InfoWeapons
SECURE INTUITIVE RELIABLE

IPv6 Security – Foiling the Wiley Hacker

ROOTCON 5 – Cebu Philippines

Lawrence E. Hughes
Chairman & CTO, InfoWeapons

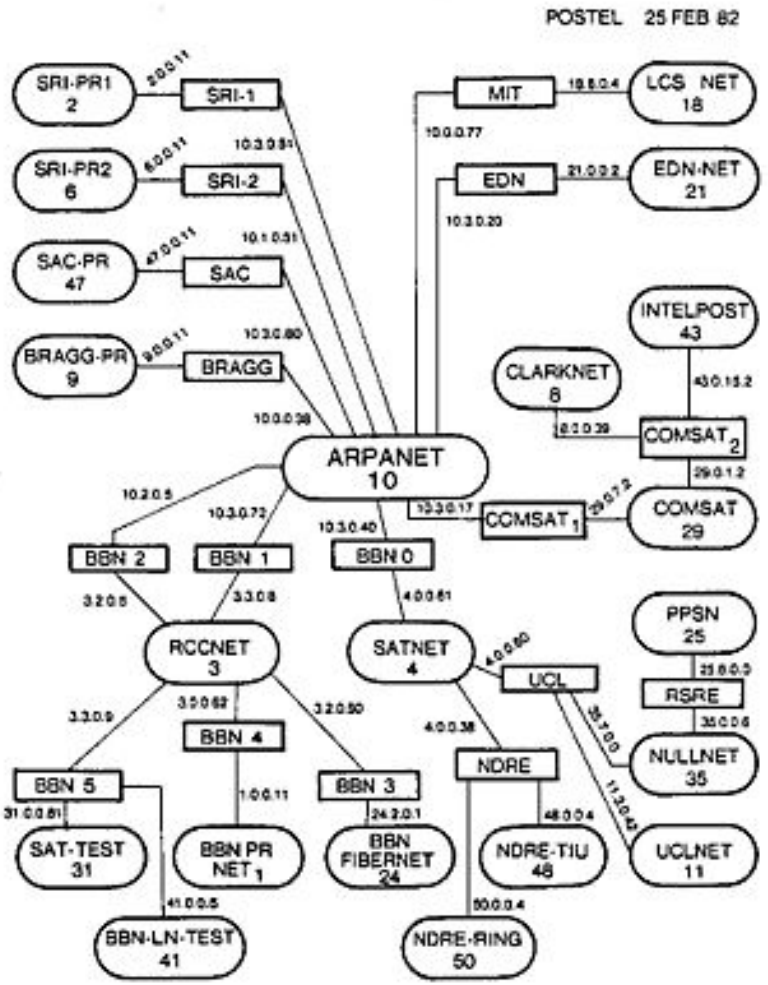
9 September 2011

Is IPv6 Here?

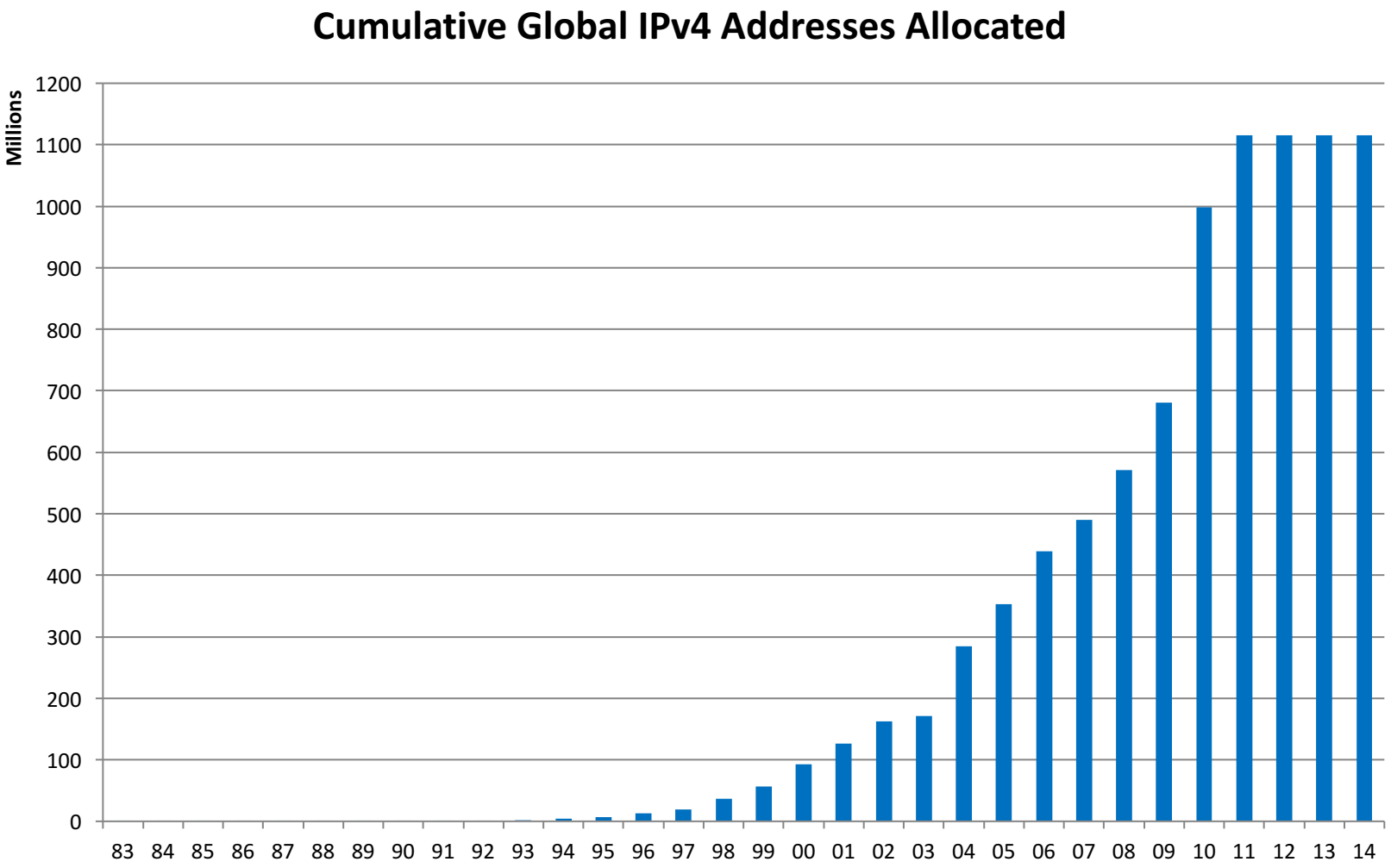
- The IANA IPv4 address pool ran out of global IPv4 addresses on 3 February 2011, which was a major event in the history of the Internet. IANA supplies addresses to the five Regional Internet Registries (APNIC, RIPE NCC, ARIN, LACNIC and AfriNIC). This generated some interest in IPv6.
- The first RIR (APNIC) reached their final "/8" block (16.7M addresses) on 15 April 2011, and formally ended normal IPv4 allocation for all of AsiaPac (about half the world's population). Since that date, an organization can only obtain a /22 (1024 addresses) or less, and then only for use in IPv6 transition (e.g. LSN, DS-Lite, etc.)
- RIPE NCC will be next (very early 2012), then ARIN (2013/2014). LACNIC and Afrinic will have IPv4 global addresses for several years, but for use only in those regions.
- ISPs will have some IPv4 addresses in stock when RIRs run out, but even those will soon be allocated to customers. Expect ISP customer allocations to get smaller and smaller, more private addresses to be given out, and very high prices charged for global IPv4 addresses as this progresses.

