THE HACKERS 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

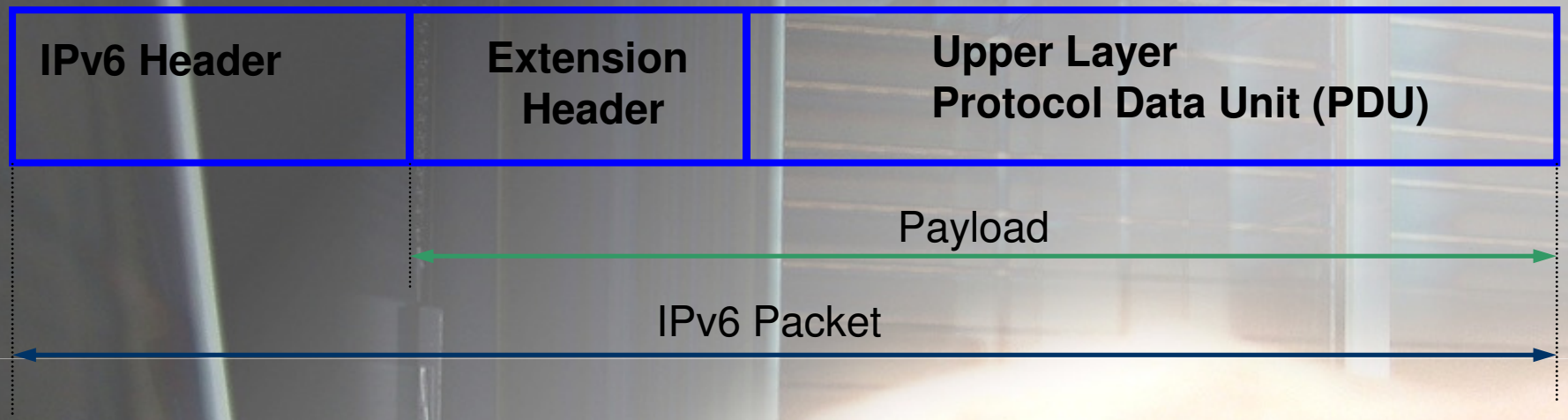
1. Short Introduction to IPv6
2. The THC IPv6 Attack Suite
3. Security relevant changes IPv4<>IPv6
4. Security Vulnerabilities in IPv6 so far
5. Implementation Vulnerabilities in IPv6
6. New Research & Future

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
- Auto-configuration of IP addresses and networking
- Hierarchical address structure
  - Reduces operational costs
- Integrated security features

IPv6 Header Structure

IPv6 Layer Structure



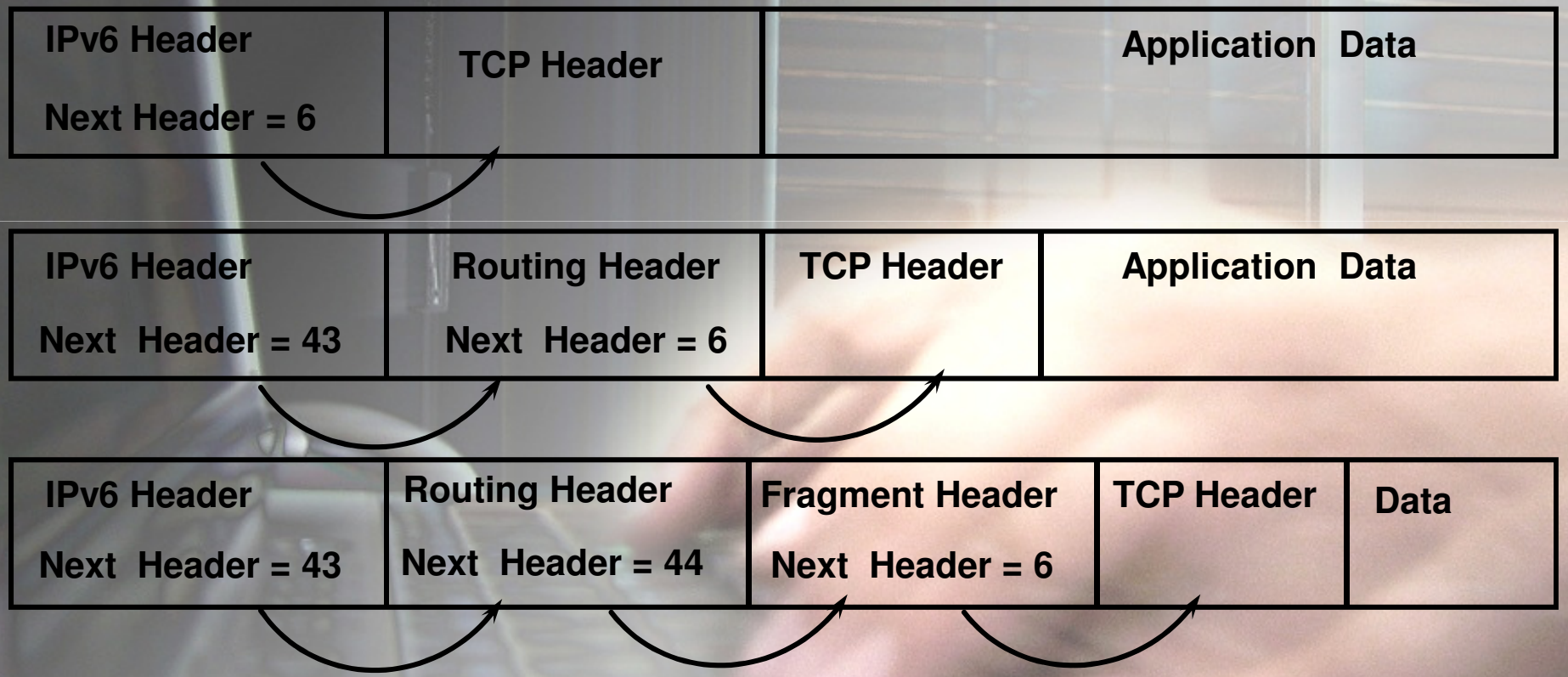
IPv6 Header ≡ **40 Bytes**

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



Blackhat usage of IPv6 today

Backdoor deployment (history now)

- Enable IPv6 6to4 tunneling
- Run Backdoor on IPv6 address
- Not detected by port scanning
- Harder to analyze traffic

Inter-Communication

- Warez exchange, IRC and bouncing

Worms

- Rbot.dud, Rabat, Maroc – Mars 2007

Availability of Hacker Tools so far …

Not many Hacker tools exist for IPv6:

- **Port Scanning: nmap, halfscan6, …**
- **Port Bouncers: relay6, 6tunnel, nt6tunnel, asybo, …**
- **Denial-of-Service (connection flooding): 6tunneldos**
- **Packet fun: isic6, scapy6, libnet (partially implemented only)**

More expected when IPv6 deployment is wider.

