Effective use of Snort on Large Networks

The OxCERT Approach

David Ford
OxCERT, University of Oxford

Plan of Action

- I'm going to focus on our usage of Snort
- Why?
- How?
- Some technical details, but our process is the focus of the talk.

Oxford University Computing Services

www.oucs.ox.ac.uk

Background to Our Team

- Founded in 1994
- Originally a number of volunteers from across the University who were interested in network security
- Now 4 full time staff members at the Computing Services

Robin Stevens (Team Leader)
Jonathan Ashton
Mark Duller
David Ford

How large a network?

- 10Gbps backbone
- Over 2Gbps actual traffic to/from the outside world
- 100+ connected "units" each with their own independent networks (making it impossible to separate by class of host)

OxCERT's role and remit

- OxCERT's responsibility lies at the University central backbone level.
- We do not see what happens beyond an individual Unit's boundary

Oxford University Computing Services

www.oucs.ox.ac.uk

Our Role

- *"To protect the integrity of the University backbone network and to keep services running"*
- This can encompass a wide range of threats
- But we're not the network police, we don't directly deal with copyright infringements, people viewing inappropriate materials, etc

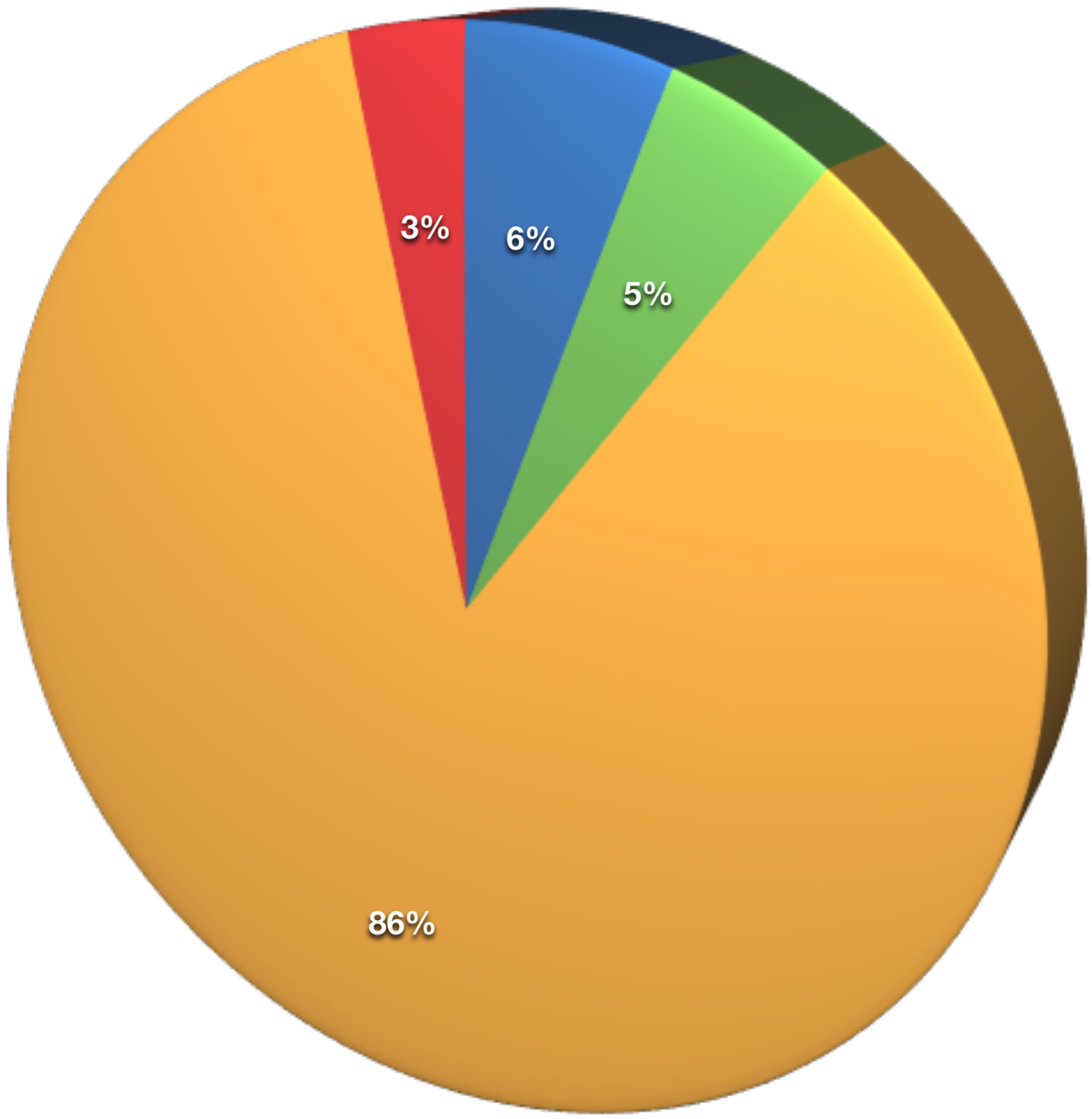
Oxford University Computing Services

www.oucs.ox.ac.uk

Why Snort?

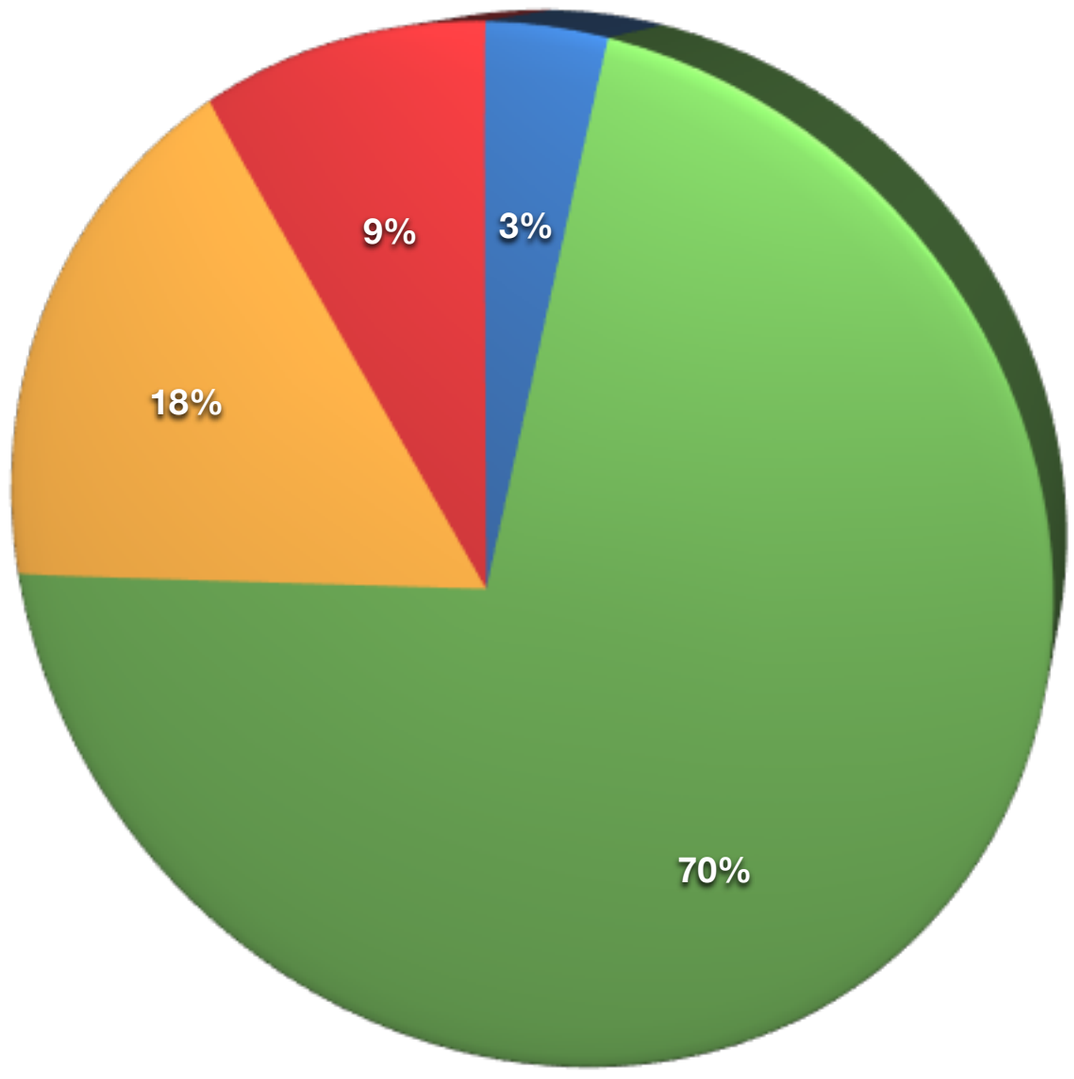
- Historically Network Flows were our key means of incident detection
- Ideal to trace IRC bots, port scanners, DoS type traffic, etc.
- Much harder to deal with HTTP botnets, or compromises over port 80
- When I started in the team in 2006 our usage of Snort was largely for a very small number of specific exploits

