NAT logging basics

David Ford
OxCERT (OUCS)

What is NAT?

- Formally - the method of modifying network address information in a packet whilst in transit
- The effect is to modify one (or more) of the Source Address, Destination Address, Source Port and Destination Port of a packet at a router
- In our environment typically used to allow a larger number of machines to share a small number of public IP addresses

Why NAT?

- Used for a number of reasons:
  - Lack of address space
  - Separation of different classes of system - this can be done in other ways
  - Perception of "security" - in practice a default deny inbound firewall has similar benefits

How does it work?

- In the typical case, internal machines are given addresses from the private address space (see RFC1918), eg 192.168.1.0/24
- The NAT device has a public IP, and an RFC1918 address
- Traffic flowing from the internal machines will have the source IP translated, and the source Port will also be translated

- The NAT device keeps a table of translated source IPs/Ports, so that when traffic flows the other way it can be translated in the opposite way:

192.168.1.24:3124 -> 163.1.2.102:22

becomes:

129.67.1.15:14675 -> 163.1.2.102:22

Where the IP "129.67.1.15" is the external IP of our NAT device and the port "14675" is selected by the NAT device

- several important things to note:
    - From within OxCERT we can see nothing beyond the last line
    - We therefore have no way of tracing the connection beyond the NAT device without your help
    - the source port of external traffic is **not** the same as that internally (this can be confusing, so remember it)

- Note, some incident types may only include limited information, for example Cease and Desists often contain:

```
Evidentiary Information:
Notice ID: 56354566
Asset: Adobe Photoshop
Protocol: BitTorrent
IP Address: 129.67.1.54
DNS: naughty.person.ox.ac.uk
File Name: Adobe Photoshop CS4 Extended
File Size: 693982815
Timestamp: 17 July 11:05:32 GMT
Port ID: 47382
```

you may have incidents where many users use a server legitimately, but one machine is using it for nefarious purposes:

- botnets on a popular ircd,
- bots using popular chat protocols etc)
- more recent http botnets/keyloggers may use common shared webservers or a hacked page on a legitimate site

What other impacts does NAT have?

Some of these also apply to proxyarp/routing firewalls

- We don’t see MAC addresses of systems
- We don’t get internal IPs
- We don’t get netbios names - this is generally the case for all systems, however some malware did give these out in the past - you can’t generally expect this

Why is logging needed?

- We’ve seen a few cases of incidents above - without logs it’s impossible to trace a system
- It’s important to be able to trace systems quickly in the event of malware - it’s no good waiting hours or days to find a system that’s scanning/ infecting other hosts
- Illegal content - being unable to trace a system when requested by law enforcement is generally a bad idea

What logging is neccessary?

- Ideally we need to trace any flow to a user/machine
- In fact, as we saw with the C&D example, ideally from a timestamp+external IP +external Port

So,

- We need to log all the flows (and preferably the NAT translations)
- And we need the map from internal IP -> MAC address/room port of the user
- Hopefully you have the latter mapping already for non-NAT systems, but make sure you do have the logs:
    - arpwatch, DHCP logs
    - Don't forget the MAC -> user mapping (eg registration forms, NAC, etc)

- Note flows may be very short lived, and particularly on a busy network may not remain within the NAT device’s state table for very long, we would strongly recommend that snapshots of a state table or similar are unsuitable as a form of NAT logging

Options for logging translations/flows

- Syslog - supported by many off the shelf NAT devices, however the format is not consistent between devices, it’s hard to parse automatically. Please check that your syslog server **and** NAT device are both NTP synchronised and check that timestamps match actual flows if you use this. Also check you have tools to filter the logs

- Argus - http://qosient.com/argus/
- A useful (and generic) starting point if you have a mirrored port
- Very useful if your hardware can’t log otherwise
- Versatile - you can also capture and process netflow with it, could form the basis for a comprehensive network monitoring solution for a unit
- But, in most configurations won’t capture translations - you want flows from both sides of NAT at minimum

- Netflow/sflow
  - Typically captures only pre or post NAT not the translations
  - Check whether it is sampled, you may miss the critical flow if you sample
  - requires hardware support for netflow/sflow, but that’s relatively common
  - You can capture the netflow data into argus if you like

- Linux
  - at a first glance, surprisingly difficult
  - Getting the right output from iptables appears to be impossible
  - but, Linux supplies a userspace conntrack tool that is designed for this

