the hacker's choice

presents:

Attacking the IPv6 Protocol Suite

van Hauser, THC

vh@thc.org

http://www.thc.org

You might know me from ...

Amap

rwwwshell

THC-Scan

Hydra

Login hacker

Keyfinder

Parasite

Covering your tracks

Hackers go corporate

Manipulate data

Anonymizing Unix Systems

Placing backdoors through firewalls

Secure Delete

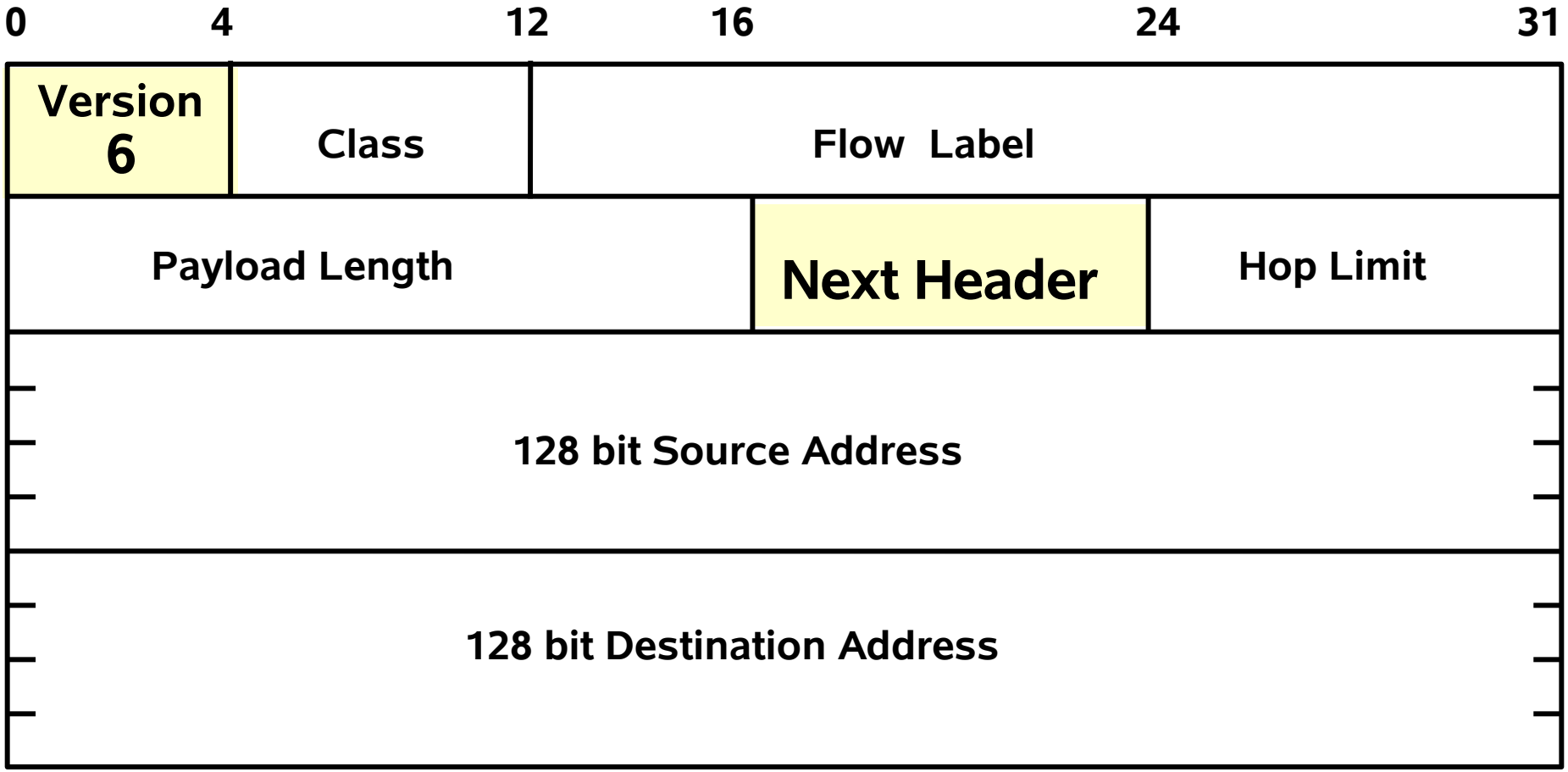
Contents

1. Very fast and short Introduction to IPv6
2. The all new THC IPV6 Attack Suite
3. Security relevant changes in IPv4<>IPv6 and Security Vulnerabilities in IPv6
4. Implementation Vulnerabilities in IPv6 so far
5. New Research & Future

