SECURITY

LAN Switch Security

What Hackers Know About Your Switches

A practical guide to hardening Layer 2 devices and stopping campus network attacks

Eric Vyncke
Christopher Paggen, CCIE® No. 2659

ciscopress.com

LAN Switch Security

What Hackers Know About Your Switches

Eric Vyncke and Christopher Paggen, CCIE No. 2659

Cisco Press

Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Table 5-2 complements Figure 5-2. It contains a description of the fields found inside a DHCP packet.

Table 5-2 *Fields Found Inside DHCP Packets*

Field	Bytes	Description
Operation Code	1	1 = request, 2 = reply
Hardware Type	1	1 = 10 Mbps Ethernet, and so on
Hardware Length	1	Length of MAC address: 6 for Ethernet
Hop Count	1	Optionally used by relay agents
Transaction ID	4	Random number chosen by client used to correlate requests/replies
Seconds Elapsed	2	Filled by client—counts seconds elapsed since beginning of transaction
Flags	2	1 bit for broadcast flag, rest is zeroed
Client IP	4	Set to zero for new requests
Your IP	4	Address offered by server
Server IP	4	Address to use in next step of bootstrap process—returned by DHCPOFFER/ACK
Gateway IP	4	Address of the relay agent
Client Hardware Address	16	MAC address of the client
Server Host Name	64	Optional
Boot File Name	128	Optional
Options	Varies	—

Notice the absence of any authentication fields or any other security-inclined information in the packet. The protocol is built on a free-for-all model. Whoever requests an IP address is free to receive one, if available. When a client wants to obtain an IP address, it crafts a DHCPREQUEST packet by populating several of its fields. The client hardware address is of notable interest, because it serves as a (de)multiplexer on the server side to identify various clients. RFC 2131 reads as follows:

> The combination of client identifier or client hardware address and assigned network address constitute a unique identifier for the client's lease and are used both by the client and server to identify a lease referred to in any DHCP message.

It is common for DHCP servers to contain many available scopes (a range of IP addresses that can be served), because servers handle requests from many different networks. To select the appropriate scope for the client's network, DHCP servers select the Gateway IP Address field as a selector. Because the client does not yet know the IP address of its

gateway (this is its default router), the Gateway IP Address field is filled by the first router relaying the client DHCPDISCOVER to the actual DHCP server(s). This DHCP relay uses the IP address of the interface that received the original DHCPDISCOVER sent by the client.

Attacks Against DHCP

With the preceding information in mind, it should be clear that two attacks are possible:

- DHCP scope exhaustion (client spoofs other clients)
- Installation of a rogue DHCP server

DHCP Scope Exhaustion: DoS Attack Against DHCP

What if a malicious client attempts to seize the entire range of available IP addresses? It does not look like anything in the protocol itself is likely to prevent this from happening. The client just needs to generate uniquely identifiable packets. It could do so by using random source MAC addresses and then sending a DHCPDISCOVER per forged MAC address.

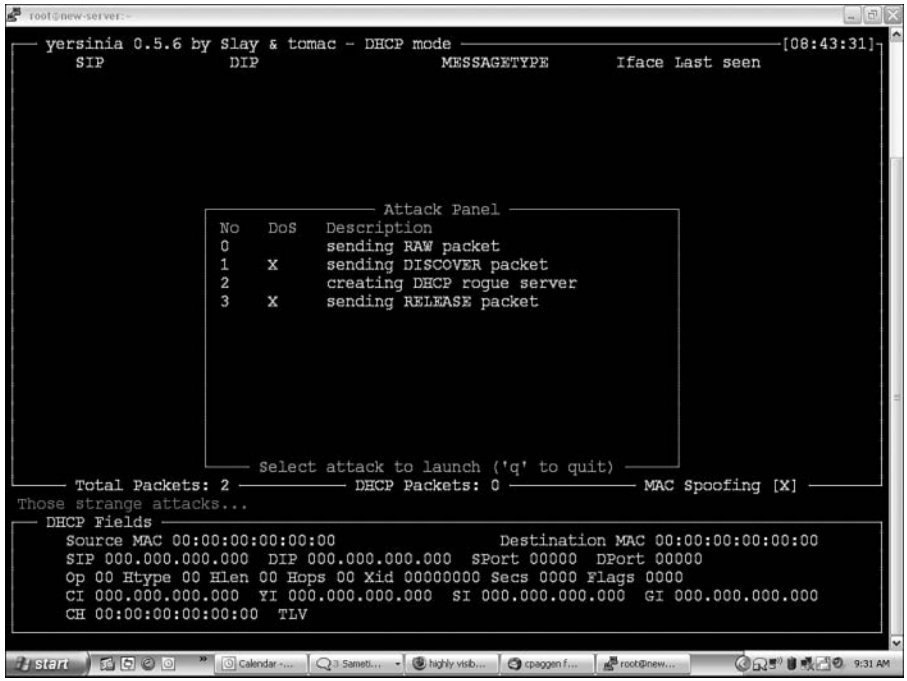
The DHCP server happily hands out the entire set of addresses available to the client's network, because it can't tell the difference between a genuine host and a spoofed one. If a legitimate client tries to obtain an IP address, it is abandoned with no IP connectivity because the entire range of addresses have already been allocated to spoofed hosts—user frustration guaranteed! At least two freely available programs exist—Yersinia and Gobbler—that do just that: Attempt to request as many leases as possible as quickly as possible.