- the conntrack tool will record the full translation, and you can parse this in various ways
- within OUCS we convert each flow to three netflow records, one representing the translation and two representing the flow destination (assuming a bidirectional flow communication)

BSD (pf)

- Here we look at PF based NAT firewalls, you can do NAT using IPFW or IPFilter
- NAT logging can be achieved through one of the directives:

```
nat log on en0 from 10.0.0.0/24 to any -> 129.67.1.15
  OR

nat log all on en0 from 10.0.0.0/24 to any -> 129.67.1.15
```

- which produces logs in a tcpdump readable format

NAT log storage

- Typically logs compress down very well - often 5-10x
- Rotate logs frequently, hourly is good if large
- Make sure your logs are synchronised with an accurate NTP source
- make sure you know what time zone they're in, and make sure you don't lose data when the clocks change
- the Conditions for Connection state a need to keep logs for 60 days

Processing

- Exactly how you process your logs will of course depend what format you're collecting and storing them in so these are only guidelines
- For argus, see my recent talk (slides on the web)
- For syslog type output, grep may be helpful. Also, try to avoid weird character sets - UTF-16 is not really needed for log files and lots of tools won't like them

What to look for?

- If you are scripting a process for analysis of your logs you probably at least want to be able to search based on:
  - destination IP/Port
  - post NAT (ie External) IP/Port
  - pre NAT (IP/Port) *(this is an unusual case)*

Other issues

- Monitoring the logs:
  - If you don’t read the logs frequently, it all too often happens that something stops working when they are needed
  - Check your logs contain data (eg check each log contains at least a certain number of lines when rotated)
  - Make sure you won’t come close to maximum file sizes for your chosen OS/partition, even if a scanner doubles the log file size

Case Studies

- This section is designed to be based more on practical examples of what can and is done within the University, they're

