

# Intrusion Detection and IPv6

Arrigo Triulzi

*arrigo@northsea.sevensesas.org*

The SANS Institute

28<sup>th</sup> April 2003

The author can be contacted at:

Arrigo Triulzi

25, Rue de Livron

1217 Meyrin

Switzerland

Telephone: +44 7956 963 288 (world-wide GSM)

E-mail: *arrigo@northsea.sevensesas.org*

# Introduction

- ◆ A short history of Network Intrusion Detection Systems (NIDS)
- ◆ A short history of IPv6
- ◆ Moving from IPv4 to IPv6
- ◆ New directions in NIDS
- ◆ IPv6 and NIDS

28/04/2003

2

This talk will concentrate on the issues related to the deployment of NIDS on IPv6 networks and in particular on the challenges which this transition will pose.

We will start with a short history which will detail some key steps in the evolution of NIDS.

We shall then describe how IPv4 compares to IPv6 and finally touch on the new directions in NIDS.

These new directions shall then be explained in terms of IPv6 deployment.

## Historical background (NIDS)

- ◆ 1988: Network Security Monitor
  - Todd Heberlein at UC Davis and LLNL
- ◆ 1991: Network Intrusion Detection
  - Evolution of NSM
  - Widespread use in US Military
- ◆ 1996: Shadow
  - Northcutt et al.
- ◆ now: Everyone!
  - ISS, Snort, NFR, Dragon, etc.

28/04/2003

3

Historically, the very first tool was (and still often is), tcpdump by LLNL.

This evolved into NSM (Heberlein at UC Davis under contract with LLNL) and then into NID.

A substantial leap took place with Shadow in 1996 (Northcutt et al.) which was then rapidly followed by commercial and free NIDS.

## Historical Background (IPv6)

- ◆ 1990: RFC1550, "request for ideas"
  - IPng: IP "New Generation"
- ◆ 1995: RFC1883, first version
  - Now called IPv6
- ◆ Who uses it?
  - Japan (WIDE initiative)
  - Others experimentally world-wide (6Bone)
- ◆ Still in flux
  - Example: DNS (A6 vs. AAAA records)

28/04/2003

4

IPv6 was born out of the famous IPv4 "address exhaustion" problem. The first RFC to put forward a new direction for IP was RFC1550 which outlined a request for white papers calling the new IP "IPng" for "New Generation".

The first version of the protocol was detailed in RFC1883 where the final name of IPv6 was chosen.

Who uses it? Well, very few people in reality. It is widespread in Japan via the WIDE initiative but elsewhere 6Bone has languished in a few research centres and keen companies or individuals.

An example? The author's mailserver has been reachable over IPv6 since December 2001 and has received only *one* SMTP connection over IPv6 since then.

The standard is still in flux. For example DNS is still being debated with IPv6 address records being moved from AAAA to the new A6 standard naming.

## IPv4 to IPv6

### ◆ Key differences:

- Simplified header
- Dramatically larger address space
- Authentication and encryption support
- Simplified routing (a lesson learned...)
- No checksum in the header
- No fragment information in the header

28/04/2003

5

The transition from IPv4 to IPv6 has brought a number of significant differences for implementers and users alike. These range from a simplified header to a finally huge address space (of course huge is what the IPv4 address space was thought to be in 1980...).

The key differences are:

- Simplified header – it was relatively clear that a large number of the fields in the IPv4 header were rarely used (for example the TOS field) so they were removed outright. Furthermore it was thought to be a good idea to make use of the fact that packing data was no longer a requirement and better alignment was brought into play to support RISC chips. All optional information is now held in “extension headers”.

- The address space was significantly enlarged from  $2^{32}$  to  $2^{128}$  possible addresses.

- Authentication and encryption support was brought into the IPv6 standard from day zero instead of becoming an afterthought like IPsec for IPv4. Both AH and ESP functionality is supported as “extension headers”.

- Routing was made classless from day zero.

- The large number of checksums in the average packet was seen as wasteful and as such the IPv6 header does not contain a checksum. Much better functionality is obtained via the use of the AH extension header instead.