Specific IPv6 protocol attacking tools?

None. Except …

The THC IPv6 Attack Suite

- An easy-to-use IPv6 packet factory library by THC J
- IPv6 protocol exploits tools can be coded in just 5-10 lines
- Lots of powerful protocol exploits included
- Linux (little endian) only
- IT'S THE ONLY ONE AVAILABLE J

The THC IPv6 Attack Suite – The Tools

- **Alive6**
  - Find all local IPv6 systems, checks 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 solicitations with lots of CGAs
- **Protocol Implementation Tester**
  - Various tests, more to come

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. Miscellaneous
8. Firewalling

1. Protocol Changes

- A few IP header content and options were removed:
  - No IP ID field
    - Nice uptime check not possible anymore L
  - No IP Record Route Option
    - No traceroute alternative anymore L
- No Broadcast addresses exist
- Multicast addresses can not be destined from remote
  - This prevents 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 now 2^64 (varies) in a subnet!

- **18.446.744.073.709.551.616 IPs per 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 => primary targets <<

2. Reconnaissance IPv6 (2/2)

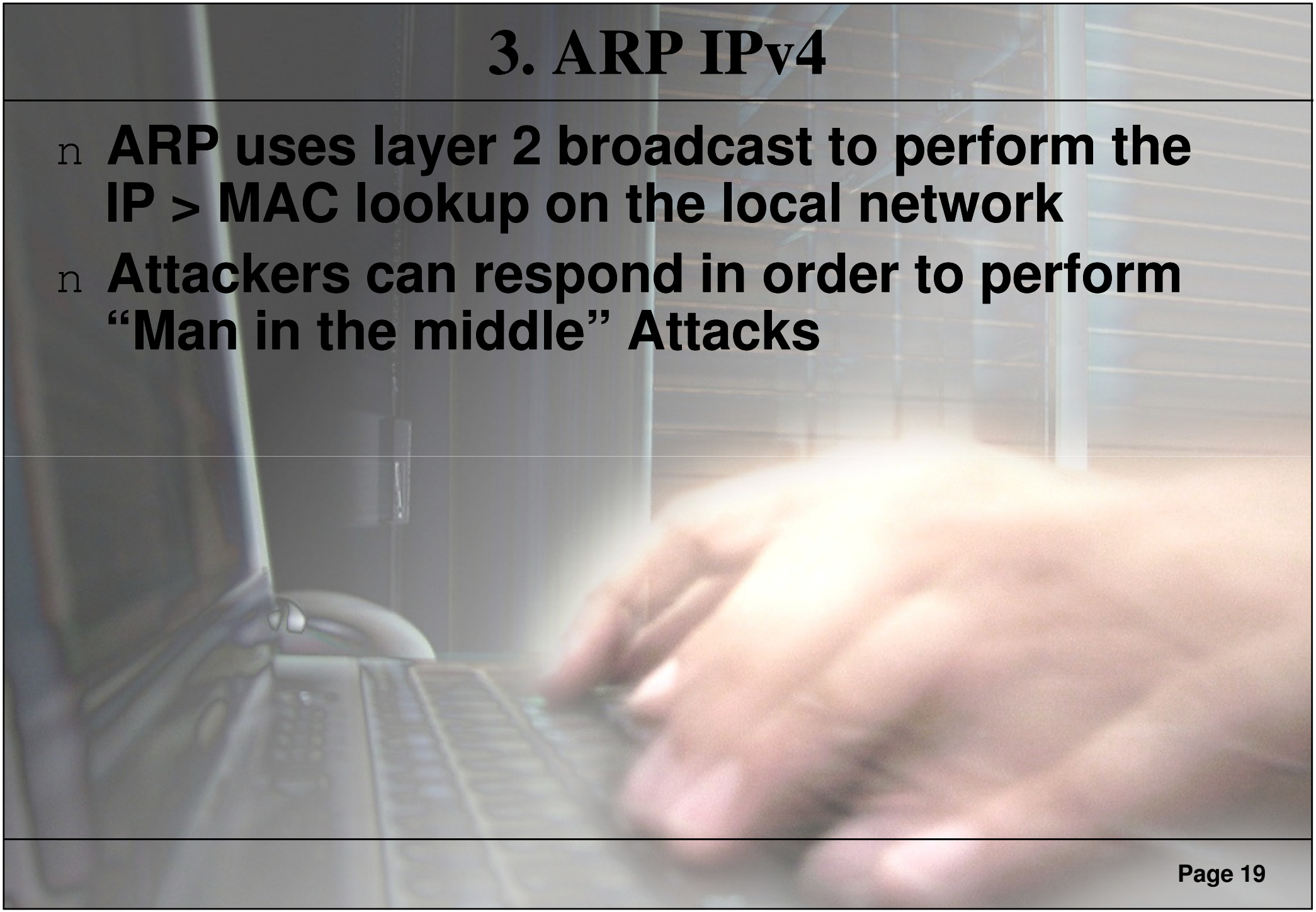
- Remote: only the public servers (via google, DNS, etc.) and anycast addresses
- New opportunities are standardized multicast addresses to identify key servers within the local network (routers, DHCP, Time, etc.)
- Local multicasts ensure that one compromised host can find all other hosts in a subnet
- 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 are unfeasible

