ARP Spoofing

Autumn 2013

Introduction

ARP spoofing, or ARP poisoning is a method used when you want to attack an Ethernet LAN. With ARP poisoning your update a target computer's ARP cache with forged ARP requests and replies. By doing this you want to change the MAC address to one that you can monitor. When this is done, the target computer sends frames that were meant for the original destination. This means the attacker can read the frames before they reach its destination, an ARP poisoning attack will be invisible to the victim and will have no clue that its being attacked.[1] After an ARP poisoning attack a Man in the middle(Mitm) attack can be attempted. When a Mitm attack has been successful, the attacker can monitor the traffic between two hosts, like a computer and the computer's router. First it sends malicious ARP replies to the router to make the router think the attacker is the victims computer. Then it will send a malicious ARP reply to the victims computer so it will think the attacker's computer is the router. With IP forwarding on the attacker's computer, he'll be able to forward any traffic it receives from the victim, to the router. So if the victim go to the internet, it will send its network traffic to the attacker, who will forward it to the router, and therefore the victim will be unaware of the attacker's interception of his traffic. With this method the attacker can sniff clear text password and hijack secured Internet sessions. For ARP spoofing to be successful, and to perform and Mitm attack, it exploits the ARP protocols vulnerabilities and the end results could be devastating if someone manages to pull it off. However, to be able to do a successful attack, the attacker have to be connected to the LAN, the hacker would need physical access or control one of the machines on the local network.

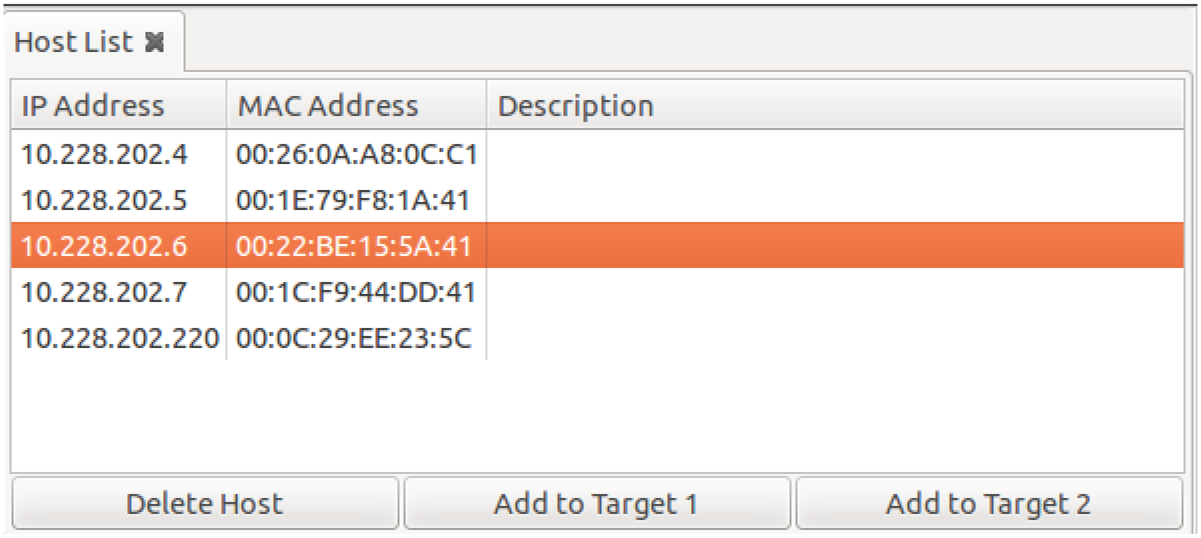
To protect yourself there are some techniques you can use, if you have a small network static IP addresses with static ARP tables. This will be hard to maintain in a large network though, for large networks its recommended to use port security features instead of static ARP tables and IP addresses. These features can let you force your switch to only allow one MAC address for each physical port on the switch. This prevents hackers trying to map more than one MAC address and even changing the MAC address of their machine. All networks, both large and small can use monitoring tools that can alert when unusual ARP communication occurs. ARPwatch is one tool you can use.[2] There is also

ArpON which is a daemon which every host uses to authenticate each host.[3]

Study

In my study I tried to perform and ARP poisoning and a Mitm attack with two virtual machines connected to a switch. The attacker have the IP address 10.228.202.164/24, connected to the port fa0/20 on the switch. The victim have the IP address 10.228.202.220/24, connected to the port fa0/13. They're both part of VLAN 100, which has the IP address 10.228.202.6/24 on the switch. The attacker runs an Ubuntu Desktop v. 12.04, and has Ettercap[4] installed which will be used to perform the ARP poisoning and the Mitm attack. The victim runs Windows XP with Service Pack 3.

