Sniffing in a Switched Network

-With A Recipe To Hack A Switch Using Ettercap and Ethereal

-Manu Garg
manugarg at gmail dot com

Problem Statement-

To gain access to main switch of your company using a machine in the same LAN.

Tools used-

Ettercap, Ethereal

Techniques Used-

Arp spoofing and Sniffing

How can we achieve our goal?

Most of the network administrators use telnet to login to a cisco machine.

Telnet is a clear-text protocol – so, if you can sniff the packets you can get to know what is other person talking to the machine.

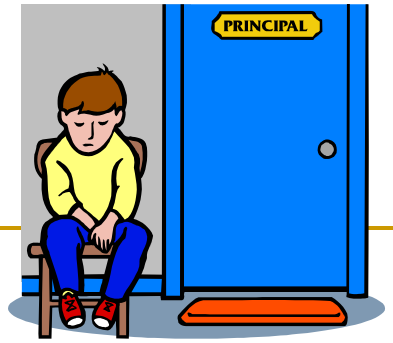
Easy then, I will just sniff the packets from the wire and get into my switch.

Hey, I don't see any traffic on the wire. What could be the reason?

You are in switched network and switches don't do any favor to the hackers. They transmit data only between the talking machines.

But this is not fair. They said I am on ethernet. Are they not supposed to use CSMA/CD then?

Ethernet has grown up buddy. It's 'switched ethernet' now.



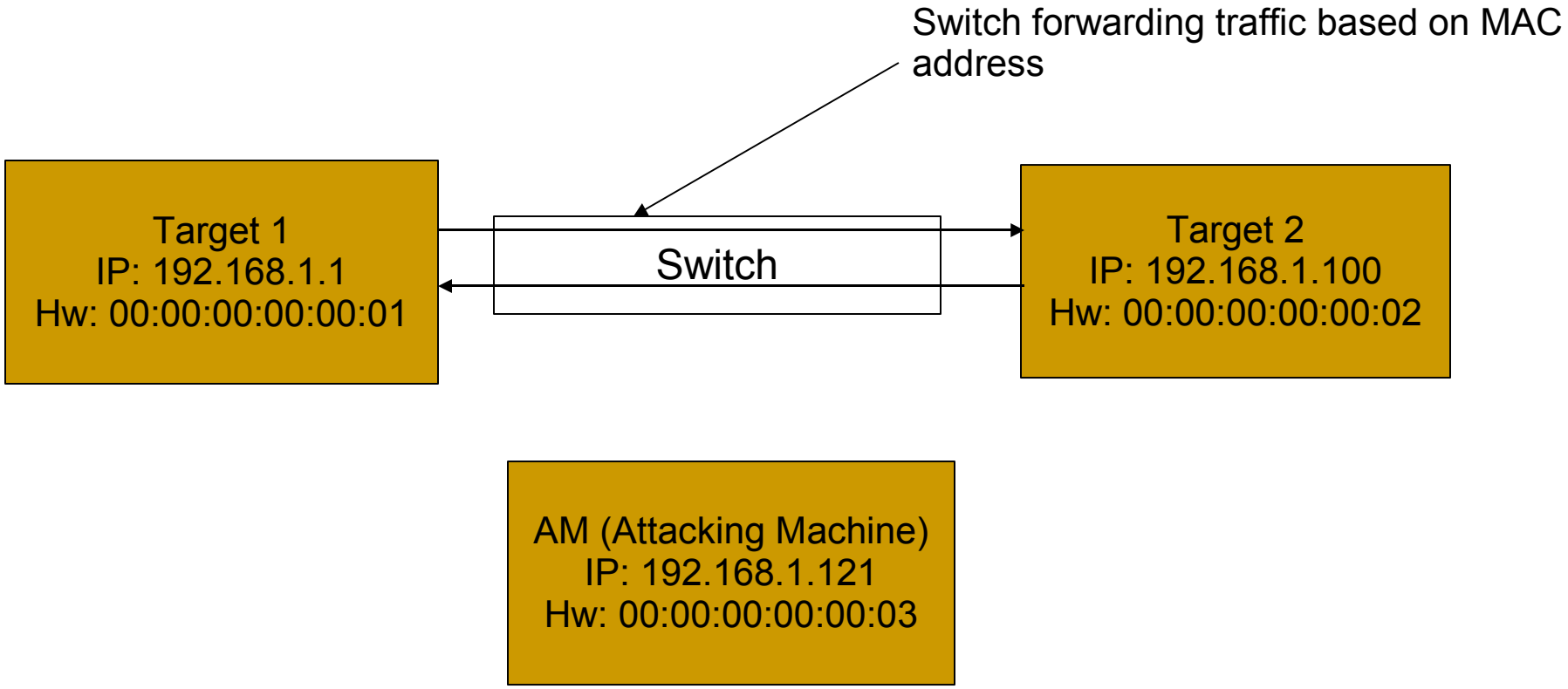
Hey wait! Don't get disappointed too soon. We hackers are not supposed to get defeated by a switch. Right?