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 packets in one fragment + a routing header for a remote router**
    - **Only works if the 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 can respond in order to perform “Man in the middle” Attacks



3. DHCP IPv4

- DHCP uses broadcast messages
- Any (rogue J) device can respond
- Feed the host with new DNS and routing information => “Man in the Middle” attack

3. ARP/DHCP IPv6

- No security added (to both)
- ICMP6 Neighbor Discovery / Neighbor Solicitation = ARP replacement
- Duplicate Address Detection based on NS allows DoS by responding to those checks
- 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 of B, it sends an ICMP6 Neighbor Solicitation to "All-Nodes" multicast address

B sees the request and responds to A with an ICMP6 Neighbor Advertisement with its MAC address

=> Like ARP But everybody can respond to the request

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 check if anybody uses 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 J

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

Routers send periodic (& soliticated) Router Advertisements (RA) to the All-Nodes multicast address

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

Anyone can send Router Advertisements!

4. Smurf IPv4

- Sending a packet to a broadcast address with spoofed source will force responses to a single target
  - Commonly ICMP echo request/reply
- Traffic amplification
- DoS for target link

4. Smurf IPv6

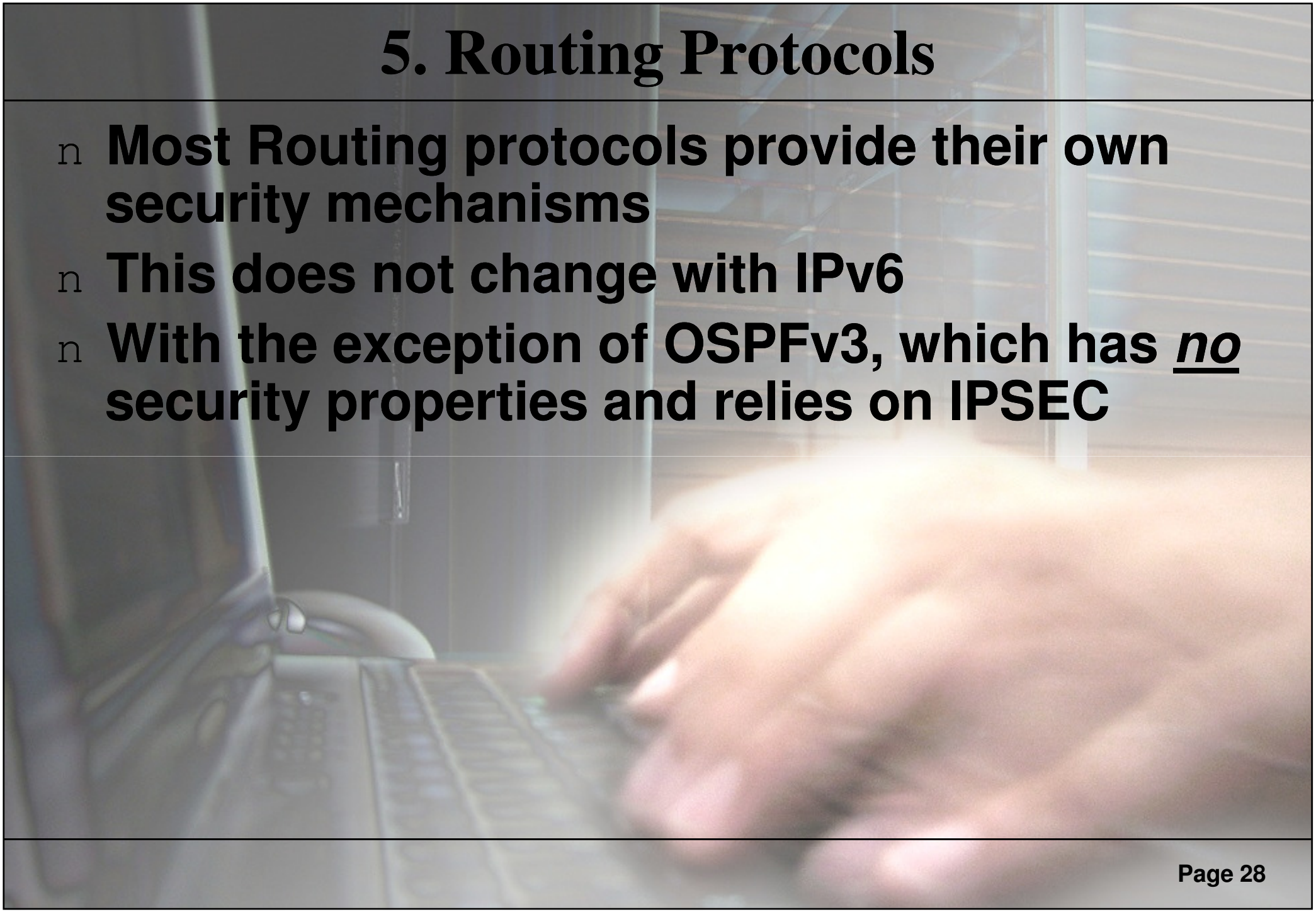
- No broadcast addresses
- Replaced with various multicast addresses
- RFC 2463 states that no ICMP response should be sent when destination is a multicast address. But exceptions are made.
  - Cisco Security Research got it all wrong J
- Exploitable?
  - Locally: YES!
  - Remote: Depends on Implementation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

4. Smurfing with the THC-IPv6 Attack Toolkit

- ***smurf6*** **– for local 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 (old Linux only)**
  - **Source is local All-Nodes multicast address (*255.255.255.255 in IPv6-speak*), destination is our target**
  - **If target has mis-implemented IPv6, it responds with an Echo Reply to the All-Nodes multicast address**
  - **FIXED in current kernels now**