InfoWeapons
SECURE INTUITIVE RELIABLE

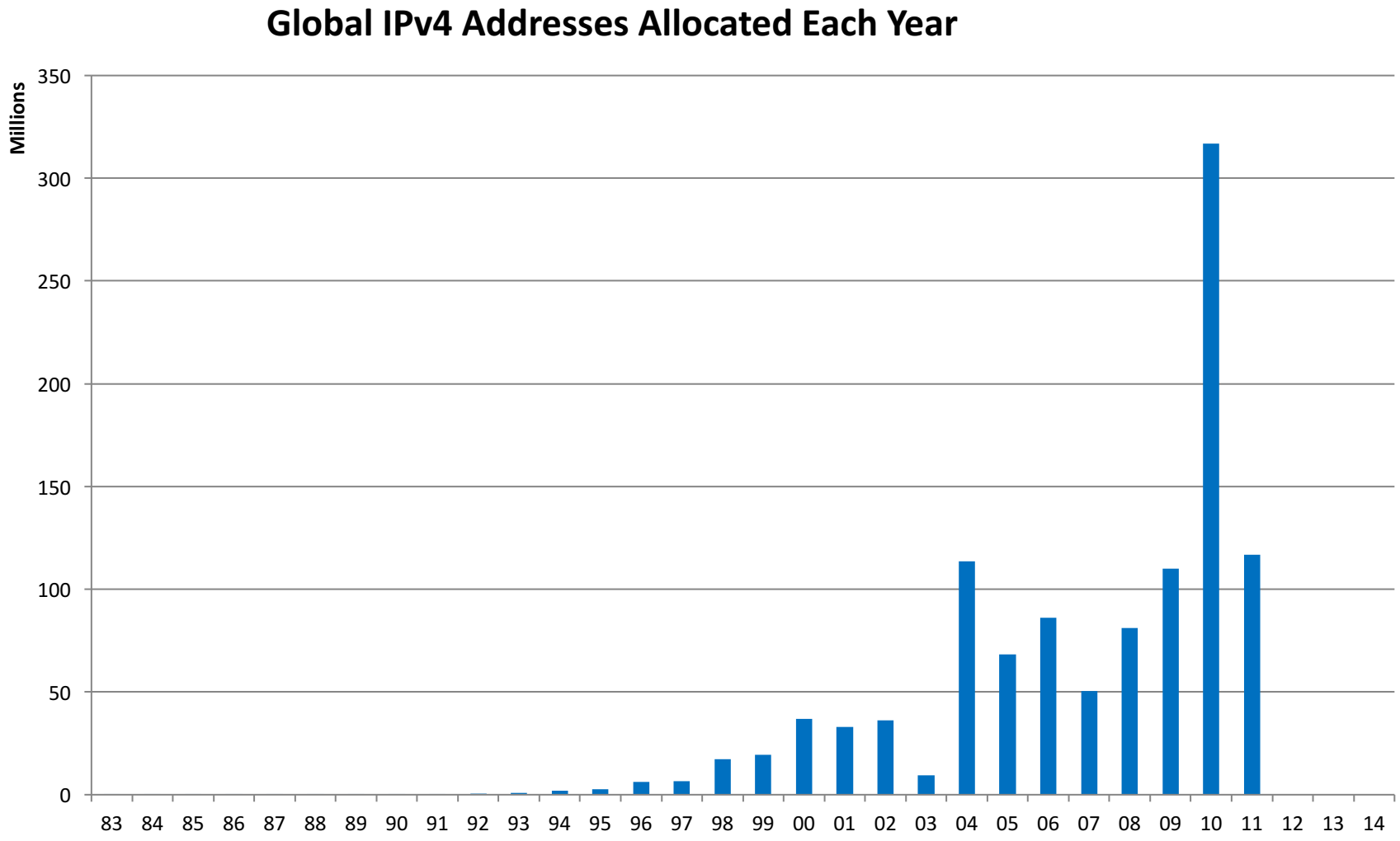
We've Been Here Before (prehistoric Internet, based on NCP)



