Recent Advances in IPv6 Security

Fernando Gont

H2HC 2012
São Paulo, Brazil. October 20-21, 2012
About...

- Security researcher and consultant at SI6 Networks
- Have worked on security assessment on communications protocols for:
  - UK NISCC (National Infrastructure Security Co-ordination Centre)
  - UK CPNI (Centre for the Protection of National Infrastructure)
- Active participant at the IETF (Internet Engineering Task Force)
Agenda

- Disclaimer
- Motivation for this presentation
- Recent Advances in IPv6 Security
  - IPv6 Addressing
  - IPv6 Fragmentation & Reassembly
  - IPv6 First Hop Security
  - IPv6 Firewalling
  - Mitigation to some Denial of Service attacks
- Tools
- Conclusions
- Questions and Answers
Motivation for this presentation
Motivation for this presentation

- Sooner or later you will need to deploy IPv6
  - In fact, you have (at least) partially deployed it, already
- IPv6 represents a number of challenges: What can we do about them?

Option #1

Option #2

Option #3

Suicide is always an option
Motivation for this presentation (II)

• We have been doing a fair share of IPv6 security research
  • Identification of problems
  • Proposals to mitigate those problems
• Part of our research has been taken to the IETF
• This talk is about our ongoing work to improve IPv6 security
Advances in IPv6 Addressing
Brief overview

- Main driver for IPv6 deployment
- Employs 128-bit addresses
- Address semantics similar to those of IPv4:
  - Addresses are aggregated into “prefixes”
  - Several address types
  - Several address scopes
- Each interface typically employs more than one address, of different type/scope:
  - One link-local unicast address
  - One or more global unicast addresses
  - etc.
Global Unicast Addresses

- The “Interface ID” is typically 64-bit long
- Can be selected with different criteria:
  - Modified EUI-64 Identifiers
  - Privacy addresses
  - Manually configured
  - As specified by transition/co-existence technologies
IPv6 Addressing
Implications on remote address scanning attacks
IPv6 host scanning attacks

“Thanks to the increased IPv6 address space, IPv6 host scanning attacks are unfeasible. Scanning a /64 would take 500,000,000 years”

– Urban legend

Is the search space for a /64 really $2^{64}$ addresses?
IPv6 addresses in the real world

- Malone measured (*) the address generation policy of hosts and routers in real networks

<table>
<thead>
<tr>
<th>Address type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLAAC</td>
<td>50%</td>
</tr>
<tr>
<td>IPv4-based</td>
<td>20%</td>
</tr>
<tr>
<td>Teredo</td>
<td>10%</td>
</tr>
<tr>
<td>Low-byte</td>
<td>8%</td>
</tr>
<tr>
<td>Privacy</td>
<td>6%</td>
</tr>
<tr>
<td>Wordy</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Others</td>
<td>&lt;1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-byte</td>
<td>70%</td>
</tr>
<tr>
<td>IPv4-based</td>
<td>5%</td>
</tr>
<tr>
<td>SLAAC</td>
<td>1%</td>
</tr>
<tr>
<td>Wordy</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Privacy</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Teredo</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>Others</td>
<td>&lt;1%</td>
</tr>
</tbody>
</table>

IPv6 addresses embedding IEEE IDs

- In practice, the search space is at most $\sim 2^{24}$ bits – **feasible**!
- The low-order 24-bits are not necessarily random:
  - An organization buys a large number of boxes
  - In that case, MAC addresses are usually consecutive
  - Consecutive MAC addresses are generally in use in geographically-close locations
IPv6 addresses embedding IEEE IDs (II)

- Virtualization technologies present an interesting case
- Virtual Box employs OUI 08:00:27 (search space: $\sim 2^{24}$)
- VMWare ESX employs:
  - Automatic MACs: OUI 00:05:59, and next 16 bits copied from the low order 16 bits of the host's IPv4 address (search space: $\sim 2^8$)
  - Manually-configured MACs: OUI 00:50:56 and the rest in the range 0x000000-0x3fffff (search space: $\sim 2^{22}$)
IPv6 addresses embedding IPv4 addr.