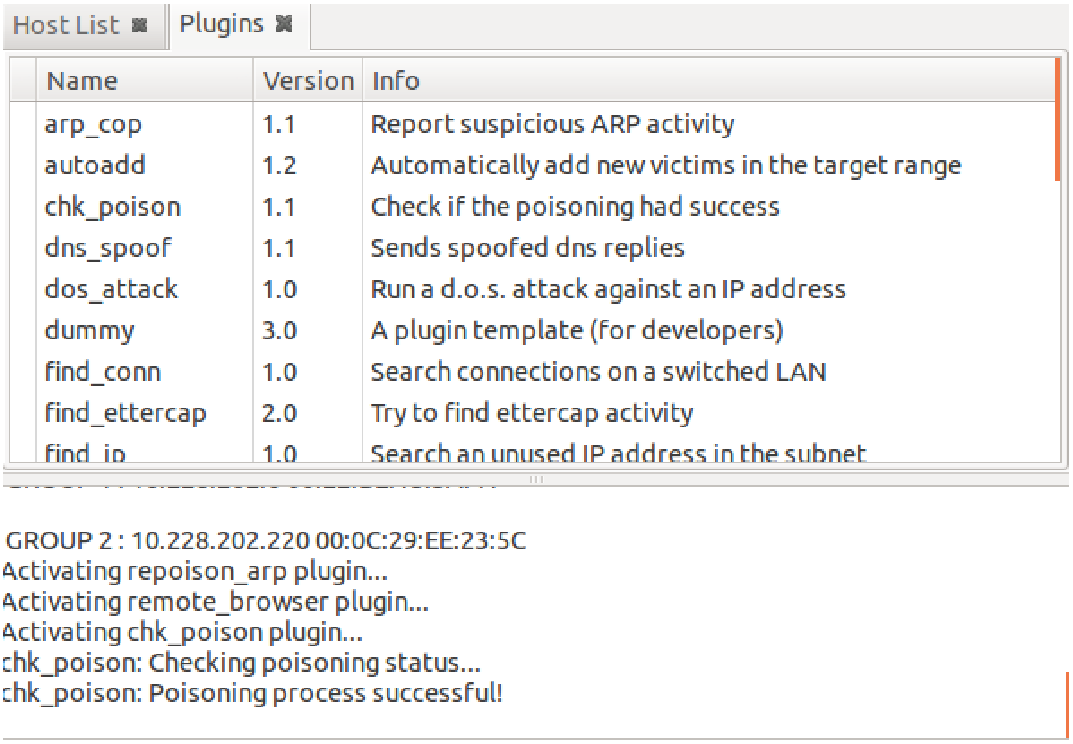
To start the attack from the attacker, I started Ettercap. In Ettercap I selected "promisc mode" in options and then Unified Sniffing on the interface eth0. After that I scanned for hosts, and found 5 hosts.



IP Address	MAC Address	Description
10.228.202.4	00:26:0A:A8:0C:C1	
10.228.202.5	00:1E:79:F8:1A:41	
10.228.202.6	00:22:BE:15:5A:41	
10.228.202.7	00:1C:F9:44:DD:41	
10.228.202.220	00:0C:29:EE:23:5C	

Here we can see four switches, where .6 are the switch we're connected to, and .220 being the victim. Select the .6 switch as target 1 and the victim, .220 as target 2. When this is done you can start sniffing. In the "Mitm dropdown", select ARP poisoning and sniff remote connections.

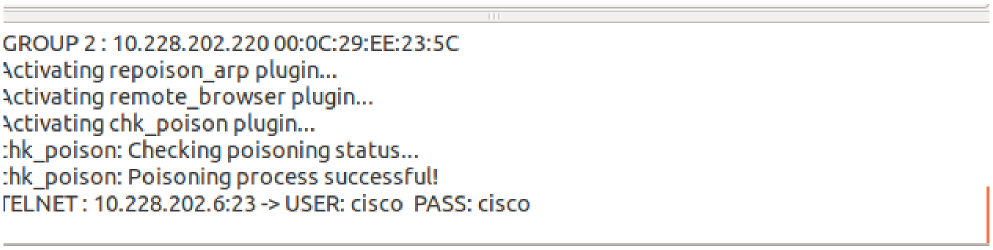
Under "plugins" you have a lot of different things you can do, everything from checking if the arp poisoning were successful to dos attacks. I started with "chk_poison", which will check if the poisoning had success, and I got this message:



Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.1	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet

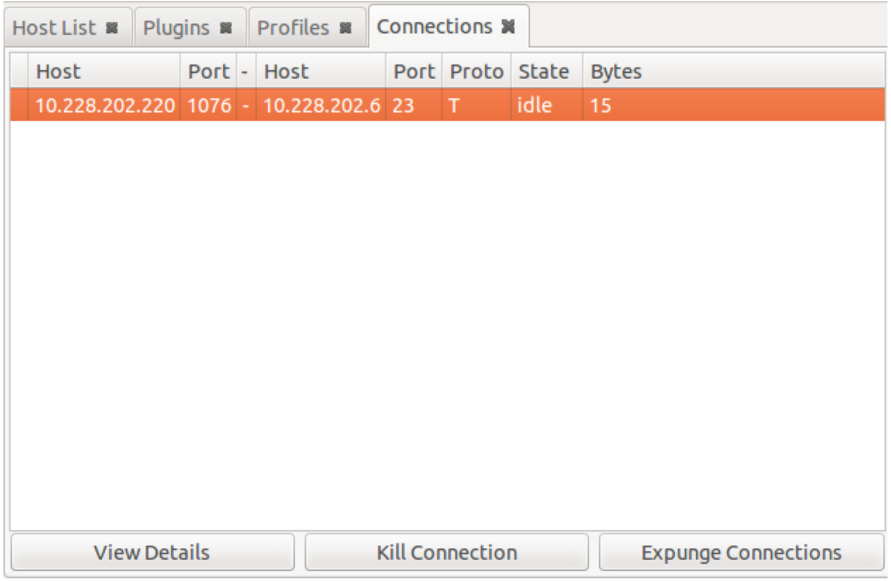
```
GROUP 2 : 10.228.202.220 00:0C:29:EE:23:5C
Activating repoison_arp plugin...
Activating remote_browser plugin...
Activating chk_poison plugin...
chk_poison: Checking poisoning status...
chk_poison: Poisoning process successful!
```

I was going to use the “remote_browser” plugin, which will return the URLs the victim have visited, but unfortunately our network didn’t have access to the internet at this time. Instead I chose to see if I can sniff a telnet connection from the victim computer, to the switch’s IP, 10.228.202.6, and Ettercap caught it:



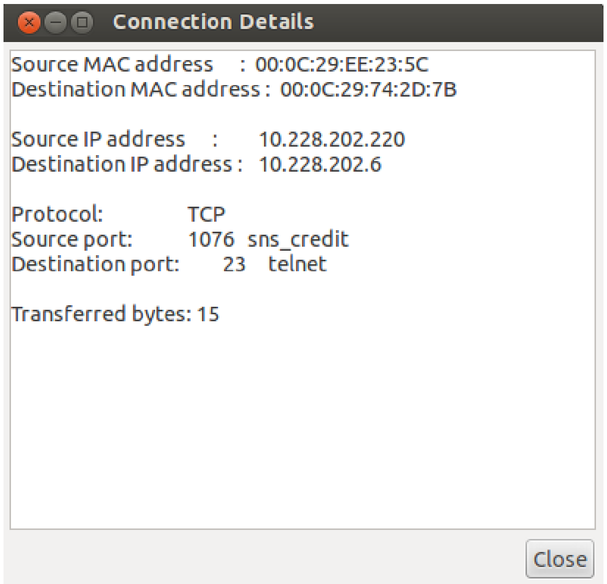
```
GROUP 2 : 10.228.202.220 00:0C:29:EE:23:5C
Activating repoison_arp plugin...
Activating remote_browser plugin...
Activating chk_poison plugin...
chk_poison: Checking poisoning status...
chk_poison: Poisoning process successful!
TELNET : 10.228.202.6:23 -> USER: cisco  PASS: cisco
```

When I saw this, i opened up “connections”, which will show all connections from the victim.