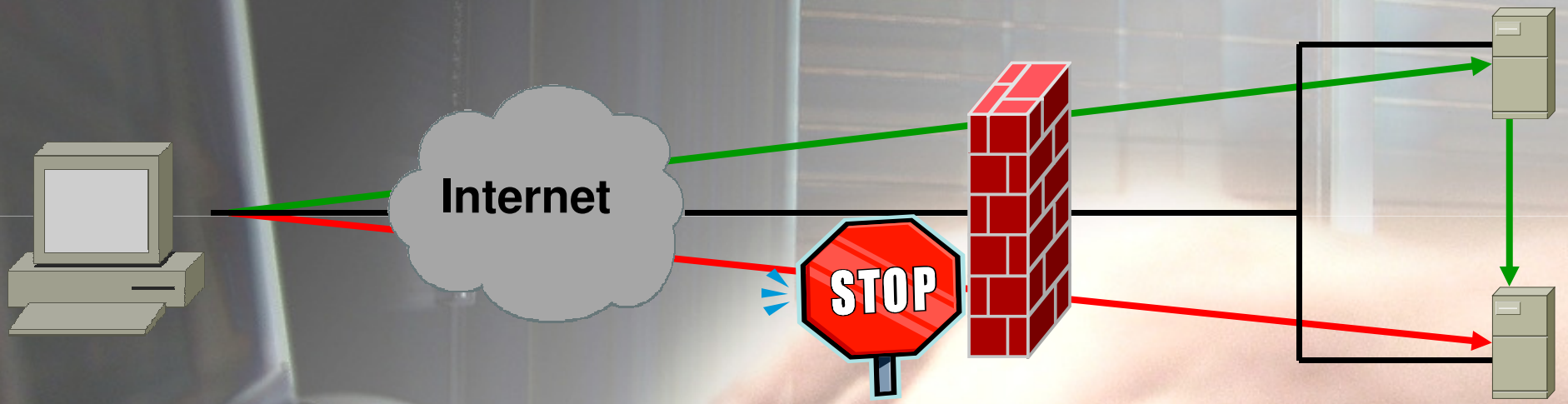
5. Routing Protocols

- 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



5. Routing Header Manipulation

Routing header attack
(like IPv4 Source Routing)



Use alive6 for checking if routing headers are allowed to target

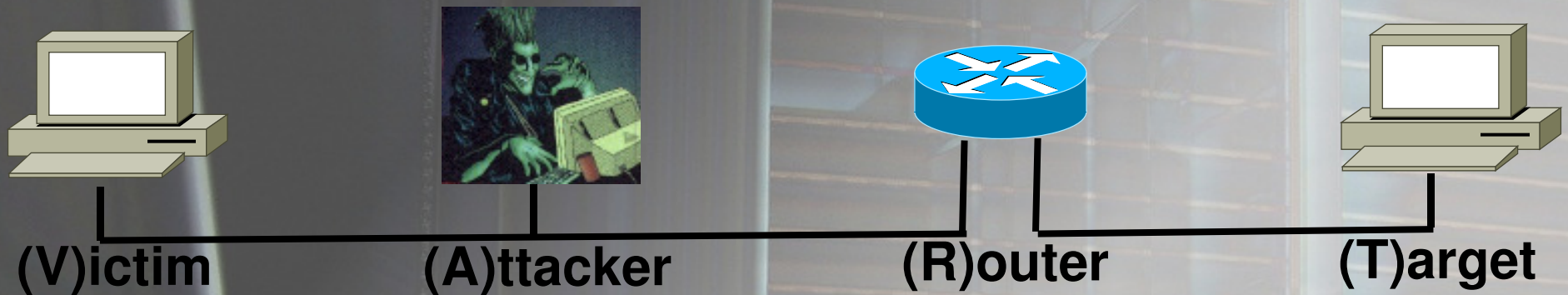
5. More fun with routing headers!

- Check if your ISP does ingress filtering
  - Send a packet from yourself to yourself via a remote system:
    - *alive6* eth0 YOUR-IP VICTIM-IP
- Find all servers in the world for an anycast address
  - Send packets to an anycast address via several remote systems:
    - *alive6* eth0 AnyCastAddr VICTIM-IP1;
      alive6 eth0 AnyCastAddr VICTIM-IP2; … etc.
- DOS network links by sending packets back and forth

5. Route Implanting with ICMP6 Redirects

- If a system is choosing a wrong local 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!
- If we fake an Echo Request, we know exactly the reply! J

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 J

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

Implementation Example – It's SIMPLE!

- 5 lines of source are enough (from redir6.c:)
- Sending an ICMP6 Echo Request[1]:
  - 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 answers with an ICMP6 Echo Reply

[1]: A ping6 packet can be gen'd+sent in 1 line, but we need do something special

Implementation Example

- Sending the ICMP6 Redirect after the ping:
  - thc_inverse_packet(ipv6->pkt + 14, ipv6->pkt_len - 14);
    - 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);
    - Functions sends the ICMP Redirect, implanting *newrouter6* for *src6*
- That's all – traffic will now be sent to *newrouter* instead!