July 2007-December 2007



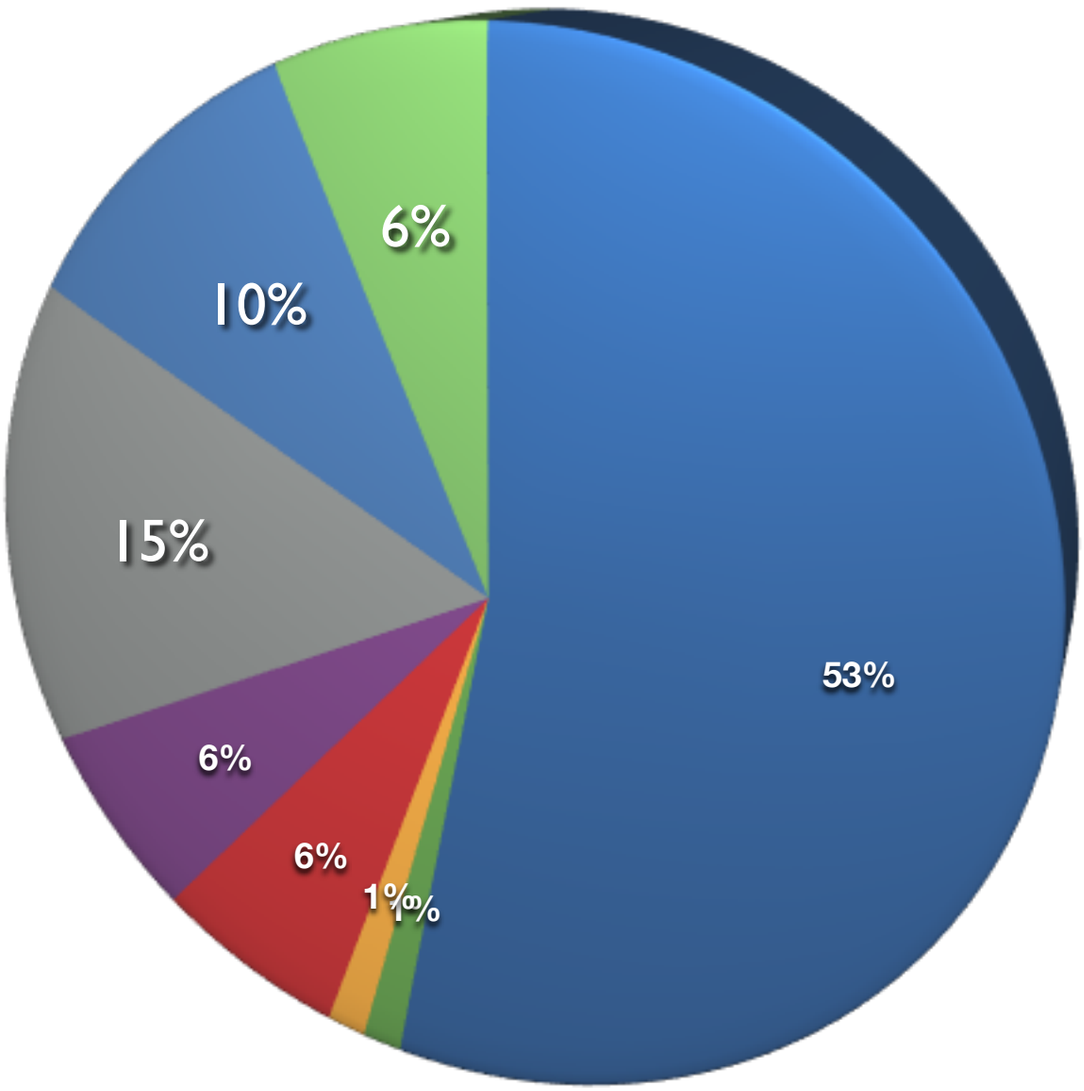
External Notification
Snort
Flows
Other

January - May 2011



Oxford University Computing Services
www.oucs.ox.ac.uk