Right. Our elite masters who designed TCP/IP protocols didn't forget us. They left an otherwise invisible path which can be seen by only their enlightened students.

So what is that path?

It's the path of arp spoofing. By using this technique you can fool your target machines to send data through your attacking machine and then you can sniff it on your attacking machine.

ARP SPOOFING!! How does it work?



Before Attack……..

Before attack, T1 and T2 are talking to each other only. Below is the arp table of the machines.

T1(192.168.1.1):

192.168.1.100	00:00:00:00:00:02
192.168.1.123	00:00:00:00:00:14

T2(192.168.1.100):

192.168.1.1	00:00:00:00:00:01
192.168.1.123	00:00:00:00:00:14

The switch understands only MAC addresses and forwards the packets to the right machines based on this MAC address. What if we manipulate the arp tables (this is called arp poisoning) on T1 and T2 so that the target MAC address in all the packets being exchanged between them, becomes the MAC address of our attacking machine. You got it right. Then switch will forward the packet to the attacking machine.

So after attack arp table should look like something below:

T1:

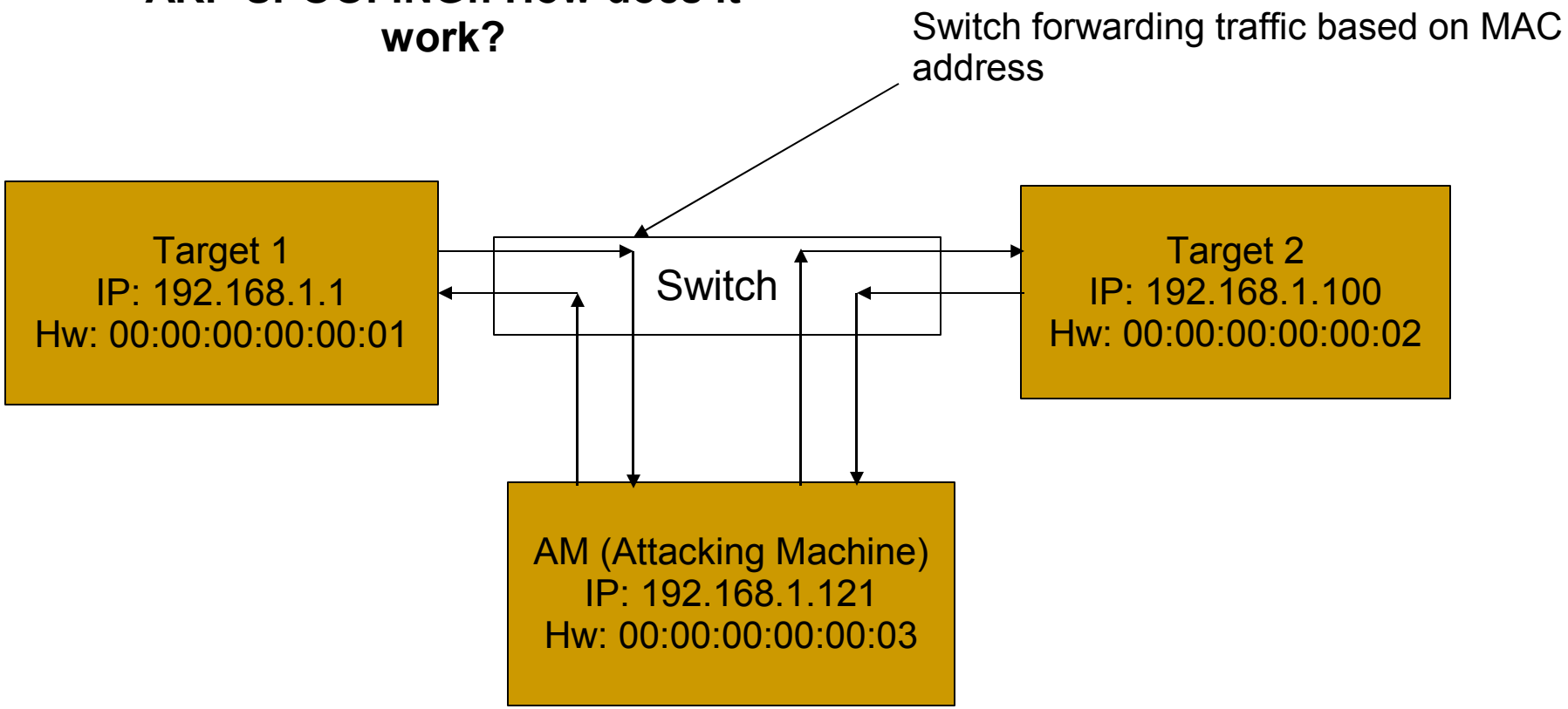
192.168.1.100	00:00:00:00:00:03
192.168.1.123	00:00:00:00:00:14

T2:

192.168.1.1	00:00:00:00:00:03
192.168.1.123	00:00:00:00:00:14

Where 00:00:00:00:00:03 is the MAC address of attacking machine.

ARP SPOOFING!! How does it work?



After Attack……..

Note: Don’t forget to enable ip_forwarding on your attacking machine otherwise you’ll break down traffic between 2 machines.

