HoneydV6

A low-interaction IPv6 honeypot

Sven Schindler

Potsdam University
Institute for Computer Science
Operating Systems and Distributed Systems

Reykjavík, July 29, 2013

Outline

Outline

Why do we need IPv6 dark- and honeynets?

- huge IPv6 address space makes brute-force network scanning impossible
- new scanning approaches in the wild?
- attacks aiming at IPv6 design weaknesses
- how to **analyse IPv6 related attacks**?

THC and si6 - IPv6 Attack Toolkits

- IPv6 attack tools like THC toolkit [3] and si6 [8] available
- fragment6 (THC) - duplicate fragments
- fake_router6 (THC) - become the default router
- rsmurf6 (THC) - remote smurf attack tool
- dos-new-ip6 (THC) - block new hosts from joining a network
- scan6 (si6) - intelligent scan approaches

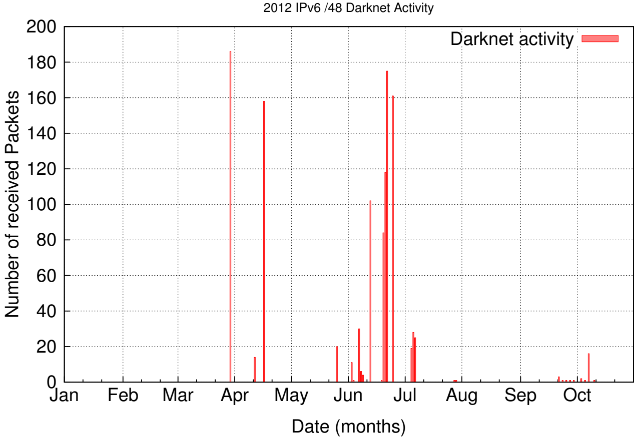
Outline

Why another IPv6 Darknet Experiment?

- /48 experiment from 2006 reported 12 ICMPv6 packets within 16 months [2]
- IPv4 class A darknet in 2004 captured 30,000 packets/second [5]
- 9 days /12 IPv6 darknet experiment received 21,000 non-malicious packets in 2010 [4]
- **started our /48 darknet experiment in March 2012 (Hurricane Electric tunnel)**

Darknet results after 9 months

- **1172 packets received**
- TCP traffic only
- most packets around IPv6 World Launch Day (6.6.2012)

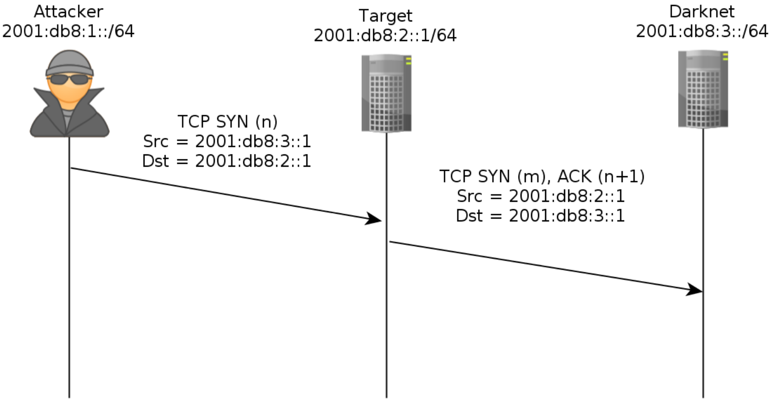


Backscatter traffic

- 1157 packets seem to be **backscatter**
- caused by misconfiguration or spoofed source addresses

Number of packets	Source port
486	auth (113)
327	ssh (22)
186	ircd (6667)
158	http (80)

Backscatter



Some interesting facts about the backscatter traffic

- port 113
  - belongs to Ident protocol (RFC1413)
  - 486 packets from 8 different sources to 457 different destinations
  - most packets contained the **same acknowledgement number**
- port 22
  - 327 packets from 8 different sources targeting 295 destinations
  - again: most packets contained the **same acknowledgement number**
- port 6667
  - 186 packets from the same source
  - again: all packets contained the **same acknowledgement number**
- port 80
  - 158 packets from the same source to different destinations
  - all packets but one with the **same acknowledgement number and target port**

→ **traffic indicates spoofed source addresses**

Darknet summary

- DoS-attacks observed?
- no connection attempts
- threat level in IPv6 network still low compared to IPv4
- attackers interest in IPv6 networks is raising

Outline

What is a virtual honeypot and why do we need it?