- They simply embed an IPv4 address in the IID
  - e.g.: 2000:db8::192.168.0.1
- Search space: same as the IPv4 search space
IPv6 “low-byte” addresses

- The IID is set to all-zeros, except for the last byte
  - e.g.: 2000:db8::1
  - There are other variants
- Search space: usually $2^8$ or $2^{16}$
Industry mitigations for scanning attacks

- Microsoft replaced the MAC-address-based identifiers with (non-standard) randomized IIDs
  - Essentially RFC 4941, but they don't vary over time
- Certainly better than MAC-address-based IIDs, but still not “good enough”
- They mitigate host-scanning, but **not** host tracking – constant IIDs are still present!
Thoughts on remote scanning attacks

- IPv6 host scanning attacks are **feasible**, but typically harder than in IPv4
- They require more “intelligence” on the side of the attacker
- It is **possible** to make them infeasible
- It is likely that many other scanning strategies/techniques will be explored
- More discussion in draft-gont-opsec-ipv6-host-scanning
IPv6 Addressing
Implications on privacy
Problem statement

- Modified EUI-64 IIDs are constant for each interface
- As the host moves, the prefix changes, but the IID doesn't
  - the 64-bit IID results in a super-cookie!
- This introduces a problem not present in IPv4: host-tracking
- Example:
  - In net #1, host configures address: `2001:db8:1::1111:2222:3333:4444`
  - In net #2, host configures address: `2001:db8:2::1111:2222:3333:4444`
  - The IID “1111:2222:3333:4444” leaks out host “identity”.
“Mitigation” to host-tracking

- RFC 4941: privacy/temporary addresses
  - Random IIDs that change over time
  - Generated in addition to traditional SLAAC addresses
  - Traditional addresses used for server-like communications, temporary addresses for client-like communications

- Operational problems:
  - Difficult to manage!
  - ipv6mon is meant to help in that area (see http://www.si6networks/tools)

- Security problems:
  - They mitigate host-tracking only partially
  - They do not mitigate address-scanning attacks
IPv6 addressing
Mitigating scanning and privacy issues
Auto-configuration address/ID types

<table>
<thead>
<tr>
<th>Stable</th>
<th>Temporary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictable</td>
<td>IEEE ID-derived</td>
</tr>
<tr>
<td>Unpredictable</td>
<td>NONE</td>
</tr>
</tbody>
</table>

- We lack stable privacy-enhanced IPv6 addresses
  - Used to replace IEEE ID-derived addresses
  - Pretty much orthogonal to privacy addresses
  - Probably “good enough” in most cases even without RFC 4941
Stable privacy-enhanced addresses

- draft-ietf-6man-stable-privacy-addresses proposes to generate Interface IDs as:
  \[ F(\text{Prefix}, \text{Interface\_Index}, \text{Network\_ID}, \text{Secret\_Key}) \]

- Where:
  - \( F() \) is a PRF (e.g., a hash function)
  - Prefix SLAAC or link-local prefix
  - \( \text{Interface\_Index} \) is the (internal) small number that identifies the interface
  - \( \text{Network\_ID} \) could be e.g. the SSID of a wireless network
  - \( \text{Secret\_Key} \) is unknown to the attacker (and randomly generated by default)
Stable privacy-enhanced addresses (II)

- As a host moves:
  - Prefix and Network_ID change from one network to another
  - But they remain constant within each network
  - F() varies across networks, but remains constant within each network
- This results in addresses that:
  - Are stable within the same subnet
  - Have different Interface-IDs when moving across networks
  - For the most part, they have “the best of both worlds”
- Document already accepted as a 6man wg item
IPv6 Fragmentation and Reassembly
**IPv6 fragmentation**

