IPv6 Intrusion Detection

Fähigkeiten und Grenzen von IPv6-basierten Angriffen
Erkennung von IPv6-spezifischen Angriffen mittels Snort-Preprocessor

Thomas Scheffler
scheffler@beuth-hochschule.de

BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Bundesministerium
für Bildung
und Forschung

12. Juni 2013

Inhaltsverzeichnis

IPv6 Security Issues

IDS/Snort

Snort IPv6 Plugin

Zusammenfassung

Stand $\sim$ 1994

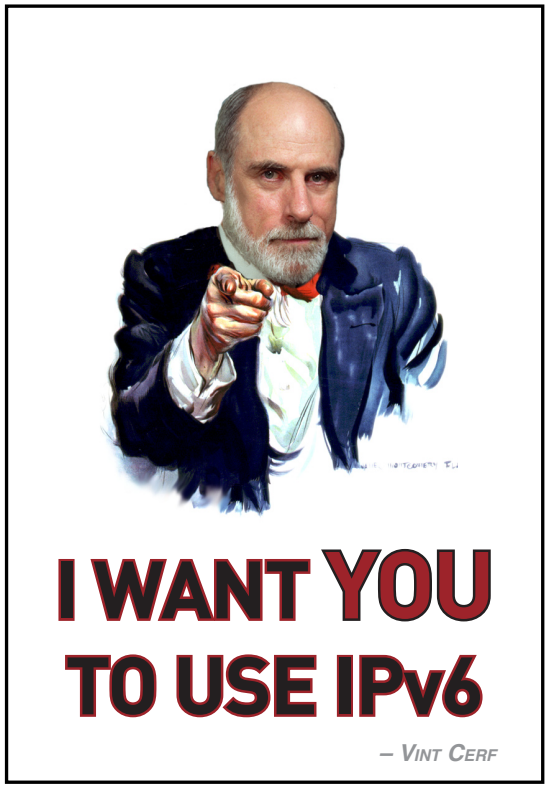
IPv4 Internet:

- Forschungs- und Uni-Netze
- bekannte Design- & Implementierungs-Fehler
- wenig Erfahrung mit Protokoll-Sicherheit
- wenig Druck zur Verbesserung

Stand $\sim$ 2011

IPv6 Internet:

- Forschungs- und Uni-Netze
- bekannte Design- & Implementierungs-Fehler
- wenig Erfahrung mit Protokoll-Sicherheit
- wenig Druck zur Verbesserung



IPv6 Sicherheit

- IPv6 Protokolldesign basiert auf RFCs von 1995/1998

$\Rightarrow$ 15 Jahre IPv4-Sicherheits-Erfahrung sind nachzuholen

IPv6 Sicherheit

- IPv6 Protokolldesign basiert auf RFCs von 1995/1998

$\Rightarrow$ 15 Jahre IPv4-Sicherheits-Erfahrung sind nachzuholen

- Viele ergänzende RFCs und Internet Drafts (IPsec, SEND, RH0 deprecation ...)
- Wenige Implementierungen / geringer Reifegrad
- Kaum genutzt in Endgeräten und für aktuelle Deployments

Angriffe auf IPv6

Das übliche:

- Wertebereiche für Felder
- Fragmentierung
- Denial of Service
- Portscans
- Fehler in Anwendungsschicht

Angriffe auf IPv6

Das übliche:

- Wertebereiche für Felder
- Fragmentierung
- Denial of Service
- Portscans
- Fehler in Anwendungsschicht

IPv6-spezifisch:

- Variable Header
- Multicast
- Routing
- v4/v6-Transition
- Autokonfiguration
- Neighbor Discovery