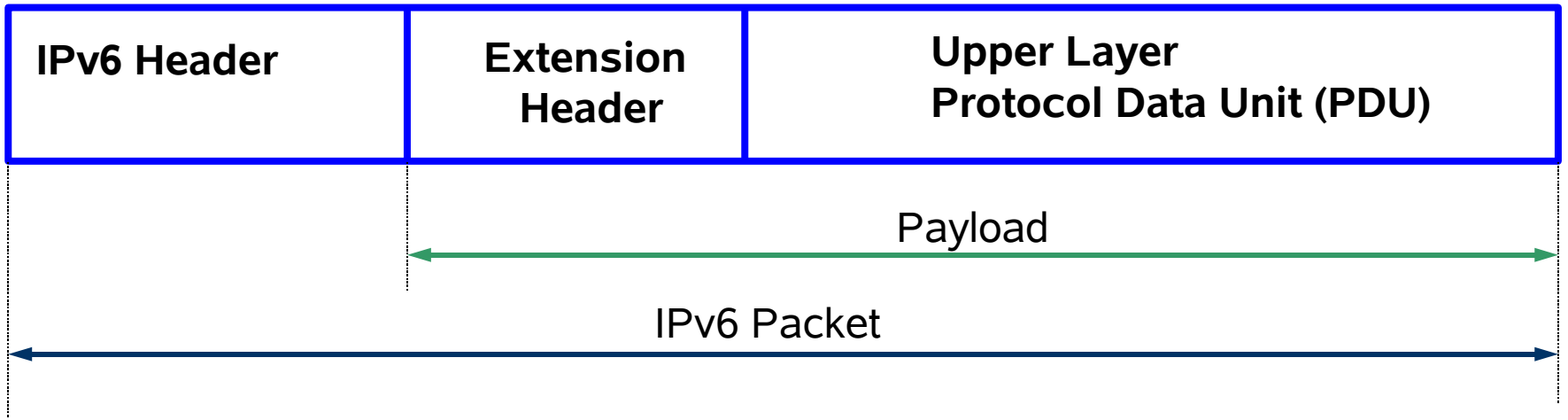
Very short and fast Introduction to IPv6

- Goals of IPv6:
  - Enough IP addresses for the next decades
    - 2^{128}=340.282.366.920.938.463.463.374.607.431.768.211.456
  - Autoconfiguration of IP addresses and networking
  - Hierarchical address structure
    - Reduces operational costs
  - Integrated security features

