

PA197 Secure Network Design

8. Network Monitoring

Eva Hladká, Luděk Matyska

Faculty of Informatics

March 26, 2019

Content

- 1 Traffic monitoring (principles and tools)
- 2 Traffic analysis
 - Tools
- 3 Netflow
 - Principles
 - IPFIX
 - Advantages and usability
- 4 Network Behavior Analysis
 - DDoS vs flash crowds

Traffic Monitoring—Principles

- Continuously monitor the computer network
- Collect information
- Perform analysis
- Send alerts
- Part of **network management**
- Wider scope than IDS
 - “natural” causes of network problems
- Traffic monitoring versus service monitoring
 - e.g. status of a particular web server

Why Traffic Monitoring?

- Information about flows in the network
 - to improve Quality of Service
 - to get global view on flows
 - flow between different networks
 - bandwidth optimization for content providers
- Information about applications and frequency of their use
 - to tune network parameters to get better performance
- To group users sharing the same network
- To allow smart logging
 - conforms to the law
 - optimize log files
- To have sufficient data for experiments
 - traffic generators
- **To detect malicious traffic**

Traffic Classification

- By port
 - applications operating on fixed port numbers
 - simple
 - unreliable
- Deep packet inspection (DPI)
- QoS based
 - rather unreliable
- Statistical classification
 - remember IDS?

Kinds of Tools

- **Diagnostic tools**
 - usually **active**
 - connectivity and reachability tests
- **Monitoring tools**
 - active or passive
 - run “on background”
 - collect events (**passive**)
 - initiate own probes (**active**)
- **Performance tools**
 - flow monitoring

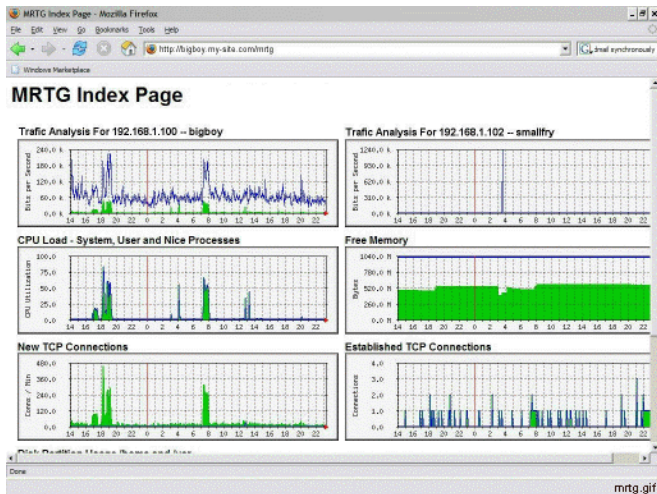
Sample Advanced Tools

- MRTG
- Wireshark
- ntopng
- SolarWinds
- ...

MRTG

- Multi Router Traffic Grapher
- Free software to monitor and measure traffic load on network links
 - written in Perl
 - available on Linux, OS X, MS Windows, UNIX, . . .
- Uses SNMP calls to send requests
 - only SNMP-enabled devices could be monitored
- Creates an HTML document with the list of graphs to display traffic from selected devices

Sample MRTG



Wireshark

- An open source packet analyzer
 - free software (under GNU GPL)
 - available for Linux, OS X, MS Windows
- Similar to tcpdump, with with extensive GUI
- Understands the structure of many network protocols
 - protocol field parsing
- Uses promiscuous mode on the monitored interface

ntopng

- Next generation of ntop
 - see <http://www.ntop.org>
- Network traffic probe
 - shows network usage in a similar way to the UNIX top command
- Use of web browser interface
 - ntop servers as a web server
- Features (selected)
 - sort network traffic according to the protocols used
 - show IP traffic distribution among various protocols
 - analyze IP traffic and sort it
 - display IP traffic subnet matrix (who is talking with who?)
 - geolocate hosts
 - store traffic statistics in RRD format

SolarWinds

- A commercial product
 - <http://www.solarwinds.com>
- Extensive suite of monitoring tools
 - multi-vendor fault, performance, and availability monitoring
 - dynamic network maps
 - customizable topology and dependency-aware intelligent alerts
 - automated capacity forecasting, alerting, and reporting
 - deep packet inspection and analysis
- Also other products
 - applications and system optimization
 - database performance acceleration
 - security and compliance enhancement

