BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

IPv6 Intrusion Detection mit Snort

Thomas Scheffler / Edurne Izaguirre
Frankfurt/Main, 20. Mai 2010

Übersicht

- Kurzvorstellung Snort
- Installation eine IPv6-fähigen Snort Systems
- Testbed Architecture
- Security and Possible Attacks in IPv6
- IPv6 Regeln
- Was funktioniert? Was funktioniert nicht?
- Fazit

Intrusion Detection System

- Intrusion Detection System IDS
  - Erkennen von Angriffen auf Computernetze, sowie unerwünschtem und potentiell gefährlichem Netznutzungsverhalten
    - Signaturbasiert
      - Beschreibung von Mustern zur Erkennung einer oder mehrerer Angriffsarten
    - Anomalieerkennung
      - Erkennung von ‘abnormalen’ Nutzungsverhalten auf Basis eines Models
    - Kombination beider Verfahren möglich
  - IDS + automatisierte Abwehrmaßnahmen = IPS (Intrusion Prevention Systeme)

IPv6 Intrusion Detection mit Snort

Was ist Snort?

Intrusion Detection System

- Firewall vs. IDS
  - IDS typischerweise nicht ‘intrusive’
  - Anaomalieerkennung für große/schnelle Netze möglich durch Paket-Sampling
  - IDS testen typischerweise einige tausend Signaturen goes more deeply into the packets and detects anomalies
  - Firewall just blocks (or not) traffic
- Network-based IDS, Host-based IDS, Distributed IDS

Was ist Snort

- **Snort**
  - Bekanntes und leistungsfähiges Open-Source Intrusion Detection und Preventionsystem (IDS/IPS)
  - Gestartet als ‘Cross-platform’ Sniffer 1998 von Martin Roesch
  - Signatur-basierte IDS
  - Januar 2001 Roesch gründet Sourcefire, Inc. um Snort auf einer kommerziellen Basis zu stellen
  - Snort 2.0 – neue Detektion Engine für Gigabit-Netzwerke (2002)
  - Experimenteller IPv6-Support (Dezember 2002)
  - Offizieller IPv6 Support mit Snort 2.8 (September 2007)
    - Seit Snort 2.8.4. werden alle Application Preprozessor unterstützt

SNORT

- **Warum SNORT?**
  - Open Source
  - Works with rules which can be customized
  - Widely used and well documented
  - It supossedly implements IPv6 support
  - Snort has proved to be stable, with no crashes, really fast and robustness
- **Definition and components**
  - Signature-based
  - Sniffer-decoder, preprocessor, detection engine, logging and alerting and output
  - Dynamic preprocessor

Components in Snort

- **Packet Decoder**
  - Libpcap: external packet capturing library
  - The packet enters the decoder depending on the link layer from it has been read
- **Preprocessors**
  - Examine packets for suspicious anomalies
  - Modify packets, that is, normalize the traffic so that the detection engine can work with them.
- **Detection Engine**
  - Evaluates a packet against all the rules included in the Snort configuration
  - A huge amount of rules => Group of rules
- **Logging, alerting and output modules.**
  - Unified logging
    - Clear separation of privileges
    - Possible to be sniffing and communicating the info same time

Installation eines IPv6-fähigen Snort-Systems

- **Installation**
  - Konfigurationsoptionen: IPv6 Support
  - ```
    ./configure --enable-ipv6 --enable-decoder-
    preprocessor-rules
    ```

```
$ snort --version

  ,,_     -*> Snort! <*-
 o"  )~   Version 2.8.5.3 IPv6 (Build 124)
  ''''    By Martin Roesch & The Snort Team:
          http://www.snort.org/snort/snort-team

          Copyright (C) 1998-2009 Sourcefire, Inc., et al.
          Using PCRE version: 6.6 06-Feb-2006
```
```

Installation

- **Installation**
  - Anpassung des Snort Systems (snort.conf):
  - Output
    - Datenbank (MySQL, …)
    - Barnyard
  - Variablen
    - HomeNet: IPv4 und IPv6
    - ExternalNet: IPv4 und IPv6