IPv6 Header Structure



IPv6 Layer Structure



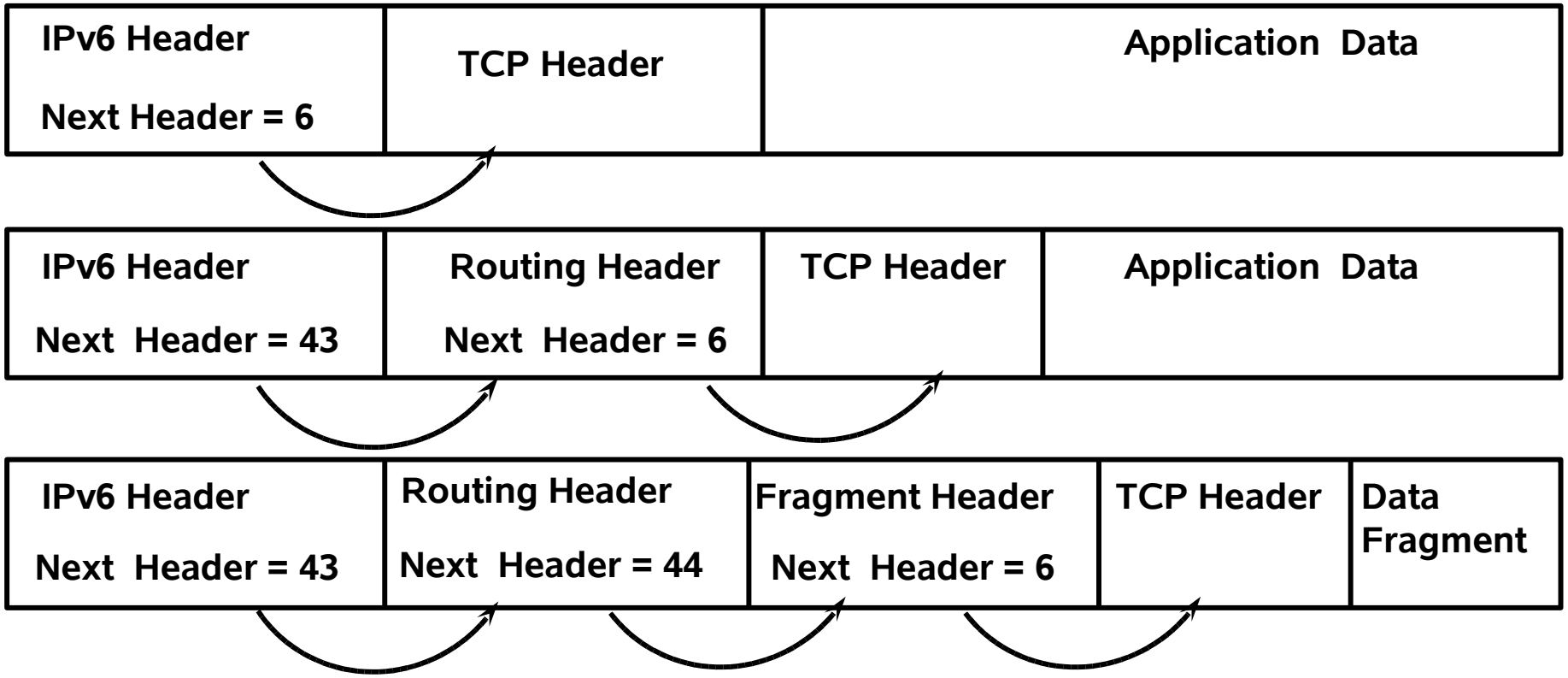
IPv6 Header ≡ **40 Bytes**

Upper Layer PDU ≤ 65535 Bytes

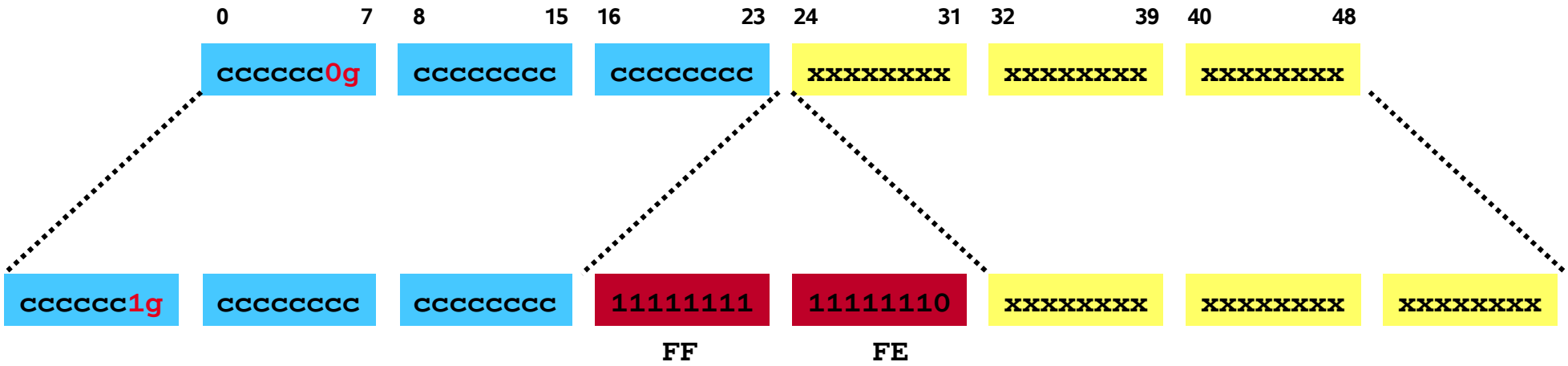
Upper Layer PDU > 65535 Bytes = Jumbo Payload

IPv6 Header Structure

Examples for Extension Headers: Hop-by-Hop = 0; UDP = 17; Encapsulated Header = 41; RSVP = 46; IPSEC (Encapsulating Security Payload = 50; Authentication Header = 51;) ICMPv6 = 58; No Next Header = 59; Destination Options = 60; OSPFv3 = 98



IPv6 Interface Identifier (EUI-64 Format) Mapping



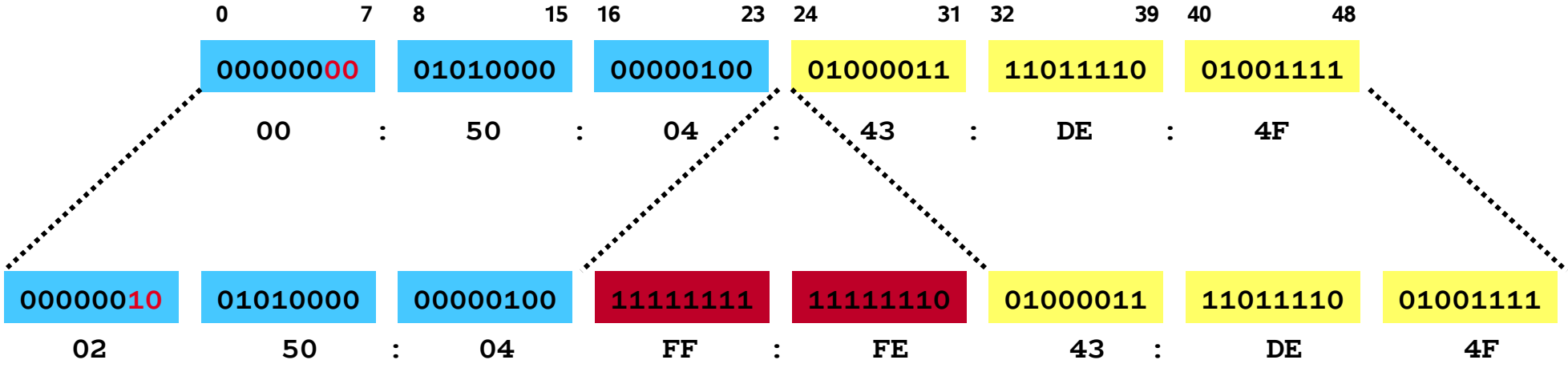
IPv6 Interface Identifier im EUI-64 Format

EUI: Extended Unique Identifier

c = company id
x = extension identifier
g = Individual/Group (G): 0 - unicast 1 - multicast

