Henry Clausen, David Aspinall, Michael Gibson

Evading stepping-stone detection with enough chaff

THE UNIVERSITY *of* EDINBURGH
informatics

EPSRC
Pioneering research and skills

The Alan Turing Institute

BT

Contribution

Large public Stepping-stone dataset:

- 90,000 connection pairs
- Chaff/delay tactics
- realistic setup

Re-evaluation of eight SSD-methods

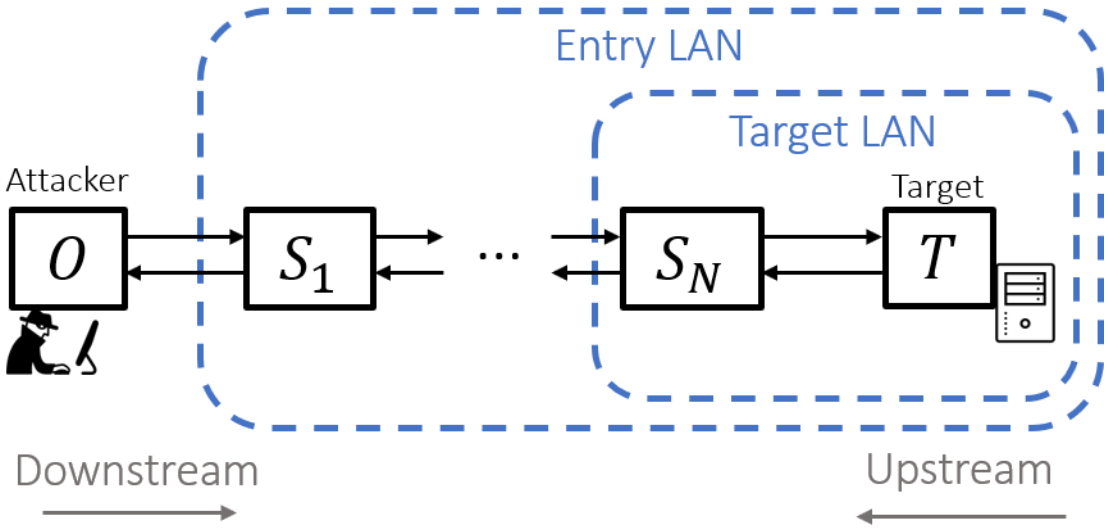
- Fair comparison of capabilities
- different settings
- Detection rates and

  AUC-scores

Stepping-stone

Relay of attack via “stepping-stone”

- Hide attack origin
- Access protected resources
- Interactive access



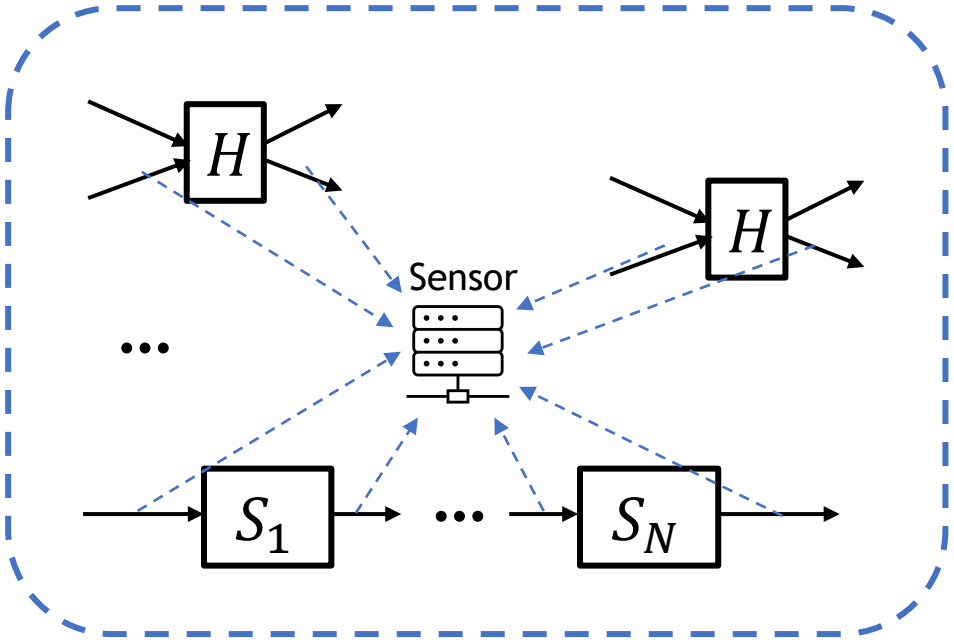
Tools

- SSH-tunnels
- Netcat backpipe
- SOCKS proxy
- …

Usually encrypted

Stepping-stone detection

- Sensor records incoming and outgoing connections
- Measure correlation between pairs



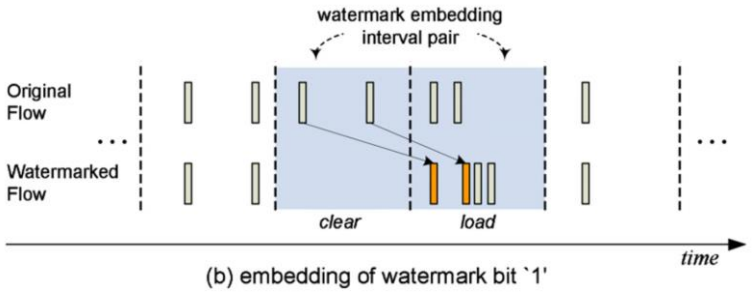
Goal

- Identify stepping-stones early before attacker
  - reaches target
  - exfiltrates data
- Trace attack back to origin

Stepping-stone detection

Most common techniques:

- Watermarking
- Packet correlation

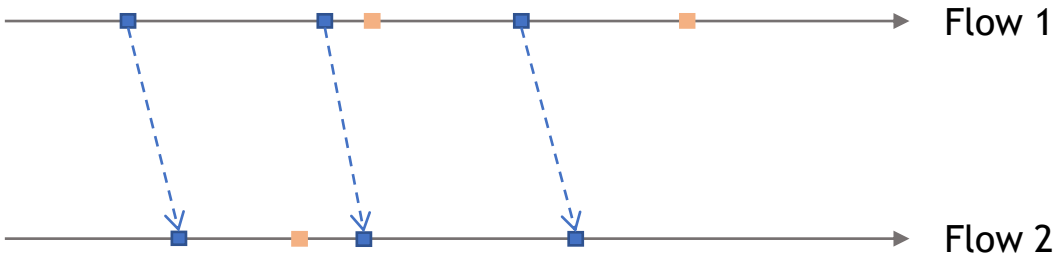


Evasive techniques

- Transfer delays
- Chaff packets

- Repacketisation
- Flow splitting

- RTT-based
- Anomaly-detection



THE UNIVERSITY of EDINBURGH
informatics

Stepping-stone detection

Most common techniques:

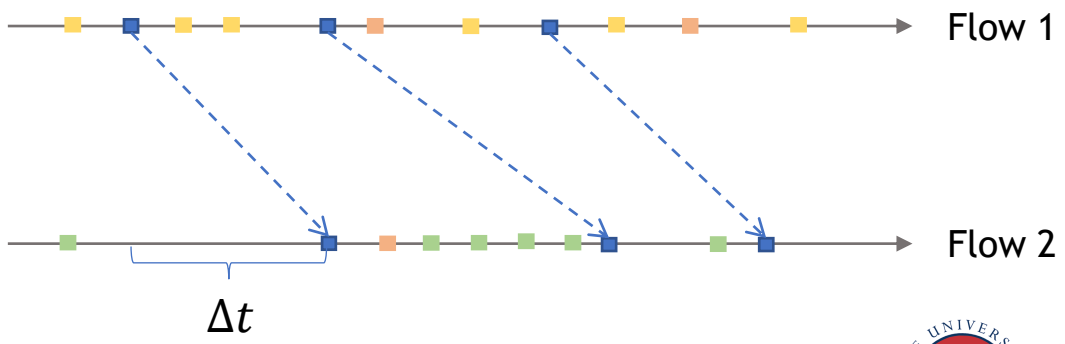
- Watermarking
- Packet correlation

- RTT-based
- Anomaly-detection

Evasive techniques

- Transfer delays
- Chaff packets

- Repacketisation
- Flow splitting



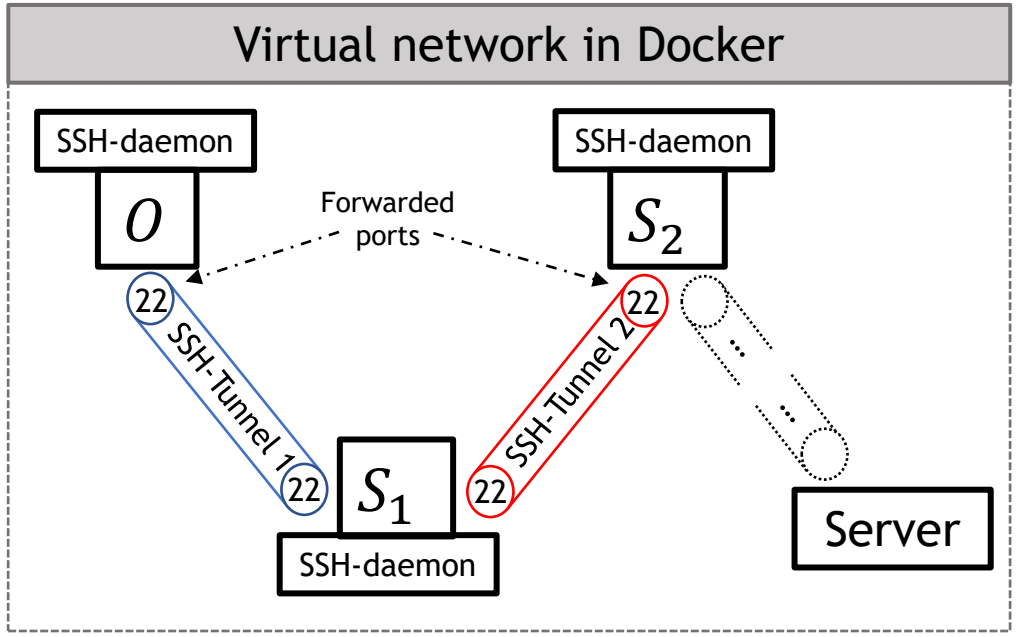
Evaluation problems

- No public data!
- Widespread use of self-generated data
  - Simplistic attack scenario
  - restrictive evasive tactics
  - Unrealistic background traffic
- No standard on number of packets
- Setup shielded from other influences