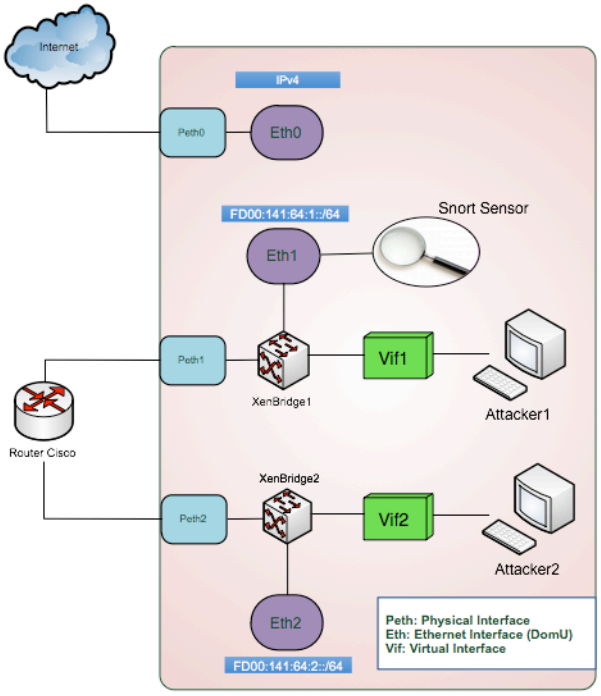
```
HOME_NET [2001::0/64,10.10.10.10]
```

  - Anpassung der Regelbasis

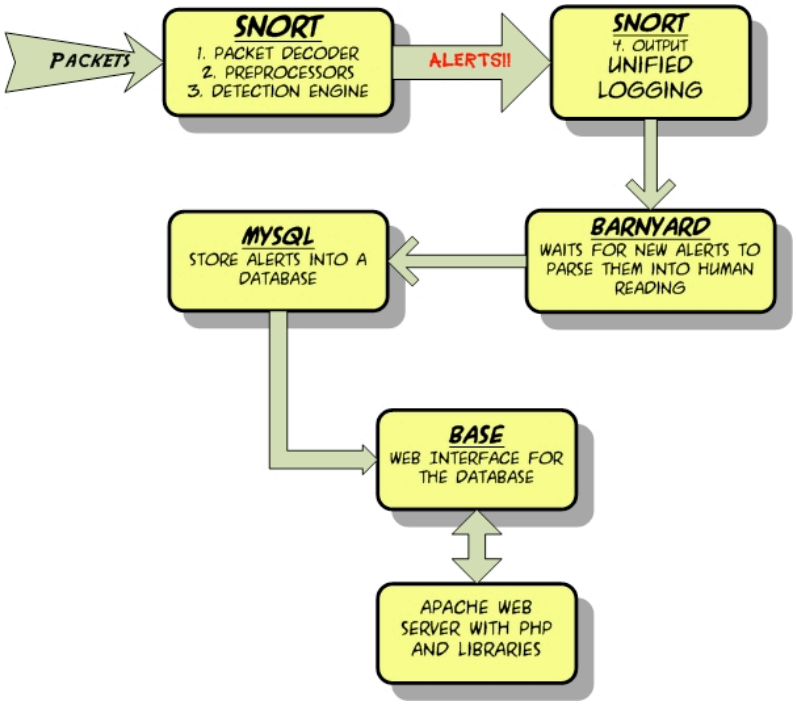
Testbed-Architektur

Virtualisiertes IPv6 Testbed

- Xen-basiert
- Snort in Dom0 installiert
  - flexible Anordnung der Sensoren möglich
- IPv6 Routing über Cisco2600
- Attacker1
  - CentOS Gast
- Attacker2
  - CentOS Gast



Was funktioniert?