5. Fragmentation

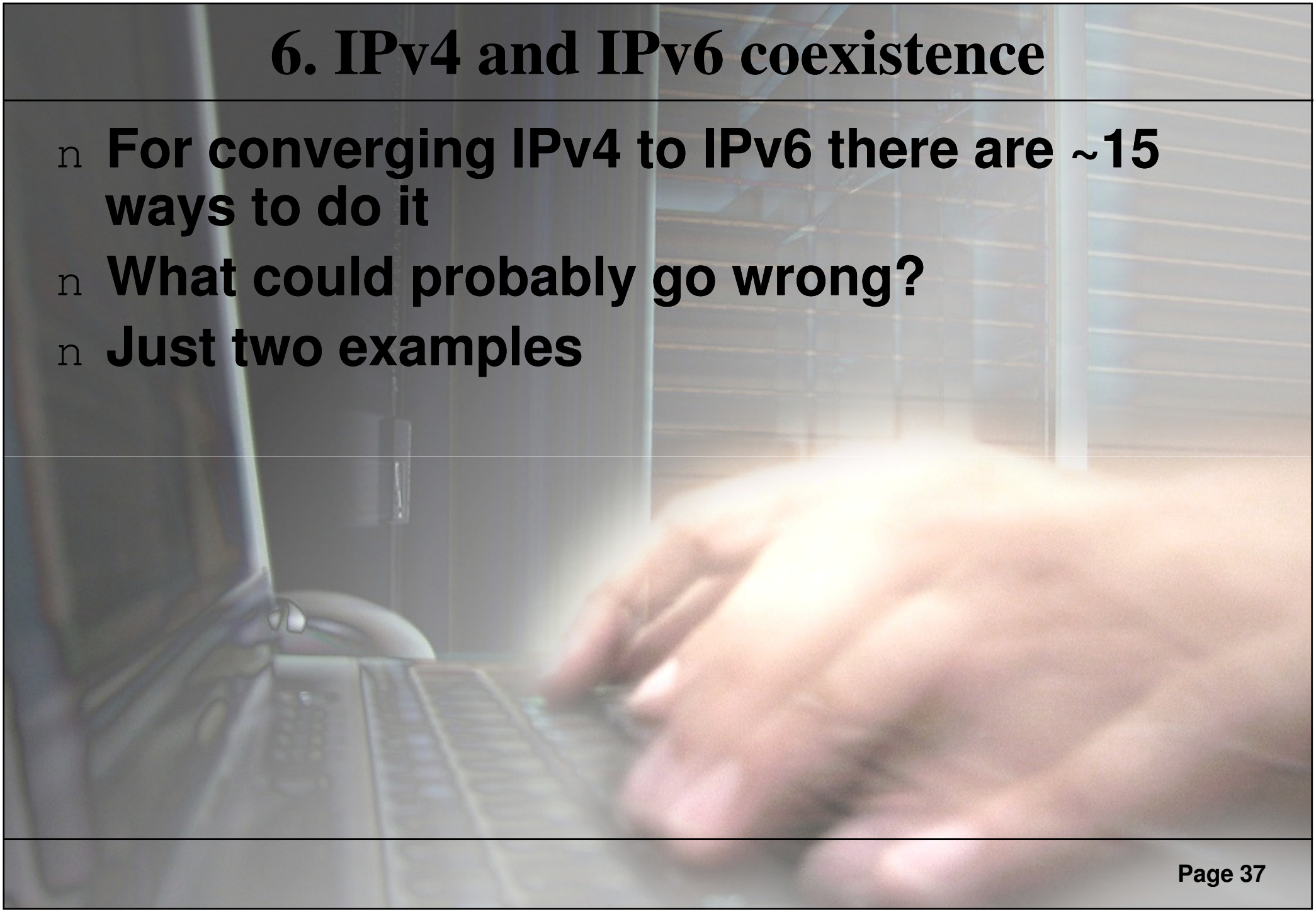
- Fragmentation is performed by source, not routers; reassembling performed by destination only
- Routers in path can not drop packets with routing header if fragmentation comes first
- Same IPv4 techniques for fragmentation, timeout, replays, etc. exist in IPv6

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 L because IPSEC is mandatory
- All implementations have the option to disable IPSEC requirement
- If this is the done, use *fake_mipv6* to redirect traffic for any mobile IPv6 node to a destination of your choice

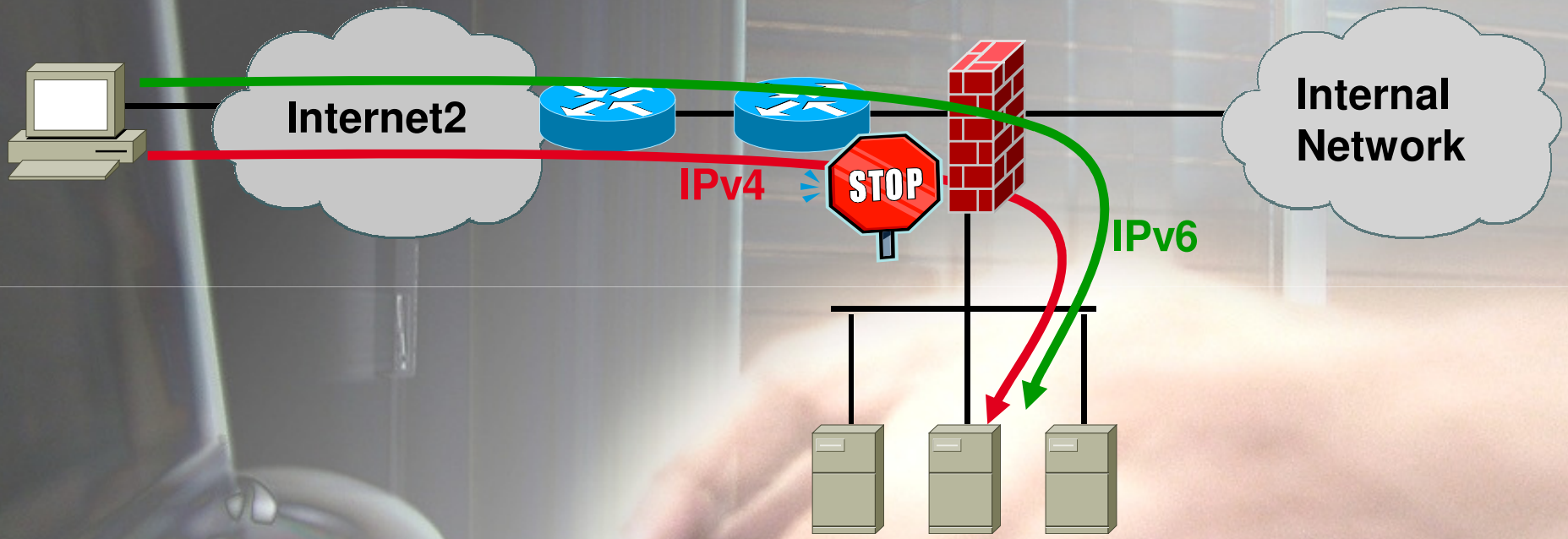
6. IPv4 and IPv6 coexistence

- For converging IPv4 to IPv6 there are ~15 ways to do it
- What could probably go wrong?
- Just two examples