Angriffe auf IPv6

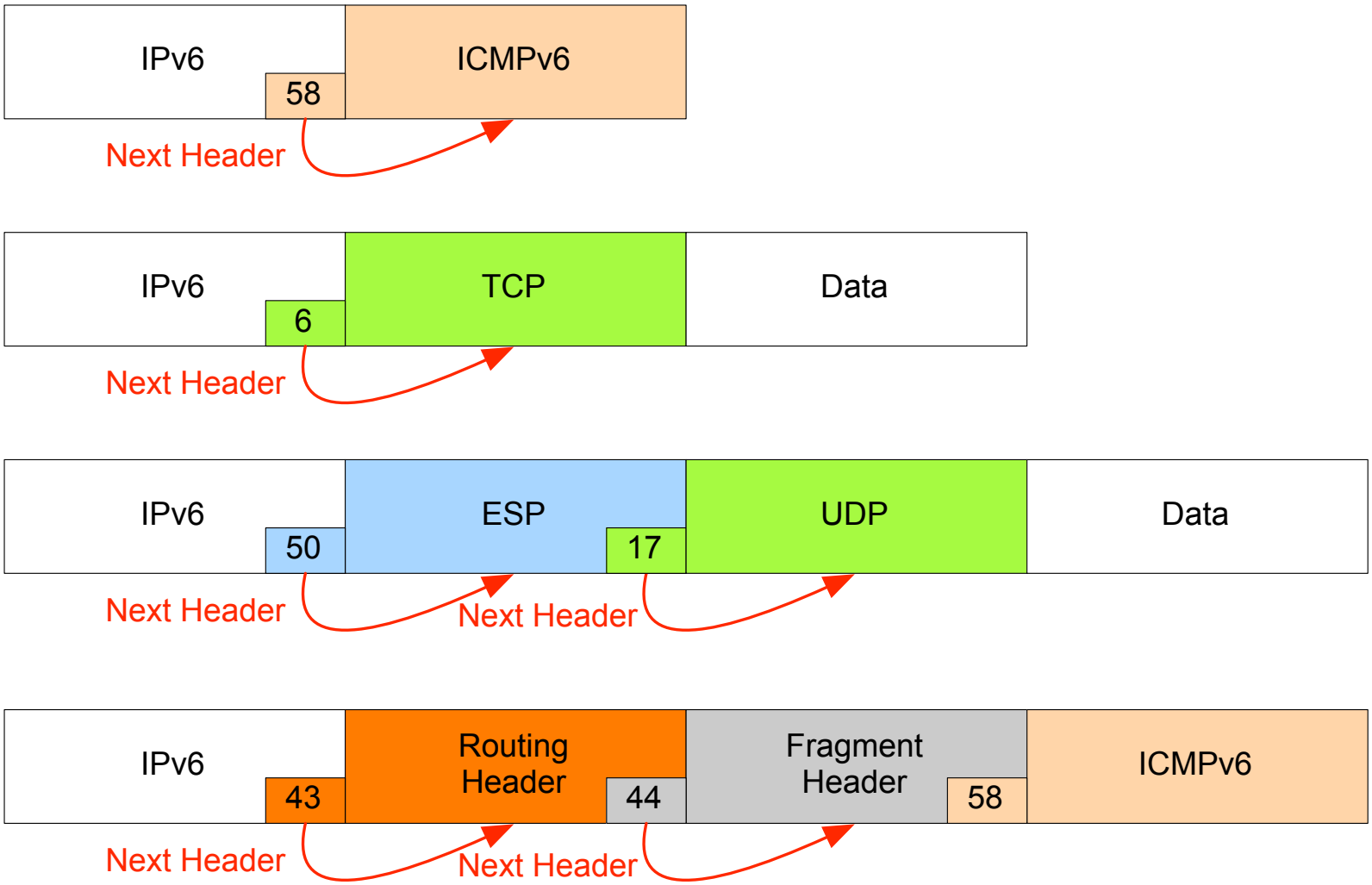
Das übliche:

- Wertebereiche für Felder
- Fragmentierung
- Denial of Service
- Portscans
- Fehler in Anwendungsschicht

IPv6-spezifisch:

- Variable Header
- Multicast
- Routing
- v4/v6-Transition
- **Autokonfiguration**
- **Neighbor Discovery**

Variable Kette von Erweiterungsheadern



Routing und Transition

Routing:

- Umfangreiche Spezifikation
- RH0 (source routing) deprecated
- RH2 für Mobile_IPv6 nötig

Transition:

- Dual-Stack: zwei Paketfilter
- Tunnelling: leichte Filter-Umgehung
- Automatic Tunnel Routing Loops

Relevanz von IPv6-basierten Angriffen

Lokale Angriffe (die große Mehrzahl bekannter Angriffe):

- Adress-Spoofing (Router, Host)
- Denial of Service / Redirect
- Reconnaisance

Relevanz von IPv6-basierten Angriffen

Lokale Angriffe (die große Mehrzahl bekannter Angriffe):

- Adress-Spoofing (Router, Host)
- Denial of Service / Redirect
- Reconnaisance

Abwehrmöglichkeiten:

- Secure Neighbor Discovery (SEND) - wenig erprobte Implementierungen, Ressource Exhaustion möglich
- Filterung von ICMPv6 Nachrichten auf Switches (RA-Guard, Prefix-Guard) - HW-Support erforderlich
- Authentisierung des Link-Layers (z.B. IEEE 802.1X), nicht in allen Netzen einsetzbar, schützt nicht a-priori vor Angriffen
- Erkennung / Filterung von ICMPv6 Nachrichten per IDS/IPS

Relevanz von IPv6-basierten Angriffen

Entfernte Angriffe (größeres Bedrohungspotential):