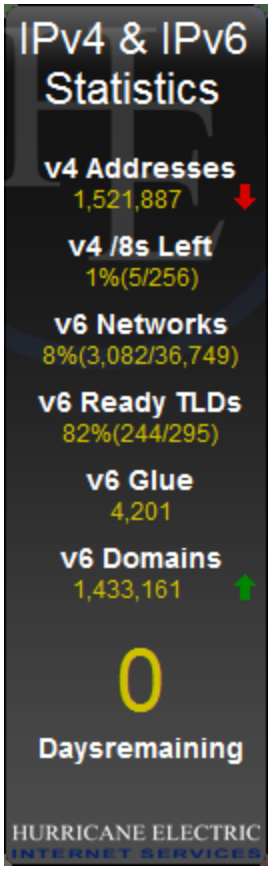
The Size of the First Internet – Linear Scale

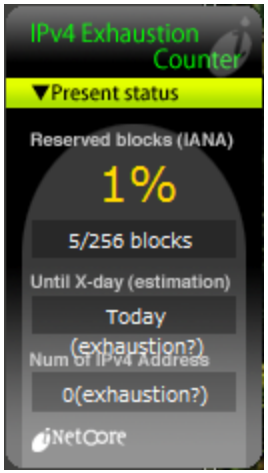


The Growth of the First Internet – Linear Scale



InfoWeapons
SECURE INTUITIVE RELIABLE

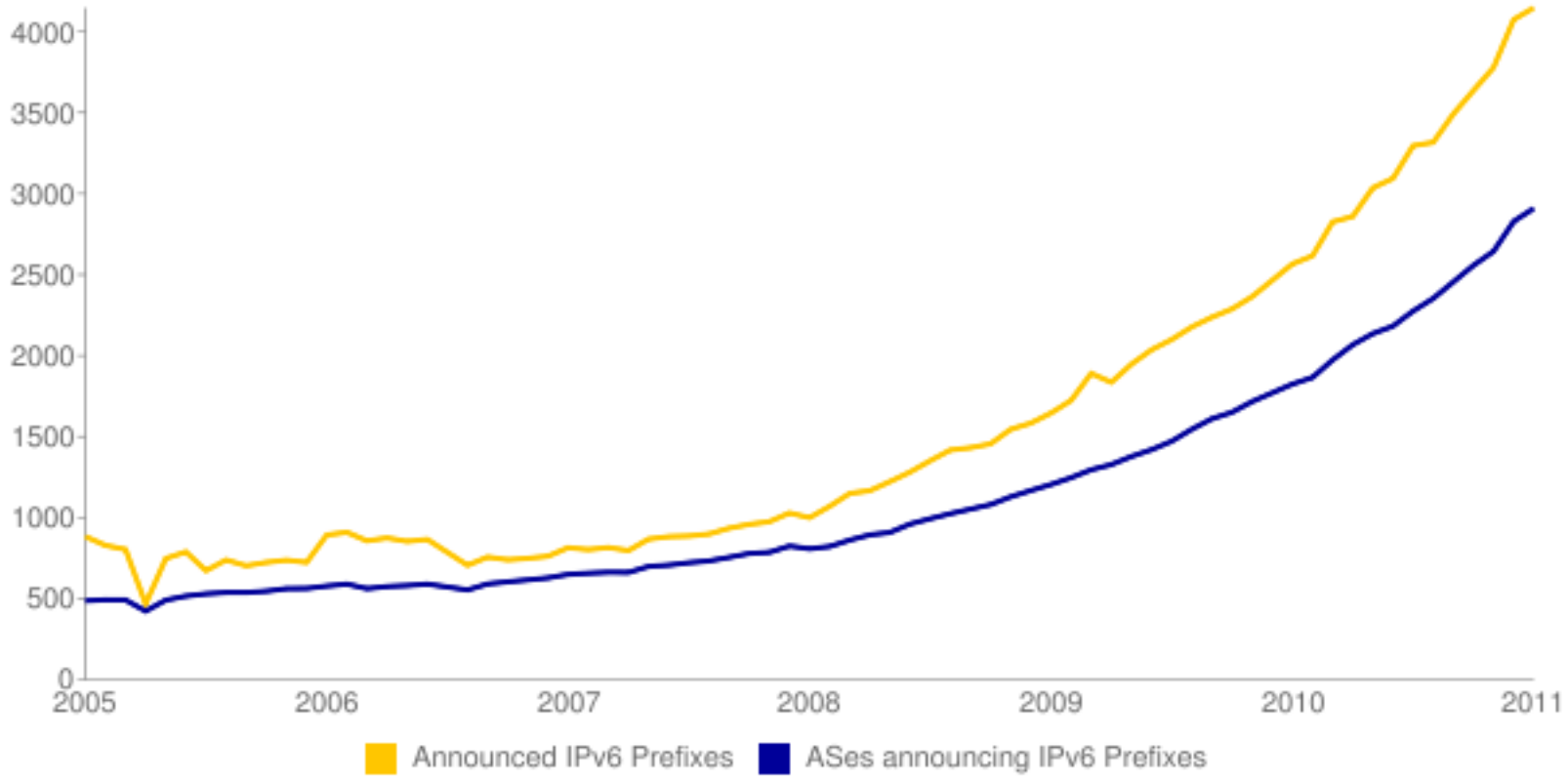




The Transition to the Second Internet (based on IPv6)

