

Hijacking a Network Connection on a Switched Network

Last updated: 9th of May 2003

The vulnerability was analyzed by:

- Kimmo Kasslin, kimmo.kasslin@hut.fi
- Antti Tikkanen, antti.tikkanen@hut.fi

Threat and Vulnerability

To hijack a network connection of our target machine we have to be able to direct the flow of network traffic from the target machine to our machine. The rest is accomplished by redirecting the packets in the kernel level.

This problem can be solved by the weaknesses of the ARP (address resolution protocol). The ARP is a stateless protocol so it is completely legal by the protocol to send ARP reply packets to the target machine even if it has not send any ARP requests yet. This makes it possible for the attacker to send forged ARP reply packets continuously to the victim where the MAC address is forged to correspond to the one of the attacker's machine. Usually when you want to sniff the traffic originating from a machine, you need to spoof the gateway of the network.

Now we are able to listen to the network traffic from the victim machine. The packet redirection is accomplished with kernel tools such as iptables on Linux.

Preconditions for the Attack

The following assumptions will be made about ARP spoofing:

- The network is a switched network.
- Both the attacker and the victim are located on the same logical network segment.
- The victim has default ARP configuration.

Attack Environment Used

Our environment will consist of:

- Windows 2000 Professional SP3 (the victim)
- Red Hat Linux 8.0 (attacker)

The network is switched. The victim and the attacker are located on the same logical network segment. No encryption is used in the network layer (IPSEC).

Analysis of the Attack

The tool required for this attack are already implemented. Dsniff is available for download from [1]. We only need one tool from this package: arpspoof. Iptables is available on most Linux distributions by default.

The ARP spoof is carried out as follows:

First you have to make sure that the attacking machine has ip-packet forwarding enabled. On RedHat Linux 8.0 this can be accomplished by executing the command:

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

From the attacking machine run the following command:

```
arp spoof -t <ip-address of the victim> <ip-address of the gateway>
```

Packet redirection is done with iptables with the following commands:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp -s <victim ip> -d <server ip> --dport <dest_port> \
-j REDIRECT --to-port <dest_port_on_attacker_machine>
```

Now traffic to the port *<dest_port>* from the victim to the server is redirected to the attacker's IP address with destination port *<dest_port_on_attacker_machine>*.

Detection and tracing

There are generally two well known ways to detect ARP spoofing attempts – monitoring the local ARP cache and monitoring the network traffic on the wire.

ARP cache monitoring on a local machine can be accomplished with the *arp*-command. It is quite trivial to notice if the gateway's MAC address has changed (assuming the real MAC address of the gateway is known). This can be done automatically with a tool called *arpwatch* [2].

Network traffic monitoring can be implemented with certain Intrusion Detection Systems. The Open Source IDS called Snort [3] is able to do this in real time.

Protection against the Attack

One of the best ways to protect machines against ARP spoofing attacks is to enforce static ARP entries on the local machines, especially the entry for the local gateway should be static.

Test results

The attack was completed successfully on a switched network environment. As a result of the successful ARP spoofing attack we were able to route all the traffic from the victim to the client through the attacking machine. This allowed us to perform the password attack against Kerberos V [4]. Kernel level redirection allowed us to route the traffic both ways between the client and the server. This man-in-the-middle situation allowed us to launch the replay attack on Kerberos V and SMB [5].

References

- [1] D. Song. Toolset dsniff. <http://naughty.monkey.org/~dugsong/dsniff/>. Referenced 10.2.2003.
- [2] LBNL's Network Research Group. <http://www-nrg.ee.lbl.gov/>. Referenced 10.3.2003.
- [3] Snort.org. <http://www.snort.org/>. Referenced 10.3.2003.
- [4] K. Kasslin, A. Tikkanen. Password attack on Kerberos V and Windows 2000.
- [5] K. Kasslin, A. Tikkanen. Replay attack on Kerberos V and SMB.