Example

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:04:43:DE:4F
          inet addr:10.2.1.1  Bcast:10.2.1.255  Mask:255.255.255.0
          inet6 addr: 3ffe:ffff:100:f101:250:4ff:fe43:de4f/64 Scope:Global
          inet6 addr: fe80::250:4ff:fe43:de4f/64 Scope:Link
          ...
```



Blackhat usage of IPv6 today

Backdoor deployment (history now)

- Enable IPv6 (6to4)
- Run Backdoor on IPv6 address
- Not detected by port scanning
- Hard to analyze if backdoor traffic is detected

Inter-Communication

- Establishing of IPv6 interconnections (via 6to4, native, ...) for warez exchange, IRC and bouncing

Availability of Hacker Tools so far …

The following Hacker tools exist:

- Port Scanning: nmap, halfscan6, …
- Port Bouncers: relay6, 6tunnel, nt6tunnel, asybo, …
- Denial-of-Service (connection flooding): 6tunneldos
- Packet fun: isic6, libnet (partially implemented only)

No IPv6 specific attack tools exist so far!

This will change when IPv6 deployment is wider

… but you do not want to wait, right?

The THC IPV6 Attack Suite

- THC has developed an easy-to-use IPv6 packet factory library
- Numerous IPv6 protocol exploits tools can be coded in just 5-10 lines
- Lots of powerful protocol exploits already included

- Current code state:
  - Linux 2.6.x only
  - Little Endian, 32-Bit
  - Ethernet and RAW mode

The THC IPV6 Attack Suite – The Tools

- Alive6
  - Find all local IPv6 systems, checks for aliveness of remote systems
- PARSITE6
  - ICMP Neighbor Spoofer for Man-In-The-Middle attacks
- REDIR6
  - Redirect traffic to your system on a LAN
- FAKE_ROUTER6
  - Fake a router, implant routes, become the default router, …
- DETECT-NEW-IPV6
  - Detect new IPv6 systems on the LAN, automatically launch a script
- DOS-NEW-IPV6
  - Denial any new IPv6 system access on the LAN (DAD Spoofing)

The THC IPV6 Attack Suite – The Tools

- SMURF6
  - Local Smurf Tool (attack you own LAN)
- RSMURF6
  - Remote Smurf Tool (attack a remote LAN)
- TOOBIG6
  - Reduce the MTU of a target
- FAKE_MLD6
  - Play around with Multicast Listener Discovery Reports
- FAKE_MIPV6
  - Reroute mobile IPv6 nodes where you want them if no IPSEC is required
- SENDPEES6
  - Neighbor solicitiation with lots of CGAs
- Protocol Implementation Tester:
  - Fragmentation + Routing Header
  - Mass Headers
  - Invalid Pointers
  - ...

Security relevant changes from IPv4 to IPv6

- Executive Summary:
  - IPv6 and IPv4 security is quite similar
  - Basic mechanisms are the same
  - Application layers are unaffected
  - IPv6 includes IPSec but currently not used
  - IPSec will not prevent application level attacks

Overview of security relevant changes

1. Protocol Changes
2. Reconnaissance
3. Local Attacks: ARP, DHCP
4. Smurfing (Traffic Amplification)
5. Routing & Fragmentation Attacks
6. IPv4 and IPv6 coexistence
7. Firewalling

1. Protocol Changes

- A few IP header content and options were removed:
  - No IP ID field
    - Nice uptime check not possible anymore ☹
  - No IP Record Route Option
    - No traceroute alternative anymore ☹
- No Broadcast addresses exist
- Multicast addresses can not be destined from remote
  - This is a big problem for remote alive scanning!

2. Reconnaissance IPv4

Network size in a subnet usually 2^8 = 256

Usual attack methodology:

1. Ping sweeps to a target remote class C (takes 5-30 seconds)
2. Port scans to an alive host
3. Vulnerability test to active ports

Wide range of tools available

- Nmap
- Amap
- Nessus
- ...

2. Reconnaissance IPv6 (1/2)

Network size increased to 2^64 *(varies)* in a subnet

