IPv6 / ICMPv6
Covert Channels
R.P. Murphy, CISSP, CEH
Overview

- IPv4
- IPv6
- RFC’s and IPv6/ICMPv6 fields
- Definition of a Covert Channel
- Assumptions
- Test Network
- v00d00N3t
  - Development
  - Capabilities
  - Testing
  - Results
- Questions
IPv4

- IPv4
  - NAT
- Limited address space
  - ~ 2009-2016
- Push to move to IPv6
  - DoD mandated by 2008
- Similar covert channel capabilities
IPv6 (IPng)

• Proposed standard NOV 17, 1994
• IPv6 is the answer to IPv4
• Huge address space
• Security by numbers
• Deployment Issues
  – Legacy equipment
  – Software modifications
  – Each device is now pingable
RFC IPv6 / ICMPv6

- RFC2119 March 1997
  - Key words for use in RFCs to Indicate Requirement Levels
    - **MUST** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
    - **Security Considerations** These terms are frequently used to specify behavior with security implications. The effects on security of not implementing a MUST or SHOULD, or doing something the specification says MUST NOT or SHOULD NOT be done may be very subtle. Document authors should take the time to elaborate the security implications of not following recommendations or requirements as most implementers will not have had the benefit of the experience and discussion that produced the specification.
RFC IPv6 / ICMPv6

• RFC2460 December 1998
  – IPv6 Specification
    • Traffic Class bits in a received packet **MUST NOT** be assumed as the same value sent by the source

• RFC3697 March 2004
  – IPv6 Flow Label Specification
    • The Flow Label value set by the source **MUST** be delivered unchanged to the destination node(s).
RFC IPv6 / ICMPv6

- RFC4443 March 2006
  - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
  - ICMPv6 (ICMP for IPv6) is used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions, such as diagnostics (ICMPv6 "ping"). ICMPv6 is an integral part of IPv6, and the base protocol (all the messages and behavior required by this specification) MUST be fully implemented by every IPv6 node.
Covert Channel Defined

• A covert channel is a mechanism that can be used to transfer information from one user of a system to another using means not intended for this purpose by the system developers.


• A covert channel is any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy.

Assumptions

- ICMPv6 traffic will be allowed (RFC4443)
- Control at both ends
- Take advantage of Dual-Stack to use Tunnel Brokers for test-bed
- Still maturing IPv6 protection technology (FW, IDS, IPS)
Test Networks

• Two networks designed and tested
  – Reflash SOHO Linksys
    • IPv6 over IPv4 Tunneling
  – ‘Slick’ IPv6
    • Controlled
Test Networks

- Linksys WRT54g
  - Firmware OpenWRT
  - Added IPv6 packages
  - IPv6 network in the home
  - 6 over 4 tunneling
  - Tunnel Broker
Automatic Configuration - DHCP

- Router Name: WRT54G
- Host Name:
- Domain Name:
- MTU: Auto
- Size: 1500

Local IP Address: 192.168.1.1
Subnet Mask: 255.255.255.0

DHCP Server: Enable
Starting IP Address: 192.168.1.100
Maximum Number of DHCP Users: 50
Client Lease Time: 0 minutes (0 means one day)
Welcome to v00d00N3t

White Russian (RC4)

* 2 oz Vodka
* 1 oz Kahlua over ice, then float the cream or
* 1/2 oz cream, milk on the top.

Current Date/Time Tue Jul 4 00:58:39 PDT 2006
SOHO IPv6 Network
RADV: 2001:618:400:5696::/64

IPv6 Traffic

Dynamically Assigned IPv4 Address
OpenWRT RC4

IPv4 Internet

BTExact Technologies
United Kingdom (UK)
IPv4: 213.121.24.85

6 Bone

Linksys WRT54G v1: 2001:470:1f00:ffff::12e7/128
Dynamically Assigned IPv4 Address
OpenWRT RC4

IPv6 Traffic

SOHO IPv6 Network
RADV: 2001:470:1f00:2519::/64
Linux FC4: 2001:470:1f00:2519:240:45ff:fe18:9aa3

IPv4 Traffic
IPv6 Traffic
6 over 4 Tunnel
IPv6

Welcome to the IPv6 Information Page!