- Source Routing
- v4/v6-Transition
- **Exhaustion des Neighbor Caches**
- **Extension Header Chaining** (Resource Exhaustion, Firewall Traversing)

Relevanz von IPv6-basierten Angriffen

Entfernte Angriffe (größeres Bedrohungspotential):

- Source Routing
- v4/v6-Transition
- **Exhaustion des Neighbor Caches**
- **Extension Header Chaining** (Resource Exhaustion, Firewall Traversing)

Abwehrmöglichkeiten:

- Filterung / Rate-Limiting von IPv6 Nachrichten - derzeitige Firewallregeln oft zu grob-granular
- Best Practices: Drop RH-0, /127 Subnetze für P2P-Links (RFC 6164)
- Erkennung / Filterung von ICMPv6 Nachrichten per IDS/IPS

Autoconfiguration und Neighbor Discovery

Design-Prämisse: sicheres und vertrauenswürdiges LAN

Simple Denial of Service:

1. Host Alice startet *Duplicate Address Detection*:
   „Benutzt jemand die Adresse *X*?“
2. Host Eve antwortet „Ich benutze Adresse *X*.“
3. `goto 1`

Routing/Man in the Middle:

1. Host Eve sendet *ICMPv6 Redirect*:
   „Hier Router Bob, für *google.com* bitte Router Eve benutzen.“

Angriffs-Sammlung: THC Toolkit

Tools/Angriffe/Tests für:

- Autoconfiguration Denial-of-Service
- Neighbor Cache Poisoning
- Routing/Redirect
- Flood-Attacks
- Multicast Listener Discovery
- DHCPv6
- `implementation6`
- ...

Zielsystem System: Snort 2.9.x

- Weit verbreitetes Open Source NIDS
- Filter-/inline-Modus (*Intrusion Prevention System*)
- Plugin APIs
- Dekoder für gängige Tunnelprotokolle
- Offizieller IPv6 Support mit Snort 2.8 (September 2007)
- ab Snort 2.8.4. werden alle Anwendungs-Präprozessoren unterstützt

IPv6 Support

im Prinzip ja, aber ...

Alle relevanten IDS haben IPv6-Support.

IPv6 Support

im Prinzip ja, aber ...

Alle relevanten IDS haben IPv6-Support.

Aber was heißt das?

- Fragment-Reassemblierung
- TCP & UDP Dekodierung
- Decoder-Warnung bei groben Protokollfehlern

IPv6 Support

im Prinzip ja, aber ...

Alle relevanten IDS haben IPv6-Support.

Aber was heißt das?

- Fragment-Reassemblierung
- TCP & UDP Dekodierung
- Decoder-Warnung bei groben Protokollfehlern

(Noch?) nicht:

- Neue Protokollfeatures (Routing Header, Jumbograms)
- Unterstützung aller Regeloptionen (`fragbits`)
- Erkennung IPv6-spezifischer Angriffe (ICMPv6/neighbor discovery)

Snort IPv6 Signaturen

Die meisten Regeloptionen und -parameter sind protokollagnostisch.

```
alert ip icmp any -> any any                                    \
    (msg:"IPv6 ICMP Echo-Request?"; itype:128;                  \
    classtype:icmp-event; sid:2000001; rev:1;)
```

Snort IPv6 Signaturen

Die meisten Regeloptionen und -parameter sind protokollagnostisch.

```
alert ip icmp any -> any any                                    \
    (msg:"IPv6 ICMP Echo-Request?"; itype:128;                  \
    classtype:icmp-event; sid:2000001; rev:1;)
```

- Keine Schlüsselwörter für neue IPv6-Felder.
- Kein Weg IPv6-only Regeln zu schreiben.

- Guter Ansatz für konsistente IPv4/IPv6 Application Rules, aber inadequate für die Erkennung von Angriffen auf den Protocol Layer.

Snort IPv6 Signaturen

Die meisten Regeloptionen und -parameter sind protokollagnostisch.

```
alert ip icmp any -> any any                                    \
    (msg:"IPv6 ICMP Echo-Request?"; itype:128;                  \
    classtype:icmp-event; sid:2000001; rev:1;)
```

- Keine Schlüsselwörter für neue IPv6-Felder.
- Kein Weg IPv6-only Regeln zu schreiben.

- Guter Ansatz für konsistente IPv4/IPv6 Application Rules, aber inadequate für die Erkennung von Angriffen auf den Protocol Layer.

$\Rightarrow$ Entwicklung eines eigenen IPv6-Plugins

Snort IPv6 Plugin (IPv6 Präprozessor)

Grundsätzliche Funktionsweise:

- Liest alle ICMPv6-Nachrichten im LAN
- Verfolgt Netz-Zustand, d. h. (MAC, IP) von:
  - On-link Routern
  - On-link Hosts
  - laufenden DADs
- Alert bei neuen Hosts, Router-Änderungen u. ä.