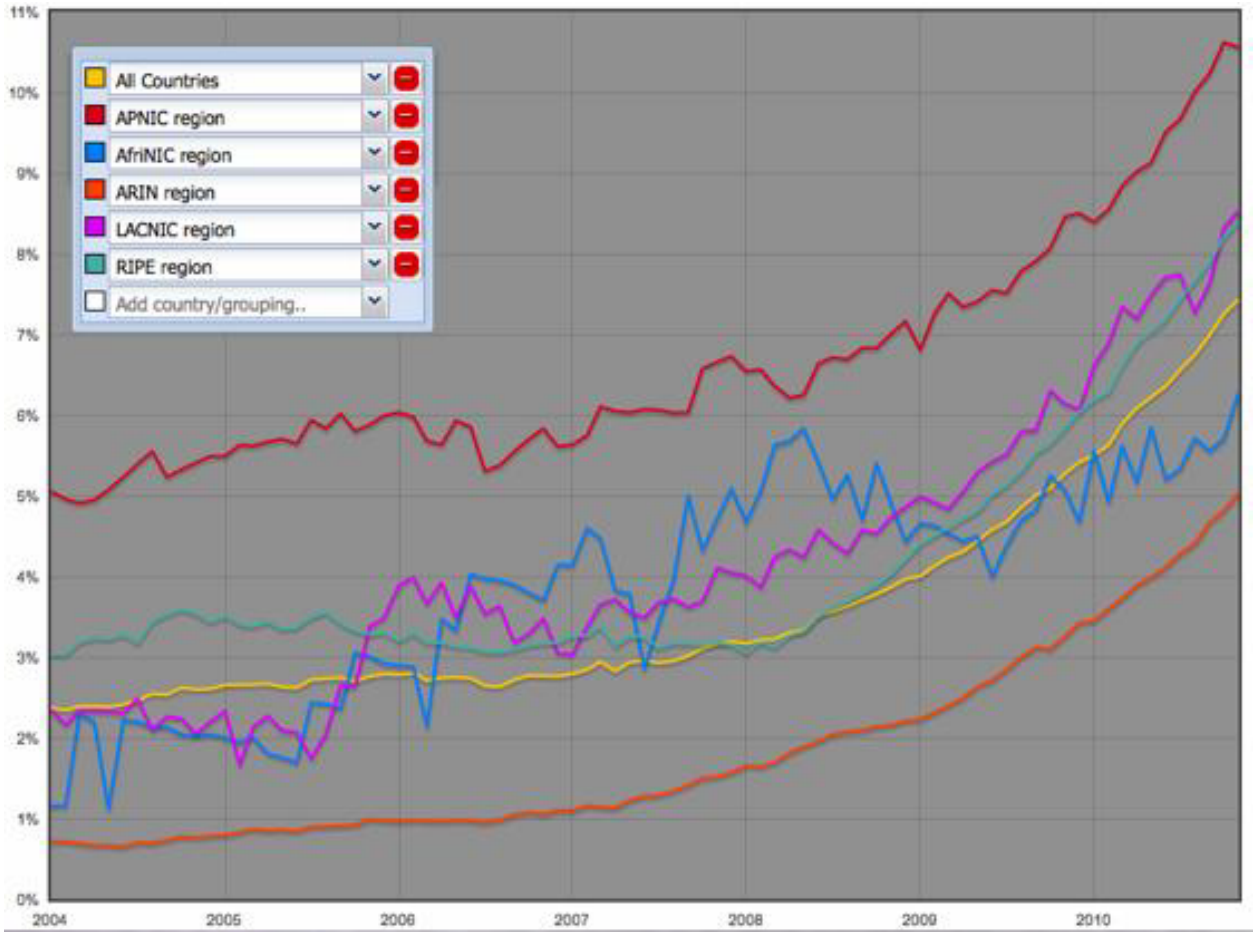
- A *Second* Internet (based on IPv6) is already growing exponentially, alongside of the *First* Internet (based on IPv4). The First Internet is going to continue operation for some time, but the exponential growth (at least of new globally routable addresses) that has characterized the First Internet to date, is about to drop to *zero*. More and more the addresses will be only *private* addresses.
- More and more nodes will be connecting to (and accepting connections from) *both* the First Internet (over IPv4) *and* the Second Internet (over IPv6). This is called Dual Stack operation. Over time, more client, server, and P2P applications (especially things that have problems with NAT, such as IPsec, VoIP, P2P, multiplayer games) will migrate to IPv6 where there is no NAT.
- Eventually (5-10 years from now?) the last IPv4 applications and services will have been migrated to IPv6, and the First Internet can be quietly shut down.

Growth of the Second Internet (IPv6) 2005 to 2011



Source: Progressing IPv6 Deployment 2011, GNKS Consult and TNO.

Growth of the Second Internet by Region



Source: Progressing IPv6 Deployment 2011, GNKS Consult and TNO.

So, Is IPv6 Here?

YES!

IPv6 Security – Areas of concern

- *Computer security* is one of the areas of significant concern to organizations planning to (or already starting to) deploy IPv6 in their networks. There are several subtopics of this fairly broad topic:
  - System security – protection at the node level. This involves host-based firewalls, Operating System vulnerabilities and understanding the threat model of a node that has enabled IPv6.
  - Network security – protection at the network level. This involves border gateway security (router ACLs, packet filtering firewalls & proxies/ALGs), intrusion detection/prevention systems, etc. It also involves understanding the syntax and semantics of the new protocols from a vulnerability viewpoint.
  - Application security – protection of network applications. This involves understanding the threat model for network aware applications (both client and server). Many IPv6-aware applications will need to accept connections directly from the outside world, not just traditional “servers” (email, web, etc).
  - Transition mechanisms – this involves issues related to dual stack architecture, tunneling of either IP in the other, and v4v6 translation.