Coincides with the malware trends we're seeing (Jan-May)



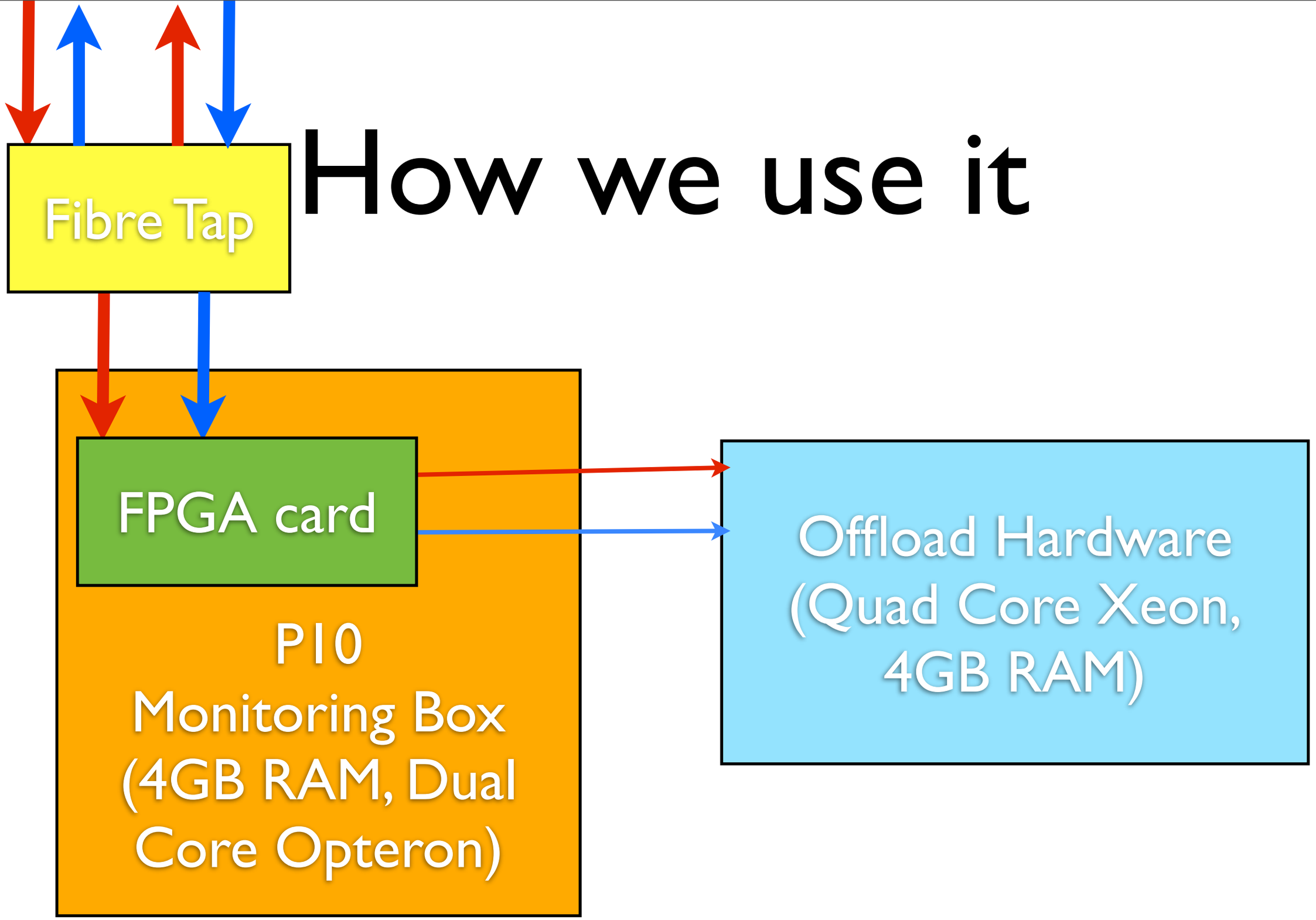
- Keylogger/Info Stealer
- Malicious DNS
- Conficker
- Port Scan
- Botnet
- Phish Replies
- SQL Injection
- Other

