Using of time characteristic in Netflow data for improvement of protocol detection

P. Piskač, J. Novotný,

`{piskac|novotny}@ics.muni.cz`

3rd NMRG Workshop on NetFlow/IPFIX Usage in Network Management
July 30, 2010, Maastricht, The Netherlands

❶ **Motivation**

❷ **Tools**

❸ **Evaluation**

❹ **Conclusion and future work**

1. **Motivation**

2. Tools

3. Evaluation

4. Conclusion and future work

Motivation

- The knowledge of network protocol distribution is very important for security applications on a computer network.
- For example - botnets represent some kind of communication with similar behavior and use small sets of network protocols.
- Information about protocols can be gathered from NetFlow but:
  - protocol recognition based only on port numbers is weak and can be simply compromised,
  - doesn't work on tunneled data.
- Despite of these disadvantages, it is possible to use NetFlow, but it needs to be extended by some other information.

Methods for extending protocol detection

- Better results can be achieved using deep packet inspection (e.g. Snort application), which:
  - + achieves good results,
  - – needs a lot of computational power, which is an issue on high speed networks,
  - – doesn't work on encrypted communication.
- Other ways **to extend** NetFlow analysis:
  - header analysis (L7 . . .),
  - analysis of first packets in a flow,
  - **methods based on time characteristic.**

Work goals

- Check protocol detection based on time characteristic analysis.
- The goals were achieved in the following steps:
  1. select and explore one protocol from packet and flow point of view,
  2. find out possibilities of detecting selected protocol using information about time characteristic,
  3. implement detection methods,
  4. create a plug-in for NfSen,
  5. make experiments.

1. Motivation

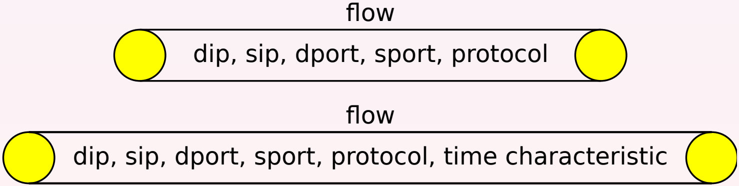
2. **Tools**

3. Evaluation

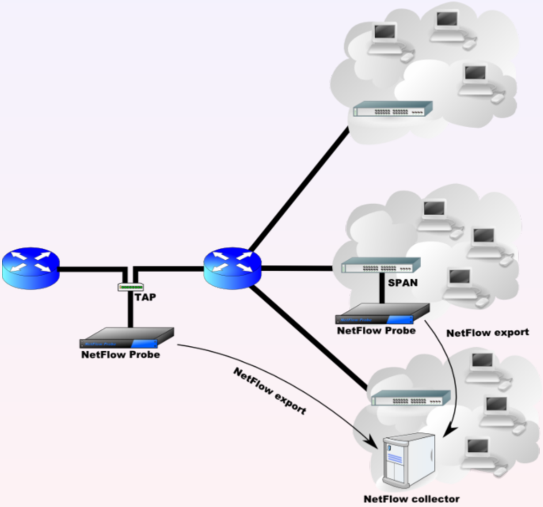
4. Conclusion and future work

Time characteristic

- Time characteristic is calculated from inter-packet gaps in a flow.
- Time characteristic of packet a flow consists of:
    - accurate time stamp of the flow begin,
    - accurate time stamp of the flow end,
    - minimal inter-packet gap in the flow,
    - maximal inter-packet gap in the flow,
    - average inter-packet gap in the flow,
    - standard deviation of inter-packet gap in the flow.


flow
dip, sip, dport, sport, protocol
flow
dip, sip, dport, sport, protocol, time characteristic

NetFlow data collecting



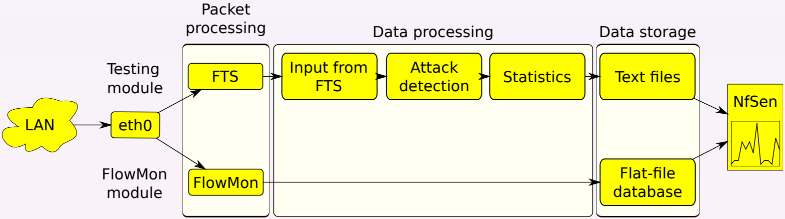
NfSen

- NfSen is an open source graphical web based front end for the nfdump NetFlow tools.
- NfSen allows you to:
  - display your NetFlow data: Flows, Packets and Bytes using RRD (Round Robin Database),
  - easily navigate through the NetFlow data,
  - process the NetFlow data within the specified time span,
  - create history as well as continuous profiles,
  - set alerts, based on various conditions,
  - write your own plug-ins to process NetFlow data on a regular interval.
- There is no necessary to develop any new tool, but we can just use NfSen with appropriate plug-in for data processing.

Getting extended NetFlow data

- Existing infrastructure of Masaryk University uses FlowMon probes and some CISCO routers. Both of them don't provide details about time characteristic.
- Time resolution is 1ms in standard NetFlow data. It is too imprecise for time characteristic.
- Flow Time Statistics (FTS) was used to get NetFlow data extended by time characteristic.
- FTS is testing tool for Liberouter project - it is not final solution suitable for real deployment.
- Important goal of the proposed work is to prove reason for extension FlowMon probes to generate time characteristic.

FTS connection



1. Motivation

2. Tools

3. **Evaluation**

4. Conclusion and future work

Choosing a protocol

- As the test protocol was chosen SSHv2 protocol because:
  - attacks (especially dictionary) on this protocol represent security threat, which should be detected,
  - the information about amount of SSH connections in a traffic is important from security reasons,
  - SSH is an open and well know protocol,
  - SSH can be used for botnet control.