IPv6 Security - Issues

- Training and experience – currently many security professionals know IPv4 protocols and issues in great depth, but they do not currently have much knowledge or real-world experience with IPv6 itself, let alone what differences exist with respect to computer security. They need to come up that learning curve quickly.
- Lack of “trial by fire” in IPv6 aware applications, Operating Systems and networks. Many of these have not had sufficient exposure to attack to uncover the weaknesses.
- There are many security engineers today, and there are starting to be engineers with good IPv6 knowledge. There are few today who understand both areas.
- Hackers – many hackers have had years to learn IPv6 and have done so. They already understand which of their attacks are no longer relevant (e.g. ones based on ARP) and have already created exploits against the new mechanisms (e.g. against ND). In particular, automated tunneling makes possible some very powerful attacks. Most important, the scandalous lack of general IPv6 knowledge among the defenders, and lack of support for IPv6 in many of their tools (many IDS/IPS systems, firewalls, etc) have given hackers an unfair edge which must be overcome.

IPv6 Security – Similarities to IPv4 Security

- IPv6 is based heavily on IPv4, and has many similarities. Many existing network threats and defenses are independent of which IP family is being used:
  - Authentication – username/password schemes are just as vulnerable over IPv6, but cryptographic authentication (e.g. strong client auth) works well
  - Privacy – unencrypted traffic over IPv6 is just as easy to sniff (Wireshark fully supports IPv6), and SSL/TLS protection has exactly the same benefits and weaknesses (this is really application layer). The good news is IPsec and IKE (Internet Key Exchange) work far better over IPv6 (no NAT to break them).
  - DNS can be attacked (if not protected with DNSSEC) just as easily over IPv6 as over IPv4. DNSSEC protects DNS equally well over IPv4 and IPv6.
  - Application weaknesses – buffer overflow attacks work the same in IPv4 and IPv6. Packet filtering without deep packet inspection can be fooled just as easily as in IPv4. Weaknesses in application layer protocols (e.g. web vulnerabilities) can be exploited in exactly the same ways as in IPv4.
  - OS and application security patches still need to be applied on a timely basis.

IPv6 Security – Differences from IPv4 Security

- Address resolution (mapping Internet Layer IP addresses to Link Layer addresses) no longer uses ARP (which lives in the *Link Layer*). In IPv6 this is done with the ND (Neighbor Discovery) protocol (which lives in the *Internet Layer*). Because this was moved to the Internet Layer, it can now be protected with IPsec. ARP exploits are no longer relevant, but ND exploits already exist. SEND (SEcure Neighbor Discovery) has been created, which is even better than protecting ND with IPsec, but is not widely supported at present (notably absent in Windows).
- Fragmentation attacks (breaking up packets to hide the attack beyond the first fragment) are more easily detected. Only the source node can fragment packets, and critical parts of the header(s) must be in the first fragment even then. If a hacker somehow fragmented packets after the source, it can usually be detected and rejected.
- No NAT to hide behind. NAT does not increase security in any way. NAT complicates tracking attacks to their source. A default “block all incoming except specific ports” stance is far more effective than relying on NAT to keep traffic out.

IPv6 Security – Differences with IPv4 Security (continued)

- Header extensions – in IPv4 there was a single IP header. In IPv6, there are a number of header extensions already defined, and even more will be defined over time. It is possible that hackers could exploit this new mechanism (e.g. force all packets to go via their node using source routing).
- Since there is no NAT to break it, IPsec (AH and ESP) work great on IPv6, even between organizations. With IKE (Internet Key Exchange) becoming mature, key management can be fully automated (through use of IPsec digital certificates), making IPsec easy to deploy, to link multiple networks or provide secure access by remote users. It could be deployed internally to secure *all* links, to defeat sniffing for good.

Restoration of the End-to-End Model

- One of the big differences between *IPv4-only* network architecture (with almost universal NAT) and the IPv6 world is the restoration of the end-to-end model. NAT made most connections into one-way (outbound only) channels. With AOL Instant Messenger, Alice and Bob both made outgoing connections through their NAT gateways to the server at AOL, which routed messages back and forth between them. Over IPv6, Alice and Bob can directly connect to each other (no intermediary gateway needed). Anyone can initiate outgoing connections, or accept incoming connections. This greatly complicates firewall design and the overall security model. Border proxies may be able to help in some cases, but it is difficult to break an SSL/TLS secured connection at the border, or do much of anything at the border with IPsec ESP encrypted traffic that terminates at an internal node.
- Skype already proves that NAT doesn't keep out incoming connections. *Any* NAT traversal scheme pretty much makes "swiss cheese" out of many IPv4 firewalls. It is easy to attack internal network nodes via Skype connections. The same is true of Teredo tunneling for IPv6 (due to its built-in NAT traversal). IPsec over IPv4 also requires NAT traversal, which introduces more security issues than the IPsec solves in the first place. IPsec works great in the IPv6 world (no NAT to break it).