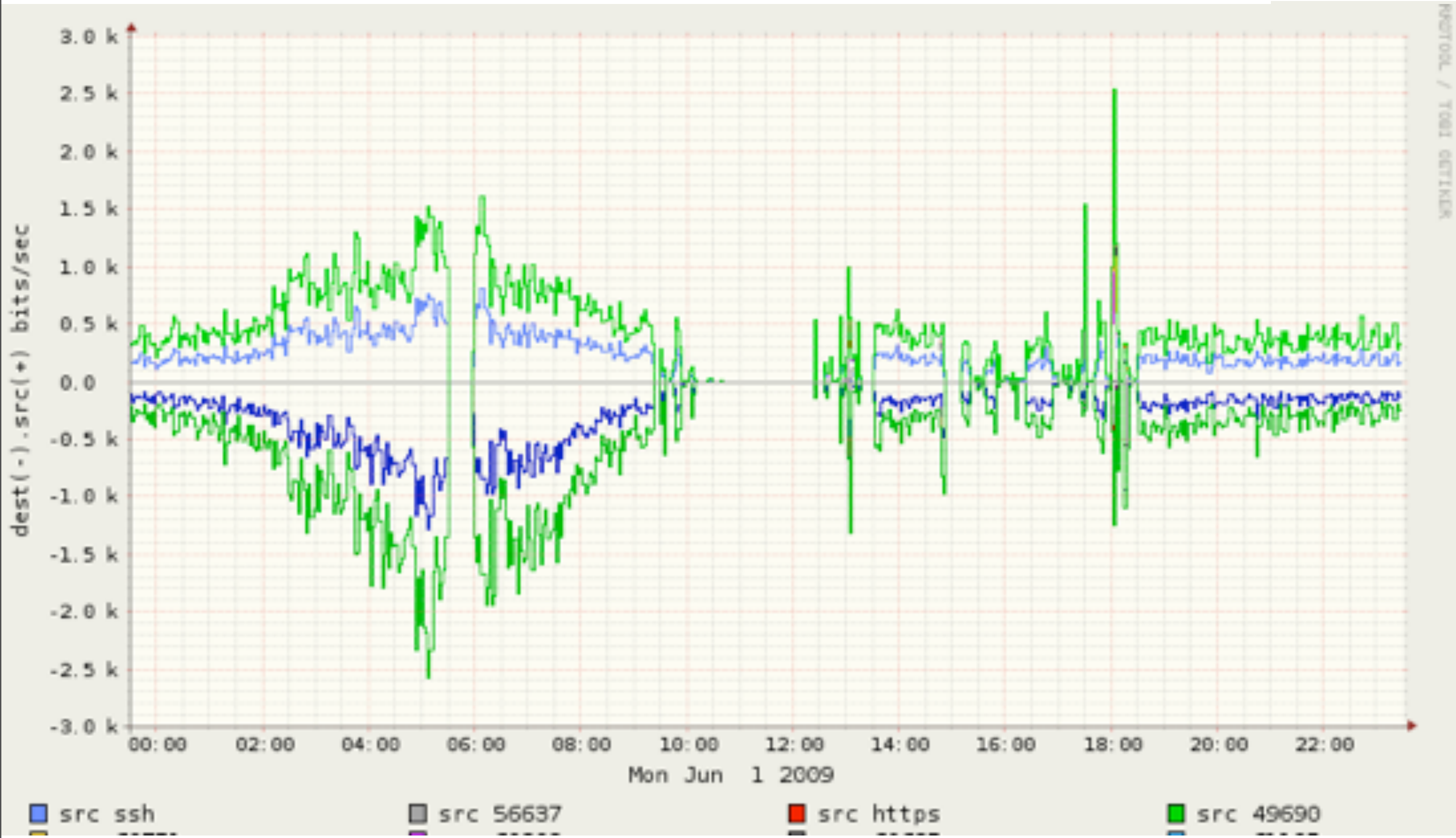
Argus

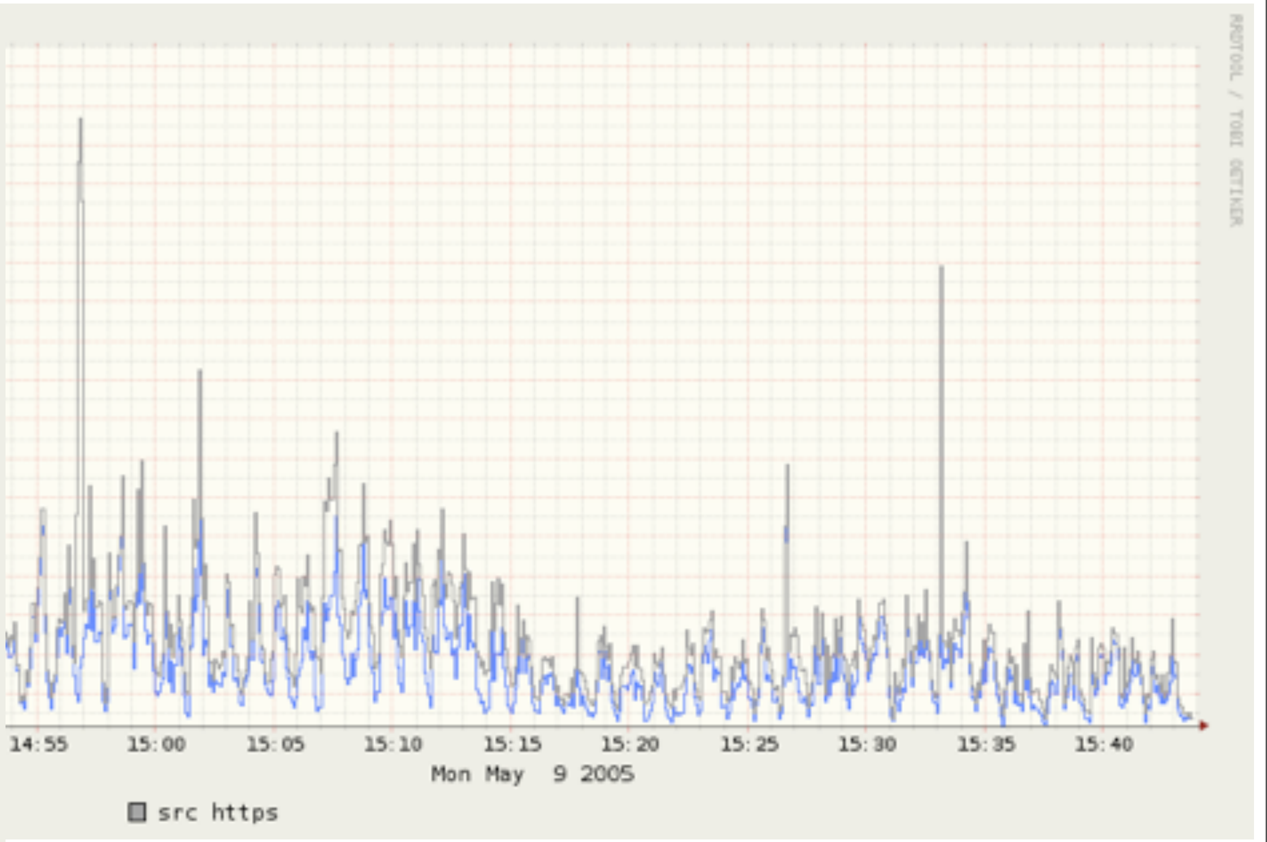
- Several units have proprietary firewall appliances that do not provide a usable or easy to use interface to retrieve NAT logs.
- However, their switches are capable of mirroring the traffic going into and out of their NAT box
- Many units use this for tools like ntop, or snort already

