Active Network Sniffing in Switched Environments

- Unlike a hub, an Ethernet switch does not broadcast all received packets to every system on the LAN.
- A switch is designed to have more intelligent than a hub. It examines the MAC address associated with each packet passing through it, sending data only to the required connection on the switch.
- When Computer A sends information to Computer B, the switch is intelligent enough to send the information out a single port, to Computer B, rather than announce it to all of the computers that are plugged into the switch.
- A switch therefore limits the data that can be captured on a passive sniffer to those packets destined for and arriving into the machine that the sniffer is running on.
- The switch maintains a table of information that maps the port on the switch to the MAC (Media Access Control) address of the computer that is plugged into each port of the switch.
- However, attackers have created a variety of tools that actively inject traffic into the LAN to support sniffing a switched environment.
- Dsniff is one of the most comprehensive and powerful, freely available packet-sniffing tool suites for capturing and processing authentication information.
- Its functionality and its comprehensive suite of utilities have made it the tool of choice used by attackers to sniff passwords and authentication information off networks.
- Dsniff is written by Dug Song (also designed FragRouter) and is available at www.monkey.org/~dugsong/dsniff.
- Dsniff runs on OpenBSD, Linux, and Solaris and there is also a Win32 port available. The core of the Dsniff suite is the sniffer program itself, called Dsniff.
- Like most other sniffers, this tool can be used to capture information passing across the network.
- But the main advantage of the Dsniff centerpiece sniffer, however, is the wide variety of protocols that it can interpret.
- Nearly every sniffer can dump the raw bits captured on the segment. However, these raw bits have little value unless the attacker can interpret their meaning by accurately parsing the information to see the various fields.
- Dsniff is designed with the capabilities for decoding a large number of Application level protocols:
- FTP, Telnet, SMTP, HTTP, POP, Poppas, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, NFS, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL*Net, Sybase SQL, and Microsoft SQL auth info.
- The ability to properly and automatically detect and interpret this enormous list of Application-level protocols is highly useful to both attackers and security professionals.
- Beyond its ability to decode all of these Application-level formats, the Dsniff suite's main feature is its ability to actively manipulate traffic.

- The Dsniff suite includes a variety of tools that will enable an attacker interact with traffic to conduct advanced sniffing attacks, such as sniffing through a switch, remapping DNS names to redirect network connections, and even sniffing SSL and SSH connections.

Sniffing a Switched LAN

- Dsniff offers two methods for sniffing data from a switched LAN. The first technique is based on MAC flooding using a Dsniff program called **Macof.**
- The Macof utility works by sending out a flood of traffic with random MAC addresses on the LAN.
- As the number of apparent different MAC addresses in use on the network increases, the switch continues to store the MAC addresses used by each link on the switch.
- This will eventually exhaust the switch's memory with bogus MAC addresses. At this point, some switch implementations start forwarding data onto all links connected to the switch, in other words the switch becomes a hub.
- An attacker can take advantage of this flaw by using Macof to flood the switch to the point where it forwards traffic to other links, and running any sniffer tool. The attacker is now sniffing a switched LAN.
- After the attacker has found a host or hosts from which he wants to sniff authentication packets, he starts spoofing the switch by sending forged ARP replies to the switch, which adds the sniffing host's IP address to the ARP cache, which maps it to the same port as the target host(s).
- The following example shows the use of Macof. In this example - i represents the interface, - s is the source IP, and -e is the target hardware address:

```
#./macof -i eth0 -s 192.168.1.100 -e 00-E0-81-04-6E-00
```

- Some switches are not vulnerable to this MAC flooding attack because they stop storing new MAC addresses when the remaining capacity of their memory reaches a given limit.
- To sniff in a switched environment where MAC flooding doesn't work, Dsniff includes a tool called **arpspoof**. As its name implies, arpspoof allows an attacker to manipulate Address Resolution Protocol (ARP) traffic on the LAN.
- ARP spoofing exploits the inherent security weaknesses of how hosts on a broadcast network maintain information about the computers around them.
- ARP Spoofing is a technique that uses forged MAC and IP addresses to masquerade as another machine in ARP cache.
- The ARP cache contains mapping information for translating a given IP address with its hardware MAC address.
- When a host wishes to communicate with another host, the requester's machine checks its ARP cache for a mapping of the host's IP address to hardware address (MAC address).
- If there is a listing in the requesters ARP cache, it proceeds to establish a connection. If the requester does not have a mapping for the host in its APP cache, it will transmit an ARP request to all hosts on the network segment.