6. IPv4 and IPv6 coexistence

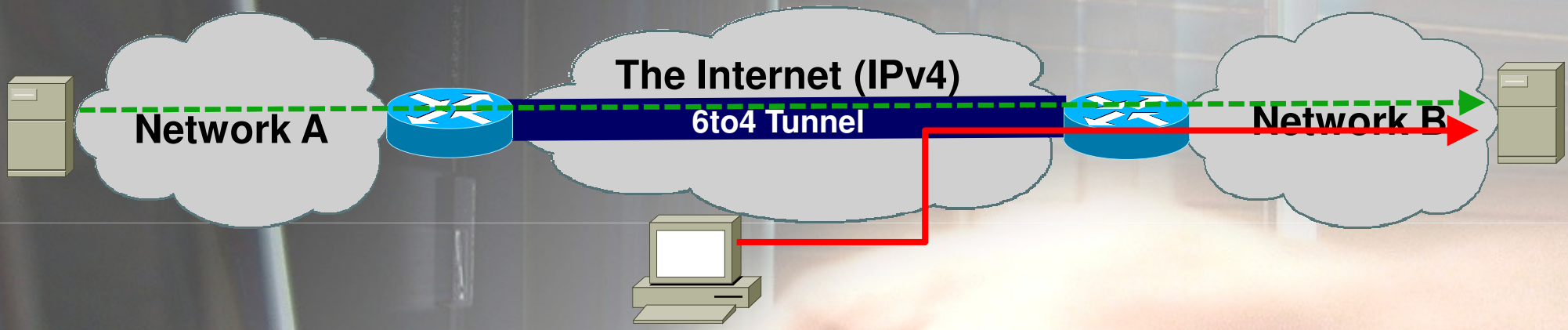
Attacks on dual stacks:



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

6. IPv4 and IPv6 coexistence

Attacks on 6to4 tunnels:



If you know the two tunnel routers its trivial to inject traffic!

IP Spoofing made very easy …

Off The Record: Attack inactive IPv6 Devices

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

- **Linux, *BSD, Vista, … have IPv6 enabled**
- **If no firewall policy for IPv6 exist = J , but:**
  - **Many OS do not allow TCP/UDP connections to their Link Local address**
- **To hack them anyway:**
  - **Use *fake_router6* with an IPv6 network prefix**
  - **Local systems will configure themselve a new IPv6 address based on the network prefix**
  - **Collect the Duplicate Address Detection packets – these are all the systems you can now attack! J**
    - **Use *detect-new-ip6* to automate this J**

7. Miscellaneous

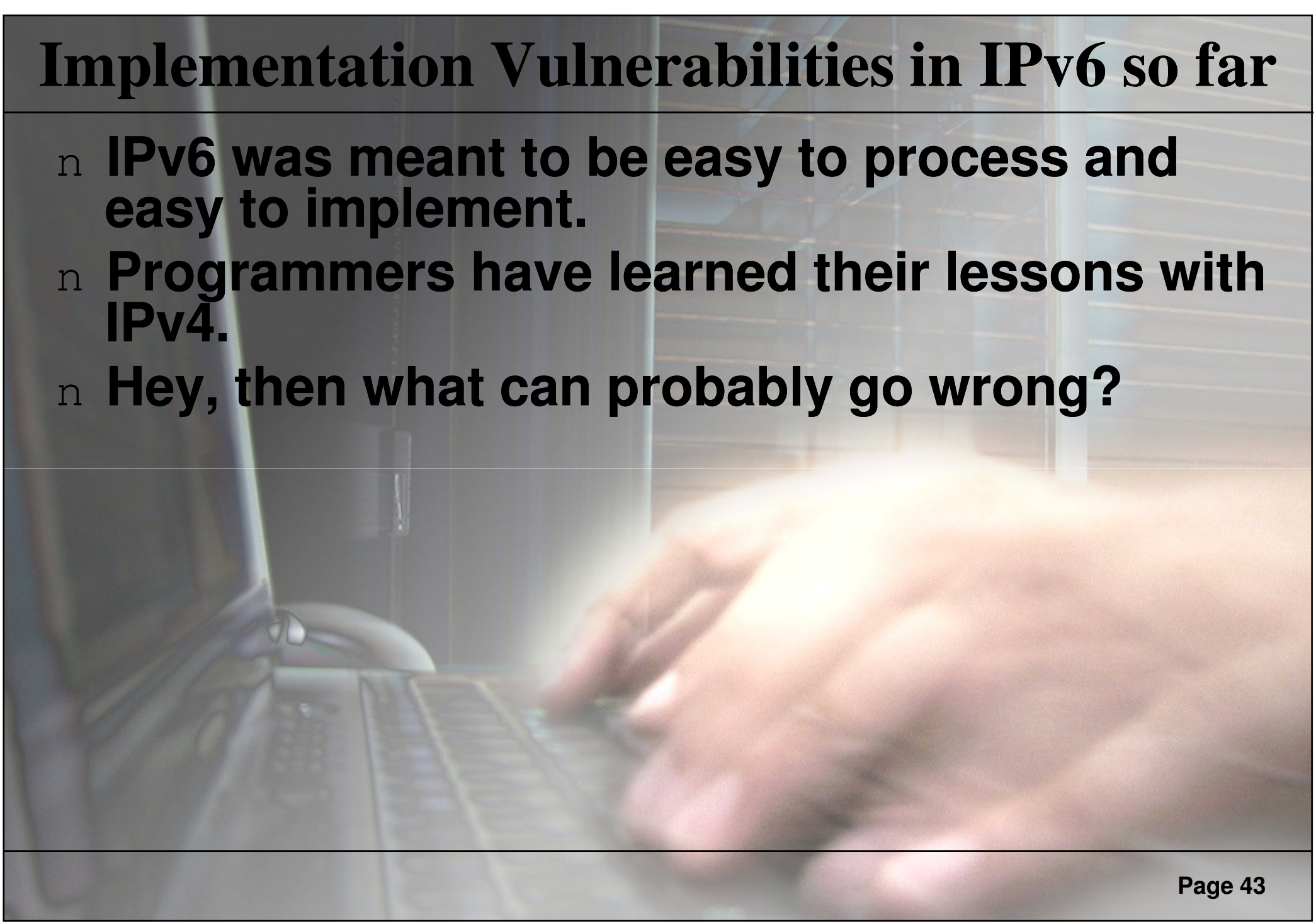
- ICMP TCP attacks do still work (tear down TCP sessions – e.g. BGP – by ICMP6 error messages, see http://tools.ietf.org/html/draft-gont-tcpm-icmpattacks-05)