- 18.446.744.073.709.551.616 possible hosts in a subnet
- Ping sweeps will consume too much time
  - Brute force: ***500 millions years***
  - Being clever + technology advances: still some months
- Public servers need to be in the public DNS
- All hosts need to be in a private DNS for admin purposes

>> DNS Servers will become primary <<
>> sources of information – and primary targets <<

2. Reconnaissance IPv6 (2/2)

- From remote, only the public servers (found via google, DNS, etc.) and anycast addresses can help.
- New opportunities are standardized multicast addresses to identify key servers within the local network (routers, DHCP, Time, etc.)
- Local multicasts will ensure that one compromised host can find all other hosts in a subnet
- Techniques to a single host remain the same (port scan, attacking active ports, exploitation, etc.)
- Remote alive scans (ping scans) as we know them on networks will become impossible

2. Reconnaissance with the THC-IPV6 Attack Toolkit

- ***alive6*** – for local/remote unicast targets, and local multicast addresses
  - Sends three different type of packets:
    - ICMP6 Echo Request
    - IP6 packet with unknown header
    - IP6 packet with unknown hop-by-hop option
    - *[IP6 fragment (first fragment) – if needed I will add this]*
  - One-shot fragmentation + routing header option:
    - Sends all packets in one fragment and a routing header for a router in the target network
    - Will only work if the target router allows routing header entries to multicast addresses – requires bad implementation!

3. ARP IPv4

- ARP uses layer 2 broadcast to perform the IP > MAC lookup on the local network
- Attackers will respond in order to perform "Man in the middle" Attacks

3. DHCP IPv4

- DHCP uses broadcast messages
- Rogue device can respond instead of a legal one
- Feed the host with new DNS and routing information in order to perform "Man in the middle" Attacks

3. ARP/DHCP IPv6

- No security added to both protocol variations
- ICMP6 Neighbor Discovery and Neighbor Solicitation = ARP replacement
- Duplicate Address Detection based on NS allows DoS against a host by responding to requests
- ICMPv6 Stateless auto configuration = DHCP light

3. ICMPv6 Neighbor Discovery

1. NS:
ICMP Type = 135
Src = **A**
Dst = All-Nodes Mulitcast Address
query= Who-has IP **B**?

***parasite6*:**
Answer to every NS, claim to be every system on the LAN ☺

2. NA:
ICMP Type = 136
Src = **B**
Dst = **A**
Data= Link Layer Address

If A needs the MAC address of B, it sends an ICMP6 Neighbor Solicitation to the All-Nodes multicast address

B sees the request and responds to A with an ICMP6 Neighbor Advertisement + its MAC address => Like ARP But everybody can respond to the request… => *parasite6*

3. ICMPv6 Duplicate Address Detection (DAD)

1. NS:
ICMP Type = 135
Src = :: (unspecified)
Dst = All-Nodes Mulitcast Address
query= Who-has IP **A**?

***dos-new-ipv6*: Answer to every NS, claim to be every system on the LAN ☺**

2.
No reply if nobody owns the IP address.

If A sets a new IP address, it makes the Duplicate Address Detection check, to see if anybody owns the address already.

Anybody can respond to the DAD checks… => ***dos-new-ipv6*** prevents new systems on the LAN

3. ICMPv6 Stateless Auto-Configuration

1. RS:
ICMP Type = 133
Src = ::
Dst = FF02::2:[limited mcast]
query= please send RA

***fake_router6*:**
Sets any IP as default router ☺

2. RA:
ICMP Type = 134
Src = Router Link-local Address
Dst = FF02::1
Data= options, prefix, lifetime, *autoconfig* flag

