DDoS Detection and Alerting

Daniel Romão

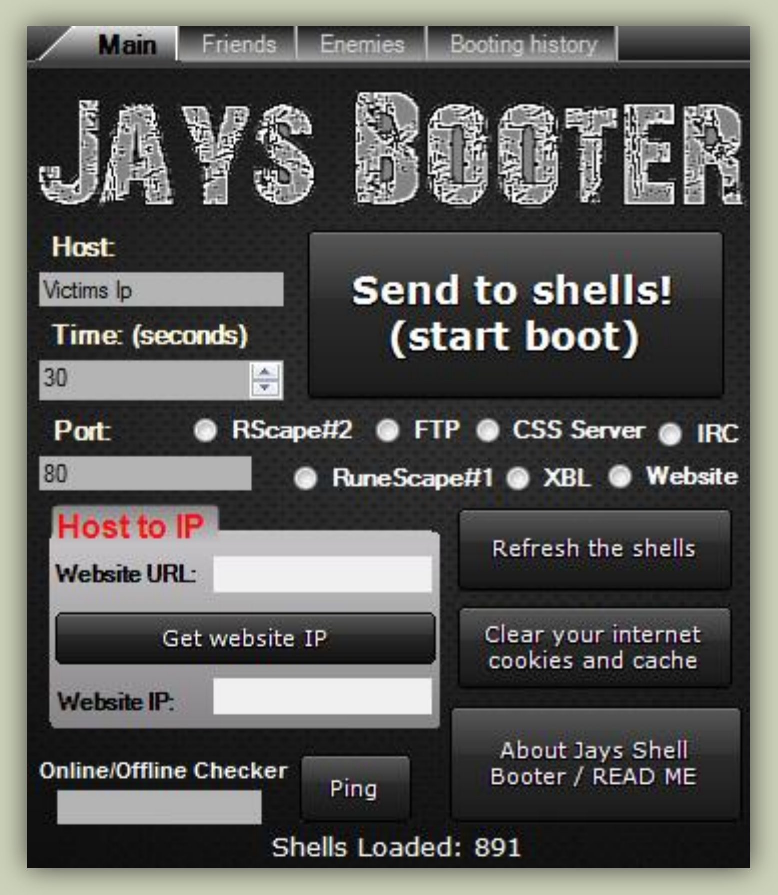
Niels van Dijkhuizen

BACKGROUND

- DDoS attacks are commonly seen in the SURFnet network
  - Mostly flooding attacks
  - Customers are heavily affected and complain

- These attacks are cheap and easily performed

BOOTERS / DDOSSERS / STRESSERS



Host: Victims Ip

Time: (seconds) 30

Port: 80

Website URL:

Website IP:

Shells Loaded: 891

CURRENT SOLUTION

- What does SURFnet currently use?
  - Fixed threshold alerting
  - IP fragmentation alerting
  - BGP off-ramping and traffic washing