Oxford University Computing Services
www.oucs.ox.ac.uk

How?

- We have a box from Force10
- Known as the P10, contains a pair of 10Gbps FPGA NICs, which can run a hardware Snort
- Matching packets can be sent to 1Gbps copper and to the local host for analysis

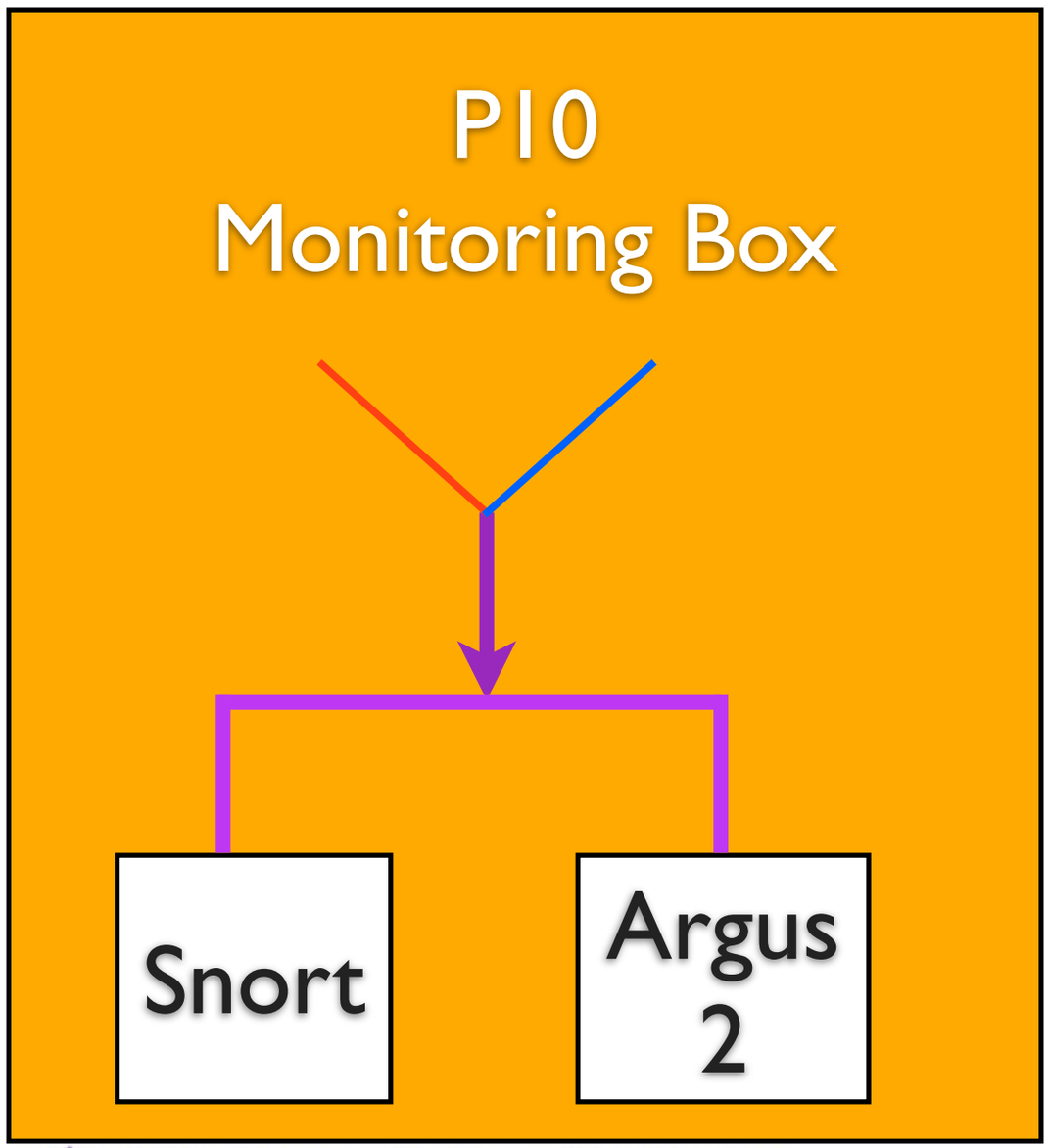
How we use it

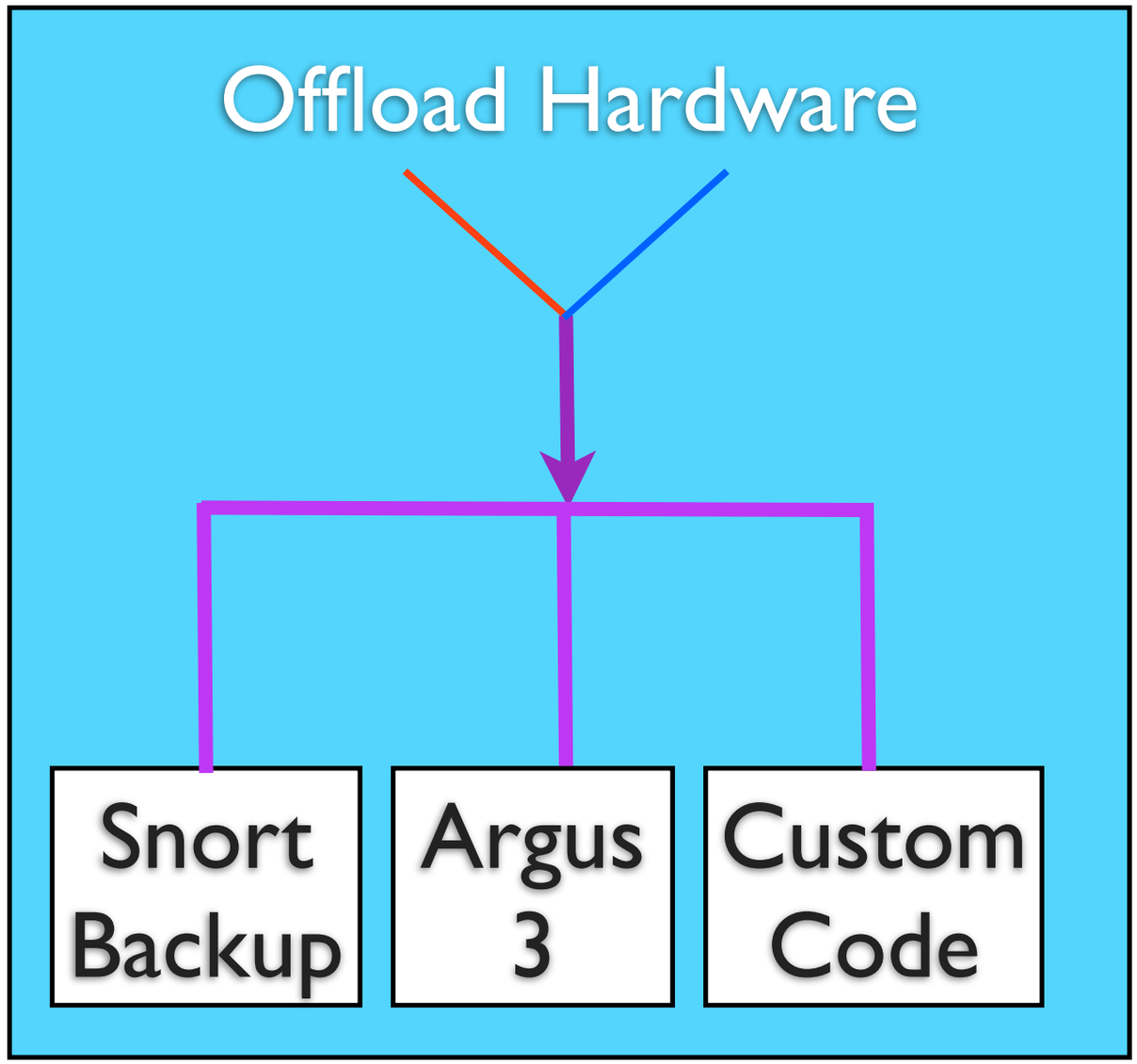


Oxford University Computing Services
www.oucs.ox.ac.uk

→ 1GBps
⟶ 10GBps

How we use it continued





Oxford University Computing Services
www.oucs.ox.ac.uk

What sort of ruleset?

- Short answer: a very much customised one
- Our general starting point is to look at our own role and remit
  "To protect the integrity of the University backbone network and to keep services running"
- How does this relate to malware we see today?

Oxford University Computing Services
www.oucs.ox.ac.uk

- This begins to allow us to decide on some things that are important to us in fulfilling our role (and lots of things that aren't)
- We focus on gathering intelligence on known compromised hosts rather than obtaining a flood of alerts to sift through
- Our ruleset achieves a false positive rate of less than 1%
- So we can then begin to know what we should be caring about detecting, for instance:

Botnets

DDoS

Keyloggers

Port Scanners

Phished Credentials

Compromised Accounts

DNS Trojans

Compromised Servers

SQL Injection

And some things that are less relevant to our remit

Producing rules

- We've covered a lot of things before we got here - that's intentional