→ Impossible to compare detection rates

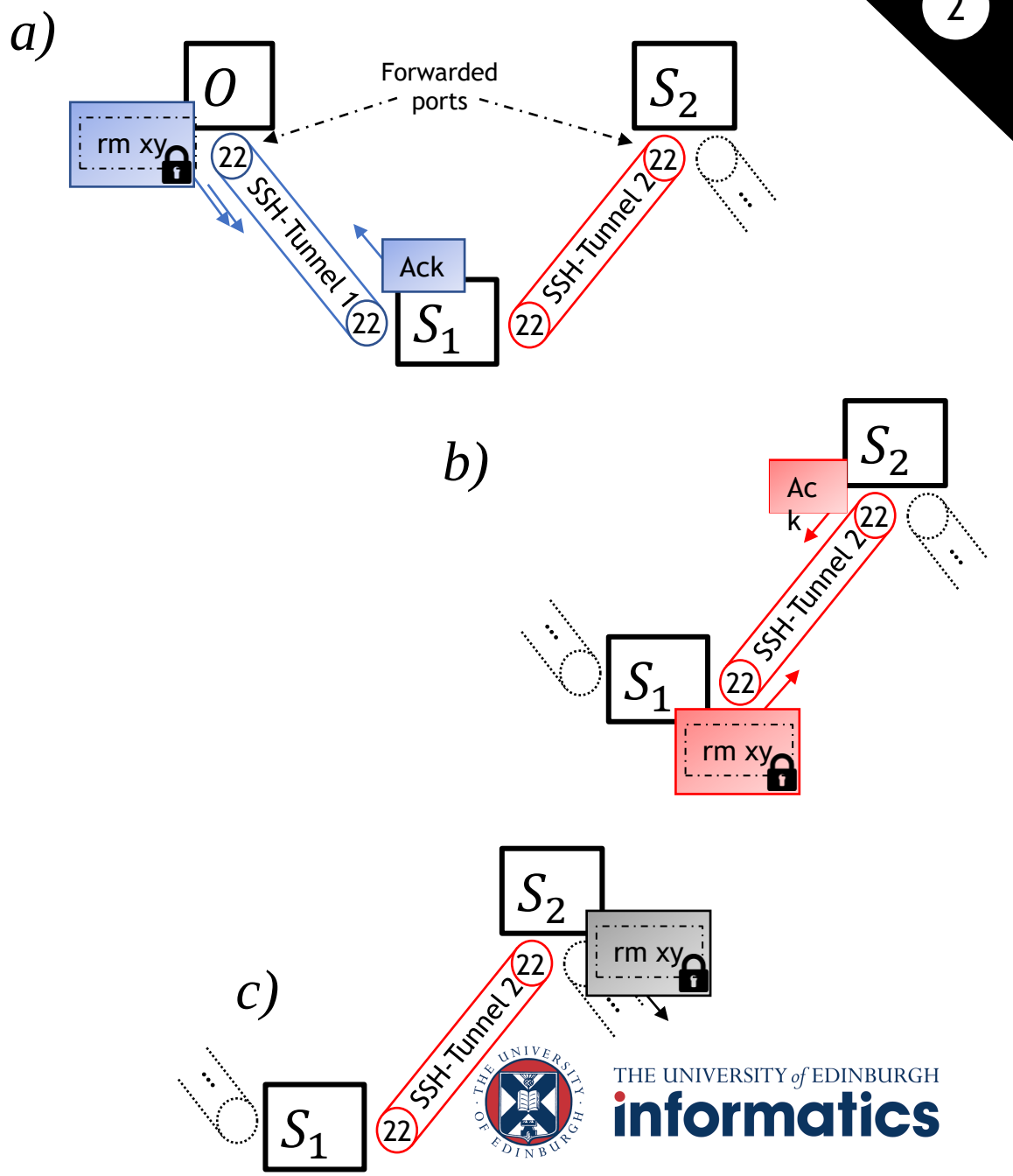
Data generation set-up

- Interactive SSH-session
  - relayed using SSH-tunnels
- SSH-script
  - commands drawn randomly
  - randomized inputs
  - sleep intervals to simulate reaction times
- Containers for reproducibility



Data generation set-up

- Interactive SSH-session
  - relayed using SSH-tunnels
- SSH-script
  - commands drawn randomly
  - randomized inputs
  - sleep intervals to simulate reaction times
- Containers for reproducibility



Data generation set-up

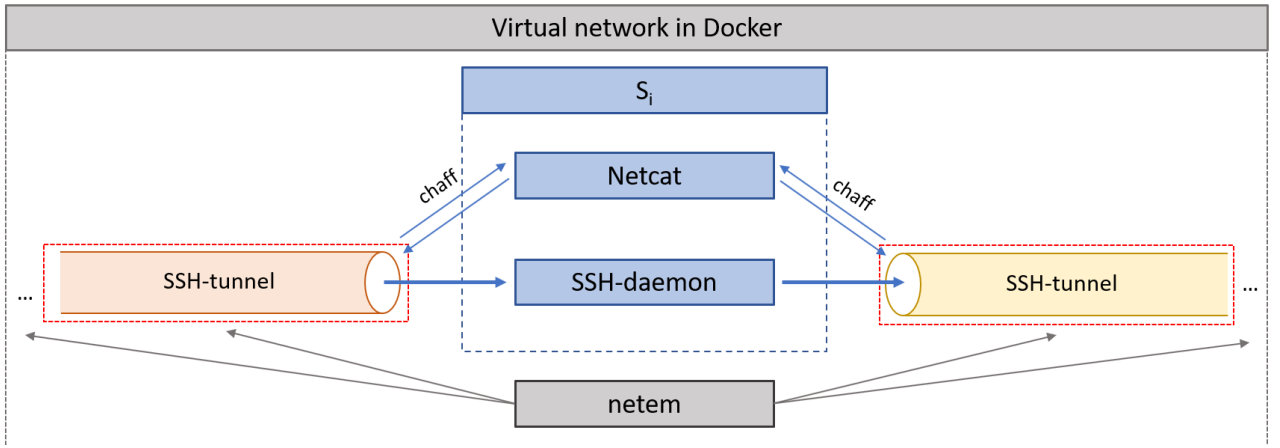
NetEm to emulate network settings

Chaff:

- Netcat
- mimics stream buffering[1]
  - Packet IAT in $[\frac{d_C}{2}, d_C]$

Jitter delays:

- NetEm
- mimics stream buffering[1]
- Δt in $[0, d_D]$
  - d_D up to 1500ms



[1] Padhye et al. (2010)