- Fragmentation is seen as wasteful, in particular as the minimum MTU is now 1270 bytes (the recommended MTU for IPv4 is 576 bytes), and is now being handled via extension headers.

# IPv6 – Simplified Header

## ◆ By example (tcpdump):

```

14:39:29.071038 195.82.120.105 > 195.82.120.99:
  icmp: echo request (ttl 255, id 63432, len 84)
0x0000  4500 0054 f7c8 0000 ff01 4c6e c352 7869  E..T.....Ln.Rxi
0x0010  c352 7863 0800 1c31 3678 0000 3e5f 6691  .Rxc...l6x...>_f.
0x0020  0001 1562 0809 0a0b 0c0d 0e0f 1011 1213  ...b.....
0x0030  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  .....!"#
0x0040  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
0x0050  3435 3637 4567

```

```

14:40:04.096138 3ffe:8171:10:7::1 > 3ffe:8171:10:7::99:
  icmp6: echo request (len 16, hlim 64)
0x0000  6000 0000 0010 3a40 3ffe 8171 0010 0007  ^.....@?...q....
0x0010  0000 0000 0000 0001 3ffe 8171 0010 0007  .....?...q....
0x0020  0000 0000 0000 0099 8000 60fe 4efb 0000  .....`.N...
0x0030  bc5e 5f3e 2f77 0100  .^_>/w..

```

28/04/2003

6

There is nothing better than tcpdump output to visualise the differences between IPv4 and IPv6. The first obvious difference is the nibble at offset zero which now contains a “6” to indicate version 6 of the protocol. What follows is a much shorter list of mandatory fields:

- Traffic class (8 bits), to be used for QoS [zero in above example].
- Flow label (20 bits), used to group datagrams for, for example, resource reservation (RSVP) [zero in above example].
- Payload length (16 bits), a value of zero indicates a “jumbo payload” which requires an extension header giving the true length [ $10_{16} = 16_{10}$  bytes in the slide].
- Next header (8 bits), a value of zero indicates no extension header. These are the same as protocol numbers for IPv4 with a number of extensions [ $3A_{16} = 58_{10}$  indicating an ICMPv6 header to follow].
- Hop limit, identical to IPv4’s TTL but now measured in hops, not seconds [ $40_{16} = 64_{10}$  hops].

This is followed by the 128-bit addresses, the extension headers if any, and the packet data.

For the record these packets are taken off a real wire and the source is an OpenBSD 3.0 system, the destination a Linux 2.2.x system.

## IPv6 – Larger address space

- ◆  $2^{128}$  possible addresses
  - In simpler terms: a lot
- ◆ Pre-partitioned
  - “where in the world is this address?”
- ◆ Organised
  - Structure, at last!

28/04/2003

7

If we now analyse the details of the new IPv6 header we should start from the simplest of changes: the increased address space. To begin with the number of possible addresses increased from  $2^{32}$  to  $2^{128}$ . In simple terms this is a huge number, enough to give every square metre of the Earth about 1 million IP addresses.

From the experience with IPv4 this time round the designers decided to pre-partition it leaving ample space free for applications yet to be invented and, more importantly, did not divide it on the basis of arbitrary classes (Like the old Class A, B and C from IPv4) but instead on the basis of “Top-Layer Aggregators”.

## IPv6 – Security support

- ◆ Same standard as IPsec for IPv4
- ◆ Authentication
  - Done via “extension headers” (AH)
  - Reference: RFC2402
- ◆ Encryption
  - Anything after ESP header is encrypted
  - Need *not* be the first extension header
  - Reference: RFC2406

28/04/2003

8

What about security? Well, security was an afterthought in IPv4 if we exclude the TOS fields which could be used to specify traffic classification (e.g. “Secret, Restricted, etc.”). This time it is in from day zero in the form of extension headers following the IPsec standard as defined for IPv4 in a number of RFCs. Some important modifications include the fact that the Authentication Header (AH) is now the recommended replacement for checksum calculations in the IP header and that the ESP (for encrypted payload) need not be the first extension header after the AH. As a matter of fact you might decide to have many extension headers which are not encrypted and only encrypt the payload itself.