You are using IPv6 from 2001:618:400:3b3b:20c:29ff:fe55:62ca
Test Networks

• ‘Slick’ IPv6 Network
  – Linux Router
    • Fedora Core 4
    • Zebra w/BGPv6
    • Router Advertisements (/etc/radv.conf)
  – Linux Clients
    • Fedora Core 4

• Windows Router
  – Server 2003 Enterprise
  – RIPv6
  – Router Advertisements
    • netsh interface ipv6 > set interface *
Test Networks

• Cisco Routers
  – 2650 (3)
    • C2600-js-mz.122_8_T5.STB.5
  – 2621XM/2610
    • C2600-ik9o3s3-mz.123-15b.bin

• IRP → RIPv6
• ERP → BGPv6
v00d00N3t Development

- It’s a PoC
- Written in C
- Creates the entire packet starting with Ethernet Layer
- Designed to subvert casual local traffic analysis
- Manipulate the IPv6 and ICMPv6 layers
- Does not cater to IPv4 AND IPv6
v00d00N3t Development

- Uses standard C libraries not USAGI
- Development system was updated weekly (kernel included)
- Test systems were updated periodically
- Test runs on FC4 and FC5
void sock_init()
{
    sock = socket(PF_PACKET, SOCK_RAW, htons(ETH_P_ALL));
}
Random MAC Address

```c
void rnd_MAC()
{
    read(dev_urandom, rand_mac, 6);
    rand_mac[0] = 0;
    snprintf(secondhalf, 64, "2%2.2x:%2.2xff:fe%2.2x:%2.2x%2.2x", rand_mac[1], rand_mac[2], rand_mac[3], rand_mac[4], rand_mac[5]);
}
```
void rnd_IPv6()
{
    char full[INET6_ADDRSTRLEN];
    char half[INET6_ADDRSTRLEN];
    char Ohalf[INET6_ADDRSTRLEN];
    inet_pton(AF_INET6, myaddress, full, sizeof(full));
    memcpy(half, full, 8);
    memset(half + 8, 0, sizeof(half));
    inet_ntop(AF_INET6, half, Ohalf, sizeof(Ohalf));
    int x = strlen(Ohalf);
    memcpy(Ohalf + (x - 1), secondhalf, sizeof(half));
    inet_pton(AF_INET6, Ohalf, full, sizeof(full));
    inet_ntop(AF_INET6, full, my_rnd_ip_addr, sizeof(my_rnd_ip_addr));
}
memset (packet, 0, 4096);
eth = (struct ether_header*) packet;
ip6 = (struct ip6_hdr*)(eth + 1);
icmp6 = (struct icmp6_hdr*)(ip6 + 1);
memcpy(eth->ether_dhost, gate_mac, ETH_ALEN);
memcpy(eth->ether_shost, rand_mac, ETH_ALEN);
eth->ether_type = htons(ETHERTYPE_IPV6);
inet_pton(AF_INET6, my_rnd_ip_addr, IPv6SRCADDR, sizeof(IPv6SRCADDR));
memcpy(&ip6->ip6_src, IPv6SRCADDR, sizeof(IPv6SRCADDR));
int send_packet(int sizer)
{
    close(sock);
    sock_init();
    if (sendto(sock, packet, sizeof(struct ether_header) +
            sizeof(struct ip6_hdr) + sizeof(struct icmp6_hdr) + sizer, 0,
            (struct sockaddr *)&sa, sizeof(sa)) < 0)
    {
        perror("There was a problem sending your packet");
        exit(-1);
    }
    sizer = 0;
}
v00d00N3t Capabilities