Host Based Firewalls

- On Windows, many third party host based firewalls have only limited support for IPv6. Some have none at all. Others may even block some mechanisms such as DHCPv6 or *Stateless Address Autoconfiguration*. If you are having problems with deploying IPv6 on your Windows node, this is one of the first things to check. The good news is that in Windows Vista and Windows 7, the built-in firewall has excellent support for IPv6 (especially if you use the *Windows Firewall with Advanced Security* admin tool). With any host based firewall you should be able to independently control traffic on any port over IPv4 and IPv6, plus have specific control over ICMPv6 messages.
- On *BSD, the *pf* kernel-based packet filter can easily be deployed as an excellent host based dual stack firewall. You can even build a full gateway firewall using it. The *pfsense* open source project has built a good GUI around pf, has very limited support for IPv6. InfoWeapons has added full IPv6 support and much other advanced functionality to *pfsense* in their SolidWall firewall.
- On Linux, netfilter/iptables is roughly equivalent to *BSD's *pf*, but is not as complete. It does have support for IPv6. Unfortunately, *pf* cannot be ported to Linux, as it is so deeply embedded in the OS kernel.

Gateway Firewalls

- In addition to all the typical gateway firewall mechanisms and controls for IPv4 (including port forwarding, NAT and BINAT), true dual-stack gateway firewalls should include the following new features:
  - Support for native dual stack service, plus tunnel endpoint support for one or more mechanisms including 6in4, TSP, 6rd, and even 4in6.
  - Configurable Router Advertisement Daemon (ideally should allow configuration of IPv6 prefix via DHCPv6 prefix delegation or other mechanisms).
  - Support for multiple internal subnets (possibly 6 or more) with different /64 prefixes into each internal subnet.
  - Packet filtering controls for IPv6 traffic independent of controls for IPv4.
  - Independent control over all ICMPv6 messages (in IPv6 you can't just block *all* ICMPv6, like you can with ICMPv4 – things will break if you do)
  - Dual stack application layer proxies for the most common protocols (HTTP, SMTP, SIP, etc) – this may become the primary way to translate between IPv4 and IPv6

Transition Mechanisms

- There are major security issues with some automated tunneling mechanisms, especially **Teredo**, **ISATAP** and **6to4**, which are all enabled by default in Windows Vista and Window 7. In a corporate network, there is no need for these, and they should be disabled or removed on all nodes. Tunnel endpoints should be *in the gateway*, not *in the LAN*, so you should block protocol 41 traffic from crossing your gateway to prevent these from working on nodes that still have them present, or tunnel mechanisms installed by users (e.g. TSP client from gogo6).
- Tunnel mechanisms like 6to4 and Teredo depend on routing traffic through external nodes that you don't even know or have any control over. They also make it easy to do IP spoofing attacks. They should *never* be used. You may wish to block all Teredo (2001:0::/32) and 6to4 (2002::/16) addresses.
- IP layer translation may be an intractable problem. When NAT-PT was deprecated, a long list of problems that will affect *any* IP layer translation was included. Recent schemes such as NAT64/DNS64 have many problems and limitations. It has all of the problems of NAT44 plus new ones due to differences in semantics in IPv4 and IPv6. Dual stack is far superior from a security and functionality viewpoint.

Network Address Scopes (Link-Local, Global, etc)

- IPv4 has a primitive, incompletely realized scope concept. The block 127/8 is interface-local. RFC 3927, “Dynamic Configuration of IPv4 Link-Local Addresses”, May 2005, introduces APIPA (Automatic Private IP Addressing), using 169.254/16. If there is no DHCPv4 server, compliant DHCPv4 clients will automatically configure link-unique addresses from 169.254/16. These have some of the same capabilities as IPv6 Link-Local addresses, but not all. All IPv4 addresses other than RFC 1918 are global.
- RFC 1918 defined three ranges of IPv4 addresses for use in any private network (hence these cannot be routed on the public Internet). These include 10/8, 172.16/12 and 192.168/16. This is basically a “site local” scope, and is widely deployed, usually behind NAT gateways. A real site local scope (fec0::/10) was defined for IPv6, but has been deprecated. Unique Local Unicast Addresses (fc00::/10) were later defined to provide a more viable site-local scope. With multicast, there are several address scopes (link-local, site-local, admin-local, organization-local and global). With unicast, the entire unicast allocation block (2000::/3) is global scope.
- The fully realized scopes in IPv6 are quite different from IPv4. They do have a major impact on security, and should be well understood by any security professional.

IPv6 Link Local Scope

- All link-local addresses fall in fe80::/10.
- All IPv6 will block them from crossing any router, into another subnet.
- Much of the automated infrastructure of IPv6 (e.g. SLAAC) is implemented with link-local unicast and link-local multicast addresses. This means that it is not possible to inject ND messages using link-local addresses from outside the local link. As further protection (say if you tried to use a higher scope address), the TTL field on all ND messages is set to the maximum value of the 8-bit field, 255 (and only messages with 255 are accepted by nodes). Each router that a packet crosses will decrement the TTL value by one, so ND messages sent from nodes outside the target link will never work.
- This means that bogus ND messages must be sent from a compromised node inside the target network link (or subnet).

SEND – Secure Neighbor Discovery - Details

- SEND is specified in RFC 3971 “Secure Neighbor Discovery (SEND)”, March 2005. The threats it addresses are discussed in RFC 3756, “IPv6 Neighbor Discovery (ND) Trust Models and Threats”, May 2004.
- SEND is currently supported in Linux, FreeBSD and Cisco IOS. There is no support in Microsoft Windows (client or server), and it would not be easy for a third party to add it (since the Windows code base is not open source, and the ND functionality is implemented in the TCP/IP network stack).
- All client, server and infrastructure nodes should support SEND for maximum benefit. It may be some time before this is possible.
- SEND is not particularly easy to deploy. Among other things, it uses digital certificates, so a PKI must first be deployed and secured. Since PKI is also required for fully automated IPsec, perhaps organizations will finally deploy it.
- In the interim, since ND lives in the Internet Layer (not Link Layer as with ARP), it is possible to secure it with IPsec (AH and/or ESP). This is described in detail in the ND standard (RFC 4861, “Neighbor Discovery for IP Version 6 (IPv6)”, September 2007. IKE is not suitable for this, so IPsec protection of ND doesn’t scale well.