Kleines 1x1 der Snort-Regeln

Eine kurze Übersicht der Snort Rules Language

- Regeln bestehen aus zwei Sektionen: Header und Options
- **HEADER:**
  - Festlegen der Aktion: alert, log (drop – im Inline-Modus)
  - Protokoll: IP, TCP, UDP and ICMP (keine IPv6 Spezifika)
  - Quell-IP Adresse
  - Ziel-IP Adresse
    - ✓ 192.168.1.1/24
    - ✓ 192.168.1.1/24,10.0.1.0
  - Quell-Ports
  - Ziel-Ports
    - Einzelne Ports: 21
    - Festlegen von Bereichen: 1:1024


```
alert ip any any -> any any (msg:"Any IP Packet"; sid:
```

Snort Regeln: Options

- Werden durch ein Semikolon abgeschlossen(;), einschließlich der letzten Option
- Der Doppelpunkt(:) kennzeichnet Option-Schlüsselworte

- Drei Hauptkategorien:
  - General, inklusive Meta-data und verschiedener Rule Optionen
  - Payload
  - Non-payload

Snort Rules: General Options

- **msg**
- **reference**
- **sid**
  - Great importance
  - Identifying uniquely the Snort rules

Less than 100	Reserved for future use
100 to 1,000,000	For the rules from www.snort.org rulesets.
Greater than 1,000,000	Use for local rules

  - Along with **rev**
- **classtype**
- **priority**
- **tag**
  - Log additional packets when a packet has triggered

Snort Rules: Payload Options

- **content**
  - Looks for a specific pattern in the packet payload and in case it matches, it triggers the rule
- **nocase**
- **depth**
  - Tells Snort to look for the specific pattern within the first X bytes
- **offset**
  - From which byte Snort should start looking for the pattern
- **distance**
  - How many bytes Snort should ignore before searching for the specific pattern after the end of the previous pattern match
- **within**
  - The pattern must be within these bytes


```
alert tcp any any -> any 23 (content: "administrator"; nocase; depth:10;)
```

Snort Regeln: NON-Payload Options

- **itype**
  - To check for a specific ICMP type value
- **icode**
  - To check for a specific ICMP code value

What is working in IPv6?

What is working in IPv6?

- **In Snort Rules**
  - *ip_proto* keywords (icmp, tcp, ip, udp) make no difference between the two protocols
  - No extra keywords for IPv6 new header fields
  - Difference in defining variables with addresses:
    - ✓ IPv4 => var 192.168.1.10/32
    - ✓ IPv6 => ipvar ff02::00

```
alert icmp 192.68.1.2 1567 -> any any (msg:"PING"; itype:8; sid
234567892;rev:1;)

alert icmp ff02::02 1567 -> any any (msg: "PINGv6";  itype:128; sid:
23456792; rev:1;)
```

What is working in IPv6?

- **SNORT**
  - Not all the preprocessors are supported in this version
  - Does not yet support the reassembly of IPv6 packets and therefore these are treated as individual, unfragmented packets
- **BASE**
  - Still working for IPv6 support
- **MySQL**
  - Handles the data for IPv4 addresses in with special data structures => IPv6 addresses have a different structure
  - Working on it