- Under normal conditions, only the host with the requested MAC address will reply with its IP. After the host transmits its IP and hardware address, a connection is established.

- The security flaw here is that after a host's IP address is mapped in a system's ARP cache, it is considered a trusted machine.

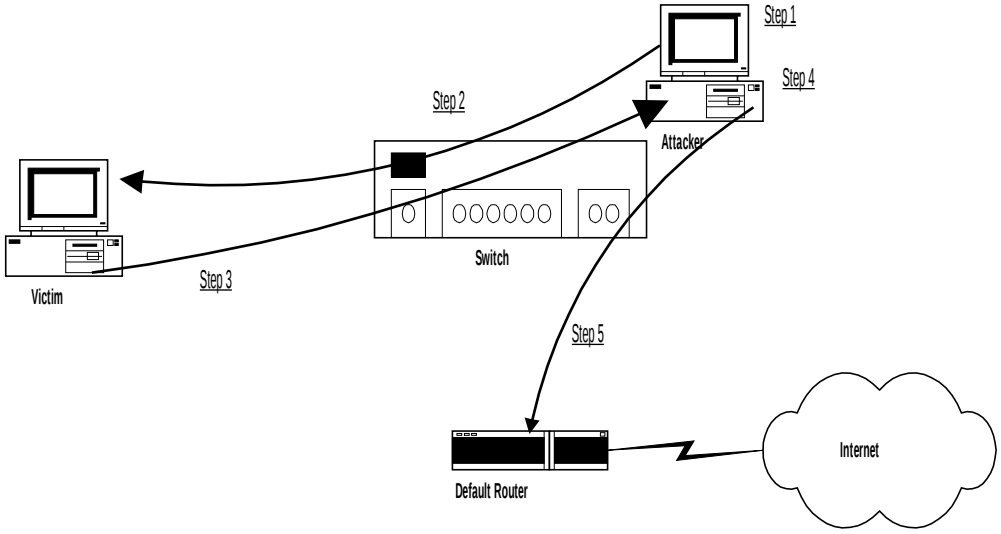
- This flaw is further compounded in that an ARP request is not necessary for a host to accept an ARP reply from a host. Many systems will except the non-requested ARP reply and update their caches with the information.

- On a switched network, a switch can be configured to assign multiple IP addresses to a single port on a switch. This enables ARP spoofing tools, such as Dsniff, to trick the switch into adding a masqueraded MAC address into its cache and connecting the attacker's machine to the same port as a target machine.

- Now that both an attacker's machine and a target are receiving broadcasted information on the switch, authentication data can again be sniffed off the line.

- To use arpspoof, the attacker must first create a map of the network, usually generated during the reconnaissance phase of the attack. The attacker obtains the IP address of the default router for the LAN.

- The basic process is illustrated in the diagram below.



- The attacker sets up the attack by configuring the IP layer of the attacker's machine to forward any traffic it receives from the LAN to the IP address of the default router. The attacker does this by activating an option available in many operating systems called IP forwarding (step 1).

- With this configuration, any traffic sent through the switch to the attacker's machine that is destined for any other IP address will be forwarded to the default router for the LAN.

- After completing this set-up phase, the attacker activates the Dsniff arpspoof program, which sends fake ARP replies to the victim host (step 2).

- The attacker's fake ARP message changes the victim's ARP table by remapping the default router's Layer-3 (IP) address to the attacker's own Layer-2 (MAC) address.