Evaluation data

Connection pairs from S_N

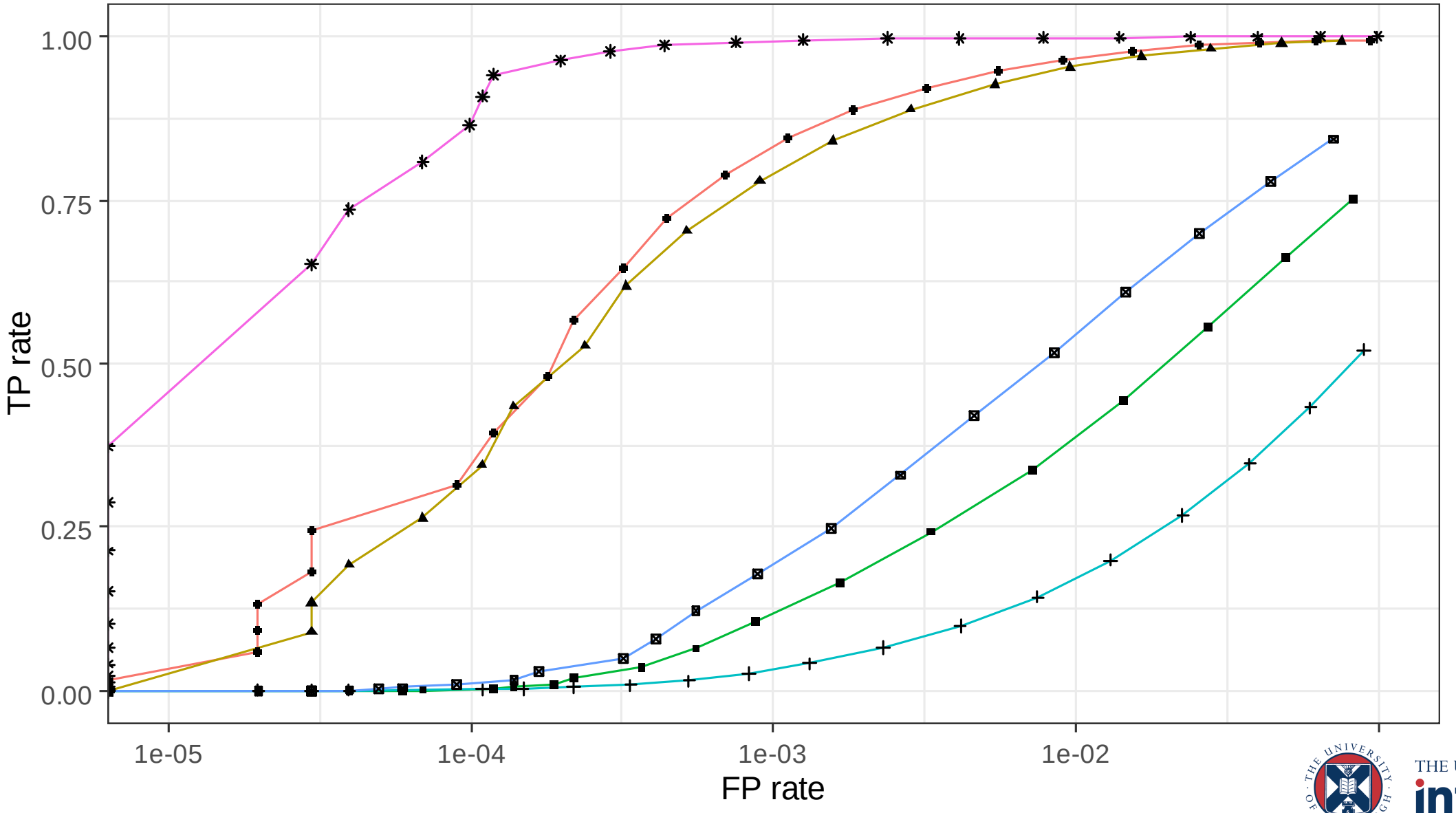
- 1,400 packets

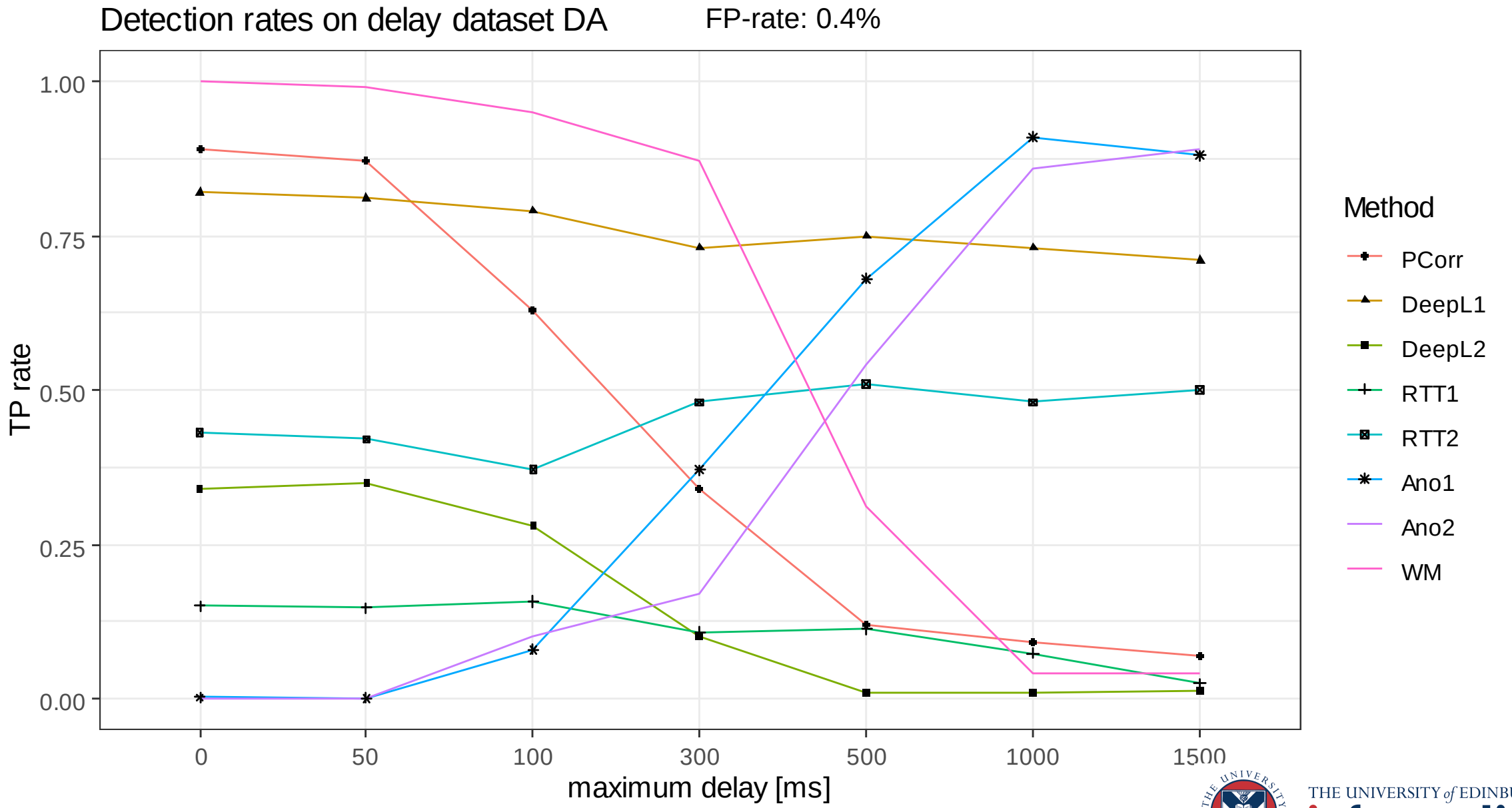
	Label	#conn	purpose
SS data	BA	30,000	Baseline attack
	DA	30,000	Delays with varying d_{D}
	CA	30,000	Chaff with varying d_{C}
Background data	CAIDA	60.000	General background
	SSH	20.000	Similar to attack commands
	Multim.	20.000	Similar to chaff pert.