Szenarien

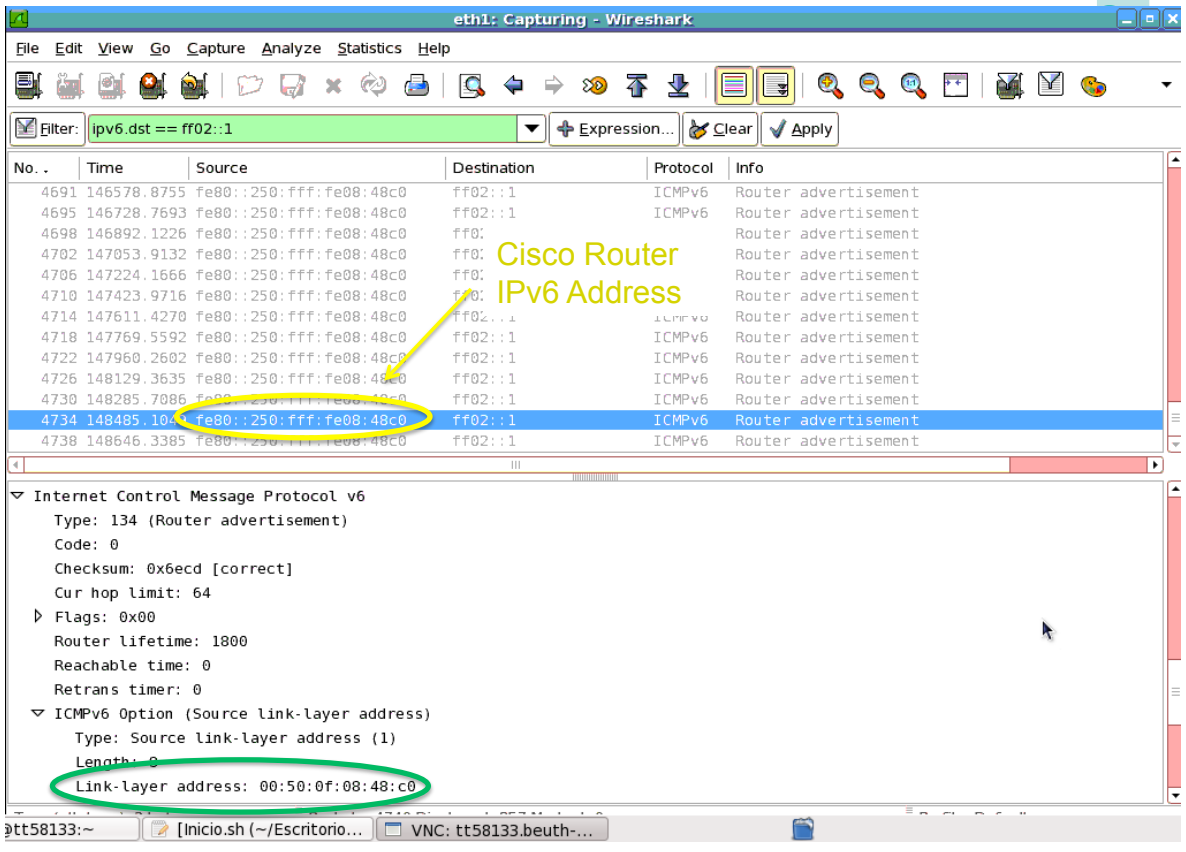
Mögliche Erkennungsszenarien mit Snort

1. Erkennung von Routing Headern
2. Erkennung eines Rough Routers
3. Erkennung von Portscans
4. Erkennung von unerwünschten Inhalten

Erkennung von Routing Headern

- Snort-Regeln arbeiten auf L4-Ebene
- Erkennung des Next-Headers möglich (aber nicht bei geschachtelten Headern)
- Kein direkter Zugriff auf Datenstrukturen im Header

A Possible Solution for Fake Routers



A Possible Detection of Rough Routers

- **Write our own rule:**

1. Define the variables:
   ipvar CISCO_ROUTER fe80:250:fff:fe08:48c0
   ipvar ALL_NODES FF02::1
2. Define which information we are going to use
3. Define the Options

```
alert icmp $CISCO_ROUTER any ->  $ALL_NODES any
(msg:"ROUTER ADVERTISEMENT"; icode:0; itype:134;
content: !"|00 50 0f 08 48 c0|"; offset:14; depth:20;
content: !"|fc 00 01 41 00 64 00 01 00 00 00 00 00 00
00 00|" ; distance:24; sid:23456790; rev:2;)
```


IPv6 Intrusion Detection mit Snort

Wie testen wir das IPv6 IDS?

Wie testen wir das IPv6 IDS?

- **Tools:**
  - Scapy
    - Python-basiertes Paketcrafting-Tool
    - http://www.secdev.org/projects/scapy/
  - Bekannte Test- und Angriffswerkzeuge
    - Nmap
    - ~~Nessus~~
  - …

Fazit

Fazit

- ⇨
- Future approach
  - Iptables and firewall (SnortSam)
  - Write our own preprocessor

Literatur und Verweise