Example of IGP Exploits

Chris Russell
chris@infosecalliance.com

Information Security Alliance, Inc.
P.O. Box 2987
Culver City, CA 90231-2987

Revision 1
October 07, 2001

ABSTRACT

This article illustrates the relative simplicity of exploiting dynamic route and network control protocols in order to redirect the flow of network traffic and open up doors to other security vulnerabilities that might have been otherwise inaccessible.

1 Introduction

Many facilities fail to consider the security ramifications of dynamic route protocols. Many of these protocols implement little or no security, and all of them can be exploited if not configured properly. Protocol hacking tools are widely available, such as Angst [1], Dsnif [5], IRPAS [8], and Nemesis [13], making it significantly easier for hackers to intercept traffic, sniff switched networks, perform man-in-the-middle attacks, hijack sessions, launch denial of service attacks, and open up doors to other security vulnerabilities that might have been otherwise inaccessible.

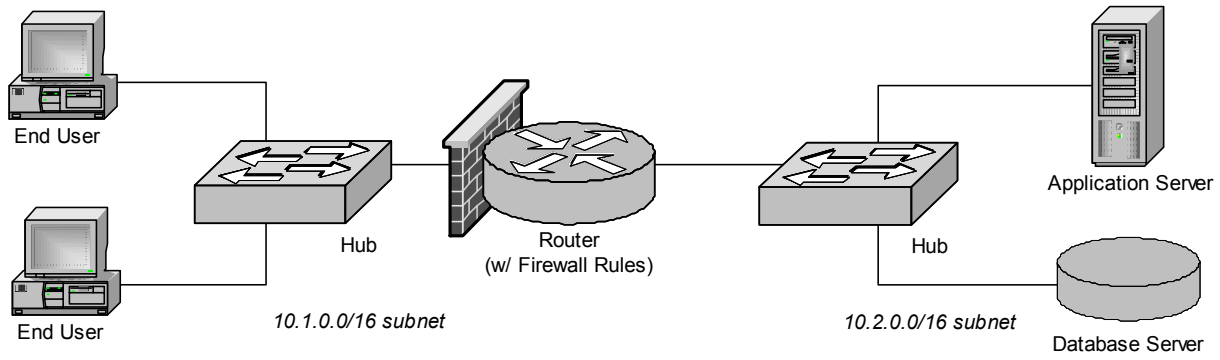
This article expands upon an attack scenario presented by FX [6] at DEFCON 9 in order to illustrate how easily dynamic routing and related protocols can be exploited.

For more detailed information on the inherent vulnerabilities of these route protocols, along with best practices recommendations for securing the network, refer to the ISA whitepaper *"Security Analysis of Dynamic Route Protocols"* [17].

2 Once Upon a Time…

…a company installed an internal firewall within their network to protect their database server. No one could access to the database without going through the firewall, not even trusted employees.

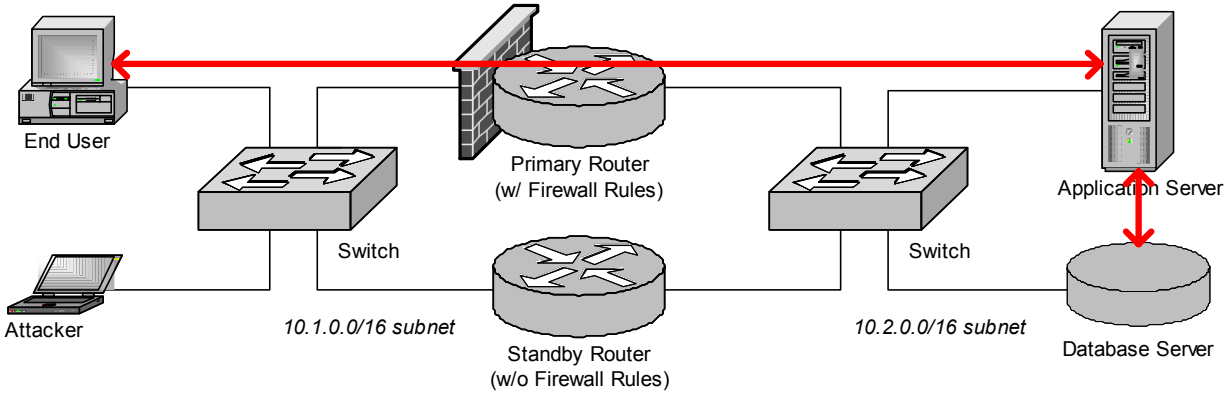
Example of IGP Exploits



The firewall was so effective at doing its job that no one ever thought to harden the database server itself. And so, the server was allowed to run r-commands, NFS, and other insecure protocols. So long as the firewall blocked ports 513 (rlogin), 514 (rexec), and 2049 (NFS), they reasoned, there was nothing to fear.