- IPv6 fragmentation performed only by hosts (never by routers)
- Fragmentation support implemented in “Fragmentation Header”
- Fragmentation Header syntax:

```
<table>
<thead>
<tr>
<th>8 bits</th>
<th>8 bits</th>
<th>13 bits</th>
<th>2b</th>
<th>1b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next Header</td>
<td>Reserved</td>
<td>Fragment Offset</td>
<td>Res</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Fragment Identification

- Security Implications of predictable Fragment IDs well-known from the IPv4 world
  - idle-scanning, DoS attacks, etc.
- Amount of fragmented traffic will probably increase as a result of:
  - Larger addresses
  - DNSSEC
- But no worries, since we learned the lesson from the IPv4 world... – right?
### Fragment ID generation policies

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeBSD 9.0</td>
<td>Randomized</td>
</tr>
<tr>
<td>NetBSD 5.1</td>
<td>Randomized</td>
</tr>
<tr>
<td>OpenBSD-current</td>
<td>Randomized (based on SKIPJACK)</td>
</tr>
<tr>
<td>Linux 3.0.0-15</td>
<td><strong>Predictable</strong> (GC init. to 0, incr. by +1)</td>
</tr>
<tr>
<td>Linux-current</td>
<td>Unpredictable (PDC init. to random value)</td>
</tr>
<tr>
<td>Solaris 10</td>
<td><strong>Predictable</strong> (PDC, init. to 0)</td>
</tr>
<tr>
<td>Windows 7 Home Prem.</td>
<td><strong>Predictable</strong> (GC, init. to 0, incr. by +2)</td>
</tr>
</tbody>
</table>

GC: Global Counter      PDC: Per-Destination Counter

At least Solaris and Linux patched in response to our IETF I-D – more patches expected!
Fixing predictable Fragment IDs

- draft-gont-6man-predictable-fragment-id:
  - Discussed the security implications of predictable Fragment ID
  - Proposes a number of algorithms to generate the Fragment ID
- Ongoing work at the 6man wg
  - Has not yet been adopted by the 6man working group
IPv6 Fragment Reassembly

- Security implications of overlapping fragments well-known (think Ptacek & Newsham, etc.)
- Nonsensical for IPv6, but originally allowed in the specs
- Different implementations allow them, with different results
- RFC 5722 updated the specs, forbidding overlapping fragments
- Most current implementations reflect the updated standard
- See http://blog.si6networks.com
**IPv6 “atomic” fragments**

- ICMPv6 PTB < 1280 triggers inclusion of a FH in all packets to that destination (not actual fragmentation)
- Result: IPv6 atomic fragments (Frag. Offset=0, More Frag.=0)
Issues with IPv6 atomic fragments

• Some implementations mix “atomic fragments” with queued fragments
• Atomic fragments thus become subject of IPv6 fragmentation attacks
• How to leverage this issue:
  • Trigger atomic fragments with ICMPv6 PTB messages
  • Now perform IPv6 fragmentation-based attacks
Mitigating issues with atomic fragments

- draft-ietf-6man-ipv6-atomic-fragments fixes the problem:
  - IPv6 atomic fragments required to be processed as non-fragmented traffic
- Document has passed WGLC
  - Should be published as an RFC this year
## Handling of IPv6 atomic fragments

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Atomic Frag. Support</th>
<th>Improved processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeBSD 8.2</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FreeBSD 9.0</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Linux 3.0.0-15</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetBSD 5.1</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetBSD-current</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>OpenBSD-current</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Solaris 11</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Vista (build 6000)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Windows 7 Home Premium</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

At least NetBSD and OpenBSD patched in response to our IETF I-D – more patches expected!
IPv6 First Hop Security
IPv6 First Hop Security