Konfiguration

in `snort.conf`

```
preprocessor ipv6:                                  \
    net_prefix 2001:0db8:1::/64                     \
    router_mac 00:16:76:07:bc:92                    \
    host_mac   ...                                  \
    max_unconfirmed 32768                           \
    max_routers 8                                   \
    expire_run 20                                   \
    keep_state 180
```

IPv6 Checks

SID	Message
1	RA from new router
2	RA from non-router MAC address
3	RA prefix changed
4	RA flags changed
5	RA for non-local net prefix
6	RA with lifetime 0
7	new DAD started
8	new host in network
9	new host with non-allowed MAC address
10	DAD with collision
11	DAD with spoofed collision
12	mismatch in MAC and NDP source linkaddress option
13	ipv6: extension header has only padding options (evasion?)
14	ipv6: option lengths != ext length

IPv6 Regeloptionen für Snort

Ziel:

- Felder des IPv6-Headers für Signaturen zugänglich machen
- Basis-Header, Erweiterungs-Header, Neighbor Discovery-Optionen

IPv6 Regeloptionen für Snort

Ziel:

- Felder des IPv6-Headers für Signaturen zugänglich machen
- Basis-Header, Erweiterungs-Header, Neighbor Discovery-Optionen

Funktionsweise:

- Definition von Callbacks für Options-Schlüsselwörter
- Aufruf mit Regel-Parametern und aktuell empfangenen Paket
- Rückgabe `match`/`no_match`

Regeloptionen des IPv6-Plugins

`ipv`	IP version
`ip6_tclass`	Traffic Class
`ip6_flow`	Flow Label
`ip6_exthdr`	Extension Header
`ip6_extnum`	Number of Extension Headers
`ip6_option`	Destination-/HbH-Option
`ip6_optval`	Destination-/HbH-Option Value
`ip6_rh`	Routing Header
`icmp6_nd`	Neighbor Discovery (bool)
`icmp6_nd_option`	Neighbor Discovery Option

Anwendung der IPv6 Regeloptionen

```
alert icmp any any -> any any (itype:8;   ipv: 4; \
  msg:"ICMPv4 PING in v4 pkt"; sid:1000000; rev:1;)
alert icmp any any -> any any (itype:8;   ipv: 6; \
  msg:"ICMPv4 PING in v6 pkt"; sid:1000001; rev:1;)

alert icmp any any -> any any (itype:128; ipv: 4; \
  msg:"ICMPv6 PING in v4 pkt"; sid:1000002; rev:1;)
alert icmp any any -> any any (itype:128; ipv: 6; \
  msg:"ICMPv6 PING in v6 pkt"; sid:1000003; rev:1;)

alert ip  any any -> any any (ip6_rh: !2;          \
  msg:"invalid routing hdr"; sid:1000004; rev:1;)
```

Plugin Performance

- Zustandslose Checks sind schnell:
  Plugin liest `struct SFSnortPacket`
- Zustände verfolgen kostet Zeit und Speicher:
  $\Rightarrow$ DoS Gefahr, es werden daher Limits gesetzt (anpassbar)

Plugin Performance

- Zustandslose Checks sind schnell:
  Plugin liest `struct SFSnortPacket`
- Zustände verfolgen kostet Zeit und Speicher:
  $\Rightarrow$ DoS Gefahr, es werden daher Limits gesetzt (anpassbar)

$\Rightarrow$ ähnliche Performance wie andere Plugins (SSL, SMTP, ...)

$\Rightarrow$ Snort-Dekoder ist hauptsächliches Bottleneck

Zusammenfassung

Snort-IPv6-Plugin:

- Als dynamische Bibliothek installierbar (kann zu einer existierenden Snort Installation hinzugefügt werden)
- Basis für neue IPv6-spezifische Signaturen
- Snort & IPv6-Plugin erkennen THC Angriffe
- Das fundamentale Problem bleibt ungelöst: unsicheres Ethernet

Zusammenfassung

Snort-IPv6-Plugin:

- Als dynamische Bibliothek installierbar (kann zu einer existierenden Snort Installation hinzugefügt werden)
- Basis für neue IPv6-spezifische Signaturen
- Snort & IPv6-Plugin erkennen THC Angriffe
- Das fundamentale Problem bleibt ungelöst: unsicheres Ethernet

$\Rightarrow$ Plugin wurde durch das Projekt veröffentlicht und findet Beachtung (IX-Artikel)

Kontakt

Email: scheffler@beuth-hochschule.de