Honeypot definition

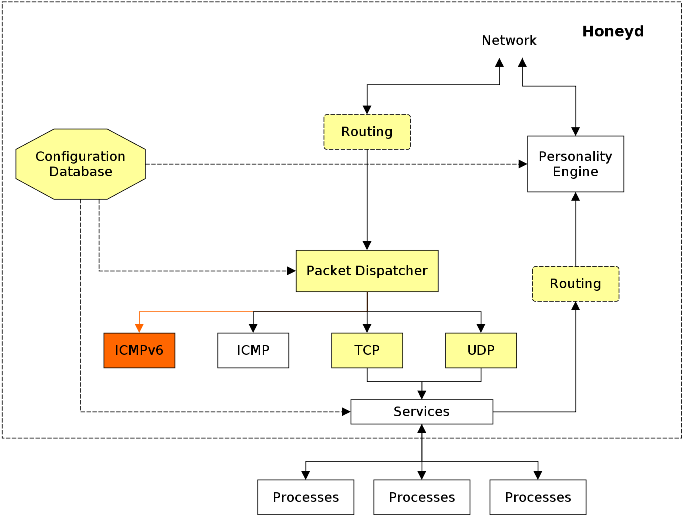
A virtual honeypot is a security device with the only purpose of attracting attackers, so that their attacks can be analysed. This can be something like a computer or even a mobile phone. The system itself has no real production value [7].

- provides level of interaction
- classification based on level of interaction
  - high-interaction honeypot drawback: hardware requirements
  - low-interaction honeypots to **simulate multiple hosts on single machine**
- Dionea is able to simulate a single IPv6 connected machine [1]

Honeyd

- open source low-interaction honeypot by Niels Provos
- custom network stack
- **simulate entire networks**
- supports OS fingerprinting
- provides framework for service scripts
- latest release v 1.5c does not support IPv6
- Tiny Honeypot, SCADA HoneyNet Project based on Honeyd

Honeyd architecture[6]



Requirements

- allow to define virtual IPv6 hosts
- create hierarchical IPv6 networks
- allow nmap, ping6 and traceroute6 to find virtual hosts
- log IPv6 communication between attacker and honeypot
- keep IPv4 support

Adapting the configuration of virtual hosts

Example IPv4 configuration

```
create windows
set windows default tcp action reset
add windows tcp port 21 "scripts/ftp.sh"

set windows ethernet "aa:00:04:78:98:76"

bind 192.168.1.5 windows
bind 192.168.1.6 windows
```

- configuration parser modified to accept IPv6 addresses
- IPv6 and IPv4 templates managed in splay tree

Implementing the Neighbor Discovery Protocol and ICMPv6

- IPv6 utilizes **NDP** instead of ARP
- send and process neighbor solicitations
- send router solicitations
- process router advertisements
- **ICMPv6** echo request/reply
- ICMPv6 Time Exceeded and Destination Unreachable

Modifying packet processing

- new IPv6 dispatcher
- updated routing engine to **simulate networks**
- **extension header** processing
- fragmentation logging of length and offset
- TCP and UDP functionality updated

How to find an IPv6 honeypot?

- linear IPv6 address scan is impossible
- attacker needs to find hosts
- **dynamically create new virtual hosts on demand**
- all connection attempts logged
- observe new scan approaches

Configuration of random IPv6 request processing

Configuration

```
create randomdefault
set randomdefault default tcp action reset
add randomdefault tcp port 21 "scripts/ftp.sh"
add randomdefault tcp port 80 "scripts/web.sh"
set randomdefault ethernet "aa:00:04:78:98:78"

randomipv6 0.5 randomdefault 256

randomexclude 2001:db8::1
randomexclude 2001:db8::2
randomexclude 2001:db8::3
```

Performance tests - HTTP get request measurements

- generated log file containing 20.000 HTTP GET request from different source addresses
- 600 requests per second
- honeyd configured to simulate single host (IPv4 and IPv6 connected)
- web.sh script on port 80

1.5c (IPv4)	V6 (IPv4)	V6 (IPv6)
212.57	214.00	205.75

Table: Comparison of the number of HTTP GET requests per second that Honeyd 1.5c and HoneydV6 is able to handle without any packet loss.