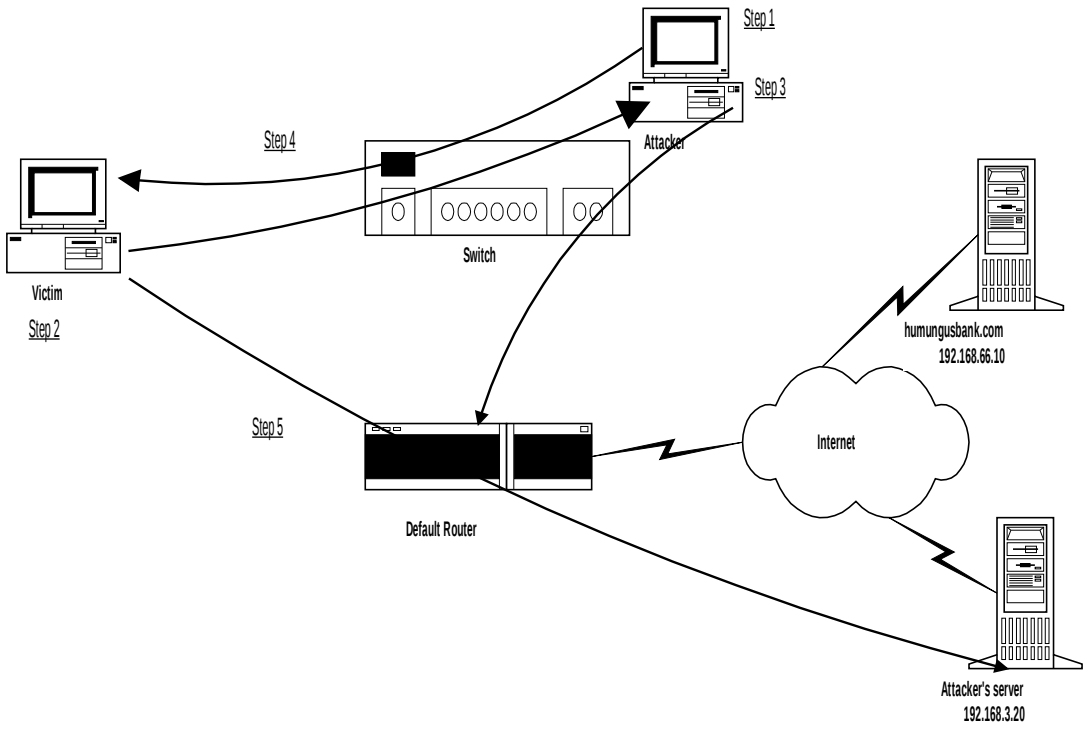
- In effect, the attacker tells the victim that to access the default router, use the attacker's MAC address, thereby poisoning the ARP table of the victim.
- Once the poisoned ARP message takes effect, all traffic from the victim machine to the outside world will be sent to the attacker's machine (step 3).
- The victim host sends the data, forwarding it to what it thinks is the default router, but using the attacker's MAC address.
- The attacker sniffs the information from the line using any kind of sniffing tool (step 4).
- Finally, the attacker's machine forwards the victim's traffic to the actual default router on the LAN, because we configured the attacker's machine for IP forwarding in the first step (step 5).
- Upon reaching the actual default router on the LAN, the traffic is transmitted to the outside world. In essence, the arpspoof program redirects the traffic so that it bounces through the attacker's machine on its way to the outside world.
- The attacker is now sniffing in a switched environment. This is why the IP forwarding set up is crucial. If IP forwarding were not enabled on the attacker's machine, the victim machine would not be able to send any traffic to the outside world, resulting in an inadvertent denial-of-service attack.
- It is important to note that this arpspoof technique doesn't target the switch itself. Instead, arpspoof manipulates the mapping of IP address to MAC address in the victim machine's ARP table to allow sniffing in a switched environment.

Sniffing and Spoofing DNS

- Dsniff includes a program called **dnsspoof** that lets an attacker send a false DNS response to a victim, which will make the victim access the attacker's machine when they intended to access another (valid) machine.

- Suppose **humungusbank.com** is an online bank. If a user wants to surf to **humungusbank.com,** the attacker can trick the client into connecting to the attacker's Web server, where the attacker could display a fake bank login screen, gathering the victim's userid and password.

- The diagram shown illustrates how Dsniff's DNS spoofing works.



- **Step 1**

  - The attacker starts the **dnsspoof** program from the Dsniff suite. This program sniffs the LAN, looking for DNS queries for specific hosts, such as **humungusbank.com**.

  - If the LAN is constructed with a hub, the DNS queries are captured off of the LAN using passive sniffing.

  - If the LAN is switched, the **arpspoof** program can be used to capture them from the target.

- **Step 2**

- The victim tries to resolve the name **humungusbank.com** using DNS, perhaps by trying to surf to the bank's Web site.