- Security mechanisms/policies employed/enforced at the first hop (local network)
- Fundamental problem: lack of feature-parity with IPv4
  - arpwatch-like Neighbor Discovery monitoring virtually impossible
  - DHCP-snooping-like RA blocking trivial to circumvent
IPv6 First Hop Security (II)

- Fundamental problem: complexity of traffic to be “processed at layer-2”
- Example:
Bringing “sanity” to ND traffic

- draft-ietf-6man-nd-extension-headers forbids use of fragmentation with Neighbor Discovery
  - It makes ND monitoring feasible
  - Turns out this is vital for SEND (otherwise it could be DoS'ed with fragments)

- Work in progress:
  - Has been adopted as a 6man wg item
  - Should be published as an RFC “shortly”
RA-Guard

- Meant to block RA packets on “unauthorized” switch ports
- Existing implementations trivial to circumvent
- draft-ietf-v6ops-ra-guard-implementation contains:
  - Discussion of RA-Guard evasion techniques
  - Advice to filter RAs, while avoiding false positives
- Can only be evaded with overlapping fragments
  - But most current OSes forbid them
  - And anyway there's nothing we can do about this :-)
- Should be published as an RFC “shortly”.
IPv6 firewalling
Problem statement

- Specs-wise, state-less IPv6 packet filtering is impossible:
  - The IPv6 header chain can span multiple fragments
  - This makes state-less firewalling impossible
First step away from “insanity”

- **draft-ietf-6man-oversized-header-chain** fixes this problem:
  - The entire IPv6 header chain must be contained in the first fragment
  - i.e. packets with header chains that span more than one fragment may be blocked – don't send them!

- Work in progress:
  - Already adopted by the 6man WG
  - Should be published as an RFC “shortly”

- There's an insanely large amount of work to be done in the area of IPv6 firewalling
IPv6 implications on IPv4 networks
VPN leakages

• Typical scenario:
  • You connect to an insecure network
  • You establish a VPN with your home/office
  • Your VPN software does not support IPv6

• Trivial to trigger a VPN leakage
  • Spoof RA's or DHCPv6-server packets, to set the recursive DNS server
  • Simply trigger IPv6 connectivity, such that dual-stacked hosts leak out
  • Even legitimate dual-stacked networks may trigger it

• More details in draft-gont-opsec-vpn-leakages
Tools
IPv6 security tools

- For ages, THC-IPv6 ([http://www.thc.org](http://www.thc.org)) was the only IPv6 security tools publicly available
- We've produced “SI6 Networks IPv6 toolkit”
  - A brand-new security assessment/trouble-shooting toolkit
  - Runs on Linux, *BSD, and Mac OS
- Available at: [http://www.si6networks.com/tools/ipv6toolkit](http://www.si6networks.com/tools/ipv6toolkit)
  - GIT repository at: [https://github.com/fgont/ipv6-toolkit.git](https://github.com/fgont/ipv6-toolkit.git)
SI6 Networks' IPv6 toolkit

- scan6: An IPv6 address scanner
- frag6: Play with IPv6 fragments
- tcp6: Play with IPv6-based TCP segments
- ns6: Play with Neighbor Solicitation messages
- na6: Play with Neighbor Advertisement messages
- rs6: Play with Router Solicitation messages
- ra6: Play with Router Advertisement messages
SI6 Networks' IPv6 toolkit (II)

- rd6: Play with Redirect messages
- icmp6: Play with ICMPv6 error messages
- ni6: Play with Node Information messages
- flow6: Play with the IPv6 Flow Label
- jumbo6: Play with IPv6 Jumbograms
Some conclusions
Some conclusions

- Many IPv4 vulnerabilities have been re-implemented in IPv6
  - We just didn't learn the lesson from IPv4, or,
  - Different people working in IPv6 than working in IPv4, or,
  - The specs could make implementation more straightforward, or,
  - All of the above? :-)  

- Still lots of work to be done in IPv6 security
  - We all know that there is room for improvements
  - We need IPv6, and should work to improve it
Questions?
Thanks!

Fernando Gont
fgont@si6networks.com

IPv6 Hackers mailing-list
http://www.si6networks.com/community/

www.si6networks.com