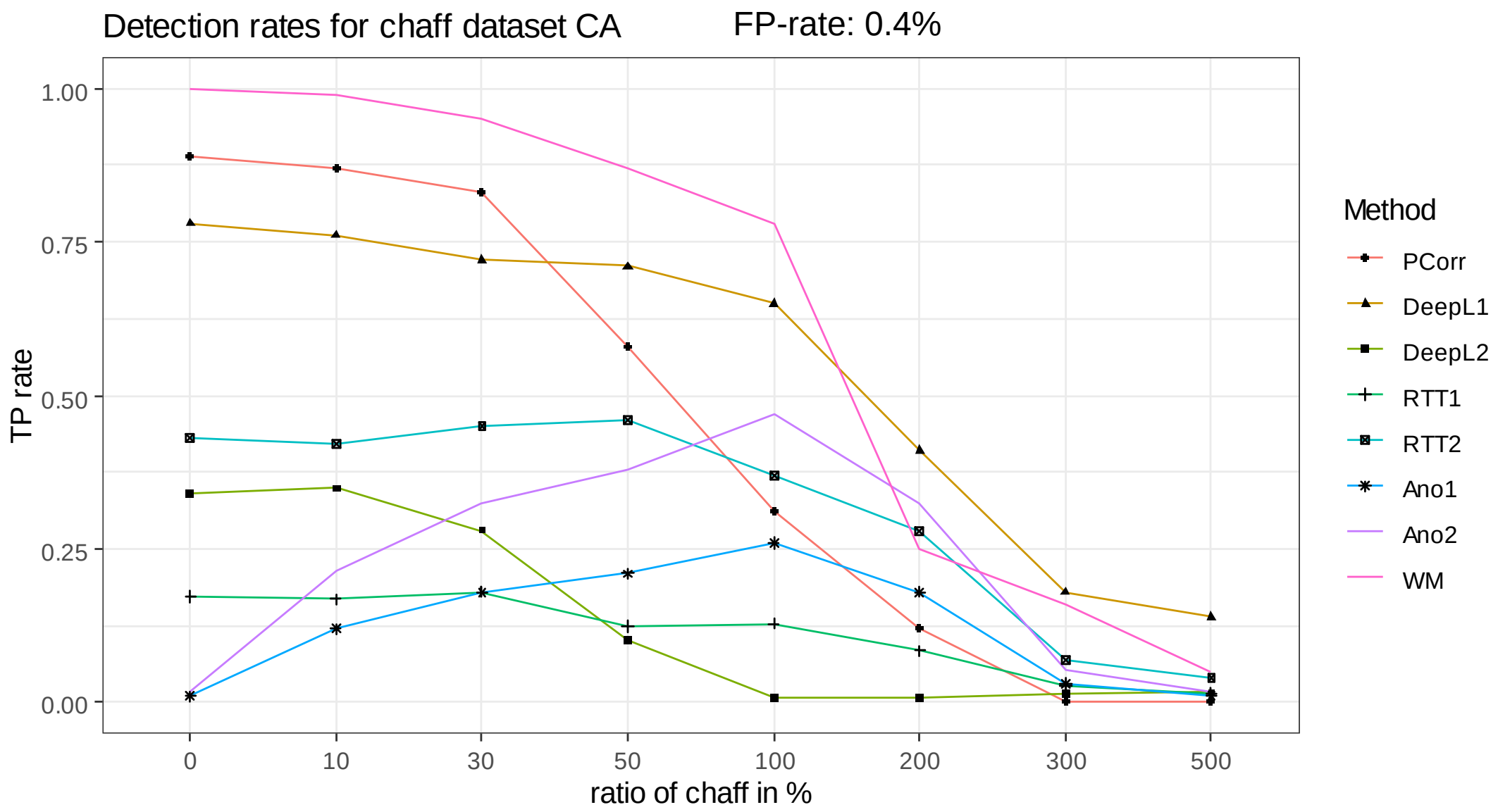
Selected methods

Label	TP	FP	Robustness	Category
PContext (2011)	100%	0%	jitter/chaff	Packet correlation
DeepCorr (2018)	90%	0.0002%	small jitter	Neural networks
WuNeur (2010)	100%	0%	-	
Rwalk (2015)	-	-	chaff	RTT-based
Crossover (2016)	85%	5%	-	
Ano1 (2011)	99%	1%	jitter/chaff	Anomaly-based
Ano2 (2011)	95%	0%	jitter/chaff	
WM (2011)	100%	0.5%	jitter	Watermarking