Routers send periodic as well as soliticated Router Advertisements (RA) to the all-nodes multicast address FF02::1

Clients configure their routing tables and network prefix from advertisements. => Like a DHCP-light in IPv4

But anyone can send Router Advertisements! => *fake_router6*

4. Smurf IPv4

- Sending a packet to a broadcast address with spoofed source will force response to on single target, e.g. with ICMP echo request/reply
- Traffic amplification
- DoS for target link

4. Smurf IPv6

- No broadcast addresses
- Replaced with various multicast addresses
- RFC 2463 states that no ICMP response should be sent when destination was a multicast address. However, exceptions are made.
  - Cisco Security Research got it all wrong ☺
- Exploitable?
  - Locally: YES!
  - Remote: Depends on Implementation of Routing Headers, Fragmentation etc.

4. Smurfing IPv6 with the THC-IPV6 Attack Toolkit

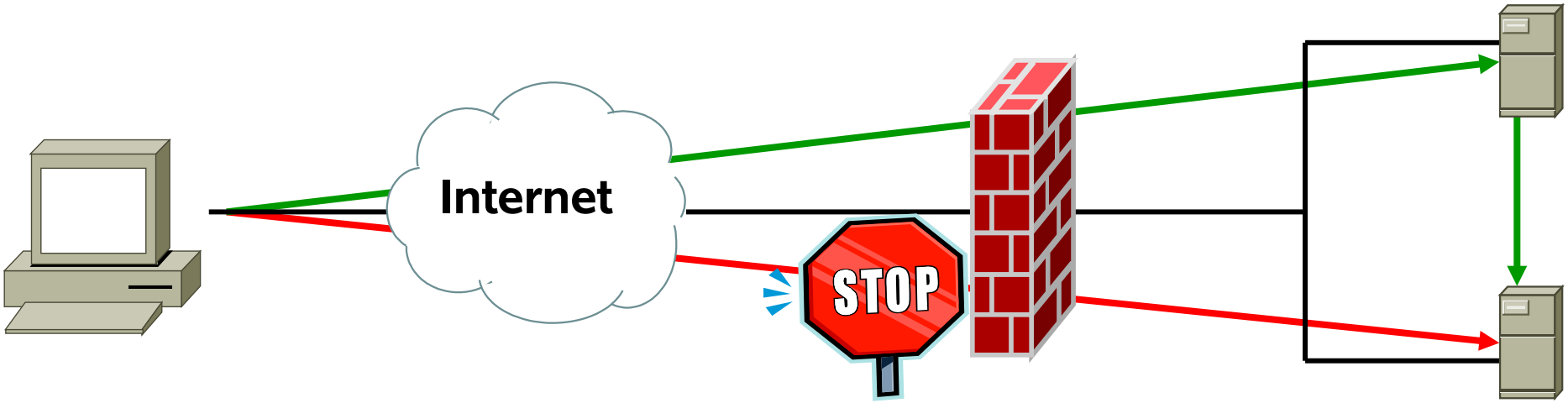
- ***smurf6*** – for local initiated smurfs
  - Source is target, destination is local multicast address
  - Generates lots of local traffic that is sent to source
- ***rsmurf6*** – reverse smurf, exploits mis-implementations (e.g. Linux)
  - Source is all-nodes multicast address (*255.255.255.255 in IPv6-speak*), destination is our target
  - If target has mis-implemented IPv6 (e.g. linux), it responds with an Echo Reply to the all-nodes multicast address, generating lots of traffic
  - On a local LAN, 1 packet in a network with 100 Linux servers generated 10000 processed packets altogether!

5. Routing Protocols

- Most Routing protocols provide their own security mechanisms
- This does not change with IPv6
- With the exception of OSPFv3, which has ***no*** security properties and relies on IPSEC usage

5. Routing Header Manipulation

Routing header attack (like IPv4 Source Routing)

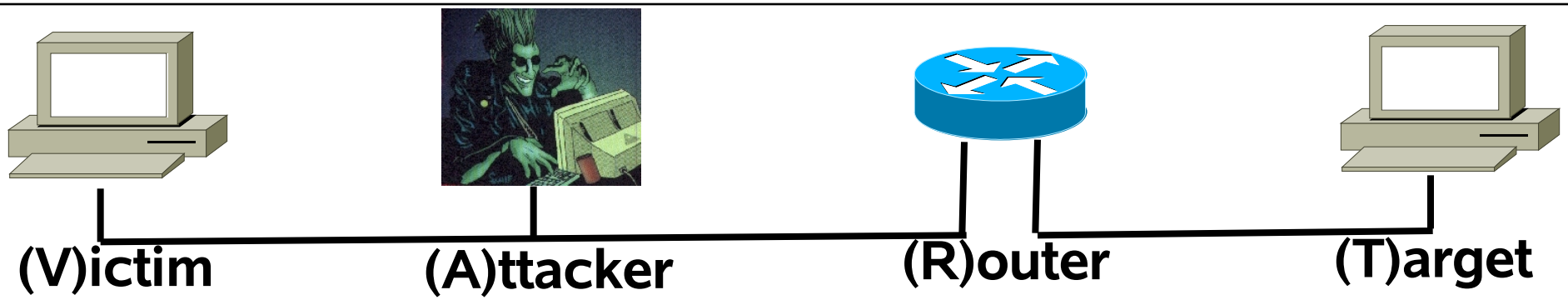


Use alive6 for checking if routing headers are allowed to target

5. Route Implanting with ICMP6 Redirects

- If a system is choosing a wrong router for a packet, the router tells this to the sender with an ICMP6 Redirect packet.
- To prevent evil systems implanting bad routes, the router has to send the offending packet with the redirect.
- If we are able to guess the full packet the system is sending to a target for which we want to re-route, we can implement any route we want! But how?
- Easy – if we fake an Echo Request, we know exactly the reply! ☺

5. Route Implanting with ICMP6 Redirects



1. (A)ttacker sends Echo Request:
   Source: (T)arget, Destination: (V)ictim
2. (V)ictim received Echo Request, and send a Reply to (T)
3. (A)ttacker crafts Redirect,
   Source: (R)outer, Destination: (V)ictim,
   redirects all traffic for (T) to (A)

Performed by ***redir6*** in the THC-IPV6 Attack Toolkit ☺

Same concept for ***toobig6*** to reduce the MTU of a (V)ictim ☺

The THC IPV6 Attack Suite – Implementation Example

- Implementation is simple!
- 5 lines of source are enough (from redir6.c:)
- Sending an ICMP6 Echo Request:
  - pkt = thc_create_ipv6(interface, PREFER_GLOBAL, &pkt_len, *target6*, ***victim6***, 0, 0, 0, 0, 0);
  - thc_add_icmp6(pkt, &pkt_len, ICMP6_PINGREQUEST, 0, 0xdeadbeef, NULL, 0, 0);
  - thc_generate_and_send_pkt(interface, NULL, NULL, pkt, &pkt_len);
- Victim6 will answer with an ICMP6 Echo Reply

The THC IPV6 Attack Suite – Implementation Example

- Sending an ICMP6 Redirect after the ping:
  - thc_inverse_packet(ipv6->pkt + 14, ipv6->pkt_len - 14);
    - This function inverses the Echo Request Packet to an Echo Reply Packet