- Flags, Flags, and more Flags...
  - d → Destination IPv6 address
  - r → Receive mode
  - k → Keyboard entry mode
  - f → Send a file
  - i → Interface identification
  - g → Gateway MAC address
  - b → Throttle by bytes (per packet)
  - t → Throttle by time (1 second intervals)
  - x → 4 digit PIN for send and receive
  - h → Help menu
v00d00N3t Capabilities

- Send data (keyboard or text file)
- Obscure data (ROT-13)
- Random source MAC and IPv6 address
- Determine gateway MAC address
- Throttle by bytes and/or time
- Receive data
v00d00N3t Capabilities

- Requires 4 digit PIN for sender and receiver, allowing multiple streams
- ICMPv6 ID tells receiver how many bytes out of payload to read
- ICMPv6 SEQ tells receiver if it should read the packet or not
v00d00N3t Testing

- Validate that the packets would survive on a ‘slick’ 6 network
- Validate that the packets would survive in the ‘wild’, basically uncontrolled environment
- Still not tested for survivability in an IPv6 production environment with IDS/IPS/FW etc…
```bash
[root@blackmagic -]# ifconfig
eth0   Link encap:Ethernet  HWaddr 00:0C:29:55:62:CA
inet addr:192.168.1.252  Bcast:192.168.1.255  Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe55:62ca/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:4574  errors:0  dropped:0  overruns:0  frame:0
TX packets:3580  errors:0  dropped:0  overruns:0  carrier:0
collisions:0  txqueuelen:1000
RX bytes:548238 (535.3 KiB) TX bytes:558190 (545.1 KiB)
Interrupt:17  Base address:0x1080

lo    Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
```

```bash
1  2001:618:400:3b3b::1 (2001:618:400:3b3b::1)  0.000 ms  0.426 ms  0.436 ms
3  uk6x-core-hopper-g0-2.ipv6.btexact.com (2001:618:1::1)  157.266 ms  157.557 ms  158.732 ms
4  v6-tunnel-ignite-de.ipv6.btexact.com (2001:7f8:2:8015::3)  186.005 ms  185.122 ms  185.360 ms
6  ge-0-0-0-100-bcr2.fra.cw.net (2001:5000:0:13::2)  208.716 ms  208.414 ms  ge-1-0-0-200-bcr2.fra.cw.net (2001:5000:0:14::)
7  211.476 ms
8  so-1-2-0-dcr2.fra.cw.net (2001:5000:0:f::1)  247.625 ms  211.164 ms  207.932 ms
9  so-4-0-0-dcr1.amd.cw.net (2001:5000:0:e::2)  208.582 ms  207.719 ms  211.347 ms
10  so-3-0-0-zcr1.amd.cw.net (2001:5000:0:12::2)  212.102 ms  219.530 ms  217.735 ms
11  nl-ams04a-rel-fe-0-0.ipv6.aorta.net (2001:7f8:1::500:6830:1)  213.759 ms  214.829 ms  210.101 ms
12  nl-ams06d-rel-t-2.ipv6.aorta.net (2001:730:1::c)  213.005 ms  214.406 ms  213.400 ms
13  hurrican.net-gwl.ipv6.aorta.net (2001:730:1::2f)  322.959 ms  323.803 ms  328.081 ms
14  2001:470:1ff:4:2e0:feff:fe07::0000 (2001:470:1ff:4:2e0:feff:fe07::0000)  323.108 ms  322.725 ms  323.923 ms
```
Options:

Required
- d Destination IPv6 address
- r Receive mode
- i Interface to communicate from
- k Send text via keyboard
- f Location of the file you want to send
- x 4 digit PIN required for packet sending/receiving

Optional
- g Gateway MAC address
- b Amount of characters to send per packet
- t Amount of delay (in seconds) between sending packets
- h This menu

Example: Send a file

# v00d00N3t -d 2006:3820:40:2a03:d843:55dc:3944:d3d2 -i eth0 -x 1234 -f /root/send.txt

Example: Send a file then remain in console mode

# v00d00N3t -d 2006:3820:40:2a03:d843:55dc:3944:d3d2 -i eth0 -g 00:12:fd:34:69:ff -x 1234 -f /root/send.txt -k

Example: Send text via keyboard

# v00d00N3t -d 2006:3820:40:2a03:d843:55dc:3944:d3d2 -i eth0 -x 1234 -k

Example: Receive incoming text

# v00d00N3t -r -i eth0 -x 1234

Quitting, you need to select an interface from the list below that has a routable IPv6 address.