OK. I guess, now you know what ARP spoofing is. Now the bigger question - How do we poison ARP table on T1 and T2?

Explanation starts from another question. How do the hosts build this arp table? This table is built using arp protocol. ARP protocol has 2 kinds of packets – ARP request and ARP reply.

When a machine (say, A) wants to know MAC address of another machine (say, B), it sends an ARP request asking “who has IP address B”. This is a broadcast, i.e. sent to FF:FF:FF:FF:FF:FF. It is picked up by all the machines in the LAN and only the machine possessing IP address B sends an ARP reply. It is sent to machine A only. Machine A stores this MAC address in it’s ARP table.

Now you must be getting some clue. Yes, we are going to poison ARP table of T1 and T2 by sending them ARP replies. Most of the machines are generous enough and respect the ARP reply packet even when there was no request for particular IP address.

There are some machines, for example SunOS, which make an ARP entry only if there is a request for the one or if that ip is already in the ARP table. We can make them request for particular IP address by sending them an ICMP Echo packet from that IP address.

Ok. So you want me to create these ARP replies myself and manage all this.

No dude. We have blessings of our fellow hackers.

Alberto Ornaghi (a.k.a. ALoR) and Marco Valleri (a.k.a. NaGA) from Milan have created a nice tool called **Ettercap** based on equally cool library '**libnet**' from Mike.