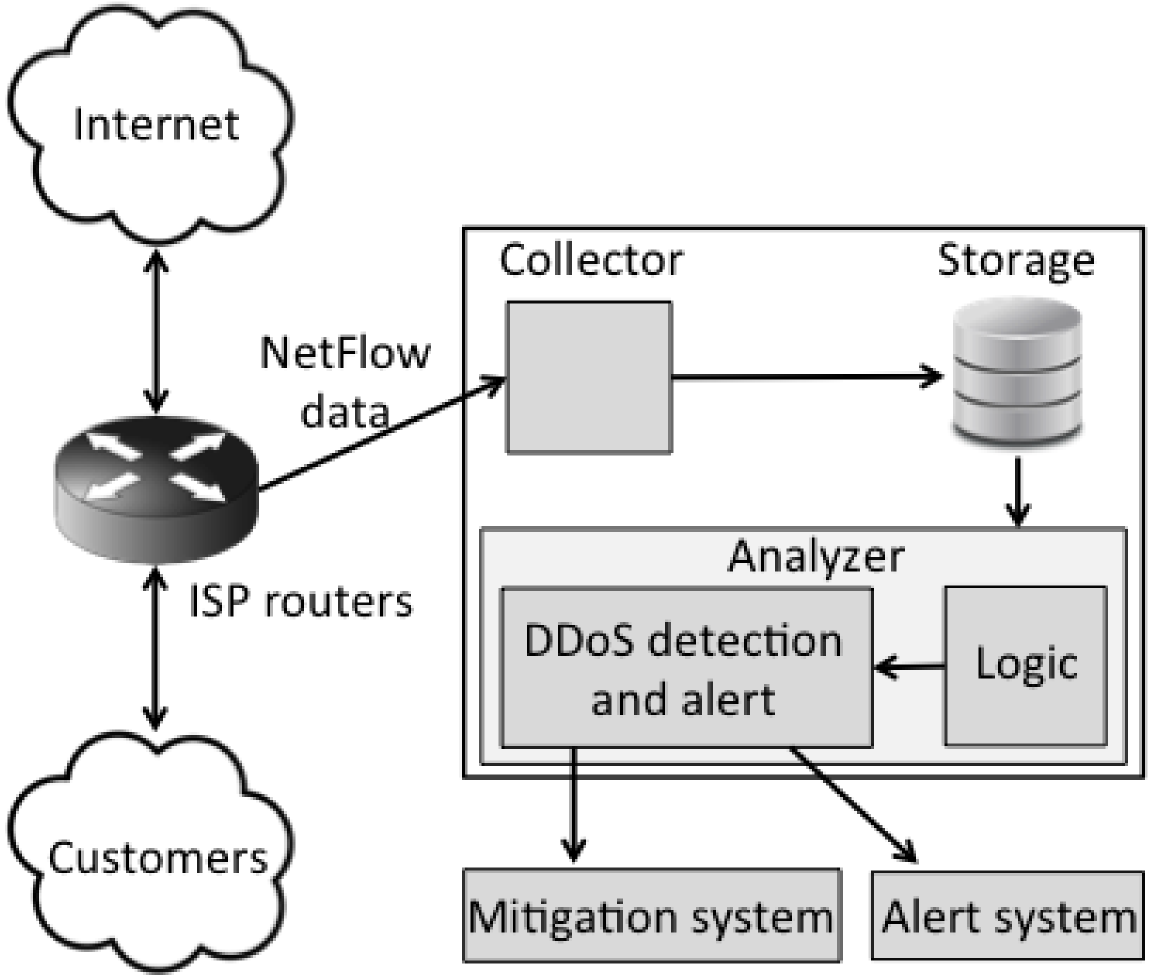
Can we make it better?

RESEARCH QUESTIONS

“Can we derive DDoS mitigation rules from the available production data in near real-time in order to alert and mitigate?”

- What kind of DDoS attacks can we detect?
- Can we detect them in near real-time?
- Can we extract enough information for mitigation?