## IPv6 – Routing and friends

- ◆ Simplified routing
  - Classless routing from day zero
  - “Top-Level Aggregators” (TLAs)
- ◆ No checksums
  - Already available in other layers
  - “Real” checksum available via AH
- ◆ No fragment info in base header
  - Fragmentation via extension headers

28/04/2003

9

Routing is one of the banes of the current Internet architecture. The propagation of routes and, in particular, the lack of use of aggregation by major ISPs mean that routing tables are huge and growing. IPv6 was designed from the beginning to be class-less and in particular the address allocation scheme means that you will no longer be able to take a bit of the address space and carry it with you. There will be Top Level Aggregators (TLAs) which will delegate parts of their huge address space as need be.

Another significant enhancement is the complete lack of checksum in the header. Why? Well, in the current IPv4 protocol there are lots of redundant checksums: IP checksum, TCP checksum, application-level checksum. In IPv6 the idea is that if you really need a checksum then you want a cryptographically strong checksum and should use an AH via an extension header. Otherwise there is no point in wasting both space and compute time.

Finally fragmentation has been tackled in a rather “final” way: fragmentation is ultimately bad, as such it should not be encouraged. Therefore fragmentation is only available via an extension header, the base header provides no support for fragmentation.

## Directions in NIDS

- ◆ White-listing
- ◆ Ubiquity
- ◆ Data management
- ◆ Network knowledge
- ◆ Deploying on IPv6

28/04/2003

10

The second part of this talk is about Network Intrusion Detection Systems and more precisely about the direction which they should be taking to tackle the problems which IPv6 will pose.

## NIDS – White-listing (I)

- ◆ Describing *badness* does not work!
  - “Badness” is an infinite concept
  - How do you catch zero-day attacks?
- ◆ Consider acceptable traffic flows
  - You should know your network
  - The number of authorised flows is smaller than you think

28/04/2003

11

Whitelisting is a simple concept to describe: instead of attempting to describe all that is bad in your network try and describe what is good and define “not good” as bad.

The current rules system in IDS is broken. Why? Because it relies on a concept of “badness”, i.e. defining traffic in terms of what is not allowed. There is a small problem with this: when does badness end? As more and more attacks are developed more and more “blacklisting” rules will be developed. But just as more and more rules will need to be developed more and more attacks are going to follow.

So how do we catch new attacks, the so-called “zero-day attacks”? You don’t because by definition nobody has seen them. This means that your IDS is completely useless as it only tells you about attacks you already know about.

This is all very well in theory but what does whitelisting mean in practice? Simple, you should consider acceptable traffic flows. Why? Because if you don’t know your network enough to be able to say “it is normal for user A to connect to server B” then you have a much bigger problem than worrying about an IDS. Once you actually start working on whitelisting you will notice that the number of authorised flows is actually much smaller than you might have first imagined.

## NIDS – White-listing (II)

- ◆ Steep learning curve
  - It takes time to describe normality
  - It is a boring job!
- ◆ Is it worthwhile?
  - You no longer play “catch-up” with rules
  - Zero-day attacks become visible

28/04/2003

12

Of course as with all novel ideas there are downsides:

Whitelisting has a steep learning curve because it takes a long time to describe normality, especially if this has never been done like in networks which have grown “organically” as needs required. The even larger problem is that it is a boring job. Given the choice what is more exciting? Is it the looking at normal network traffic and defining it or is it having a lab full of machine where you test out new attacks? Of course the latter so whitelisting requires *discipline*.

What do you gain by whitelisting?

You gain a lot: for example the ability to see zero-day attacks! Why? Because having defined what is valid any attack will be seen as invalid traffic, hence become an alarm. Secondly you no longer need to play catch-up with rules: instead of having to pray that someone has written a rule for a new attack *and* that you have time to download it before being attacked you can rest in the knowledge that your whitelisting will alert you.