Then one day, several months later, management pondered what would happen if the router were to fail. "That router is a single point of failure!", they exclaimed, and instructed their sysadmin to install a backup router immediately.

The poor I.T. staff scrambled to install a second router. They cloned the configuration of the first router, enabled Cisco's HSRP redundant router protocol, set the primary router to be active, and the standby router to be passive. Thus, if the primary router ever fails, then HSRP automatically activates the standby router to take over all routing.



Unfortunately, they forgot to install firewall rules on the standby router! They were so focused on implementing network redundancy that they forgot about security, and no one ever noticed. So long as the primary router was active, the database was safe. But if it ever failed, then the HSRP would make the standby router active, thus exposing the database server to attack.

And then the company hired Eve. She only had one thing on her mind, to steal information from the corporate database. However, two obstacles stand in her way:

- Eve's laptop is connected to a switch, which blocks her from sniffing the network.
- The database server is protected behind the primary router/firewall.

Eve exploits three insecure protocols in order to sniff the network, intercept a user's database session, and ultimately shutdown the primary router, thus activating the standby router. Once the standby router is active, Eve is free to hack into the database server.

Specifically, the three exploits performed by Eve are:

- Exploit #1: Eve sends an ARP Flood to the switch, thus slagging its ARP cache and essentially turning it into a hub. Thus she can sniff the network. This is a similar technique to CVE candidate CAN-1999-0667, "The ARP protocol allows any host to spoof ARP replies and poison the ARP cache to conduct IP address spoofing or a denial of service." [1]
- Exploit #2: Eve broadcasts Spoofed RIP Advertisements to alter route tables and intercept user sessions through the firewall. This vulnerability is identified as CVE-1999-0111: "RIP v1 is susceptible to spoofing".
- Exploit #3: Eve sends Spoofed HSRP messages to switchover the primary firewall to a misconfigured standby firewall, thus giving him complete access into the secure network. There are currently no CVE vulnerabilities files against the HSRP protocol, although there should be.

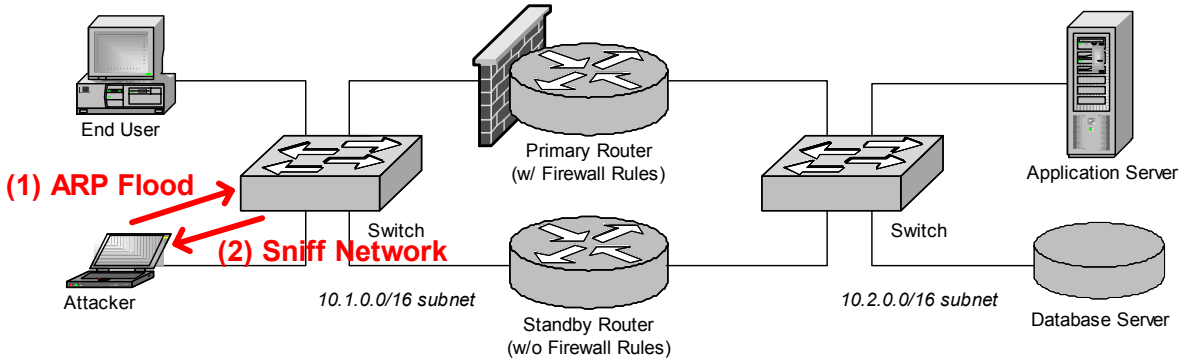
All three of these exploits are *protocol-based attacks*. In other words, they do not rely on software bugs or misconfigured systems, but rather exploit well-defined, supported features of the protocols. As such, they cannot be "fixed" by simply installing a software patch or update.