  - thc_redir6(interface, *oldrouter6*, fakemac, NULL, ***newrouter6***, mac6, ipv6->pkt + 14, ipv6->pkt_len - 14);
    - This functions sends an ICMP Redirect, implanting ***newrouter6*** instead of the old default router *oldrouter6* for *src6*
- That's all – traffic will now be sent to *newrouter* instead!

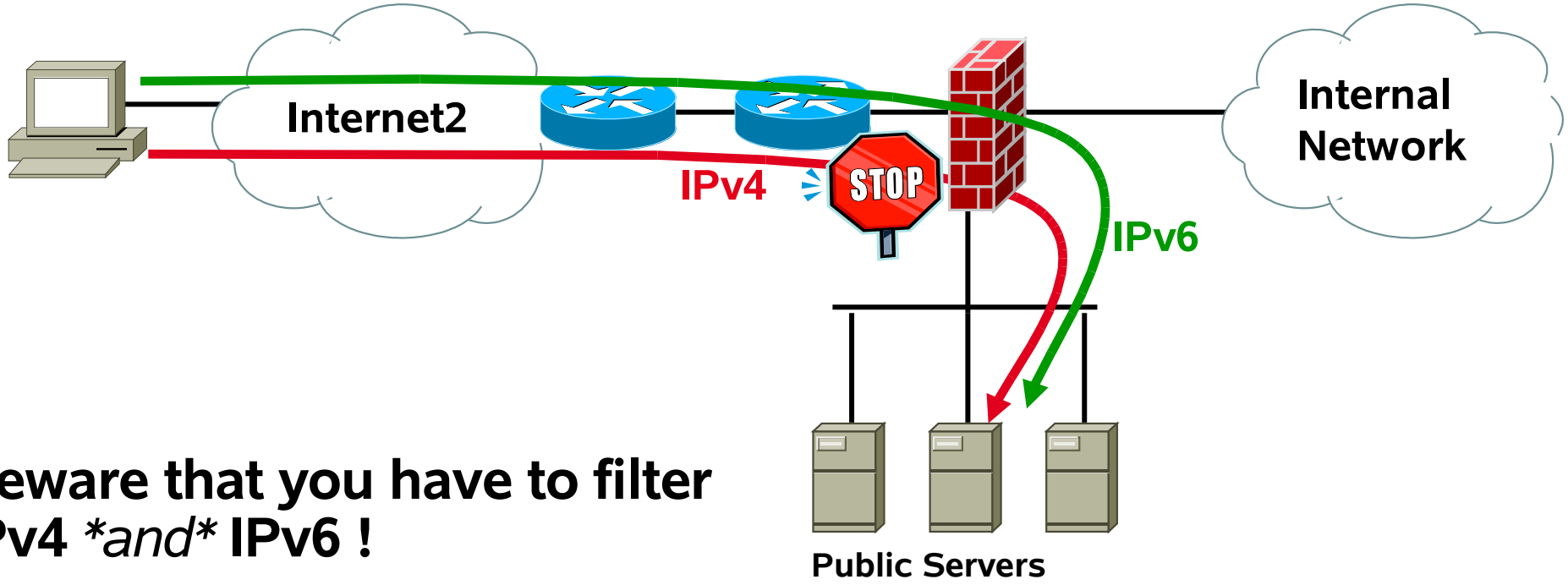
5. Fragmentation

- Fragmentation is performed by source, not routers; reassembling performed by destination only
- Routers in path will not be able to drop packets with routing header if fragmentation comes first and routing header afterwards, after reassembling.
- Same techniques for fragmentation, timeout, replays, etc. apply

5. Mobile IPV6

- Mobile IPV6 allows nodes to travel to different networks, while keeping TCP, UDP etc. connections alive – pretty cool
- Protocol specification is secure ☹ because IPSEC is mandatory
- All implementations reviewed however have the option to disable the IPSEC requirement
- If this is the done, use *fake_mipv6* to redirect traffic to any mobile IPv6 node to a destination of your choice …

6. Dual stack attack



Beware that you have to filter IPv4 *and* IPv6 !

Off The Record: Hacking inactive IPv6 Devices (HERE!)

Little hint *(e.g. for hacking at a conference *g*)*:

- Most Linux and *BSD systems now have IPv6 automatically enabled
- If they have not specified a DROP policy for IPv6 they are open, but:
  - Many OS do not allow TCP/UDP connections to the Link Local address
- To hack them anyway:
  - Start *fake_router6* with an arbitary IPv6 network prefix
  - Local systems will configure themselve a new IPv6 address based on the network prefix
  - Just collect the Duplicate Address Detection packets – these are all the systems you can now attack! ☺
    - Use *detect-new-ip6* to automate this ☺

7. Firewalling IPV6

- IPv6 will change how firewalls will work
- Many ICMP6 messages must be allowed through the firewalls to allow IPv6 to work (e.g. toobig)
- IPSEC hides data and upper layer protocols
- Much higher risks: Lots of different extension headers and options make it hard for a firewall to:
  - filter correctly
  - get it right not to BOF or DOS

Implementation Vulnerabilities in/with IPv6 so far

- Python getaddrinfo Function Remote Buffer Overflow Vulnerability
- FreeBSD IPv6 Socket Options Handling Local Memory Disclosure Vulnerability
- Juniper JUNOS Packet Forwarding Engine IPv6 Denial of Service Vulnerability
- Apache Web Server Remote IPv6 Buffer Overflow Vulnerability
- Exim Illegal IPv6 Address Buffer Overflow Vulnerability
- Cisco IOS IPv6 Processing Remote Denial Of Service Vulnerability
- Linux Kernel IPV6_Setsockopt IPV6_PKTOPTIONS Integer Overflow Vulnerability
- Postfix IPv6 Unauthorized Mail Relay Vulnerability
- Microsoft IPV6 TCPIP Loopback LAND Denial of Service Vulnerability
- Microsoft Internet Connection Firewall IPv6 Traffic Blocking Vulnerability
- Microsoft Windows 2000/XP/2003 IPV6 ICMP Flood Denial Of Service Vulnerability

Implementation Vulnerabilities in/with IPv6 so far

- Ethereal OSI Dissector Buffer Overflow Vulnerability
- SGI IRIX Snoop Unspecified Vulnerability
- SGI IRIX IPV6 InetD Port Scan Denial Of Service Vulnerability
- Apache Web Server FTP Proxy IPV6 Denial Of Service Vulnerability
- Sun Solaris IPv6 Packet Denial of Service Vulnerability
- Multiple Vendor HTTP Server IPv6 Socket IPv4 Mapped Address Handling Vulnerability
- BSD ICMPV6 Handling Routines Remote Denial Of Service Vulnerability
- Cisco IOS IPv6 Processing Arbitrary Code Execution Vulnerability
- Cisco IOS IPv6 Processing Arbitrary Code Execution Vulnerability