Interface: lo
Address: 

Interface: eth0

Interface: eth0
Address: 

[root@blackmagic ~]#
[root@blackmagic ~]# ./v00d00N3t -d 2001:470:1f00:2658:240:45ff:fe18:9aa3 -x 1234 -i eth0 -k
Your address is: 2001:618:400:3b5b:20c:29ff:fe55:62ca
Attempting to find the GW MAC..
Found GW MAC: 0:14:bf:b4:db:4
You are in console mode, type your message and press return to send.

[root@blackmagic ~]# ./v00d00N3t -r -i eth0 -x 4321
Your address is: 2001:618:400:3b5b:20c:29ff:fe55:62ca
Attempting to find the GW MAC..
Found GW MAC: 0:14:bf:b4:db:4
Receive Mode
[root@blackmagic ~]# ./v00d00N3t -d 2001:470:1f00:2658:240:45ff:fe18:9aa3 -x 1234 -i eth0 -k
Your address is: 2001:618:400:5b5b:20c:29ff:fe55:62ca
Attempting to find the GW MAC..
Found GW MAC: 0:14:bf:b4:db:4
You are in console mode, type your message and press return to send.
this is the first test

[root@blackmagic ~]# ./v00d00N3t -r -i eth0 -x 4321
Your address is: 2001:618:400:5b5b:20c:29ff:fe55:62ca
Attempting to find the GW MAC..
Found GW MAC: 0:14:bf:b4:db:4
Receive Mode
This is from Steve
Results

- The packets survived each test run
- Sent ‘Echo Reply’ messages with a payload of 1440 bytes in payload with no problem
- Larger files were broken up by the host and sent in increments
- Sent packets with a throttle set for 1 byte per 5 minutes
- Used 2 different Tunnel Brokers for testing
Frame 12824 (94 bytes on wire, 94 bytes captured)

  - Source: Cisco-Li_b4:db:04 (00:14:bf:b4:db:04)
  - Type: IPv6 (0x86dd)

Internet Protocol Version 6

- Version: 6
- Traffic class: 0x96
- Flowlabel: 0x94141
- Payload length: 40
- Next header: ICMPv6 (0x3a)
- Hop limit: 247
- Source address: 2001:470:1f00:2658:276:12ff:fe0b:9100

Internet Control Message Protocol v6

- Type: 129 (Echo reply)
- Code: 105
- Checksum: 0xd16b [correct]
- ID: 0xff54
- Sequence: 0xe198
- Data (32 bytes)

```
0000  00 0c 29 55 62 ca 00 14 bf b4 db 04 86 dd 69 69 ..)ub... .......f
0010  41 41 00 28 3a f7 20 01 04 70 1f 00 26 58 02 76  AA.(.:.:p..&X.v
0020  12 ff fe 0b 91 00 02 01 06 18 04 00 3b 3b 02 0c  
0030  29 ff fe 55 62 ca 81 69 d1 6b ff 54 e1 98 79 20  
0040  73 62 65 20 76 67 66 20 6e 6f 76 74 67 66 6c 0a  sbe vgf novyvgl.
0050  67 62 20 65 72 7a 68 79 67 76 63 gb ernq zhygvc
```
References

- http://www.rfc-editor.org
- http://openwrt.org/
- http://wiki.openwrt.org/IPv6_howto
- http://www.he.net/index.html