ROC-curves on dataset BA




Detection rates on delay dataset DA
FP-rate: 0.4%
TP rate
1.00
0.75
0.50
0.25
0.00
0
50
100
300
500
1000
1500
maximum delay [ms]
Method
PCorr
DeepL1
DeepL2
RTT1
RTT2
Ano1
Ano2
WM



Disproves chaff robustness claims by PCorr, RTT1, and both anomaly methods!

THE UNIVERSITY of EDINBURGH informatics

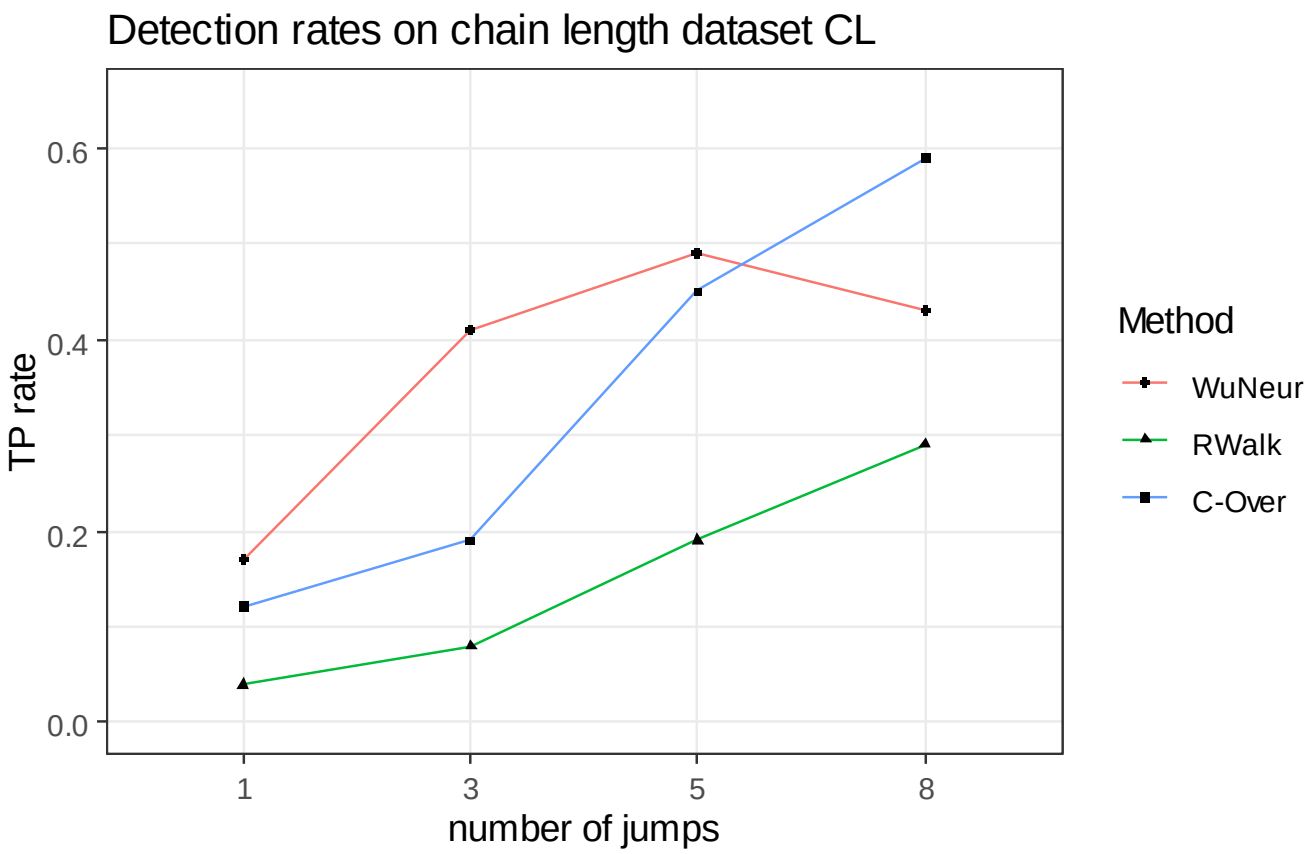
Limitations

- No behavioural/graph-based models
- No flow-splitting
- No store-forward-stepping stones
- Data might need updates for future methods

Conclusion

- Large public dataset
  - Realistic interactions
  - Evasive tactics
  - github.com/detlearsom/detgen/stepping-stone-data
- Evaluation of current state-of-the-art
  - Lower overall detection rates
  - Lack of robustness against chaff
  - Watermarking and deep-learning performs best