- **Step 3**
  - The attacker sniffs the DNS query from the line.
- **Step 4**
  - The attacker immediately sends a fake DNS response.
  - This response will deceive the victim into believing that **humungusbank.com** should resolve to 192.168.3.20 (the IP address of attacker's server in the outside world), instead of 192.168.66.10 (the real bank's Web site).
  - The victim machine will cache this incorrect DNS entry.
  - At some later time, the real response from the real DNS server will arrive, but be ignored by the victim machine because it already has the DNS mapping for **humungusbank.com.**
- **Step 5**
  - The victim's browser makes a connection with the system at 192.168.3.20, which it thinks is **humungusbank.com**.
  - Unfortunately, this is the attacker's system, pretending to be the bank.

Signature of the Attack

- Dsniff is a passive attack on the network, so it leaves few signs of its existence. Because Dsniff focuses on capturing authentication information, an attacker is likely to place the program on a host close to server that receives many authentication requests.
- Common targets are hosts and gateways that are situated between two different network segments. One benefit for security analysts is that Dsniff places the host machine's network interface in promiscuous mode, which will show up on sniffer detectors.
- Another sign of Dsniff can be large amounts of disk space being consumed. Depending on Dsniffs configuration and the amount of network authentication traffic, the file that Dsniff uses to store the capture data can grow quite large.
- Signs of ARP spoofing are frequent changes to ARP mappings on hosts and switches. Administrators may also see an abnormal amount of ARP requests. Numerous invalid entries in ARP tables can also be a sign of ARP spoofing activity.

Detection and countermeasures

- Dsniff itself does not show up on IDSs or security audit logs because it is not intrusive. Dsniff also does not show up as a network resource log because it only looks at the first few bytes of a packet.
- The only real way to prevent the sniffing of traffic by Dsniff would be to deploy and use data encryption, like IPSec.
- Ensure that programs such as telnet are replaced with programs such as SSH, that do not transmit authentication information in clear text. All programs that have the ability to encrypt authentication and session information should be implemented.
- Although there are no sure ways of protecting a network from Dsniff and ARP spoofing, there are several different methods that can be used to mitigate the vulnerability.
- First, security analysts should use one or more of the commercial or freely available tools to search the network for sniffers and machines that are in promiscuous mode.
- Anti-sniff measures the reaction time of network interfaces. From these reaction times, Anti-sniff is able to extrapolate whether a host's network interface is in promiscuous mode.
- There are also some freely available tools that can help monitor and detect ARP spoofing as well. A tool that can be used is ARPWatch.
- ARPWatch is a free UNIX utility, which monitors IP/Ethernet mappings for changes.
- Another method that can be used to defend against these forms of attacks is the use of static ARP mappings. Many operating systems allow for APP caching to be made static instead of timing out every couple of minutes.
- This method is effective for preventing ARP spoofing, although it requires manual updating of the ARP cache every time there is a hardware address change.
- Security analysts and network administrators can conduct baselines on the amount of ARP traffic that is sent across the network. From these baselines, administrators can monitor whether abnormal amounts of ARP traffic are being sent.

Additional References

- Christopher R. Russel has a paper that can be found on the SANS web page that goes into detail about how to use dsniff once it is installed, called "Penetration Testing with dsniff" (http://www.sans.org/infosecFAQ/threats/dsniff.htm).
- As described by Dug Song, dsniff contains the following utilities, which are explained in more depth in the man pages, in the dsniff readme, and in Russel's article "Penetration Testing with dsniff."
- The following URLs provide information about techniques used in sniffing switched-based networks and steps to mitigate the security threats:

- http://www.sans.org/infosecFAQ/switchednet/sniffers.htm
- http://www.securitysoftwaretech.com/antisniff/

- The following is a list if the additional utilities provided with dsniff.