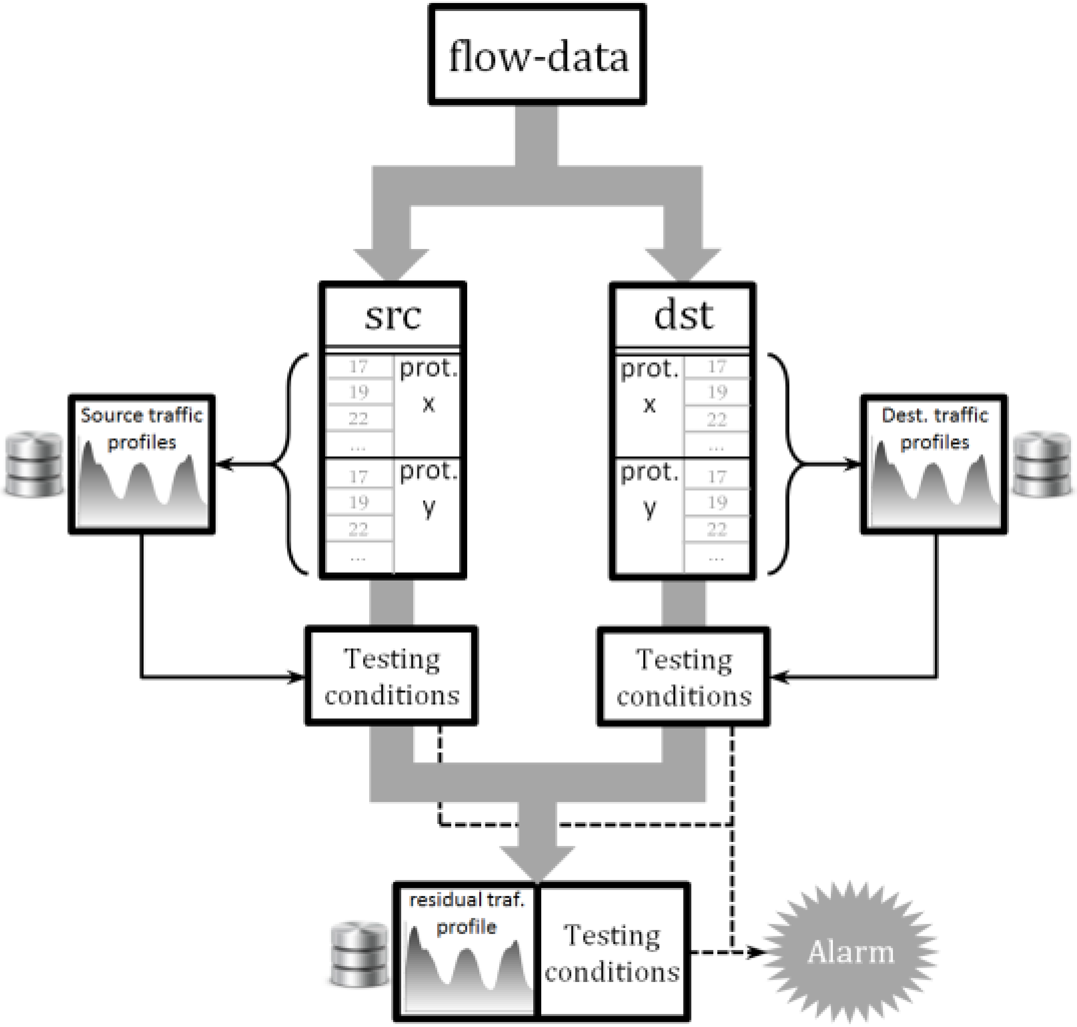
WHAT WE PROPOSED



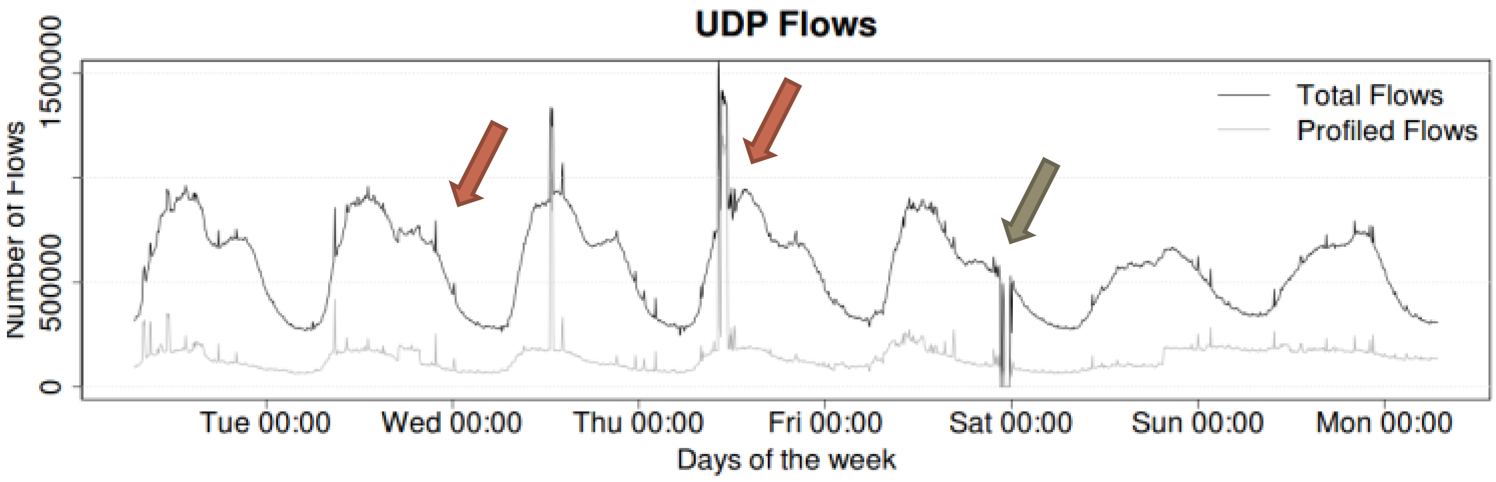
APPROACH

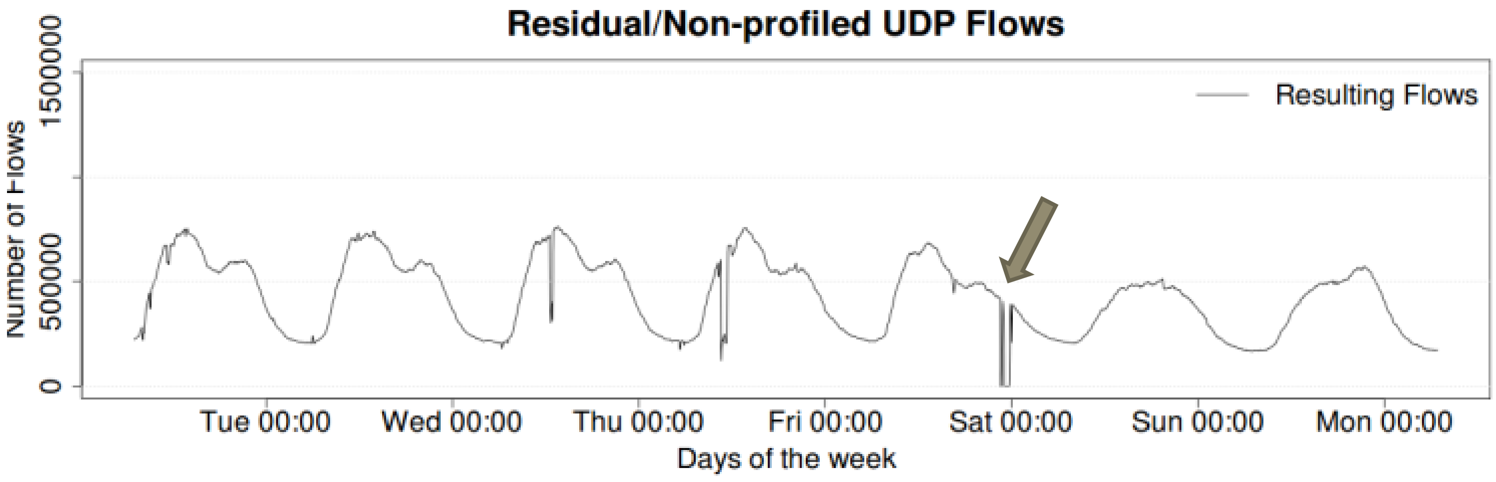
1. Collect one week NetFlow data
   - One on hundred sampling
2. Filter interesting application protocols
   - 53/udp (DNS), 123/udp (NTP), 80/tcp (HTTP), …
3. Categorize traffic by behavior
4. Create baselines
   - Application protocols
   - Rest of the traffic (icmp, tcp, udp)

MODEL


flow-data
src
dst
17
19
22
...
prot.
x
prot.
y
Source traffic profiles
Dest. traffic profiles
Testing conditions
Testing conditions
residual traf. profile
Testing conditions
Alarm