Additional results

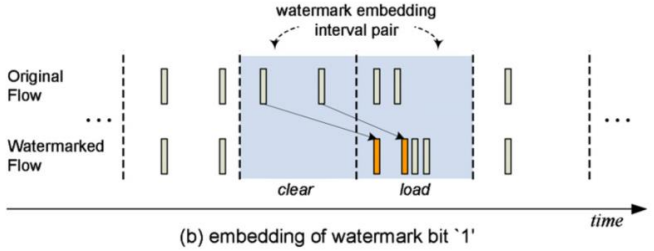


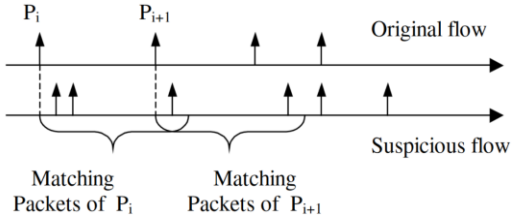
WAN-influence

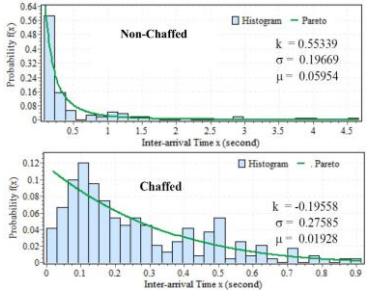
	Value	Deviation from average				
		Deep Corr	WuNeur	RWalk	COver	WM
RTT	5ms	-0.2%	+41.3%	-42.3%	-36%	+0.03%
	70ms	-5.6%	-5.8%	+35.1%	+51%	-2.2%
Packet loss	0%	+1.2%	+1.3%	+2.1%	+4.3%	+0.02%
	7%	-9.1%	-1.1%	-3.1%	-7.3%	-9.7%

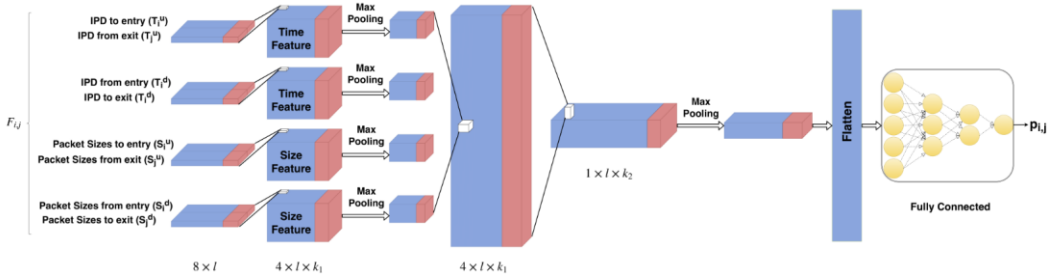
THE UNIVERSITY of EDINBURGH informatics

Models

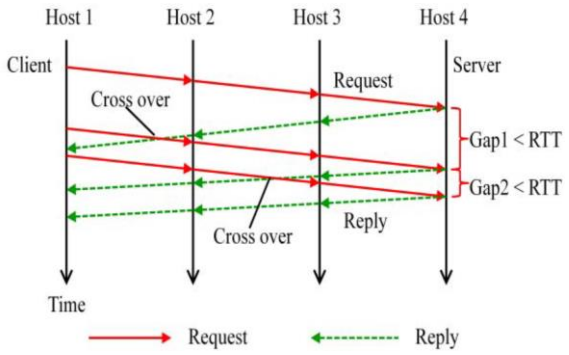


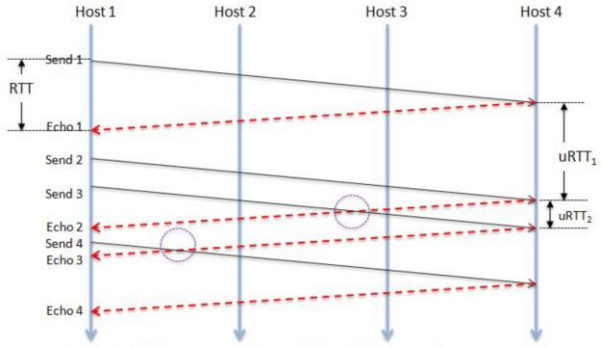


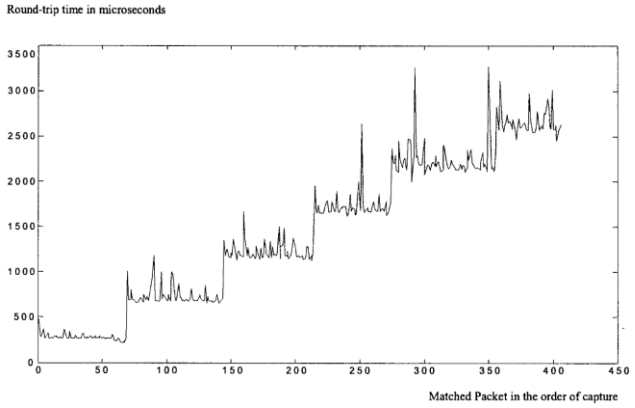




DeepCorr 2018, Nasr et al.







THE UNIVERSITY of EDINBURGH
informatics

Agenda

(1) ML and network data

(2) Problems in current datasets

(3) Containerization

(4) Traffic generation suite

(5) **Example use-case**

(6) Limitation & conclusion

Agenda

(1) ML and network data

(2) Problems in current datasets

(3) Containerization

(4) Traffic generation suite

(5) Example use-case

(6) **Limitation & conclusion**

Agenda

(1) Stepping-stones and detection

(2) Data generation process

(3) Evaluation

(4) Limitation & conclusion

Agenda

(1) Stepping-stones and detection

(2) Data generation process

(3) Evaluation

(4) Limitation & conclusion

Agenda

(1) **Stepping-stones and detection**

(2) Data generation process

(3) Evaluation

(4) Limitation & conclusion

Agenda

(1) Stepping-stones and detection

(2) **Data generation process**

(3) Evaluation

(4) Limitation & conclusion

Agenda

(1) Stepping-stones and detection

(2) Data generation process

(3) **Evaluation**

(4) Limitation & conclusion

Limitations

Not replicated well:

- Network-wide distribution
- long-term temporal structures

Data volume huge

- preprocessing required

Manual implementation

Conclusion

- Our traffic generation suite fuels ML through:
  - High degree of traffic variability
  - Ground truth labels through activity isolation
  - Scalability
  - Modularity
- github.com/detlearsom/detgen/
- Future work:
  - capture of syslogs
  - streamlined data coalescence

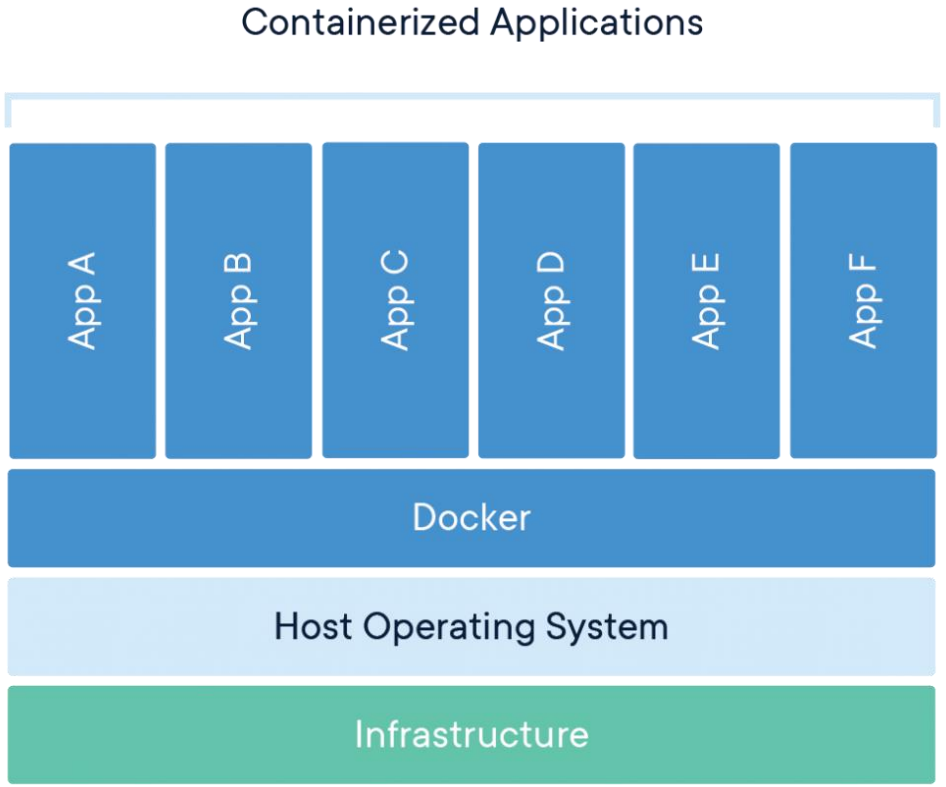
Containerization

Programs/process as standalone virtualised standard units

Advantages:

- lightweight
- runs uniformly
- safe through isolation

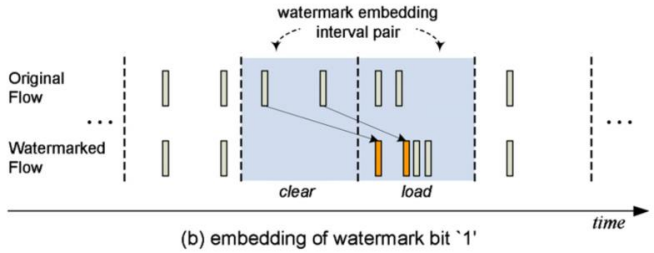
Containers can be arranged in virtual networks

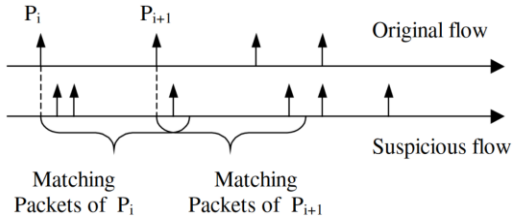


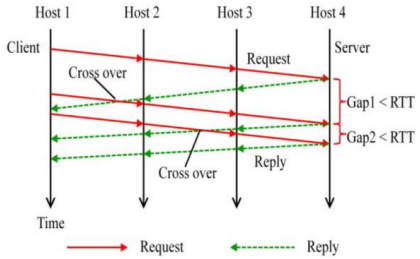
Stepping-stone detection

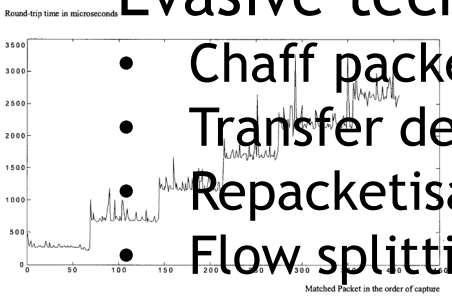
Most common techniques:

- Watermarking
- Packet correlation
- ML-based flow correlation
- RTT-based
- Anomaly-detection



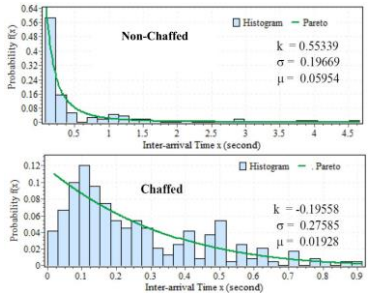






Evasive techniques

- Chaff packets
- Transfer delays
- Repacketisation
- Flow splitting



THE UNIVERSITY of EDINBURGH
informatics

Evaluation data

Connection pairs from S_N

- 1,400 packets

	Label	#conn	purpose
SS data	BA	30,000	Baseline attack
	DA	30,000	Delays with varying d_{D}
	CA	30,000	Chaff with varying d_{C}
	CL	4,000	Varying chain length
Background data	CAIDA	60.000	General background
	SSH	20.000	Similar to attack commands
	Multim.	20.000	Similar to chaff pert.