Ettercap is pretty versatile and tool of choice for MITM attacks including ARP spoofing. Ettercap also has sniffing capabilities, but I prefer to use it only for spoofing. For sniffing, I prefer using **Ethereal**. Main reason for this is the use of pcap format for storing packets by ethereal. Pcap is a pretty old format and there are many tools available to analyze pcap files.

So, use ettercap for arp spoofing. Enable ip forwarding in kernel to maintain the connection between victim machines. And start sniffing using tethereal (text interface to ethereal). That's all you need to do.

Recipe

Let's try to do everything said till now. Let's look at our problem once again

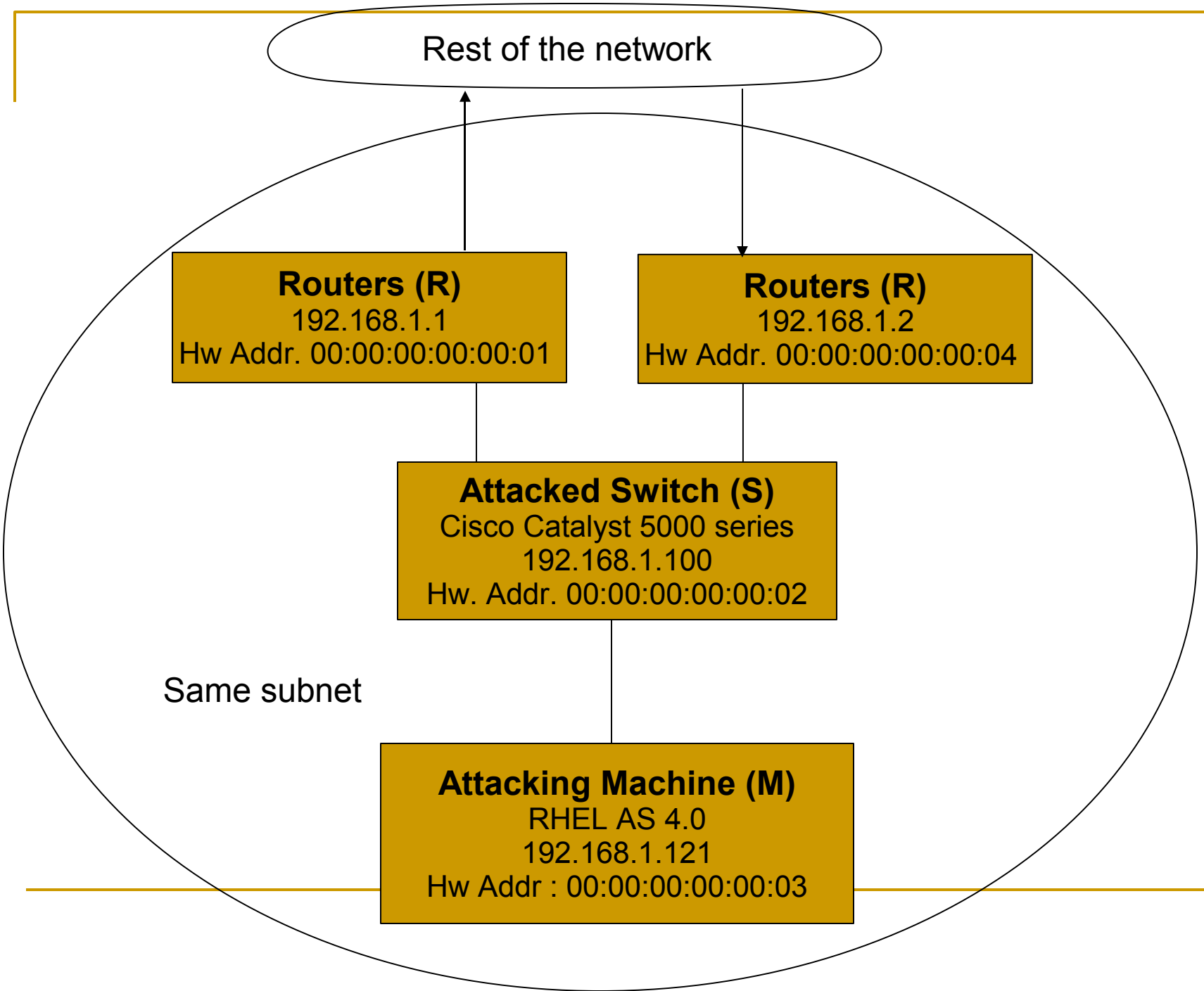
We want to get into the main switch of our company. This will allow us to configure the switch ports the way we want.