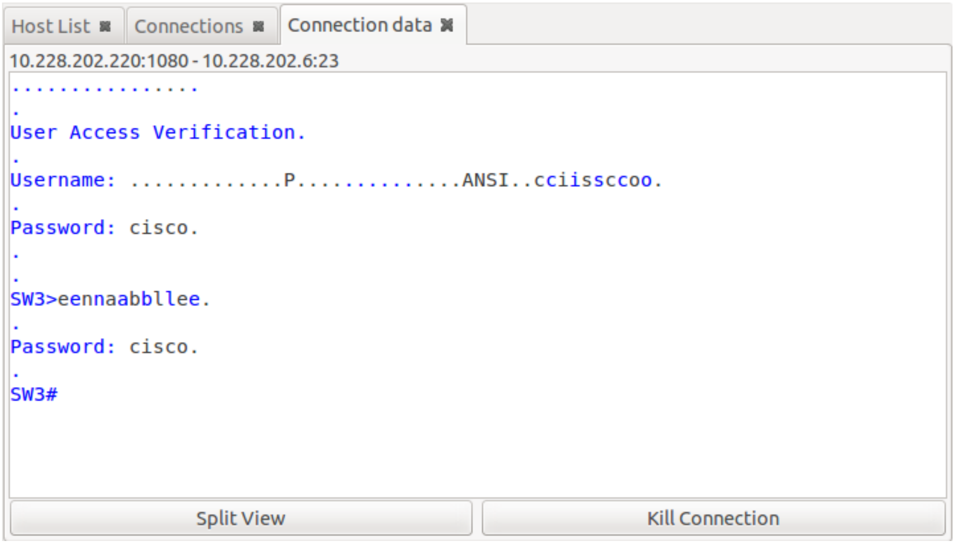
Host	Port	-	Host	Port	Proto	State	Bytes
10.228.202.220	1076	-	10.228.202.6	23	T	idle	15

As you can see, it uses Port 23, which is telnet, you're also able to get a more detailed layout of the connection if you right click and press "View details", and it looks like this:



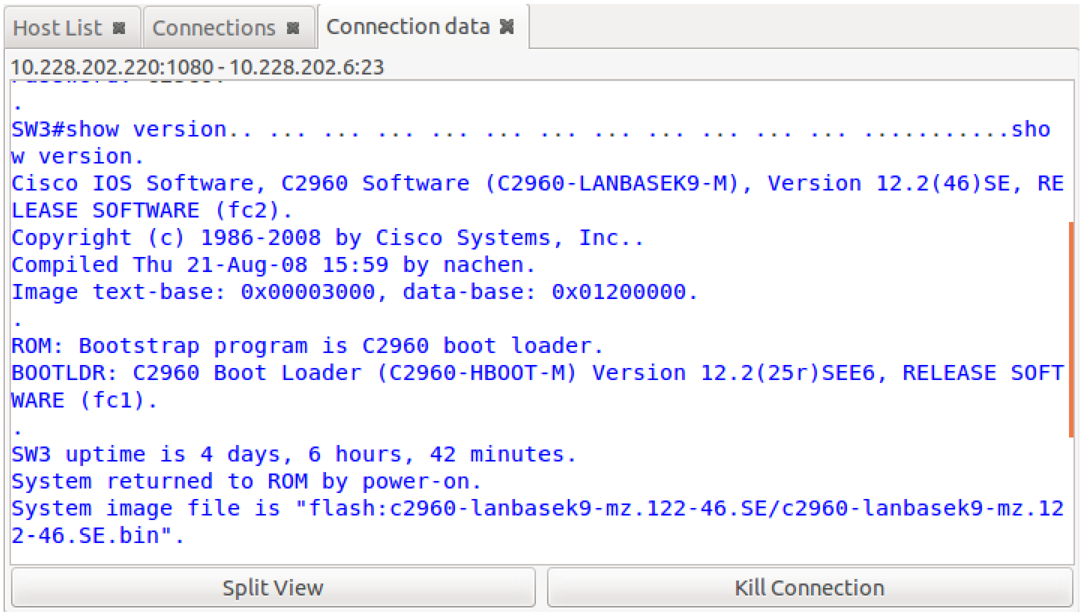
```
Source MAC address     : 00:0C:29:EE:23:5C
Destination MAC address : 00:0C:29:74:2D:7B

Source IP address    :     10.228.202.220
Destination IP address :  10.228.202.6

Protocol:          TCP
Source port:       1076  sns_credit
Destination port:      23    telnet

Transferred bytes: 15
```

If you go back to connections and double click on the telnet connection, you'll get a split screen and see exactly what the victim have been entering into his terminal. It can look a bit messy at the beginning, but you can clearly understand what's going on.