Outline

Conclusion and Future work

- HoneydV6 is the **first low-interaction honeypot which can simulate entire IPv6 networks on a single host**
- may be used to add IPv6 support for low-interaction honeypots based on honeyd
- new protocols implemented (NDP, ICMPv6)
- random IPv6 request processing helps to understand new scan approaches
- OS fingerprinting and tunnel support not yet implemented
- working on shellcode detection engine
- currently running at a major German hosting company
- HoneydV6 source code available on www.idsv6.de
- Questions?

Conclusion and Future work

- HoneydV6 is the **first low-interaction honeypot which can simulate entire IPv6 networks on a single host**
- may be used to add IPv6 support for low-interaction honeypots based on honeyd
- new protocols implemented (NDP, ICMPv6)
- random IPv6 request processing helps to understand new scan approaches
- OS fingerprinting and tunnel support not yet implemented
- working on shellcode detection engine
- currently running at a major German hosting company
- HoneydV6 source code available on www.idsv6.de
- Questions?

Conclusion and Future work

- HoneydV6 is the **first low-interaction honeypot which can simulate entire IPv6 networks on a single host**
- may be used to add IPv6 support for low-interaction honeypots based on honeyd
- new protocols implemented (NDP, ICMPv6)
- random IPv6 request processing helps to understand new scan approaches
- OS fingerprinting and tunnel support not yet implemented
- working on shellcode detection engine
- currently running at a major German hosting company
- HoneydV6 source code available on www.idsv6.de
- Questions?

References

[1] Dionaea.
dionaea catches bugs.
http://dionaea.carnivore.it/, nd.

[2] Matthew Ford, Jonathan Stevens, and John Ronan.
Initial Results from an IPv6 Darknet.
In *ICISP '06: Proceedings of the International Conference on Internet Surveillance and Protection*, page 13, Washington, DC, USA, 2006. IEEE Computer Society.

[3] Marc Heuse.
THC IPv6 attack tool kit.
http://www.thc.org/thc-ipv6/, nd.

[4] Geoff Huston.
Background Radiation in IPv6.
https://labs.ripe.net/Members/mirjam/background-radiation-in-ipv6, October 2010.

[5] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson.
Characteristics of internet background radiation.
In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 27–40, New York, NY, USA, 2004. ACM.

[6] Niels Provos and Thorsten Holz.
Virtual Honeypots - From Botnet Tracking to Intrusion Detection.
Addison-Wesley, 2008.

[7] Christian Seifert, Ian Welch, and Peter Komisarczuk.
Taxonomy of honeypots.
Technical report, Victoria University of Wellington, Wellington, 2006.

[8] SI6 Networks.
SI6 Networks' IPv6 Toolkit - A security assessment and troubleshooting tool for the IPv6 protocols.
http://www.si6networks.com/tools/ipv6toolkit, 2012.

Pitfalls

scope IDs in link-local addresses

```
static void addr_remove_scope_id(struct addr* ip6)
    {
  if (ip6->addr_data8[0]==0xfe && ip6->addr_data8
      [1]==0x80) {
    /* delete scope id */
    ip6->addr_data8[2]=0;
    ip6->addr_data8[3]=0;
  }
}
```

Pitfalls

use of dynamic arrays

```
struct interface {
  TAILQ_ENTRY(interface) next;

  struct intf_entry if_ent;
  int if_addrbits;
  struct event if_recvev;
  pcap_t *if_pcap;
  eth_t *if_eth;
  int if_dloff;

  char if_filter[1024];
};
```

Performance tests - throughput measurements

- PRIMERGY TX200 S5 Server with an Intel Xeon processor 5500 series and 4096 MB of RAM running Ubuntu 12.04
- benchmark client was installed on a Lenovo ThinkPad L520 with an Intel i5-2450M CPU and 4096 MB of RAM
- computers connected via Brocade FWS648G FastIron switch using Gigabit Ethernet

Filesize	1.5c (IPv4)	V6 (IPv4)	V6 (IPv6)
50 MB	15.98 s	16.19 s	16.33 s
100 MB	31.85 s	31.94 s	32.36 s

Table: Comparison of transmission time in seconds between the original Honeyd version 1.5c and HoneydV6 - median values of 5 test runs