Yersinia

Yersinia is the Layer 2 hacker's Swiss-army knife, as discussed in Chapter 3, "Attacking the Spanning Tree Protocol." Yersinia is named after *Yersinia pestis*, which is a bacteria that causes plague. As its name implies, Yersinia is mainly an attack tool against several Layer 2 protocols: Spanning Tree Protocol (STP), Institute of Electrical and Electronics Engineers (IEEE) 802.1Q, IEEE 802.1X, and, of course, DHCP (even if DHCP is not a Layer 2 protocol, strictly speaking).

Figure 5-3 shows a Yersinia attack screen.

Figure 5-3 *Yersinia's DHCP Attack Screen*



NOTE For more information on Yersinia, see Chapter 3.

Gobbler

Gobbler specializes in DHCP-only attacks. From its documentation,[2] Gobbler is described as follows:

> A tool designed to audit various aspects of DHCP networks, from detecting if DHCP is running on a network to performing a denial of service attack. The Gobbler also exploits DHCP and Ethernet to allow distributed spoofed port scanning with the added bonus of being able to sniff the reply from a spoofed host. This tool is based on proof of concept code "DHCP Gobbler" available from networkpenetration.com.

Gobbler even goes a step further than Yersinia. Certain DHCP servers periodically send Address Resolution Protocol (ARP) requests or Internet Control Message Protocol (ICMP) echo packets to probe for IP addresses that the server might have reclaimed. Servers do not perform this check for security purposes; instead, they do this because, sometimes, clients do not release their assigned IP address when shutting down.

The author(s) of Gobbler observed this DHCP server behavior and equipped Gobbler with the capability to counteract by responding to ARP requests!

Example 5-1 represents Gobbler's command-line interface (CLI) Help menu.

Example 5-1 *Gobbler's Help Menu*

```
[root@linux-p4]# ./Gobbler

The Gobbler (Alpha release 2.0.1) from NetworkPenetration.com
-------------------------------------------------------------
Scanning Options
-A <b,g,n,s,w> Arp scan (b)cast (g)obble (n)et-broadcast (s)pec* (w)rong
-C <g,s> Create a host (g)obble (s)pecified*
-D Detect DHCP service / rogue servers on network
-G Gobble attack -  DoS DHCP server via IP exhaustion / MAC spoofing attack
-M <d,l,o> DHCP mitm attack ns mitm (l)eaving subnet (o)ther ip range
-N <IP> None gobbled SYN scan*
-P <IP> SYN scan using a gobbled IP address
-Q <IP-r,m,n,1a:2b:3c:4d:5e:6f> Src IP-MAC (r)andom (m)ulticast (n)on-spoofed
-R <135-139,445,a,o,s,n> Port range (a)ll (o)sstm (s)ervices (n)nmap
-S Start sniffer
-T Traceroute to target (use with -P or -N)
-U ICMP ping target (use with -P or -N)
-X Nmap OS detection (use with -P or -N)
-Z Port 0 OS detection (use with -P or -N)

Misc
-a <x> Amount of pings (use with -U)
-c Closed ports displayed at end of portscan (all ports opposed to 20)
-d Filtered ports displayed at end of portscan (all)
-e <x> End of scan sleep for x seconds - wait for replies (default 2)
-f Fast mode - possible errors with port lists
-g Don't release gobbled IP's (might be handy when portscanning)
-h Don't ICMP ping target... useful if a firewall is blocking ICMP pings
-i <if> Interface (use before -Q if non spoofed mac)
-j Jump past rescanning filtered ports (useful when scanning all ports)
-l <x> Size of icmp echo request (default 32)
-n <x> Number of spoofed source hosts used in -P and -Cg
-o / -O <port num> Open port on spoofed host o(tcp) (O)udp
-r Don't reply to ICMP ping requests
-s <port> Source port for SYN scanner (Default: random)
-t Tag mac addresses for gobbled hosts(each will end in 4e:50)
-u <x> Closed UDP port used in OS detection (default port 1)
-v Verbose (may be used 3 times for crazy amounts of debugging info)
-V Display linked list after every update (used when gobbling a IP address)
-w Remove warnings at start of various scans

Examples
Gobbled scan single dynamically assigned host: Gobbler -P 192.168.1.1 -R n
Gobbled scan multiple src hosts: Gobbler -P 192.168.3.1 -R 21-23,445 -n 4
Non-gobbled scan: Gobbler -N 10.0.0.1 -Q 10.0.0.50-r -Q 10.0.0.51-r -R n -f
Sniffer: Gobbler -i eth0 -S -v                Arp scan: Gobbler -i fxp0 -Ag
Detect rogue DHCP server: Gobbler -D -i eth0  DHCP DoS: Gobbler -G -i fxp0
```