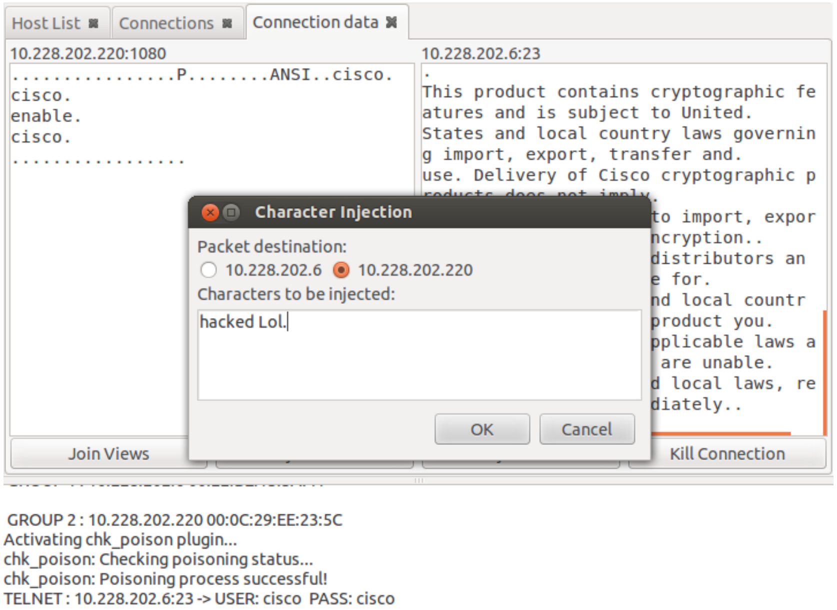
```
10.228.202.220:1080 - 10.228.202.6:23
................
.
User Access Verification.
.
Username: ............P.............ANSI..cciissccoo.
.
Password: cisco.
.
.
SW3>eennaabbllee.
.
Password: cisco.
.
SW3#
```

You got the username and the password before when the victim first telneted into the switch, and now you got the enable password, which is cisco. The dots(.), means that the victim have pressed enter(injected a <cr>) and is not part of the password or username. Now you can go back to "Split view", there you can inject data or even files. First I tried to inject some data, and I sent the command "show version", but it didn't really work. "Show version" just appeared in the victim's terminal without being executed. It appears that I forgot to inject a <cr>. So, I tried again, but before sending "show version" again, I pressed enter into text field so I sent "show version" and "<cr>". Then show version got executed and started showing its output:



```
10.228.202.220:1080 - 10.228.202.6:23
.
SW3#show version.. ... ... ... ... ... ... ... ... ... ... ... ...........sho
w version.
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 12.2(46)SE, RE
LEASE SOFTWARE (fc2).
Copyright (c) 1986-2008 by Cisco Systems, Inc..
Compiled Thu 21-Aug-08 15:59 by nachen.
Image text-base: 0x00003000, data-base: 0x01200000.
.
ROM: Bootstrap program is C2960 boot loader.
BOOTLDR: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(25r)SEE6, RELEASE SOFT
WARE (fc1).
.
SW3 uptime is 4 days, 6 hours, 42 minutes.
System returned to ROM by power-on.
System image file is "flash:c2960-lanbasek9-mz.122-46.SE/c2960-lanbasek9-mz.12
2-46.SE.bin".
```

When I saw that it worked, I tried to inject data to the victims computer. So instead of sending it to 10.228.202.6, I sent it to 10.228.202.220.