8. Firewalling IPv6

- IPv6 changes how firewalls work
- No NAT necessary – and perhaps unfeasible
- Many ICMP6 messages must be allowed through the firewalls to allow IPv6 to work (e.g. toobig, errors, …)
- IPSEC hides data and upper layer protocols
- Lots of different extension headers and options make it hard for a firewall to:
  - filter correctly (not too much, not too less)
  - get it right not to BOF or DOS

Implementation Vulnerabilities in IPv6 so far

- IPv6 was meant to be easy to process and easy to implement.
- Programmers have learned their lessons with IPv4.
- Hey, then what can probably go wrong?



Implementation Vulnerabilities in 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

Implementation Vulnerabilities in IPv6 so far

- Microsoft Internet Connection Firewall IPv6 Traffic Blocking Vulnerability
- Microsoft Windows 2000/XP/2003 IPv6 ICMP Flood Denial Of Service Vulnerability
- Ethereal OSI Dissector Buffer Overflow Vulnerability
- SGI IRIX Snoop Unspecified 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

Implementation Vulnerabilities in IPv6 so far

- 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
- 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

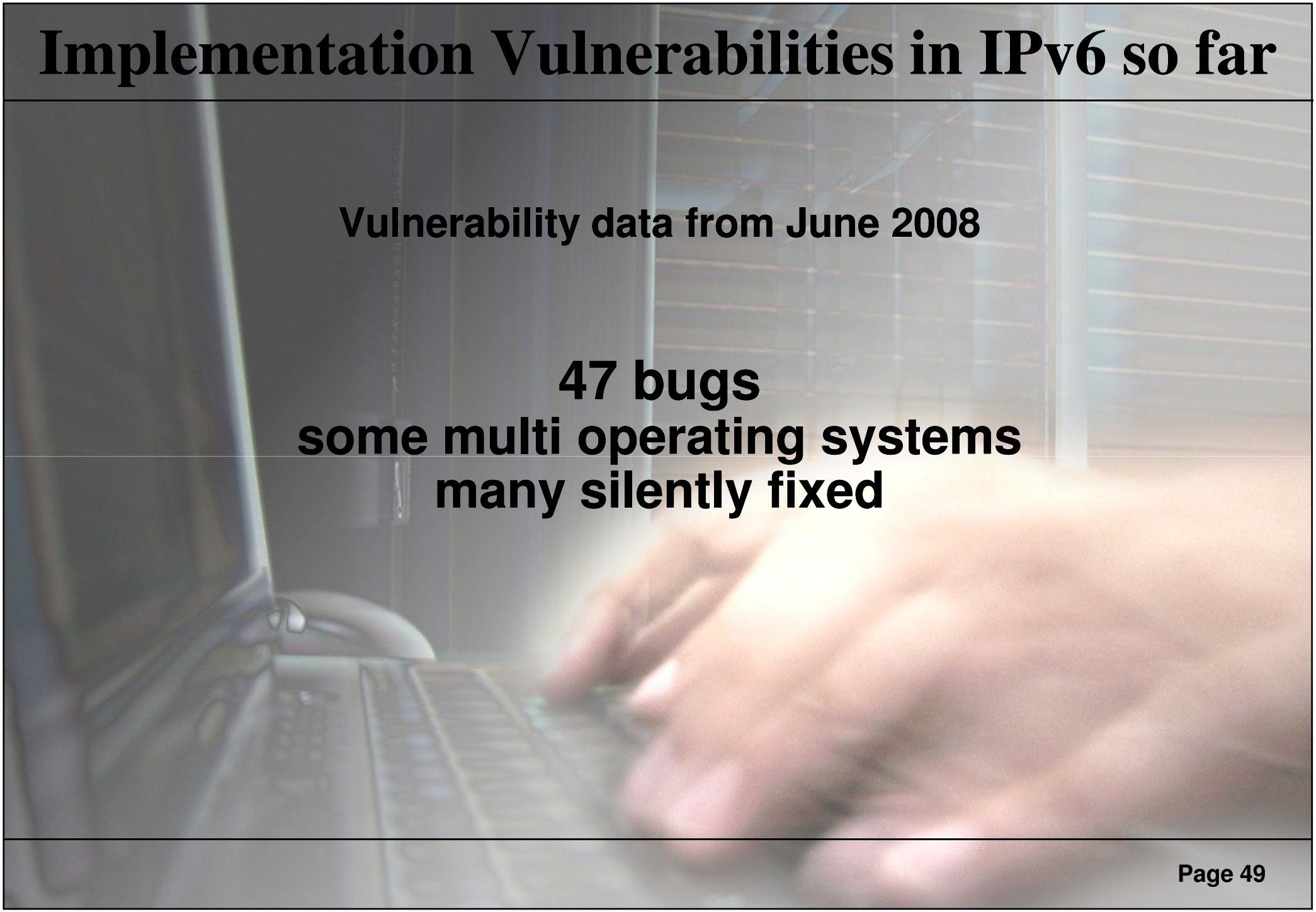
Implementation Vulnerabilities in IPv6 so far

- 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
- Windows Vista Torredo Filter Bypass
- Linux Kernel IPv6 Seqfile Handling Local Denial of Service Vulnerability
- Linux Kernel Multiple IPv6 Packet Filtering Bypass Vulnerabilities
- Cisco IOS IPv6 Source Routing Remote Memory Corruption Vulnerability
- Linux Kernel IPv6_Getsockopt_Sticky Memory Leak Information Disclosure Vulnerability
- Linux Kernel IPv6 TCP Sockets Local Denial of Service Vulnerability