FINDING NEW ANOMALIES



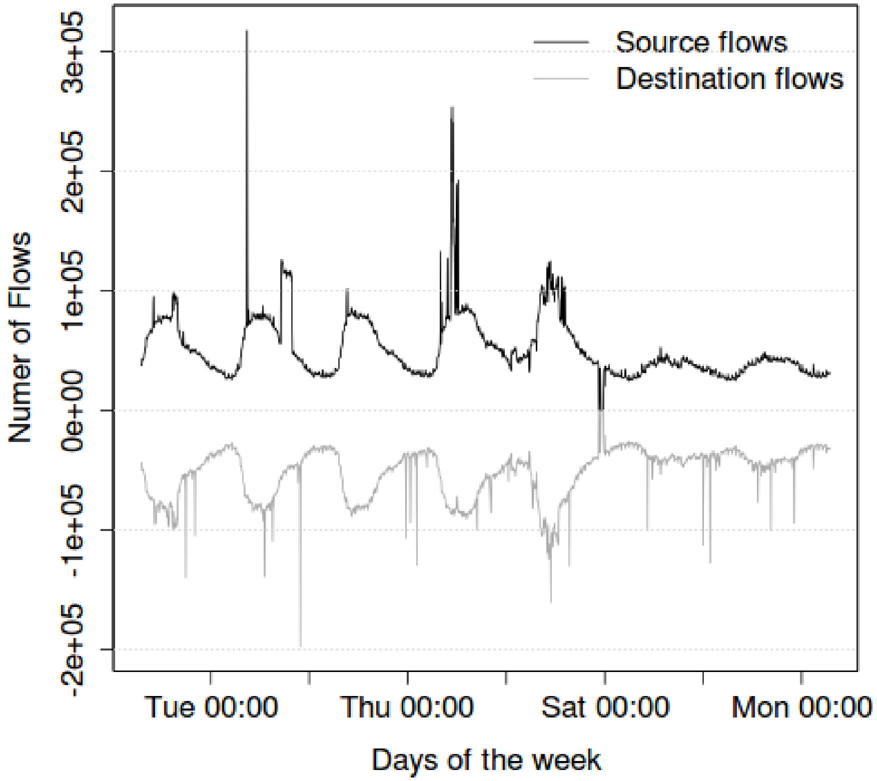


ANALYSIS

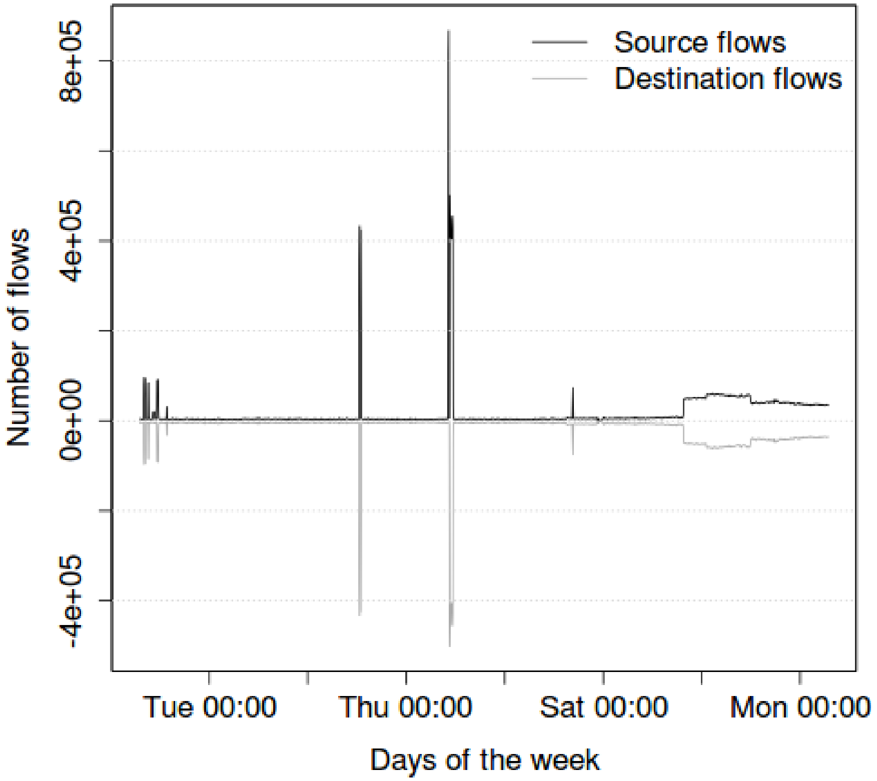
- Correlations:
  - Bytes per packet
  - Source – Destination ratios (symmetry)

- Categories identified:
  - Regular traffic without noise (e.g. HTTP/TCP)
  - Regular traffic with noise (e.g. DNS/UDP)
  - Non-regular traffic (e.g. NTP/UDP)