- Linux Kernel IPV6 Unspecified Denial of Service Vulnerability
- HP Jetdirect 635n IPv6/IPsec Print Server IKE Exchange Denial Of Service Vulnerability

Implementation Vulnerabilities in/with IPv6 so far

- 6Tunnel Connection Close State Denial of Service Vulnerability
- HP-UX DCE Client IPv6 Denial of Service Vulnerability
- Multiple Vendor IPv4-IPv6 Transition Address Spoofing Vulnerability
- ZMailer SMTP IPv6 HELO Resolved Hostname Buffer Overflow Vulnerability
- Linux Kernel IPv6 FlowLable Denial Of Service Vulnerability
- Linux Kernel IP6_Input_Finish Remote Denial Of Service Vulnerability
- Juniper Networks JUNOS IPv6 Packet Processing Remote Denial of Service Vulnerability
- Sun Solaris 10 Malformed IPV6 Packets Denial of Service Vulnerability
- Sun Solaris Malformed IPv6 Packets Remote Denial of Service Vulnerability

(data from September 2006)

Implementation Vulnerabilities in/with IPv6 so far

Place reserved for Oracle

DOS is common

- DOS-ing is easy
  - Implementation is hard, DOS is common
  - Flooding
    - router advertisements (clients)
    - neighbor advertisements (clients and routers)
    - Router solicitation (routers)
    - multicast listener discovery (routers)
    - … etc.

DOS is common

- DOS-ing is easy
  - Fun with routers: force packet forwarding processing in CPU rather than ASIC
    - Hop-by-hop extension header, especially:
      - router alert option
    - multicast listener discovery
    - Usually anything with more than two extension headers is processed in CPU
  - Hop-by-Hop router alert + upper layer processing bugs can be VERY interesting *g*
  - Crypto CPU hog exploits
    - E.g. Sending Neighbor solicititation with lots of CGAs (***sendpees6***)

Research and Implementation Tests

Tested: Linux 2.6, Windows XP SP2, Cisco IOS 12, FreeBSD 5.3

2. **Responding to packets to multicast destinations (Echo Request)**
   - Vulnerable: Linux, FreeBSD
3. **Responding to packets to multicast destinations (Invalid Header Options)**
   - Vulnerable: ***ALL*** => Status: Can be configured on BSD
4. **Responding to packets from multicast address sources**
   - Vulnerable: Linux => Status: FIXED
5. **Routing header to multicast address**
   - Vulnerable: none
6. **Fragmentation and following Routing Header**
   - Vulnerable: ***ALL***
7. **One-Shot Fragmentation**
   - Vulnerable: ***ALL***

Upcoming IPv6 Security Research from THC

- Multicast Fun
  - Global Multicast FF0E:: exploitation
  - MLD/PIM/etc. spoofing
- IPv4 <> IPv6 co-existance solutions
  - Security weaknesses in Tunneling

Upcoming IPv6 Threats and Chances

1. **Specific attack tool development for IPv6**
   - No special difference to existing IPv4 attack tools
2. **Worms**
   - TCP/IP Worms (e.g. Slammer types) will not be as effective anymore – globally
   - E-Mail Worms will stay
   - Messenger and P2P Worms will come
3. **DNS Server will become primary targets**
4. **Attacks will move to attack Clients from compromised servers in a LAN**
5. **When IPSEC is widely deployed, certificate stealing will be primary security concern**

Conclusion Internet Security with IPv6

So far no known new risks with IPv6, but some security improvements against IPv4:

- Alive-Scanning and TCP/IP Worming will be much harder
- IP Record Route Option removed, no uptime check
- Easier network filtering and attack tracing

Introduction of IPSEC will not make IPv6 secure, but will make attack tracing easy, and sniffing + Man-in-the-Middle very difficult

Some implications unclear yet, research needed

Have fun!

Thank you!

Download from:
www.thc.org/thc-ipv6