IPv6 Unicast Addresses: Subnet Prefix + Interface Identifier

- The low 64-bits of each 128-bit IPv6 unicast address (global or link-local) is called the *interface identifier*. It must be unique within a given local network link. The first 64-bits of each IPv6 unicast address is called the *subnet prefix*. All link-local unicast addresses use the special subnet prefix *fe80*. Global addresses must have a 64-bit value valid in their subnet, that is globally unique (e.g. *2001:470:3d:3000::/64*).
- Every IPv6 node can generate a 64-bit interface identifier that is unique in that link (e.g. *a81e:950c:9077:defb*). This is used to create a link-local node address (*fe80::a81e:950c:9077:defb*) and (with the help of a router advertising the subnet prefix) a global unicast node address (*2001:470:3d:3000:a81e:950c:9077:defb*).
- It is also possible to manually assign any number of *static global unicast addresses* to a node, using the subnet prefix and any interface identifier unique within that link, e.g. *2001:470:3d:3000::1:100* and *2001:470:3d:3000:172:25:0:1*. These are *not* aliases.
- If your network includes a stateful DHCPv6 server, and the router advertises its availability (M and O flags set in Router Advertisement message), the node will obtain *yet another* global unicast address from its pool, e.g. *2001:470:3d:3000::4:1001*, with an expiration date (optionally via reservation, just like DHCPv4).

IPv6 Unicast Addresses: EUI-64 based Interface Identifiers

- The original IPv6 specification (RFC 2460) defined using the *EUI-64* algorithm to generate the interface identifier. This generates a 64-bit value using the interface's 48-bit MAC address. EUI-64 splits the MAC address into two 24 bit fields, inserts the 16 bit value *fffe* between them, then sets the 7th bit to 1. If the MAC address is *00:90:0B:1B:57:62*, then the EUI-64 value is *290:bff:fe1b:5762*. EUI-64 interface identifiers can always be recognized by the *fffe* in the middle. Anyone can recover your MAC address from an EUI-64 global address. When you use EUI-64, your MAC address (which normally never leaves your local link) can go out into the world, which is a security concern.
- If you download copyrighted material or hack someone using an EUI-64 based address, it could be traced to you and used to prove you did it. There is no NAT to hide behind.
- Anyone could link any online activity together with other activity using the same address as having come from the same node (or at least the same MAC address).
- A hacker could use EUI-64 addresses from network connections with the same subnet prefix to help "map" that network as preparation for an attack.

IPv6 Unicast Addresses: Randomized Interface Identifiers

- To address these security issues, the IETF released RFC 3041, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6”, January 2001. It was replaced with RFC 4941 (same title), in September 2007.
- This introduces two new concepts:
  - *Randomized interface identifiers* – the 64 bit interface identifier is chosen from the 2^{64} possible identifiers, rather than from the relatively tiny number of possible EUI-64 identifiers. The MAC address no longer leaves the subnet.
  - *Temporary addresses - i*dentifiers that *change* over time – when an automatically assigned address expires (its *preferred lifetime* reaches zero), a new randomly chosen interface identifier is used to generate a new address. The old and new address will both accept incoming connections until the old one’s *valid lifetime* reaches zero. The network admin can choose how often such changes happen, and the length of the overlap period (by choosing appropriate preferred and valid lifetimes for the subnet). The newly generated address can be advertised in DNS to make the change invisible to other users.

IPv6 Unicast Addresses: Randomized Interface Identifiers

- By default, Microsoft (Vista, Win7, Server 2008 and Server 2008R2) use *randomized interface identifiers*, but you can change it back to EUI-64 via *netsh* commands. They also support *temporary addresses* by default (in addition to the more stable global unicast address), but this can also be disabled.
- By default, FreeBSD and Linux use EUI-64 interface identifiers, but you can enable *randomized interface identifiers* and/or *temporary addresses* in both OSes.
- It's possible a hacker (or law enforcement agency) could use a Trojan attack to disable the privacy extensions on nodes (to take advantage of stable EUI-64 based addresses). Most users would never notice the difference.
- In theory a border gateway could be designed to block EUI-64 based addresses from leaving the network. This would require all internal nodes making outgoing connections to use the privacy extensions or manually assigned static addresses.

Are you safe if you configure only IPv4?

- Say you are running Windows Vista, Windows 7, Linux, Mac OS X, etc. You have only configured IPv4. When you use *ipconfig* or *ifconfig*, no IPv6 addresses appear (or only a link-local address). None of your network gear (routers, firewalls, IDS, etc) support IPv6. Think you're safe?
- On Windows, Teredo will happily make connections through tunneled IPv6, without any configuration (at least 6to4 and ISATAP require *some* infrastructure to be set up). When Google enabled IPv6 on YouTube, some 250,000 connections a day began almost immediately. Most of those were via Teredo, without the user's knowledge. Most people's security systems (firewall, IDS/IPS) are not even aware of Teredo.
- A hacker could send a bogus Router Advertisement into your net, and many of your nodes will silently configure IPv6 and the hacker can communicate with them. Again, your network security gear may not even see this traffic happening.
- Even without getting your nodes to do SLAAC (causing global Ipv6 addresses to appear in ipconfig or ifconfig), every node that has IPv6 enabled can communicate via its link-local address, at least with other nodes in the link. A hacker could easily implant a Trojan in your subnet to communicate with your nodes via link-local addresses.