Implementation Vulnerabilities in IPv6 so far

- Linux Kernel IPv6_SockGlue.c NULL Pointer Dereference Vulnerability
- Multiple: IPv6 Protocol Type 0 Route Header Denial of Service Vulnerability
- Linux Kernel Netfilter nf_conntrack IPv6 Packet Reassembly Rule Bypass Vulnerability
- Sun Solaris Remote IPv6 IPSec Packet Denial of Service Vulnerability
- Linux Kernel IPv6 Hop-By-Hop Header Remote Denial of Service Vulnerability
- KAME Project IPv6 IPComp Header Denial Of Service Vulnerability
- OpenBSD IPv6 Routing Headers Remote Denial of Service Vulnerability
- Cisco IOS Dual-stack Router IPv6 Denial Of Service Vulnerability
- Multiple Platform IPv6 Address Publication Denial of Service Vulnerabilities

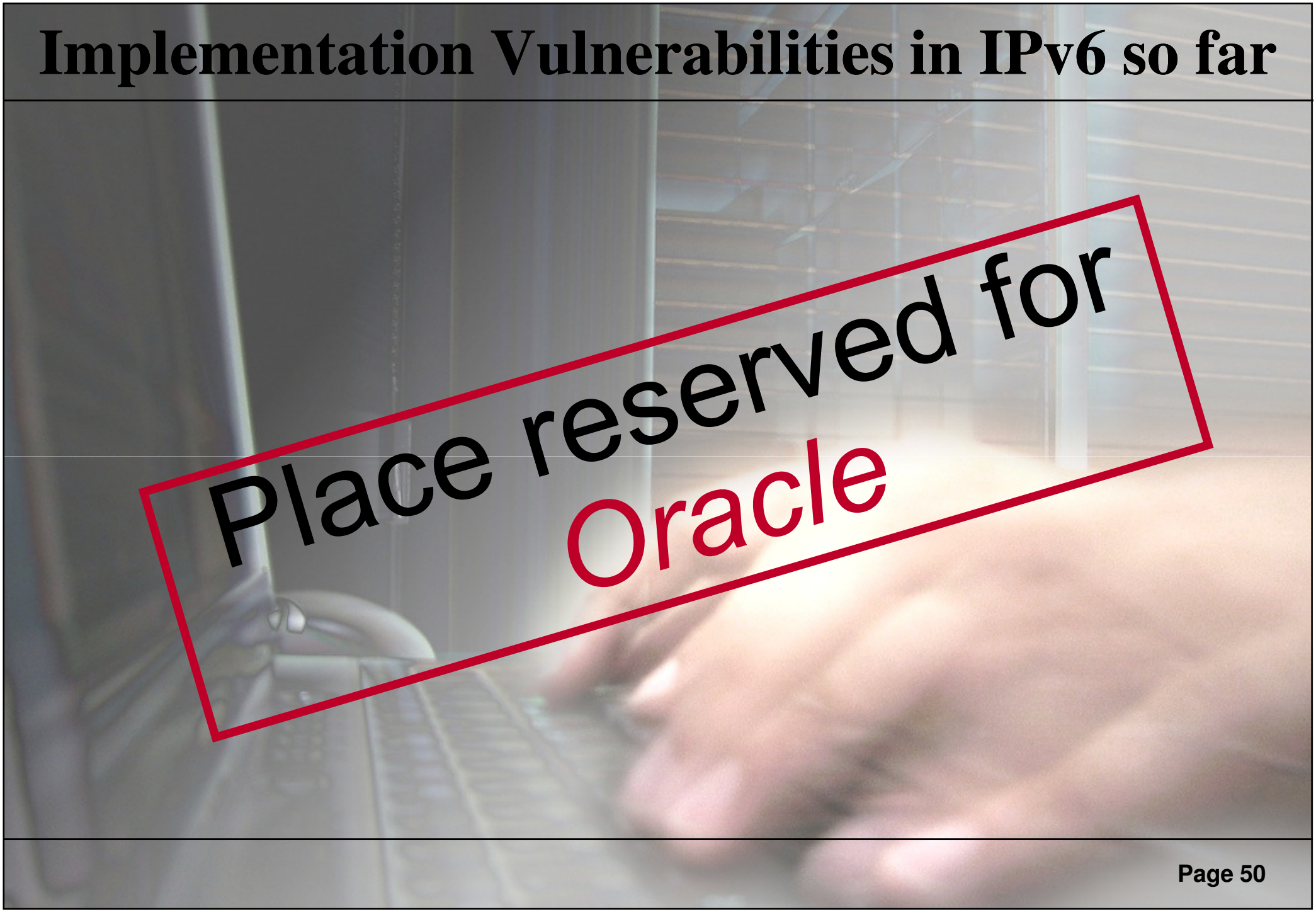
Implementation Vulnerabilities in IPv6 so far



Vulnerability data from June 2008

47 bugs
some multi operating systems
many silently fixed

Implementation Vulnerabilities in IPv6 so far



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 2.6.9, Windows XP SP2, Cisco IOS 12, FreeBSD 5.3

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

Upcoming IPv6 Security Research from THC

- Firewall IPv6 implementation tests J
  - Ipfilter6, ipfw
  - FW-1, Netscreen, PIX
- 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 real differences to existing IPv4 attack tools
2. Worms
   - TCP/IP Worms (e.g. Slammer types) will not be as effective anymore – globally
   - All other worms will stay (E-Mail, Messenger, P2P, Forum, Social Network)
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 serious new risks with IPv6, but some security improvements against IPv4:

- Alive-Scanning & TCP/IP Worming will harder
- No IP Record Route Option & 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

IPv6 BREAKTHROUGH IS NEAR!!!

“The Great IPv6 Experiment“

Free porn for everybody so people start to use IPv6!
It worked with VCR, the web, so why not for IPv6?!

http://www.ipv6experiment.com/

Have fun!

Thank you!

new version !

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