3 Exploit #1: ARP Flood

3.1 Description of the Attack

Eve's first problem is sniffing the network. The network switch automatically filters out all unicast packets that are not specifically addressed to Eve's MAC address.

Therefore, she sends a flood of ARP replies ("I am") with random IP and MAC addresses, hoping to overload the switch. The switch accepts the unsolicited replies and inserts them to its ARP cache. The cache quickly fills up with bogus MAC addresses, flushing out older addresses in FIFO (first in first out) or LRU (least recently used) order.



Since all of the valid MAC addresses for devices on the network have been flushed out of the cache, the switch no longer knows how to route the packets and therefore degrades down to behaving like a hub, forwarding copies of all packets it receives into all of its ports.

Eve can now sniff the network, searching for passwords, trust relationships, etc., so long as she continues flooding the switch with spoofed ARP replies.

[1] The general ability to sniff a network is identified as CVE candidate CAN-1990-0530, "A system is operating in "promiscuous" mode which allows it to perform packet sniffing".

Eve uses `ARP0c2` [2] to generate the continual (nonstop) ARP flood:

```
# ARP0c2 –I eth0 –f
```

3.2 Protocol Description

The ARP protocol is defined in the network layer, encapsulated directly inside the Ethernet frame. It uses Ethernet type 0x0806, specified in bytes 12:13 in the Ethernet or 802.3 header.

For TCP/IP, the ARP header is 28 bytes long:

Bytes 0:1	Hardware type (1 = Ethernet)
Bytes 2:3	Protocol type (0x0800 = TCP/IP)
Byte 4	Hardware address length (6 = 48 bit Ehternet MAC addresses)
Byte 5	Protocol address length (4 = 32 bit IP addresses)
Bytes 6:7	Opcode (2 = Reply)
Bytes 8:13	Sender Ethernet address
Bytes 14:17	Sender IP address
Bytes 18:23	Target Ethernet address
Bytes 24:27	Target IP address

For ARP flooding, the sender and target Ethernet and IP addresses are all filled with random addresses.

3.3 Signature of the Attack

A tcpdump listing of ARP0c2 produces:

```
0:0:0:0:0:0 > a3:3c:dd:de:16:51 null I (s=4,r=0,R) len=24
                0604 0002 a33c ddde 1651 772c 8667 5661
                6e4f 27c9 772c 8667
0:0:0:0:0:0 > b9:c4:29:20:be:14 null I (s=4,r=0,R) len=24
                0604 0002 b9c4 2920 be14 8726 aef9 21e6
                675a 560a 8726 aef9
0:0:0:0:0:0 > 95:34:e8:ab:85:5f null I (s=4,r=0,R) len=24
                0604 0002 9534 e8ab 855f d60b c52d 6b34
                7b92 fd34 d60b c52d
0:0:0:0:0:0 > 57:27:54:16:3b:da null I (s=4,r=0,R) len=24
                0604 0002 5727 5416 3bda 3be8 d35b cf3b
                b426 454a 3be8 d35b
0:0:0:0:0:0 > 59:2d:f4:dd:8b:ca null I (s=4,r=0,R) len=24
                0604 0002 592d f4dd 8bca e851 f654 8571
                e682 a53d e851 f654
```

This is a very unusual signature for ARP messages. Close inspection of the Ethernet frame's data payload (listed in hex) reveals it to be a properly formed ARP reply, but from the output of tcpdump, this is obviously a crafted datagram. (To be honest, I've never seen tcpdump output like this before!)

Even though ARP spoofs generated by `ARP0c2` can be easily identified, it would be simple to modify it to generate normal looking ARP datagrams (albeit containing random IP and MAC addresses).

Other indicators include unusually heavy network traffic on the network and bandwidth degradation, since the switch is now operating more like a hub.

3.4 How to Protect Against It

Some switches allow static ARP tables to be defined. The MAC addresses of all known devices would be preloaded into the switch; however, this is exceptionally impractical.

It is possible that a layer 3 switch may be resistant to this specific attack, since it switches based on network addresses (IP addresses) rather than link addresses (Ethernet MAC addresses). However, it may be similarly vulnerable to layer 3 attacks, such as IP flooding? An intelligent switch with subnet information and anti-spoof detection could theoretically detect this and protect against it.