SolarWinds example screen

[Export to PDF](#)
 Thursday, July 17, 2014 12:48:50

Router - All Details

Average Response Time & Packet Loss

Avg Resp Time

Packet Loss

Top 10 Flapping Routes

DESTINATION NETWORK	COSE	FLAPS	NEXT HOP	PROTOCOL	LAST CHANGE
10.0.0.0	24	10	Router2 lab.localOSPF		about 1 day ago (7/15/2014 5:57:04 PM)
10.0.0.0	24	7	Router2 lab.localBGP		about a day ago (7/15/2014 8:57:51 PM)
10.0.0.0	24	4	Router3 lab.localOSPF		about a day ago (7/16/2014 2:57:14 AM)
10.0.0.0	24	4	Router2 lab.localBGP		about 2 days ago (7/14/2014 5:58:47 PM)
10.0.0.0	24	3	Router2 lab.localBGP		about 2 days ago (7/14/2014 2:58:47 PM)

Min/Max/Average Response Time

Core Router

RESPONSE TIME IN MILLISECONDS

12:00 AM 3:00 AM 6:00 AM 9:00 AM 12:00 PM

12:00 PM 3:00 AM 6:00 AM 9:00 AM 12:00 PM

- Average Response Time Core Router
- Percentile 95%
- Min/Max Response Time Core Router

Average CPU Load & Memory Utilization

Avg CPU Load

Memory Used

Default Route Changes (1 records)

NEXT HOP	ROUTE CHANGE	TIME OF CHANGE
11.0.0.1	added	about 10 hours ago (7/16/2014 6:50:17 PM)

Network Latency & Packet Loss

Core Router

RESPONSE TIME IN MILLISECONDS

100 ms

0 ms

6:00 PM 17 Jul 6:00 AM 12:00 PM

12:00 PM 3:00 AM 6:00 AM 9:00 AM 12:00 PM

12:00 PM 3:00 AM 6:00 AM 9:00 AM 12:00 PM

- Response Time Core Router

Hardware Details

Hardware Status: **Critical**

Manufacturer: Cisco

Model: PT2004S1000V

Service Tag: 3N17014 12:47:30 PM

Last Poll Time: 3/17/2014 12:47:30 PM

Current Hardware Health

SENSOR NAME	STATUS	VALUE
Fan	Critical	
Power Supply	OK	
Temperature	OK	

Netflow Origin and Principles

- Introduced by CISCO
- Available at CISCO routers to collect IP traffic at interfaces
- Analysis of netflow traffic can help
 - to determine source and destination of traffic
 - class of service
 - congestion
- Components
 - **flow exporter**
 - router: aggregates packets into flow
 - sends them to collector
 - **flow collector**
 - reception, storage and preprocessing
 - **analysis application**

Network Flow

- NetFlow version 5
- Flow is a unidirectional sequence of packets that all share the following 7 values:
 - ① ingress interface
 - ② source IP address
 - ③ destination IP address
 - ④ IP protocol
 - ⑤ source port for UDP or TCP; 0 for other protocols
 - ⑥ destination port for UDP or TCP; type and code for ICMP; 0 for other protocols
 - ⑦ IP type of service

Routing information is not included as it may change during flow lifetime (e.g. due load balancing)

- Also user defined key are allowed in advanced implementations

Sampled NetFlow

- NetFlow designed to process all packets
 - router implementation
- Performance implications for high bandwidth links
- **Sampled NetFlow**
 - only one packet in n is processed
 - **deterministic** sampling: exactly each n -th packet
 - **random** sampling
 - more complex patterns
 - per flow sampling
 - sampling rate per router or per interface
- Sampling introduces errors
 - INVEA-TECH probes for wire speed at multigigabit networks

IPFIX

- **IP Flow Information Export**
- IETF protocol
- Standard of export for IP flow information from routers, probes, ...
- Based on NetFlow version 9
- Defined in the following RFCs: 5103, 7011–7015
- IPFIX flow
 - packets that share same properties observed in a specific timeframe
- Basic Architecture contains
 - **metering process** collects data at an **observation point**
 - **exporter** sends collected flow information to a **collector**
 - A **many-to-many** relationship exists between collectors and exporters
 - IPFIX is **push protocol**

Advantages

- Unobtrusive
 - the attackers can't detect flow monitoring
 - can slow down high traffic bandwidth
 - esp. not sampled monitoring
- Relatively easy to implement
 - information taken from routers
 - probes in the network
- Substantial processing power required
 - for real-time monitoring
 - more extensive analysis possible off-line for limited time periods

Usability

- Observing limits and security policies
 - users' compliance with network use policy
 - service use (for network optimization)
- QoS monitoring
 - passive, but potentially biased
- Traffic accounting

Security related

- P2P network/service detection
- IP port scanning detection
 - TCP RESET packets increase for vertical scan
 - high increase of ICMP Host Unreachable packets for horizontal scan
- DoS attacks detection
 - e.g. TCP SYN-flood attack
 - flash-crowd effect (see later)
- Worms and viruses spread detection
 - high number of unexpected open connections to other computers

Network Behavior Analysis

- Detection of unusual actions through traffic monitoring
- Monitor network inside an organization
 - many monitoring points
 - aggregation
 - trends spotting
 - including e.g. bandwidth fluctuation
- Machine learning methods
 - **what is normal** behavior?
- Complements IDS, firewalls, ...