- Actually writing the rules is a small part of producing an targetted/efficient/useful/low false positive Snort setup
- But how do we do it? (once you know what you're looking for)

The data is out there

- But probably not in the format you desire
- Lots of resources will give you analyses of malware behaviour - and we're moving into that realm ourselves as well
- A few places we find invaluable for writing our own rules:

Oxford University Computing Services

www.oucs.ox.ac.uk

Starting points

- Anubis /Threatexpert (and their blogs)
- Zeustracker + Amada.abuse.ch
- isc.sans.org
- Some AV vendors (although they tend to be more inclined to obsfucate URLs, which is unhelpful for what we are doing)
- Google
- Other ac.uk sites (IDS-SIG etc)

What to look for

- Our approach is to hunt for known compromised hosts we want things to indicate a compromise, not merely an exploit attempt!
- URLs - ideally matching the following criteria:
- Are unique to infected machines
- Or, where the user agent/URL/referrer combination is unique to infected hosts

Common behaviour you may see

- One or more of:
- Downloading of a malware configuration file
- Updating a malware sample
- sending data back to a controller
- You may wish to take different steps for each
- An IRC channel name (or more commonly a tcpdump/ngrep style output)

Things to be aware of

- IP resolution may not do what you think - hosts file entries are not uncommon
- If you have a copy of the malware this may be easy to verify, otherwise you may want to ensure your rules match on the Host: header as well as what you think the IP is
- Hosting data on so called “fast-flux” networks is also possible, which may nobble static IPs in rules

Scripting

- You almost certainly want to script lots of this up to do things automatically
- A few ideas:
- Automatically downloading and parsing data from some known good sources
- A database of URLs and connection destinations
- you might want to consider a separate caching DNS resolver to avoid any local DNS redirection

Example: SQL Injection

- Very common these days against custom written code
- after several web server compromises with logs it was clear a small number of tools were used to attack servers
- after checking half a dozen or so cases we had a good idea of the stages of an attack and how to detect the various stages

- We knew what was happening and why, so were in a good position to notify IT Staff that their servers were likely to be vulnerable
- Added benefit of being able to say "variable X in script Y is not sanitised properly"

Future Work

- For us this process is an ongoing one
- Improving our detection via our new upcoming malware lab
- devising ways to make inserting new bad URLs/IRC Channel names and similar easier
- scalability of our system (due to our flow/ snort setup on a single piece of hardware)

Oxford University Computing Services

www.oucs.ox.ac.uk

Comments/Questions

http://www.oucs.ox.ac.uk/network/security

david.ford@oucs.ox.ac.uk

security@oucs.ox.ac.uk