continues

Example 5-1 *Gobbler's Help Menu (Continued)*

```
Note: all options with a * require -Q
Note: MITM -M is in the early stages of coding
Note: When performing a DoS attack the gobbler crashes

WARNING read README.1ST before using the Gobbler
If you do not understand what you are doing, do NOT use this program!
[root@linux-p4#
```

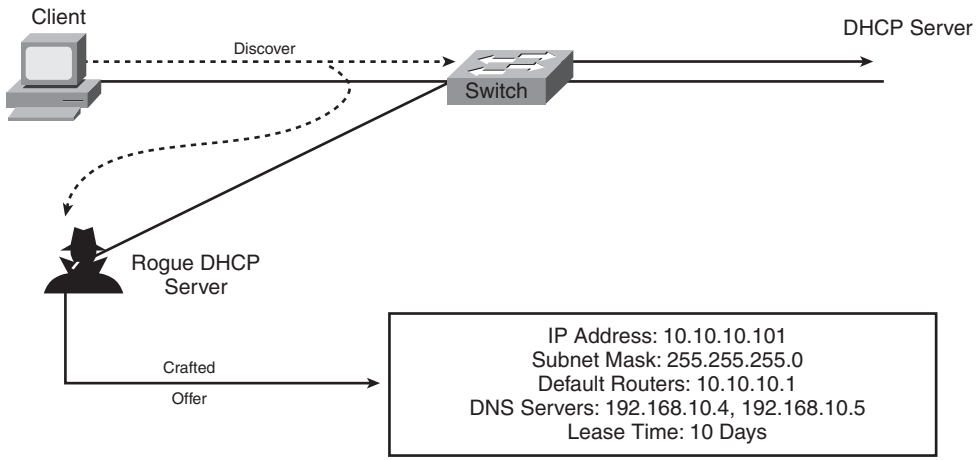
NOTE All of Example 5-1's lines are just options for Gobbler: Many of them exist because Gobbler is a powerful attack tool against DHCP.

At the end of the day, both Yersinia and Gobbler make it all too easy to attack DHCP servers.

Hijacking Traffic Using DHCP Rogue Servers

Another DHCP exploit with devastating results consists in installing a covert DHCP server on a LAN segment, as Figure 5-4 shows.

Figure 5-4 *DHCP Rogue Server*



If a rogue DHCP server is installed on the LAN, by default, it receives DHCPDISCOVER messages from clients seeking to acquire an IP address.

At this point, it is a race condition between the rogue DHCP server and the legitimate server. Because of its proximity to the clients, the rogue server probably has the upper hand. At this point, all bets are off: The rogue server can hand out options of its choosing to clients.

Which DHCP Server Will the DHCP Client Use?

When the DHCP client receives several DHCPOFFERs from different servers, which offer should it use?

In general, a DHCP client remembers the IP address it used before and, if there is an offer for this address (DHCP server being stateful offers the same IP address to the same client, if the IP address is available), the DHCP client uses this offer.

When all offers are unrelated to the client's previous IP address, the client simply uses the first offer received.

Many times, hosts obtain their domain name and domain name server IP address through DHCP. Convincing a host to use a specific (compromised) DNS server is close to the holy grail of LAN security—or insecurity, depending on your point of view!

An attacker can now attract victims to forged websites that are exact replicas of the original ones. Here, they capture credentials, account information, and other sensitive information.

Countermeasures to DHCP Exhaustion Attacks

The solution to the first type of DHCP attack (DoS by grabbing the entire available scope of addresses) depends on the hacker's knowledge of the protocol. By default, DHCP starvation tools use a random source MAC address every time they request a new IP address from the DHCP server (one new MAC per DHCPDISCOVER). Identifying this type of attack is straightforward: A sudden increase in the number of dynamically learned MAC addresses from a given LAN port is a clear indication. Under normal circumstances, there should be no more than one or two MAC addresses dynamically learned per LAN port.

When using IP telephony solutions, it's possible to see up to three addresses for a short duration. For example, when a Cisco IP phone is plugged into a port and a host (a PC or laptop) is directly connected to the phone, up to three MAC addresses can appear on the port. The phone's MAC address appears temporarily in the data VLAN so that the switch and the phone can exchange Cisco Discovery Protocol (CDP) packets.

The IP phone and switch use CDP for automatic voice and data VLAN assignment. After the VLAN negotiation is complete, the phone's MAC address appears in the voice VLAN. The host's MAC address pops up in the data VLAN.