

Automaton Models for Netflow Analysis Fingerprinting and Classifying Participants

NMRG Workshop, Prague, Czech Republic Friday, July 24th 2015

Christian A Hammerschmidt, christian.hammerschmidt@uni.lu

Interdisciplinary Centre for Security, Reliability and Trust University of Luxembourg



Automaton Models

Short Overview





Fingerprinting with Automatons

Prediction, Classification, and Visualization (I)



Prediction

- predicting next states
- detecting outliers and anomalies

Classification

classifying flows

identifying type of activity or infection

unsupervised

(semi-) supervised

Fingerprinting with Automatons

Prediction, Classification, and Visualization (I)



Prediction

Classification

- predicting next states
- detecting outliers and anomalies

- classifying flows
- identifying type of activity or infection

unsupervised

(semi-) supervised

Fingerprinting with Automatons

Prediction, Classification, and Visualization (II)



animation of automaton

Challenges

NetFlow Data as a (Regular) Language



Header	←NetFlow Version 9 Header: 32 bits	→		
First Template FlowSet	Version 9 Count = 4 (FlowSe	ts)		
Template Record	System Uptime			
First Record FlowSet	UNIX Seconds			
(Template ID 256)	Package Sequence			
First data Record	Source ID			
Second Data Record				
Second Template Flow Set	←Template FlowSet 16 bits →	<−Data Flow	/Set: 32 bits →	
Template Record	FlowSet ID = 0	FlowSet	Length =	
Template Record	Length = 28 bytes	► ID = 256	64 bytes	
Second Record FlowSet	Template ID = 256	 192.16 	8.1.12	
(Template ID 257)	Field Count = 5	 10.5.1 	2.254	
Data Record	IPv4_SRCADDR (0x0008)-///	 192.16 	192.168.1.1	
Data Record	Length = 4 /// 4	► 50	5009	
Data Record	IPv4_DSTADDR (0x000C)-/1//	► 5344	5344365	
Data Record	Length = 4 ///	192.168.1.27		
	IPv4_NEXT_HOP (0x000E)-///	10.5.12.23		
	Length = 4 //	192.16	192.168.1.1	
	PKTS 32 (0x0002)	74	748	
	Length = 4	3889	388934	
	BYTES 32 (0x0001)	192.16	192.168.1.56	
	Length = 4	10.5.12.65		
		192.16	58.1.1	
		5	5	
		653	34	

¹http://www.cisco.com/c/dam/en/us/td/docs/ios/ipv6/configuration/ guide/ip6-netflow_v9.fm/_jcr_content/renditions/ip6-netflow_v9-1.jpg

C. Hammerschmidt (SnT)

Automaton Models for NetFlows

1

Challenges

NetFlow Data as a (Regular) Language



From regression of numeric values to classification:

via clustering to obtain few representatives

or through discretization

via binning to obtain a discrete state space

What to choose?

Method Learning State Structure from Data





²Taken from [2]

C. Hammerschmidt (SnT)

Automaton Models for NetFlows

SnT 2015-07-24 6 / 13



Experiments (on time-aggregated flow data):

- 1. predicting statistics for next flows
- 2. classifying flows on unlabeled data
- 3. classifying flows on labeled data³

³Using a botnet traffic data set[1]

Evaluation

Generated Automatone





Evaluation Excerpt



Data Set Experiment Error / F₁ / FPR

C. Hammerschmidt (SnT)

Automaton Models for NetFlows

Conclusion

Conclusion and Future Work



Results

- structure learning on netflow data is feasible
- initial results look very promising
- this is still work-in-progress and offers a number of ways to improve

Further Research

- compare performance to other fingerprinting solutions
- apply a more expressive automaton model

Conclusion

Conclusion and Future Work



Results

- structure learning on netflow data is feasible
- initial results look very promising
- this is still work-in-progress and offers a number of ways to improve

Further Research

- compare performance to other fingerprinting solutions
- apply a more expressive automaton model

Future Work and Extensions

Currently Ongoing Research





⁴Taken from [2]

C. Hammerschmidt (SnT)

Automaton Models for NetFlows

SnT 2015-07-24 11 / 13



Thank You!

Time for questions.

C. Hammerschmidt (SnT)

Automaton Models for NetFlows

SnT 2015-07-24 12 / 13

References I



- García, S. and Grill, M. and Stiborek, J. and Zunino, A. An empirical comparison of botnet detection methods Computers & Security, 2014.
- S. E. Verwer, C. Witteveen, M. M. De Weerdt.
 Efficient identification of timed automata: Theory and practice, March 2010.



Heule, M.J.H., Verwer, S.,

Software model synthesis using satisfiability solvers. Empirical Software Engineering 18, 825–856., 2013