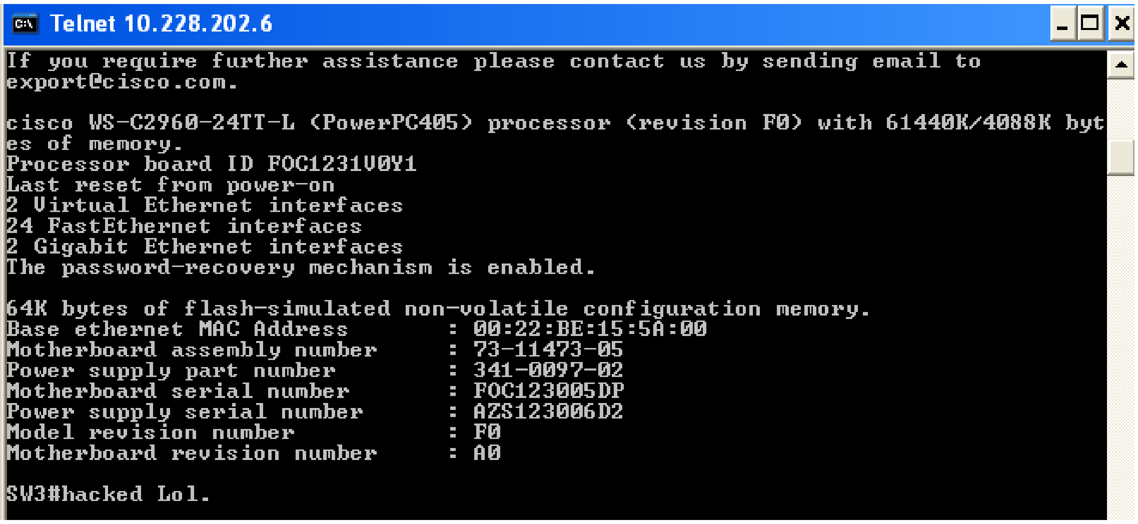
```
10.228.202.220:1080
.................P........ANSI..cisco.
cisco.
enable.
cisco.
.................
```

```
10.228.202.6:23
.
This product contains cryptographic fe
atures and is subject to United.
States and local country laws governin
g import, export, transfer and.
use. Delivery of Cisco cryptographic p
roducts does not imply.
to import, expor
ncryption..
distributors an
e for.
nd local countr
product you.
pplicable laws a
 are unable.
d local laws, re
diately..
```

Packet destination:
○ 10.228.202.6 ● 10.228.202.220
Characters to be injected:
hacked Lol.

```
 GROUP 2 : 10.228.202.220 00:0C:29:EE:23:5C
Activating chk_poison plugin...
chk_poison: Checking poisoning status...
chk_poison: Poisoning process successful!
TELNET : 10.228.202.6:23 -> USER: cisco  PASS: cisco
```

When I clicked OK the victim got this:



```
If you require further assistance please contact us by sending email to
export@cisco.com.

cisco WS-C2960-24TT-L (PowerPC405) processor (revision F0) with 61440K/4088K byt
es of memory.
Processor board ID FOC1231V0Y1
Last reset from power-on
2 Virtual Ethernet interfaces
24 FastEthernet interfaces
2 Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.

64K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address       : 00:22:BE:15:5A:00
Motherboard assembly number     : 73-11473-05
Power supply part number        : 341-0097-02
Motherboard serial number       : FOC123005DP
Power supply serial number      : AZS123006D2
Model revision number           : F0
Motherboard revision number     : A0

SW3#hacked Lol.
```

So I managed to ARP poison, and perform and Mitm attack.

Summary

As I mentioned in my introduction, if someone manage to pull a ARP poisoning and Mitm attack it can have devastating consequences. The insight I got from doing this study, is how easy it is to perform without any defense like port security, ArpOn or ARPwatch. The switch got all its passwords encrypted as far as I know, but that didn't help at all since it catches whatever the victim enters. With a graphical tool like Ettercap pretty much anyone can be successful with an ARP poisoning and a Mitm attack, since it is so easy to use and there's also a lot of documentation, guides and such available on the internet.

I wouldn't say that Ettercap can only be used to harm a network, I think it can help you secure your network aswell. You can use it to retrieve information about hosts on the LAN, see if they have any open ports, or just spy on your employees to see if they actually work.

It can be very harmful, but in my opinion it is pretty easy to defend yourself against it ARP poisoning and Mitm attacks, you just have to shutdown every unused port so no one can plug themselves into an unused port on your switch, and use port security features on the used ports, and/or use a tool like ARPwatch or have every host run ArpOn daemon. In my opinion, the easier and quicker way to go would be to configure port security on the used ports of the network. I do also believe port security is the most secure option to use, ARPwatch delivers every "suspect" ARP activity, and could maybe react on something that ain't something harmful. If you're going to use ArpOn, every host must have it. If someone manages to shut it down, maybe because they do not know what it is or does, you have a leak in your security. With port security you control the security and do not have to rely or check up on every ArpOn host to see if it actually runs as it should.

If you have any of those defense methods, especially port security, I do not see how someone could make a successful attack, even if its an angry employee connected physically to the network.

References

[1] http://www.webopedia.com/TERM/A/ARP_spoofing.html, aquired 2014-01-13

[2] http://www.watchguard.com/infocenter/editorial/135324.asp, aquired 2014-01-13

[3] http://arpon.sourceforge.net/, aquired 2014-01-13

[4] http://ettercap.github.io/ettercap/, aquired 2014-01-13