Protocol detection

- Detection works on comparison two vectors - pattern vector and unknown connection vector.
- A vector is created from extended flow information.
- Data included in a vector:
    - information about time characteristic,
    - number of transferred bytes and packets,
    - information about 3rd and 4th network layers.
- Key issue is to find pattern vector - for test purposes it was created by “hand” using data observation.

Choosing of pattern vector

- Pattern vector can be chosen from real or testing environment.
- Testing environment minimizes latency and other network influences.
- Real environment uses data with a lot of different influences. It makes finding of the right vector more complex (according "noise" in data).
- Pattern vector for SSH protocol has been chosen from testing environment according to results of the tests.

Operations with vectors

- There is a lack of information about any method used for time characteristic in the literature.
- We need to use methods from other area.
- Vectors were compared using:
  - average distance between vectors $d(p,q) = \frac{\sum_{i=1}^{N}(\|p_i - q_i\|)}{N}$,
  - root-mean-square distance $d(p,q) = \sqrt{\frac{\sum_{i=1}^{N}(p_i - q_i)^2}{N}}$,
  - euclidean distance $d(p,q) = \sqrt{\sum_{i=1}^{N}(p_i - q_i)^2}$,
  - angle between vectors $d(p,q) = \frac{\sum_{i=1}^{N}(p_i \times q_i)}{\sqrt{\sum_{i=1}^{N}(p_i^2)}\sqrt{\sum_{i=1}^{N}(q_i^2)}}$.

Test results

- We were not capable to classify SSH protocol because user interaction brings a lot of random data, that countermeasures all vectors.
- But the tests show, that there is a possibility to detect some dictionary attacks on SSH.
- Detection of dictionary attacks was chosen to prove the method, which uses NetFlow data extended by time characteristic.

Accuracy of dictionary attacks detection

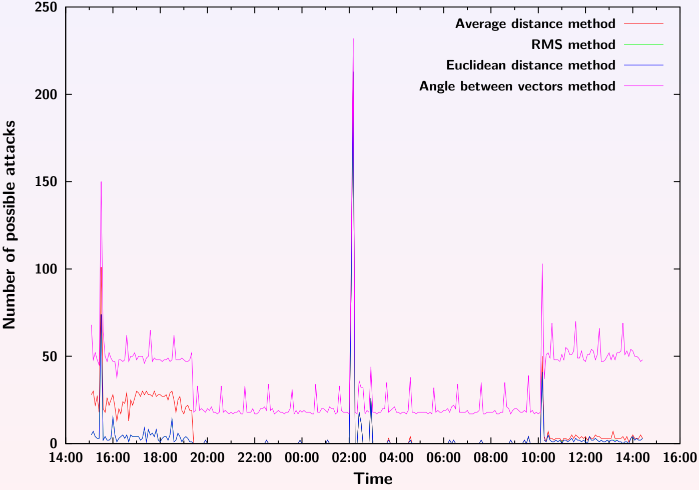
Pattern	Average distance		RMS Distance	
	TAR[1]	FAR[2]	TAR	FAR
	%	%	%	%
Testing	91	8	91	10
Real	88	3	88	3

Pattern	Euclidean metrics		Angle between vectors	
	TAR	FAR	TAR	FAR
	%	%	%	%
Testing	91	10	94	25
Real	87	2	78	19

[1] TAR - True Acceptance Rate

[2] FAR - False Acceptance Rate

Practical example



1. Motivation

2. Tools

3. Evaluation

4. **Conclusion and future work**

Conclusion

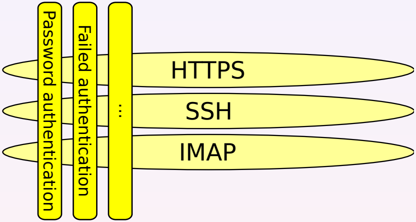
- This field of interest has not been deeply explored yet.
- Some protocols (e.g. HTTPS, IMAP) are very similar to SSH from time characteristic point of view.
- Vector comparison methods give very similar results with exception of angle between vectors method.
- It has been explored, that password based authentication protocols look very similar.
- This method works for revealing dictionary attacks.

Future work

- Extend probes and all NetFlow monitoring infrastructure by:
    - time characteristic support,
    - more precise resolution of NetFlow time information,
    - IPFIX for data export,
- Make tests on high speed networks.
- Extend test vector by minimal, maximal, average and standard deviation of packet size,
- Look for other information, which can improve protocol detection.
- Implement adaptable vectors.

Future work - continue

- Categorize protocols (and their variants) into groups according to their time characteristic.

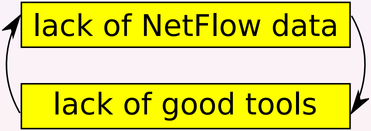


- Use huge randomness in time characteristic of SSH protocol for its detection.
- Detect other protocols, i.e. VOIP, P2P, IRC (botnet controlling)...

Conclusion and future work

Information about time characteristic represents interesting method for protocol detection, which deserves deeper inspection.

Deadlock similar to origin of NetFlow deployment.



lack of NetFlow data

lack of good tools

Try to break the deadlock.

Thank you for your attention!

Using of time characteristic in Netflow data for improvement of protocol detection

Questions?

Pavel Piskac et al.
piskac@ics.muni.cz

Project CYBER
Project code: OVMASUN200801
http://www.muni.cz/ics