Finally, installing a network intrusion detection system (NIDS) probe can detect this attack, either by statistical analysis or more simply by detecting ARP replies containing erroneous network addresses (in this case, networks other than 10.1.0.0/16).

3.5 Variant #1

Instead of generating ARP replies, sending a flood of IP datagrams (either TCP or UDP) with random IP and MAC addresses accomplishes the same effect as an ARP flood. This may be accomplished using `macof` [5]:

```
# macof –r –i eth0 –n 100000
```

which produces the following tcpdump listing:

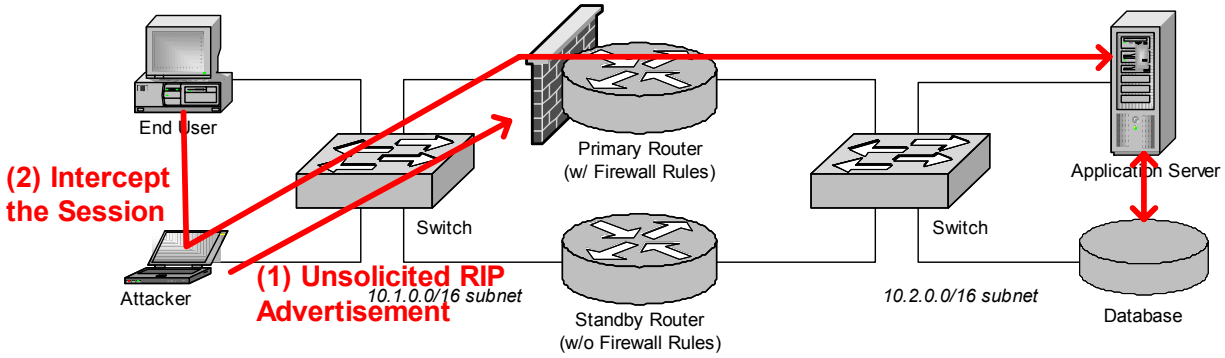
```
174.183.209.156.26355 > 133.36.213.136.50073: . 0:0(0) win 65535 (DF) [tos 0x10]  (ttl 64, id 0)
165.113.175.249.25334 > 67.134.251.70.20810: . 0:0(0) win 65535 (DF) [tos 0x10]  (ttl 64, id 0)
155.161.109.154.1731 > 136.39.6.168.2007: . 0:0(0) win 65535 (DF) [tos 0x10]  (ttl 64, id 0)
144.181.52.123.54255 > 103.147.17.67.11487: . 0:0(0) win 65535 (DF) [tos 0x10]  (ttl 64, id 0)
4.111.234.237.57042 > 35.13.219.53.34409: . 0:0(0) win 65535 (DF) [tos 0x10]  (ttl 64, id 0)
```

4 Exploit #2: Spoofed RIP Advertisements

4.1 Description of the Attack

After sniffing the network, Eve decides to intercept all traffic from the end user (at 10.1.7.11) to the application server (at 10.2.0.5). She has a few methods available to choose from, but selects RIP since the hosts are configured for RIP v1 and it is unlikely to setoff any alarms.

Eve sends the end user a RIP packet specifying a new route directly to address 10.2.0.5 with a metric (hop count) of 1. The end user's computer gobbles up the unsolicited route advertisement and proceeds to route all future packets to the application server through Eve as a gateway. Eve may then perform man-in-the-middle attacks by reading and potentially altering the packets before forwarding them on to their final destination.



In this example, Eve is only intercepting the traffic in one direction. In similar fashion, she may also setup routes with the primary router to intercept and route traffic back in the other direction.

Nemesis [13] is used to issue a RIP v1 command to route all traffic from host 10.1.7.11 to the server 10.2.0.5 through the Eve's computer at 10.1.123.123:

```
#./nemesis-rip -c 2 -V 1 -a 1 -i 10.2.0.5 -m 1 -V 1 -S 10.1.123.123 -D 10.1.7.11
```

Or, to generate a RIP v2 route with proper netmask information, use:

```
#./nemesis-rip -c 2 -V 2 -a 1 -i 10.2.0.5 –k 0xffffffff-m 1 -V 1 -S 10.1.123.123 -D 10.1.7.11
```

Eve must also configure her computer to route the packets onward. Otherwise, this would result in an inadvertent routing black hole and the packets never reach their destination.