Argus (2)

- Argus provides a simple addition to this that will capture flows, and store them in a standard format
- However, see the Argus talk for potential pitfalls
- Also, it doesn’t capture translations, but in some cases the presence of both sides is sufficient (**warning:** we can envisage cases where it won’t!)

```
15:28:09.51  tcp 10.0.3.2.2134 <?> 208.78.34.24.80 CON 1 7 130 5783
15:28:11.52  udp 10.0.3.2.4324  -> 129.67.1.1.53   INT 1 0 5   64
15:28:11.52  udp 10.0.3.2.5258  -> 129.67.1.1.53   INT 1 0 5   64
```





Syslog based logging

- There are several products in use within the University that can log NAT translations to syslog, examples include:
  - Cisco NAT devices
  - sonicwall NAT devices
  - others we've not been told the name of

Syslog examples

- Each product has it’s own different format:

```
2009-07-11,00:01:08,id=firewall time=”2009-07-11 00:01:03” fw=163.1.1.3 pri=6
c=1024 m=537 msg=”Connection Closed” n=0 src=10.0.3.32:80 dst=209.85.227.104
proto=tcp/http sent=54 rcvd=5674

2009-07-11,00:01:09,id=firewall time=”2009-07-11 00:01:04” fw=163.1.1.3 pri=6
c=1024 m=537 msg=”Connection Closed” n=0 src=10.0.3.28:80 dst=209.85.227.147
proto=tcp/http sent=54 rcvd=5674
```

```
2009-04-28 21:45:39 [10.1.4.5] [Local Use Four] [Informatonal] 45:38:
%ASA-5-245454: Built Outbound TCP connection 3243434254 for OUTSIDE:
209.85.227.147:80/80 (209.85.227.147:80/80) to INSIDE:10.0.3.32/3218
(163.1.1.3/32254)
```

Difficulties Experienced

- Several units have experienced difficulties with reliability:
    - data getting dropped towards the end of the day when traffic was heavy (possibly due to file sizes)
    - Syslog servers falling over
- Also, difficulties in processing the files - not really suitable for tools like notepad/ wordpad/Textedit.app

- We can and do process syslog files for units in the event of a security incident
- However our resources are insufficient to deal with this for C&Ds, or when a unit wishes to track a user for other reasons
- We would strongly recommend working out how to process log files before you need to!

Linux Based NAT (conntrack/netflow hybrid)

- This is a solution developed in house for use by the Location Independent Networks (eg Eduroam/OWL)
- It works from on a Linux based NAT solution using conntrack
- (however the theory could be extended to any type of NAT device from which logs of translations can be extracted)

- Conceptually the data is converted from the output of:

conntrack -E

- via a perl script, to form two sets of output in the “flow-tools” format:
  - 1x translation
  - 1x flow post translation

- These can be captured by any device that supports netflow, and processed, we store flow tools format and import into argus
- We are working on tools to use argus to output the following format from the data:

```
14/07/2009 14:00 tcp 10.0.3.1:3218 -> 163.1.1.3:32254 <-> 209.85.227.147:80

                               1  2   54  6436
```

- As of 13/07/2009, preliminary code is working, still a few bugs, but it is usable
- The aim is to have a sufficiently generic framework that this can be used as a standardised NAT storage model if people wish

Conclusions

- NAT is increasingly common within the University
- Care must be taken to ensure adequate logging is kept
- Several formats of logs can be used with a variety of advantages/disadvantages
- The key is to check your log collection and processing process is robust and reliable
- You can send us logs via https://malware.oucs.ox.ac.uk