So, I've Deployed Dual Stack in my Network.

- This is better than pretending that IPv6 doesn't exist. At least you can see and control IPv6 traffic on your network.
- You should still block *all* internal tunneling (e.g. Teredo) ideally both on all Windows nodes and at the border gateway (by address range).
- You should upgrade all of your network defenses and infrastructure to dual stack. It s OK for a few legacy (IPv4-only) devices such as printers to support only IPv4.
- People will want to make use of the new found end-to-end connectivity, such as allowing direct end-to-end VoIP, P2P, etc. You now have tons of incoming connections direct to end nodes that you never had to worry about before. You may also need to publish all of those nodes in your external DNS (otherwise how will other people find those nodes?)
- By the way, *you have just doubled your attack surface*. Hackers will always attack via the weakest point in your defense. For most networks, this will be the IPv6 side, until their admins learn as much about IPv6 as the already know about IPv4. Why bother going through the well secured front door (IPv4), when the back door (IPv6) is *wide open*?

IPv6 Hacking Tools – Sniffers, Packet Capture

- Snort – Intrusion detection tool - http://www.snort.org/
- WinPcap – Promiscuous mode packet capture tool for Windows (used by other tools such as WireShark) - http://www.winpcap.org/
- TCPdump / LibPCap – command line promiscuous mode packet capture tool for FreeBSD / Linux / Mac OS X - http://www.tcpdump.org/
- Windump - TCPDump for Windows - http://www.winpcap.org/windump/
- COLD - (supports IPv6 since 1.0.12) - http://www.ipv4.it/cold/
- Wireshark - GUI based packet capture and protocol analysis tool (IPv4 + IPv6) for Windows, Mac OS X - http://www.wireshark.org/

IPv6 Hacking Tools – Scanners, Redirection, Denial of Service

- Scanners
  - IPv6 security scanner - http://www.securiteam.com/tools/5EP0I1F7FM.html
  - Halfscan6 – http://freshmeat.net/projects/halfscan6
  - Nmap – http://freshmeat.net/projects/nmap/ – current version supports IPv6
  - Strobe - http://tldp.org/HOWTO/Linux+IPv6-HOWTO/ipv6-security-auditing.html
- Traffic Redirection
  - Netcat – now supports IPv6 - http://netcat.sourceforge.net/
- DoS Tools
  - 6tunneldos - http://packetstormsecurity.org/files/favorite/25398/
  - 4to6ddos - http://packetstormsecurity.org/files/view/23730/4to6.tar.gz
  - Imps6-tools - http://packetstormsecurity.org/files/25417/imps6-tools.tar.gz.html

IPv6 Hacking Tools – Packet Forgers / Complete Toolkit

- Packet forgers
  - **Scapy** - generate any IPv4/IPv6 packet (even pathological)
    - IPv6 functionality merged into main project (no longer separate scapy6)
    - Embedded in python scripting language (must learn python to use scapy)
    - http://hg.scdev.org/scapy
  - **SendIP** – send any IPv4/IPv6 packet (no need to learn python)
    - http://freshmeat.net/projects/sendip
  - **Packit** – Packet Toolkit - Network injection and capture
    - http://packetfactory.openwall.net/projects/packit
- Complete toolkit – **THC-IPv6** – attacking the IPv6 protocol suite
  - Contains many tools, runs on FreeBSD / Linux / Mac OS X
  - http://thc.org/thc-ipv6/

Final Thoughts

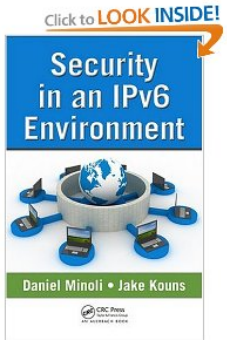
- Be aware that deploying dual stack *doubles* your “attack surface”. Hackers can now attack you via the weaker of your IPv4 or IPv6 connections. Be careful to secure any exposed IPv6 ports as carefully as you do your IPv4 ports.
- Any tunneled service potentially allows a hacker to sneak things by even “deep packet inspection” – make sure your gateway firewall knows how to look inside tunnels, including 6in4 and 4in6.
- Since IPsec will be more widely deployed in IPv6 (with no NAT to break it), you are more likely to encounter encrypted tunnels. Gateway firewalls *cannot* look inside those, or even control traffic by destination port. You may wish to limit encrypted tunnels to known trusted sources and/or terminate them at the gateway (not at an internal node). Firewall rules should be applied *after* traffic exits from any tunnel (esp. an encrypted tunnel) but *before* allowing that traffic into your internal network.
- You can’t defend a network if you don’t understand the technology, so all security professionals should become as familiar with IPv6 as they are with IPv4, and sooner rather than later.

Shameless Recruiting Plug

- Are you a talented security professional?
- Do you already know IPv6, or are willing to learn?
- Would you like to join or help build a security consulting practice (around region, not just Philippines) who focus is in new IPv6 security issues?
- Talk with me during the show.

Useful Links

- My free book (in addition to being posted on www.apnic.net, www.ipv6forum.org, etc). Includes detailed steps for deploying a dual stack testbed network using generic PCs, open source software, and free tunneled service:
  - http://www.secondinternet.org
- IPv6 information and testing (dual stack autoresponders for web, telnet, email, VoIP):
  - http://www.v6address.com
- Recommended books (available worldwide, including Kindle format):
  - “IPv6 Security”; Scott Hogg & Eric Vyncke; Cisco Press; December 2008.
  - “Security in an IPv6 Enviroment”; Daniel Minoli & Jake Kouns; CRC Press; 2009



Thank You

Please visit our website:

www.infoweapons.com

Over IPv4 *and* IPv6, of course.