4.2 Protocol Description

RIP is a very simple protocol. It uses UDP port 520, and the header is:

Bytes 0:1	Command (2 = Response, or Advertise Routes)
Bytes 2:3	Not used (set to 0x0000)
Bytes 4+	RIP v1 entry table

Each entry in the entry table is specified as follows:

Bytes 0:2	Address family
Bytes 3:4	Route tag (set to 0x0000 for RIP v1)
Bytes 5:8	Target IP address
Bytes 9:12	Target subnet mask (set to 0x00000000 for RIP v1)
Bytes 13:16	Next hop IP address (set to 0x00000000 for RIP v1)
Bytes 17:20	Metric (hop count)

The Next hop address specifies the router's IP address. If the next hop value is 0 (which is always the case for RIP v1), then the router's address defaults to the sender's IP address of the RIP datagram.

4.3 Signature of the Attack

A tcpdump of the RIP v1 command produces:

```
10.1.123.123 > 10.1.7.11.route: rip-resp 1: [family 1: 0000 0500 020a 0000 0000 0000 0000 0000
0001](1) (DF) [tos 0x18] (ttl 254, id 122)
```

A tcpdump of the RIP v2 command produces:

```
10.1.123.123 > 10.1.7.11.route: rip-resp 1: [family 1: 0000 0500 020a ffff ffff 0000 0000 0000
0001](1) (DF) [tos 0x18] (ttl 254, id 123)
```

These are all normally formatted RIP packets. Perhaps the best way to detect these as illegal (spoofed) RIP packets is to filter for any RIP responses (but not requests) originating from unexpected IP addresses, such as any address that is not from a known router on the network segment.

However, this will not catch RIP v2 packets that spoof the source IP address from a known router but uses the Next hop value to specify the Eve as the router. Filtering these packets requires a significantly more detailed analysis into the application layer of the datagram, which is beyond the capabilities of tcpdump.

4.4 How to Protect Against It

To protect against this exploit:

- Disable route discovery protocols on host machines. Use static route tables instead.

- Only use cryptographically secure IGP and EGP protocols on routers, such as OSPF or RIP v2 with MD5 authentication.
- Disable IRDP, if possible, to prevent similar exploits using IRDP route insertion.

4.5 Variant #1

Instead of exploiting layer 3 route protocols, the network traffic can be intercepted using layer 2 ARP cache poisoning, similar to the technique used by hunt for TCP session hijacking across a switch.

However, this technique is prone to detection, since modern operating systems tend to generate spoofed packet alerts whenever they detect ARP replies advertising their own IP address.

4.6 Variant #2

Instead of using an IGP protocol like RIP, it is possible to manipulate the route tables using IRDP route insertion. This technique enables Eve to selectively intercept data between two hosts (or between a host and a router) without having to intercept (and forward) all of the traffic from the router.

This approach is superior to ARP spoofing since it is generally more stealthy and less likely to setoff and alarms. However, not all hosts are configured to accept IRDP.

CVE identifies a special case of this exploit (when using DHCP) as CVE-1999-0875, "DHCP clients with ICMP Router Discovery Protocol (IRDP) enabled allow remote attackers to modify their default routes".