## NIDS – Ubiquity (I)

- ◆ Your monitoring needs to be pervasive
  - Monitor all subnets
  - Validate firewall flows
- ◆ Remote sites need to be monitored too
  - Trained security analysts are not everywhere
  - Bad guys rarely take the front door

28/04/2003

13

Another problem with current IDS technology is that you are normally looking at a small portion of your network. Proper monitoring, especially if compared with real-life monitoring, is pervasive. It doesn't just cover the front door!

You should monitor all your subnets, not just the perimeter. Furthermore: do you trust your firewall? Are firewalls absolutely without flaws? If your firewall has a software error which allows through a packet which was meant to be blocked will you ever know? With current IDS deployment you will not know and the firewall is definitely not going to log a software error! So you need an IDS on both sides and compare the output: Differential Firewall Analysis (See <http://www.alchemistowl.org/arrigo/Papers/differential-firewall-analysis.pdf>).

What about remote sites? Often people ignore them because they are "minor". They might be minor from a business point of view but they are the ideal back-doors into your system. Bad guys don't normally walk through the front door... Similarly you won't normally have trained security personnel at outlying sites which means that the chances of intrusion being undetected are quite high.

## NIDS – Ubiquity (II)

- ◆ Look at network-wide traffic statistics
  - Surge on port 80 inbound: web DDoS?
  - Surge on port 25 outbound: Outlook virus?
- ◆ See the “bigger picture”
  - Isolated incidents are no longer “isolated”
  - Patterns appear

28/04/2003

14

It is also important to look at network-wide traffic statistics! There is a lot to learn by looking at numbers, more than one might imagine. It is sometimes a very good exercise to look at port 80 and port 25 statistics. If you normally consume no more than about 25% of your bandwidth in SMTP (port 25) traffic and this suddenly jumps to 80% could this not be a new Outlook virus?

Putting it all together is the key: look at the bigger picture! Isolated incidents might not be so “isolated” and then make use of the human brain. As more data is available the analysis of the individual datum stops being important as much as the pattern matching abilities. Patterns appear – look out for them and interpret them as soon as possible.

## NIDS – Data management (I)

- ◆ There is too much data!
  - Nobody really looks at it...
  - Slow & low attacks are invisible
- ◆ Aggregate
  - Why have 40000 identical alarms?
  - A little knowledge is dangerous knowledge

28/04/2003

15

There is simply too much data. The truth is that *nobody* looks at the data coming from current IDS. It overwhelms the human and virtually guarantees that any sophisticated attack will always be missed.

Data should be aggregated: why do we accept 40000 identical alarms in an IDS? If we don't aggregate we limit our ability to extract knowledge from the data and "a little knowledge is a dangerous knowledge".

## NIDS – Data management (II)

### ◆ Correlate

- Why look at ten sensors individually?
- Data becomes knowledge in context

### ◆ Trawl

- Historical analysis
- Sophisticated pattern matching

28/04/2003

16

Just like we should aggregate why should we accept data from ten separate sources in isolation? Why is this not aggregated? Data becomes knowledge *in context*. By correlating patterns emerge, you finally see that a certain attack is hitting sites on the US East Coast but nowhere else rather than saying “we are being attacked at ten sensor sites”.

Once you have both aggregation and correlation you can finally trawl through the data and perform historical analysis. Answer questions like “have I seen this site before?”. Similarly the pattern matching, both automated and manual, which can be performed in sorted, aggregated and correlated data is infinitely more powerful.

## NIDS – “knowledge” (I)

### ◆ Judge attacks depending on target

- IIS attack against Apache should *not* alert
- \*nix attacks against \*nix should *escalate*

### ◆ Match attacks with your staff

- \*nix attacks to Windows staff is a waste of resources
- Play your best analyst on tough calls

28/04/2003

17

The final weak point in IDS technology is the issue of “knowledge”. What does an IDS know? Well, fundamentally nothing. Why do you get alerts for a Windows IIS attack against a Unix Apache host? Surely this shouldn’t happen. Similarly why is an Apache attack against an Apache host ranked exactly the same as an irrelevant attack?

You need to match the attack with the target!