arpspoof:	redirects packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies. This is an extremely effective way of sniffing traffic on a switch. Kernel IP forwarding (or a userland program which accomplishes the same, e.g. fragrouter (8)) must be turned on ahead of time.
dnsspoof:	forges replies to arbitrary DNS address / pointer queries on the LAN. This is useful in bypassing hostname-based access controls, or in implementing a variety of man-in-the-middle attacks.
dsniff:	is a password sniffer which handles FTP, Telnet, SMTP, HTTP, POP, poppas, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP MS-CHAP, NFS, VRRP, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix, ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL*Net, Sybase and Microsoft SQL protocols.
filesnarf:	saves files sniffed from NFS traffic in the current working directory
macof:	floods the local network with random MAC addresses (causing some switches to fail open in repeating mode, facilitating sniffing.
mailsnarf:	outputs e-mail messages sniffed from SMTP and POP traffic in Berkeley mbox format, suitable for offline browsing with your favorite mail reader (mail(1), pine(1), etc.).
msgsnarf:	records selected messages from AOL Instant Messenger, ICQ 2000, IRC, MSN Messenger, or Yahoo Messenger chat sessions.
tcpkill:	kills specified in-progress TCP connections (useful for libnids-based applications which require a full TCP 3-whs for TCB creation.
tcpnice:	slows down specified TCP connections on a LAN via "active" traffic shaping.
urlsnarf:	outputs all requested URLs sniffed from HTTP traffic in CLF (Common Log Format, used by almost all web servers), suitable for offline post-processing with your favorite web log analysis tool (analog, wwwstat, etc.).
webspy:	sends URLs sniffed from a client to your local Netscape browser for display, updated in real-time (as the target surfs, your browser surfs along with them, automagically). Netscape must be running on you local X display ahead of time.
sshmith:	proxies and sniffs SSL traffic redirected by dnsspoof, captures password logins and optionally allows hijacking interactive sessions.[1]
webmitm:	proxies and sniffs HTTP/HTTPS traffic redirected by dnsspoof, capturing SSL-encrypted logins and form submissions.[1]

Ettercap

- Ettercap is an open source program that combines a packet sniffer with pop/http/https/sftp and many other password crackers. It is the latest tool for sniffing switches and it has many more newer features that the older and deprecated dsniff application.

- Ettercap is a complete suite for man in the middle attacks on LAN. It features sniffing of live connections, content filtering on the fly and many other interesting capabilities for exploiting network traffic.

- It supports active and passive dissection of many protocols (even ciphered ones) and includes many feature for network and host analysis.

- It also provides other capabilities such as the use of custom filters and plug-ins, the ability to steal SSL/SSH logins, such as the logins to Gmail, Yahoo, and many other “secure” connections.

- The tool relies heavily on ARP spoofing, which has already been discussed above. Ettercap provides a very convenient and easy to use GUI.

- This tool provided the capability to easily perform man-in-the-middle (MITM) attacks in a switched LAN environment as the launch pad for many of its other functions.

- Once it is inserted in the middle of a switched connection, it can capture and examine all communication between the two victim hosts, and subsequently take advantage of the following features:

    - Character injection: Insert arbitrary characters into a live connection in either direction, emulating commands sent from the client or replies sent by the server

    - Packet filtering: Automatically filter the TCP or UDP payload of packets in a live connection by searching for an arbitrary ASCII or hexadecimal string, and replacing it with your own string, or simply dropping the filtered packet.

- SSH1 support: Capture username, password, and the data of an SSH1 connection

- HTTPS support: Insertion into an HTTP SSL session, as long as a false certificate is accepted by the user

- PPTP suite: Perform man-in-the-middle attacks against PPTP tunnels

- Kill any connection: View and kill arbitrary active connections

- It also has many useful reconnaissance tools built in, to ensure that an attacker can stealthily gain awareness of the LAN topology before launching MITM attacks:

    - Active OS fingerprinting: Directly probe a LAN host to identify its operating system, using the *nmap* database

    - Passive LAN scanning: By listening to and analyzing passing frames, collect information about LAN hosts such as the operating system, open ports, running services, and IP and MAC addresses

    - IP and MAC-based sniffing: Listen to LAN traffic in promiscuous mode and capture passing traffic. This feature is similar to common packet capture utilities, such as *tcpdump*, and allows filtering by IP or MAC address.

- Search for other ARP poisoning activity on the LAN and promiscuous mode NICs:

    - Detect other systems that are currently sniffing on the LAN, or performing ARP cache poisoning attacks.

    - Packet forge: Construct and send custom Ethernet frames and IP packets to test the responses of network devices. This function has features similar to the tool *hping*3, and may be used to manually set header flags and spoof IP and MAC address [1].

- It is important to note that attacks using this tool can be very disruptive to a live production network, so it is imperative to experiment on an isolated test network.

- This attack has several limitations. It is important to note that **Sniffer** must forward all intercepted packets to the correct victim hosts, or the result would be a denial of service, as no frames sent between the two hosts would ever reach their destination if **Sniffer** merely discarded them.

- ARP poisoning attacks will also degrade network performance, as the attacking system must intercept, analyze, and forward each frame sent between the two victims.

- Finally, one cannot poison the caches of computers on a different subnet or VLAN because ARP broadcasts only reach systems within a single Ethernet broadcast domain.