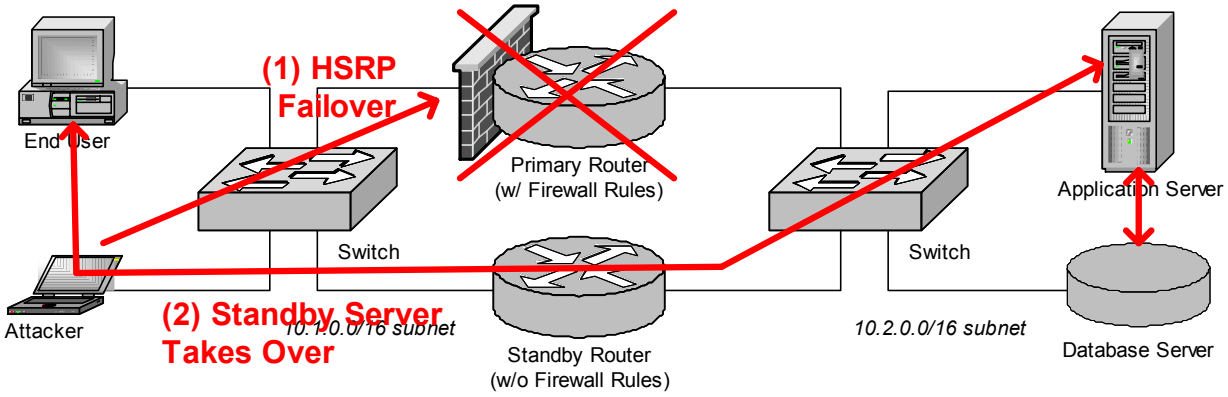
4.7 Variant #3

If Eve is connected to a link between neighboring exterior routers, then this technique can be applied to EGP protocols (such as BGP) to intercept traffic between autonomous systems, thus enabling man-in-the-middle attacks across the Internet!

5 Exploit #3: Spoofed HSRP

5.1 Description of the Attack

Eve scans the primary router for vulnerabilities, but doesn't find any. At the same time, she detects the hot standby router using HSRP and discovers no firewall rules are enabled. At last the break she was looking for!



Eve sends a spoofed HSRP message to the primary router (at 10.1.0.1) to essentially take it offline. Once offline, the standby router becomes the new primary router and automatically starts to route all traffic

between the two networks. There is no longer any firewall protection for the data center network, and Eve is free to use traditional tools and techniques to hack into the systems.

The router is put into standby mode using IRPAS [8]. If the primary router's address is 10.1.0.1, then the following would put it into standby mode and hence force a failover to the backup router:

```
# while (true)
do
        ./hsrp -d 10.1.0.1 -v 10.1.0.2 -a cisco -g 1 -i eth0
        sleep 3
done
```

5.2 Signature of the Attack

A tcpdump listing of hsrp produces:

```
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 80)
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 80)
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 82)
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 82)
```

Or, using 'tcpdump –x' to include the packet data (in hex) produces:

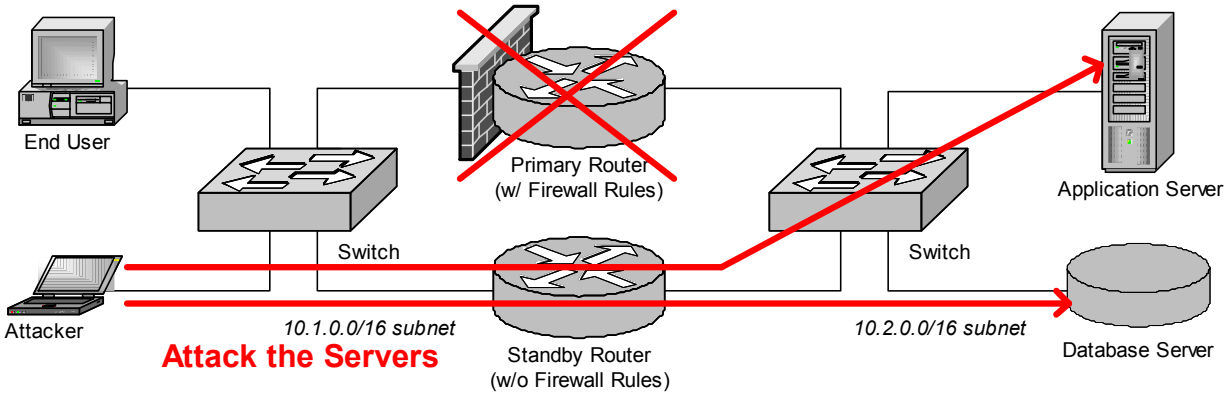
```
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 80)
                        4500 0030 005c 0000 8011 2598 0a01 0001
                        0a01 0001 07c1 07c1 001c 0000 0001 1003
                        ffff 0100 6369 7363 6f00 0000 0a01 0002
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 80)
                        4500 0030 005c 0000 8011 2598 0a01 0001
                        0a01 0001 07c1 07c1 001c 0000 0001 1003
                        ffff 0100 6369 7363 6f00 0000 0a01 0002
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 82)
                        4500 0030 005e 0000 8011 2596 0a01 0001
                        0a01 0001 07c1 07c1 001c 0000 0000 1003
                        ffff 0100 6369 7363 6f00 0000 0a01 0002
10.1.0.1.1985 > 10.1.0.1.1985: udp 20 (ttl 128, id 82)
                        4500 0030 005e 0000 8011 2596 0a01 0001
                        0a01 0001 07c1 07c1 001c 0000 0000 1003
                        ffff 0100 6369 7363 6f00 0000 0a01 0002
```

5.3 How to Protect Against It

Again, use authenticated protocols whenever possible. Since HSRP is inherently insecure, use VRRP with MD5 authentication if it is available.