You also need to match the attack to the staff at your disposal: security people are expensive, wasting them is sacrilege. Sending Unix attacks to your Windows guru is a prime example of wasting precious resources. You should also escalate depending on the severity of the attack. If you call your security guru for each ICMP Echo Request packet you are not going to have him in your staff book for long.

## NIDS – “knowledge” (II)

- ◆ Judge severity before alerting
  - 1000 “red alerts” lose their meaning
  - A 24x7 “High alert” becomes “normality”
- ◆ Follow attack flows
  - Don’t wait for the network to be flooded
  - Find internal sources

28/04/2003

18

Furthermore you need to judge severity before alerting. There is little worse than 1000 red alerts a day: the screen is a “sea of red” and the concept of “red alert” loses its meaning. Similarly being on “high alert” 24x7 doesn’t take you very far because “high alert” becomes “normality” and people no longer react to it.

Once you have knowledge you can also follow attack flows, instead of reacting to a complete meltdown of your network see as the attack happens and try and close off paths to isolate the attack from the rest of the network.

You can also find internal sources of attacks once you have ubiquitous monitoring.

## NIDS – IPv6 (I)

- ◆ Deploy a small test IPv6 network
  - Cheap: use Linux or \*BSD
  - Simple: native on OpenBSD
  - Painless: mistakes remain internal
- ◆ Use real services
  - Don't play with telnet only
  - Try at least a webserver and mailserver

28/04/2003

19

So how do you do NIDS on an IPv6 network? Well, first of all test everything. All you need is a small network, a few PCs running Linux or OpenBSD. It is painless and nobody will ever know if you make a mistake. Use real services, not telnet. If you are not prepared to deploy real services like e-mail and a web server then you will not learn much.

## NIDS – IPv6 (II)

- ◆ See what attacks look like
  - IPv4 NIDS don't detect IPv6 attacks
  - Make sure you have an IPv6 router
  - Learn tcpdump!
- ◆ Follow developments
  - focus-ids @ SecurityFocus
  - Snort CVS head

28/04/2003

20

You should then start looking at attacks which are well-known in the IPv4 world. What do their counterparts in IPv6 look like? Of course quite a few of them will be just the same (e.g. HTTP attacks). Others will be more interesting (fragmentation attacks, for example).

You really need to learn tcpdump well to play with this!

Furthermore follow developments in the IDS community, for free you can subscribe to focus-ids, a mailing list at SecurityFocus or indeed look at the Snort CVS head.

## Observations

- ◆ IPv6 is (very) slowly coming
  - Simple structure means better performance
  - Be prepared for *lots* of data
- ◆ NIDS are becoming a commodity
  - Less research in esoteric protocols
  - More attention to user interfaces at the price of representing complex systems

28/04/2003

21

IPv6 is slowly coming: it has a much better structure which means better performance but also carries with it the fact that more and more hosts will connect. This means *lots* of data!

NIDS are instead becoming a commodity. Vendors want you to go to the shelf and next to your anti-virus buy a pretty IDS box. This means that there is less and less research into protocols and more in fancy interfaces. Also vendors don't want you to use whitelisting because that will mean that you will not buy their rule update service!

## Goals of Security Analysts

- ◆ Have a small rate of false-positives
- ◆ Must not lose grasp of the network as a whole
- ◆ Must see from multiple sensors aggregated with given criteria
- ◆ Must have dedicated forensic tools

28/04/2003

22

What should a security analyst want in life? Simple: as few as possible false positives. He should not lose grasp of the network as a whole but always see it as a single large entity. It should also be possible to see security data aggregated across the whole network with a given criteria and *must* have dedicated forensic tools at his disposal!

## Conclusions

- ◆ IPv6 is not revolutionary in itself but the masses of data requires a strategy
- ◆ Current NIDS are unprepared

Unless things change security analysts  
will have a very tough time

28/04/2003

23

So what does all this mean? Well, IPv6 is not revolutionary but the great amount of data requires a strategy. What is worse is that current NIDS are unprepared and, even worse, getting less and less prepared as marketing strategies take them elsewhere.

This means that unless things change security analysts will have a very tough time!