EXAMPLE OF BEHAVIORS



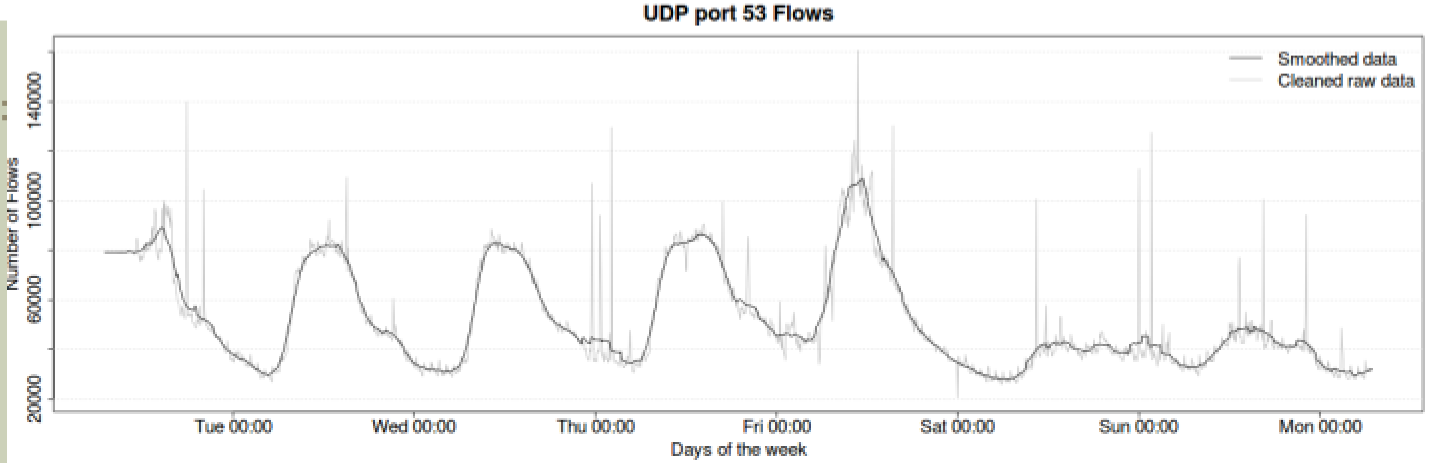
Flows for 53/udp



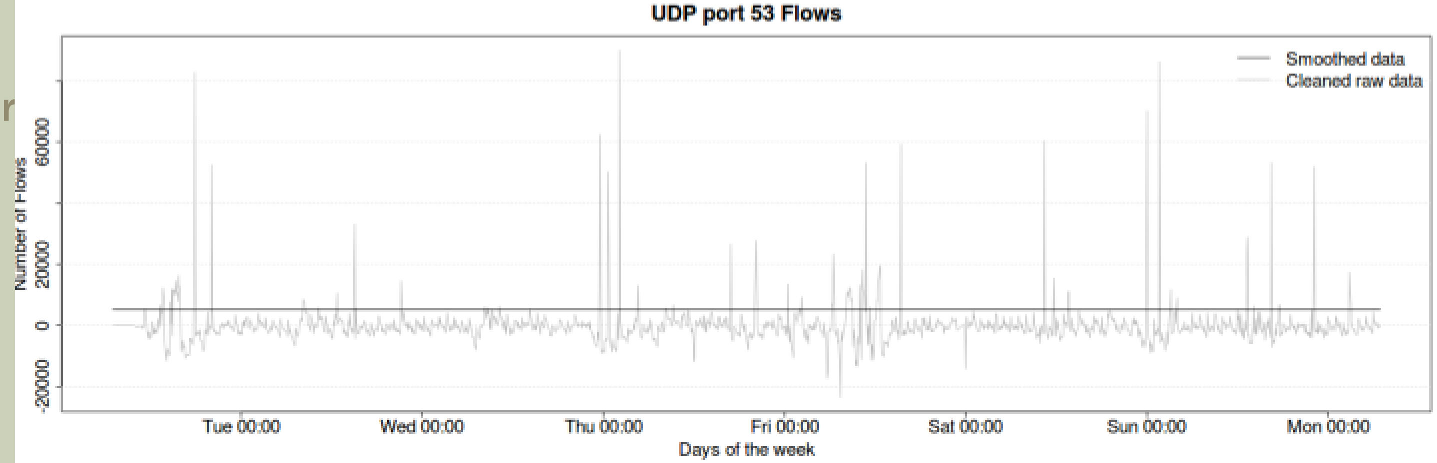
Flows for 123/udp

REGULAR WITH NOISE

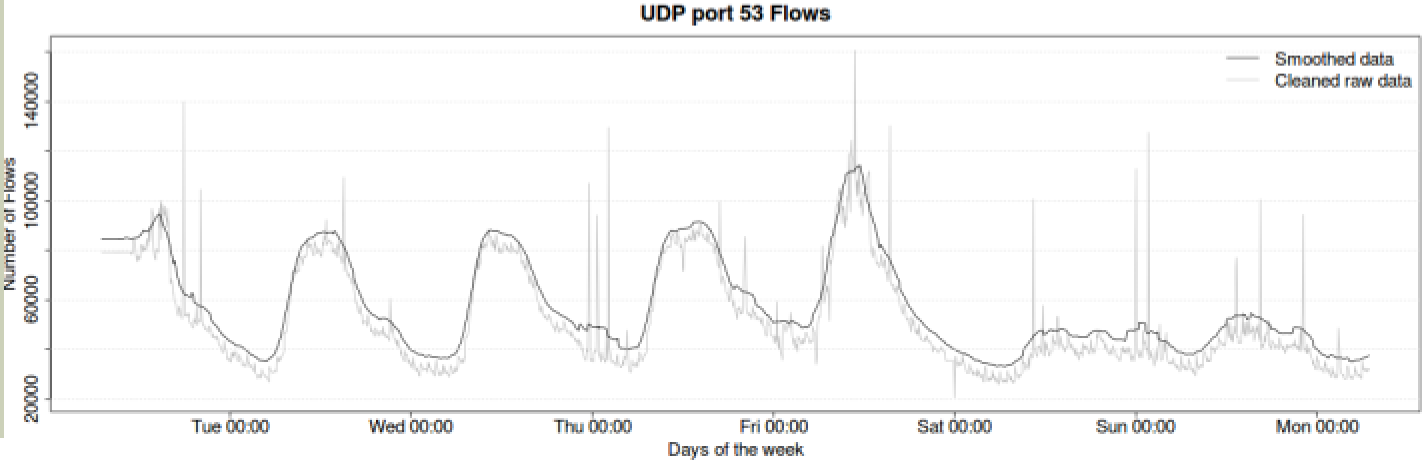
Smoothing: (friedman)



IQR rule for outliers:



Smoothing + offset:

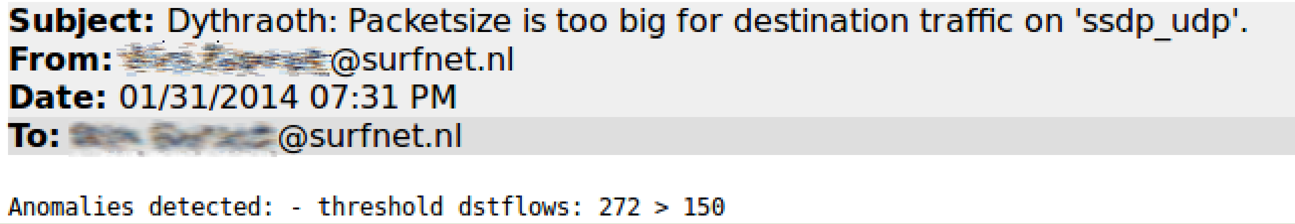


ANALYSIS (CONT.)

- For the other categories our statistical analysis was not as effective
  - Traffic without noise -> baseline but hand-picked offset
  - Non-regular traffic -> threshold

OUR PROTOTYPE

- NfSen plugin written in Perl and HTML/PHP
  - Run every five minutes
  - Run-time: 10 seconds
- Baselines and configuration stored in a SQLite database
- Adaptive baseline
  - Weighting value
- E-mail alerting



Subject: Dythraoth: Packetsize is too big for destination traffic on 'ssdp_udp'.
From: [redacted]@surfnet.nl
Date: 01/31/2014 07:31 PM
To: [redacted]@surfnet.nl

```
Anomalies detected: - threshold dstflows: 272 > 150
```

CONCLUSION

- What kind of DDoS attacks can we detect?
  - We can detect anomalies based on high volume. However…
  - Verified for profiled application protocols and rest.
  - Due to constraints, we didn’t dive into low-rate anomalies.

- Can we detect them in near real-time?
  - Yes, within a 5 minutes interval (or even faster)

- Can we extract enough information for mitigation?
  - No, but we expect that to be possible with further development of the plugin

FUTURE WORK

- Automate analysis
- Gather more information to detect the type of the anomaly
- Make the model distributed
- Integration with a mitigation system

Cool, right?

THANK YOU!