First task, find out ip address of the switch. In my case it is 192.168.1.100. You can use Nmap's OS detection feature to guess it.

Now find out how network admin communicates with the switch. In my case network admin sits offsite. So, he connects to the switch through WAN and the requests come through the gateway.

I use another machine in the same subset to observe the traffic for sometime and find out that all incoming traffic to the subnet comes through the router 192.168.1.2 and all outgoing traffic goes through the router 192.168.1.1.

So, when network guy from 192.168.101.34 logs into cisco console, packets are routed through the routers 192.168.1.1 and 192.168.1.2.


Rest of the network
Routers (R)
192.168.1.1
Hw Addr. 00:00:00:00:00:01
Routers (R)
192.168.1.2
Hw Addr. 00:00:00:00:00:04
Attacked Switch (S)
Cisco Catalyst 5000 series
192.168.1.100
Hw. Addr. 00:00:00:00:00:02
Same subnet
Attacking Machine (M)
RHEL AS 4.0
192.168.1.121
Hw Addr : 00:00:00:00:00:03

Recipe …

As you can guess, I need to put attacking machine in the path between 192.168.1.1 - 192.168.1.100 to tap outgoing packets and 192.168.1.2 – 192.168.1.100 to tap incoming packets.

So, I need to tell the router 192.168.1.2 that 192.168.1.100 is at 00:00:00:00:00:03 which is the MAC address of attacking machine. At the same time also tell switch i.e. 192.168.1.100 that 192.168.1.1 is at 00:00:00:00:00:03.

Before inviting packets to your machine, make sure you have path for them to reach their destination i.e. don't forget to enable ip forwarding. In linux you can enable ip forwarding using following command:

echo 1 >/proc/sys/net/ipv4/ip_forward

Start spoofing….

To start arp spoofing using ettercap:

ettercap -o -T -P repoison_arp -M arp:remote /192.168.1.100/ /192.168.1.1-2/

-o : only spoofing no sniffing.
-T : text mode
-P repoison_arp

Tells it to load plugin repoison_arp. This plugin re-poisons arp table at some intervals

-M arp:remote /192.168.1.100/ /192.168.1.1-2/

Tells it to start MITM attack with 192.168.1.100 in first target group and 192.168.1.1, 192.168.1.2 in second target group.

I'll suggest you to run ettercap in screen terminal, so that you can detach from screen and forget about it for some time.

I used "ettercap NG-0.7.2". You can download it from http://ettercap.sourceforge.com.

Start Sniffing …

Now you are all set to start sniffing. Use following command to start sniffing and write packets to a file:

 # tethereal -afilesize:100000 -w /tmp/cisco.pcap -f "host 192.168.1.100 and not arp and not icmp”

-afilesize:100000
limits the file size to 100MB.
-w /tmp/cisco.pcap
writes packets to /tmp/cisco.pcap
-f “host 192.168.1.100 and not arp and not icmp”
is the filter string. It tells to collect the packets either coming from or going to 192.168.1.100 and not to collect any arp or icmp packets.

Use some social engineering. Find out when network team is going to work on the switch or any other host which you want to break into. Leave your tools running while they do it.

Later on you can analyze the capture file by opening it in ethereal and you can just follow the telnet stream to find out the password.

That's all it takes.

A *person* is smart. *People* are dumb, panicky, dangerous animals and you know it.
--Ed Solomon