6 The End

Now that network traffic is redirected to go through the standby firewall, Eve is free to attack the database server directly. By spoofing r-commands and NFS, she is able to gain full access the database server and steal their data.



And no one will suspect a thing. After the unexpected failover, they may even celebrate the success of their redundant router, unaware that it is being used at that moment to hack the database.

7 References

[1] Angst (an active sniffer with integrated ARP flooding). Vers. 0.4b. 7 Oct 2001 <http://angst.sourceforge.net>.

[2] ARP0c. Vers. 2.3. 7 Oct. 2001 <http://www.phenoelit.de/arpoc>.

[3] Deering, S. RFC 1256, ICMP Router Discovery Messages. 7 Oct. 2001 <http://www.ietf.org/rfc/rfc1256.txt>.

[4] Dittrich, Dave. Some TCP/IP Vulnerabilities. 7 Oct. 2001 <http://www.staff.washington.edu/dittrich/talks/agora/index.html>.

[5] Dsnif (arpspoof and macof). Vers. 2.3. 7 Oct. 2001 <http://www.monkey.org/~dugsong/dsniff>.

[6] FX, and Phenoelit. Routing & Tunneling Protocol Attacks. Presentation at DEFCON 9. 7 Oct. 2001 <http://www.phenoelit.de/stuff/routing.pdf>.

[7] Hedrick, C. RFC 1058, Routing Information Protocol. 7 Oct. 2001 <http://www.ietf.org/rfc/rfc1058.txt>.

[8] Internet Routing Protocol Attack Suite (IRPAS). Vers. 0.8. 7 Oct. 2001 <http://www.phenoelit.de/irpas/docu.html>.

[9] Knight, S., et. al. RFC 2338, Virtual Router Redundancy Protocol. 7 Oct. 2001 <http://www.ietf.org/rfc/rfc2338.txt>.

[10] Li, T., B. Cole, P. Morton, and D. Li. RFC 2281, Cisco Hot Standby Router Protocol (HSRP). 7 Oct. 2001 <http://www.ietf.org/rfc/rfc2281.txt>.

[11] Malkin, G. RFC 2453, RIP Version 2. 7 Oct. 2001 <http://www.ietf.org/rfc/rfc2453.txt>.

[12] Mirte Corporation. Common Vulnerabilities and Exposures Database. 7 Oct. 2001 <http://www.cve.mitre.org/cve>.

[13] Nemesis Packet Injection tool suite. Vers. 1.32. 7 Oct 2001 <http://jeff.wwti.com/nemesis>.

[14] Phenoelit. Internet Routing Protocol Attack Suite (IRPAS) Documenation. 7 Oct. 2001 <http://www.phenoelit.de/irpas/docu.html>.

[15] Plummer, David C. RFC 826, An Ethernet Address Resolution Protocol. 7 Oct. 2001 <http://www.ietf.org/rfc/rfc0826.txt>.

[16] Postel, J. RFC 792, Internet Control Message Protocol. 7 Oct. 2001 <http://www.ietf.org/rfc/rfc0792.txt>.

[17] Russell, Chris. Security Analysis of Dynamic Route Protocols. 1 Feb. 2002 <http://www.infosecalliance.com/resources/whitepapers/dynamic-route-protocols.pdf>.

[18] Stevens, W. Richard. TCP/IP Illustrated, Volume 1 – The Protocols. Massachusetts: Addison-Wesley, 1999.

[19] Wilson, Curt. Protecting Network Infrastructure at the Protocol Level. 7 Oct. 2001 <http://www.netw3.com/documents/Protecting_Network_Infrastructure.htm>.