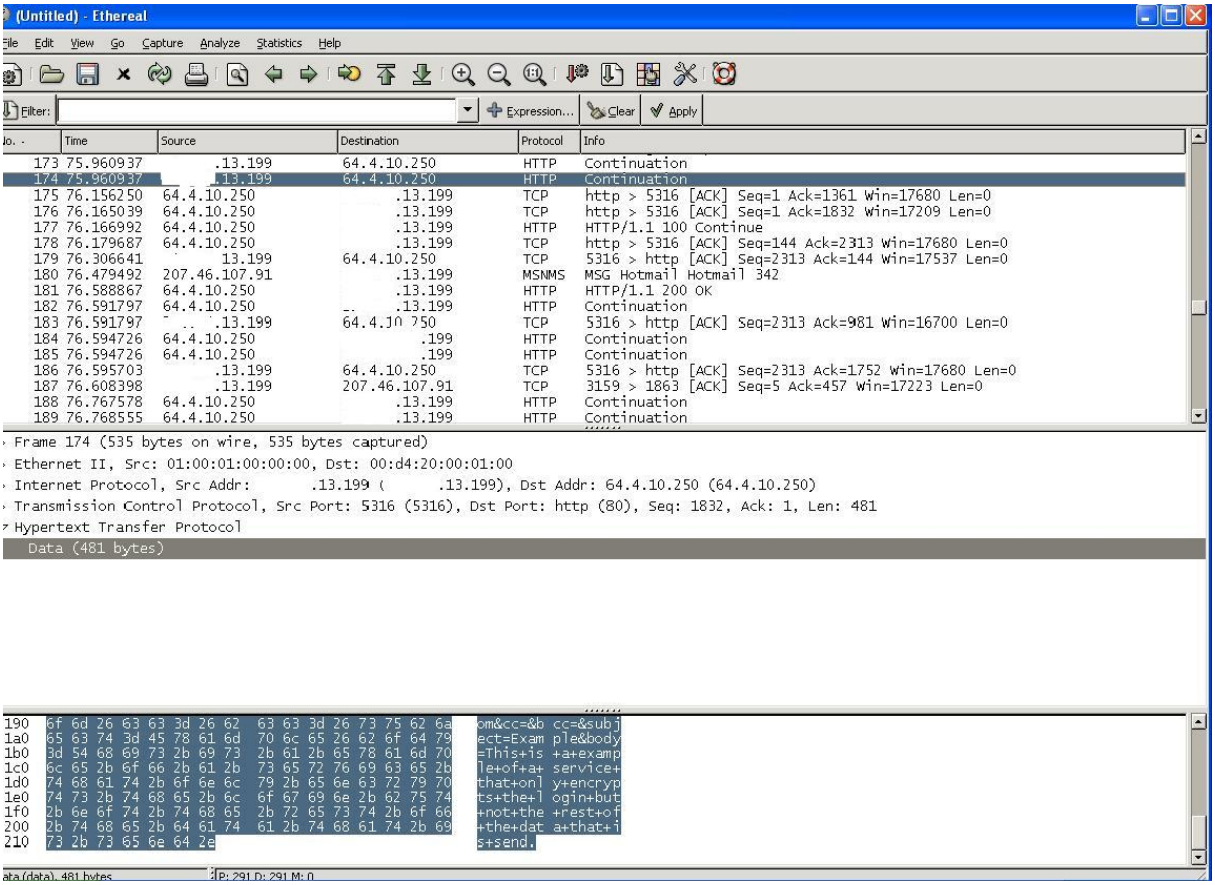
Like you already guessed it all happened without causing a DOS attack on the network because if there would have been a DOS there would be no traffic between those 2 hosts and the network would have serious performance problems.

How to Secure

Well this is hard and simple question.
Why you ask? Well cause it depends on the kind of network you are running and how important the traffic is that is being passed on the network.
But specifically for this kind of attack there are a few ways to protect. (this doesn't mean it is full proof cause on the net almost anything is possible)

- Encryption of the data being passed this is a tedious but useful way of protecting specially when used with strong algorithms with a minim strength of 128. Also remember to not only secure the login cause mostly the person arp poisoning is after the data so if you only encrypt the login but leave the rest of the send unencrypted it's useless. A example of this would e-mail services.

They have the encryption only for the login part and not for the part when one sends a e-mail or reads it.



Like you can see you can view what is send in the e-mail:
"subject=Example&body=This+is+a+example+of+a+service+that+only+encrypts+the+login +but+not+the+rest+of+the+data+that+is+send."
So if some one sends sensitive information and the other person just happens to be sniffing at the good moment then well he gets the data.

- Another method would be to make the switch identify such a attack. Switch can try to look for how many times one MAC address is being used. So when spoofing a lot of targets (in example when wanting to sniff traffic between the all the clients and the switch) the switch would detect several the same MAC addresses and could then reject them from the network.
- And the last method is just locking up the arp table but that means in some cases hundreds of entries so this last option would only be for small networks where you can know for sure you don't forget any arp entry.

This where some of the methods that you can use to secure this arp poisoning method.

Last words

Well we can say we learned a lot about this.
Some thing that where answered during this test:

- Network performance does decrease, Meaning if you try to poison all hosts to capture all the traffic between the switch and the hosts you will get a network performance decrease cause your comp has to reroute all the traffic and that depends on the computer speed and the upload speed you have.
- It seems easy when done with the right tools but the possibility to create a DOS on the network still exists. Taking the example that you poison all the host and supposing your computer can't handle all that traffic it causes a DOS on the network.
- The impact of this kind of attack is pretty big. Just imagine several cases where the network was designed where they thought only of immediate network risks and not about who is being next to who. That could create situations where normal employee/student is next to some one who sends sensitive information over the wire. This could cause severe loss of money or just pride.
- One more thing, we also tried using the switch as the start point of poisoning. And that didn't work out well. It seems it caused the network to fall apart. Because no one was able to ping or do anything on the network. So this caused in our test lab a DOS attack on the network

Well this kind of ends this paper about our experience with bringing arp poisoning in practice. On the next page you will see some references and thanks to the persons who deserve it.

All we got to say is,

This was a very nice way of learning something and also a lot less boring then normal classes that you can get on security subjects even though there aren't many classes on that.

So peeps just keep reading and learning new things and remember, whenever you need to test or try something out that can be dangerous to the network or the integrity of the security of information of some place, use a controlled environment to test it DO NOT run like a complete idiot to the nearest computer and start testing.

Hope you all enjoyed this paper and excuse our English plz ☺

KD-Team

info@kd-team.com

References

Thanks to:

- Our teacher for letting us use the lab

Information Link

- http://www.oxid.it/downloads/apr-intro.swf

Information Papers (found by using google)

- ARP spoofing and IP hijacking By: nebunu
- WCI (it is a tool but there is also a paper explaining about the poisoning itself) By: http://www.phenoelit.de
- Sniffing Switched Network By: mefakon
- Arp-spoof, fundamento y practica By: tontete

Tools:

- Xitami Webserver http://www.xitami.com
- Serv-U ftp server Trial http://www.serv-u.com
- C&A http://www.oxid.it
- Ethereal http://www.ethereal.com