TPS Quickstart - Sniffing switched networks with Dsniff (v1.0)

Chris Vicious – chris@seclogic-as.com

Abstract:
Dsniff is an advanced suite of utilities which are commonly used for network and penetration testing. This document is intended as a Quickstart guide to implementing a common Man-in-the-Middle (MitM) attack using Dsniff and MAC address spoofing. Before experimenting, please **make sure that you are authorized** to utilize these techniques on the network that you're testing. For additional warnings and information, please refer to the Dsniff FAQ below.

Background:
To understand how or why Dsniff works in the manner in which it does, it is helpful to have a little background information with regard to *shared* and *switched* network topologies.

In a *shared* network topology (e.g. using a simple hub), Ethernet frames are broadcast to every host on the local segment. Under normal circumstances, the frames are processed only by the host for which the traffic is destined. With this type of network, one may – assuming root access and a NIC that permits *promiscuous mode* – easily deploy a sniffer to any host on the segment in order to capture data.

In a *switched* network topology, Ethernet frames are sent only to the host for which the data is destined – which has historically been a relatively successful defense against rudimentary sniffing attacks. The destination host is signified within the Ethernet frame by the MAC address – a "unique" hardware address for each NIC attached to the network.

Tools Used:
TPS Linux
- Fragrouter
- Dsniff (Dsniff)

- Mailsnarf (Dsniff)
- Arpspoof (Dsniff)

Step 1 - Fragrouter:
One of the most common errors in attempting to sniff a switched network is when the attacker fails to enable IP forwarding. **Failing to enable IP forwarding can break things in a serious way**. If IP forwarding is not enabled, your attacking host essentially becomes a black hole that absorbs packets. This is especially bad if you're representing yourself as a gateway host. There are various ways to enable IP forwarding, but the easiest is to use Fragrouter (provided in TPS). To use Fragrouter to enable IP forwarding:

```
(attacker@host) $ fragrouter –B1
fragrouter: base-1: normal IP forwarding
192.168.1.2.2080 > 192.168.1.1: S 592459704:592459704(0)
192.168.1.2.2080 > 192.168.1.1: P 592459705:592459717(12)
192.168.1.2.2080 > 192.168.1.1: . ack 235437339
192.168.1.2.2080 > 192.168.1.1: P 592459717:592459730(13)
```

Step 2 - Arpspoof:
One of the simplest – and unfortunately effective – ways to sniff on a switched network is with MAC address spoofing. Essentially, your attacking host is flooding a target with ARP-replys saying "Data bound for IP address xxx.xxx.xxx.xxx (target) should go to MAC address xx:xx:xx:xx:xx:xx (attacker)". This is a basic MitM technique that is commonly used to insert your attacking host between a target host (or hosts) and some other device, such as another computer or the local gateway. To start Arpspoof on eth0 and insert your machine between your target and the gateway (where 192.168.1.1 is the gateway and 192.168.1.2 is the target):

```
(attacker@host) $ arpspoof –i eth0 –t 192.168.1.1 192.168.1.2 &
0:50:56:7b:b4:d8 0:1:2:9a:19:bd 0806 42: arp reply 192.168.1.2 is-at
0:50:56:7b:b4:d8
0:50:56:7b:b4:d8 0:1:2:9a:19:bd 0806 42: arp reply 192.168.1.2 is-at
0:50:56:7b:b4:d8

(attacker@host) $ arpspoof –i eth0 –t 192.168.1.2 192.168.1.1 &
0:50:56:7b:b4:d8 0:50:56:59:2a:a2 0806 42: arp reply 192.168.1.1 is-at
0:50:56:7b:b4:d8
0:50:56:7b:b4:d8 0:50:56:59:2a:a2 0806 42: arp reply 192.168.1.1 is-at
0:50:56:7b:b4:d8
```

Step 3 - Sniffing:
At this point, you're free to use any of the ancillary sniffer packages that are provided with Dsniff. In keeping with the spirit of this article, I'll only address a couple of the more basic utilities.

Dsniff - "dsniff is a password sniffer which handles FTP, Telnet, HTTP, POP, poppass, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, NFS, YP, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, and Oracle SQL*Net protocols." To start Dsniff on eth0 and log to an output file:

```
(attacker@host) $ dsniff –i eth0 –w output.txt
dsniff: listening on eth0
```

Mailsnarf - "mailsnarf outputs all messages sniffed from SMTP traffic in Berkeley mbox format, suitable for offline browsing with your favorite mail reader (mail(1), pine(1), etc.)." To start mailsnarf listening on eth0:

```
(attacker@host) $ mailsnarf –i eth0
mailsnarf: listening on eth0
```

Conclusion:
While there is far more that can be done with Dsniff, it provides immediate facilities for bringing to light the security shortcomings of networks upon which MAC spoofing attacks are successful. Proactively implementing *port security*, *routing security*, and the like, would have a great impact on mitigating potential attacks.

Resources:
Dsniff FAQ
http://www.monkey.org/~dugsong/dsniff/faq.html

Intrusion Detection FAQ - Why your switched network isn't secure.
http://www.sans.org/resources/idfaq/switched_network.php

Dsniff 'n the Mirror
http://www.linuxsecurity.com/docs/PDF/dsniff-n-mirror.pdf

Introduction to Networking
http://www.siliconvalleyccie.com/linux-hn/network-intro.htm