Anomaly Detection

- Basic steps
 - uses history of traffic observation to build a model of selected relevant characteristic of network behavior
 - predict these characteristics for the future traffic
 - identify the source of discrepancy between predicted and measured values
- Adaptable, no limit for the detection strength
 - artificial intelligence approach
- Error rate the main potential problem
 - single NBA methods usually prone to high number of false negatives
 - multistage collaborative methods, trust modeling etc. used to overcome this shortcoming

DDoS vs flash crowd

- Web server example
 - highly variable usage patterns
 - unexpected increase in the traffic
 - attack or information attractiveness
- DDoS attack
 - malicious activity
 - aim to shutdown the web server
 - distributed access patterns
- **Flash crowd** (Slashdot effect)
 - massive increase of traffic to a web site
 - due to sudden interest
 - often through **linking** from a popular site
- Difficult (impossible?) to distinguish

Distinguishing Flash Crowds

- An example taken from the following article
 - Ke Li et al (2009): Distinguishing DDoS Attacks from Flash Crowds using probability Metrics. *Network and System Security NSS'09*, pp. 9–17, DOI 10.1109/NSS.2009.35
- Differences between Flash crowds and DDoS attacks
 - intent: users want content, DDOS wants the site shut down
 - users coming from the whole community network or the whole Internet
 - aggregated source IP addresses resemble flat fractional Gaussian noise distribution
 - DDOS from attackers/botnet
 - aggregated source IP addresses follow Poison distribution
 - difference in traffic increase/decrease
 - users follow the spread wave (gradually increase the traffic)
 - attackers use rather short time frame during the initial phase of attack

The Basic Theory

- Based on a hybrid probabilistic method
 - using similarity between flows to distinguish normal versus flash crowd versus DDoS flows
 - similarity measured as

$$\rho(P, Q) = \sum_{i=1}^n \sqrt{p_i q_i}$$

where $P = (p_1, p_2, \dots, p_n)$ and $Q = (q_1, q_2, \dots, q_n)$ are two probability distributions

$\rho(P, Q) = 1$ for $P = Q$ and $\rho(P, Q) = 0$ when P and Q are

- total variation calculated as

$$T(P, Q) = \sum_{i=1}^n |p_i - q_i|$$

The Algorithm

- The algorithm (applied at the last router preceding the server)
 - set grouping thresholds GT_S (similarity) and GT_T (variance); each threshold has an lower and upper bound
 - calculate probabilistic distribution for each aggregated flow
 - calculate total variation $T(P, Q)$ and similarity $\rho(P, Q)$ for each two flows
 - if $T > \text{upper}(GT_T)$ and $\rho < \text{lower}(GT_S)$ the DDoS is detected from Flash crowds
 - if $\text{lower}(GT_T) \leq T \leq \text{upper}(GT_T)$ and $\text{lower}(GT_S) \leq \rho \leq \text{upper}(GT_S)$ then DDoS is detected from Normal flow
 - if $T \leq \text{upper}(GT_T)$ and $\rho > \text{lower}(GT_S)$ than Flash crowds is detected form Normal flow
 - otherwise Normal flow is assumed
 - The values for upper and lower band of thresholds GT_T and GT_S was derived from simulations and are (0, 5921, 1.1045) and (0.7220, 0.8708), resp.

A different method

- Based on article
 - P.R.Reddy et al (2013): Techniques to Differentiate DDoS Attacks from Flash Crowd. Int. J. Adv. Res. Comp. Sci. Soft. Eng., Vol 3(6), pp. 295–299.
- Uses **flow correlation coefficient**
- Similar observations as above
 - individual attack flows show an internal similarity—flow standard deviation is usually smaller than that of genuine flash crowd flows
 - smaller number of botnet nodes compared to number of genuine flash crowd users
 - each botnet node must initiate higher number of attack flows to mimics the expected number of users

Metrics used

- Flow correlation coefficient
- Packet arrival patterns
- Information distance
- In all cases, the differentiation is based on smaller variance in DDoS attack flows
 - the correlation coefficient use experimentally verified as the most promising metrics

Summary

- Traffic monitoring as a very strong mechanism
 - unobtrusive
 - not detectable by attacker
- Usable in a large range of scenarios
 - performance as well as security related
- Support from network elements needed
 - probes
 - router implementation
- NetFlow and IPFIX
- Network behavior analysis
 - example of DDoS versus Flash crowd detection
- Next session: Wireless Networks (introduction)