The Tangled Genealogy of IoT Malware

Emanuele Cozzi[1], Pierre-Antoine Vervier, Matteo Dell'Amico[1], Yun Shen[2], Leyla Bilge[2] and Davide Balzarotti[1]

EURECOM

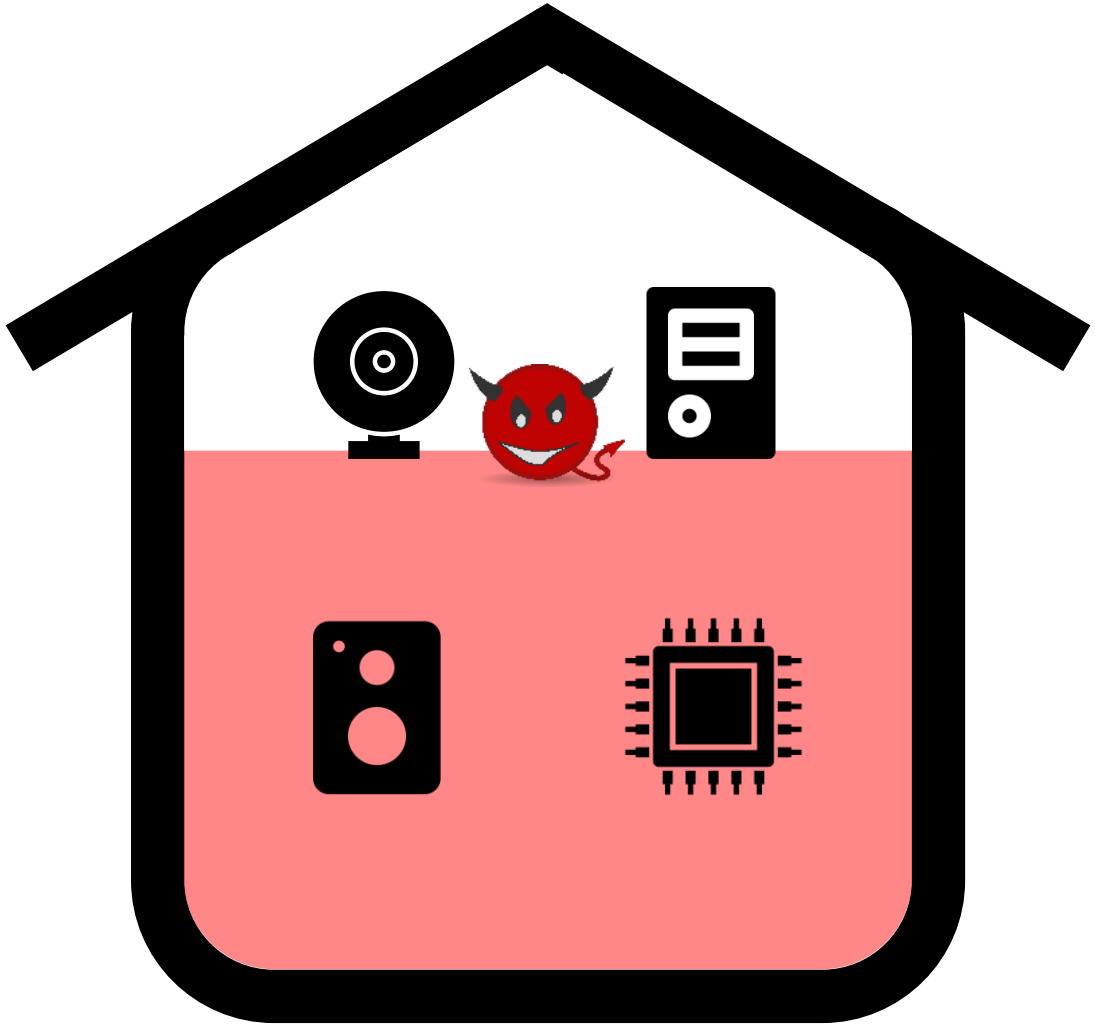
NortonLifeLock



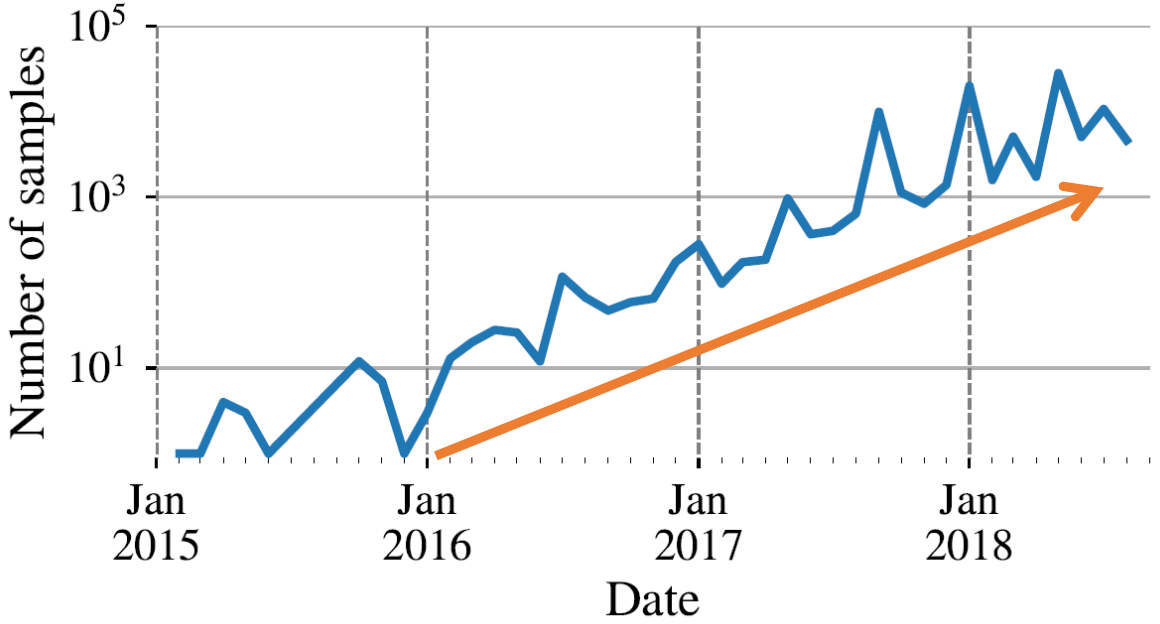


Annual Computer Security Applications Conference (ACSAC) 2020

IoT devices and malware



Submissions on VirusTotal





Community Score

40 engines detected this file

da20e2642cb4d7fa3b99bf2cb88804b44ed48e16f3c68b7c3418208f0d532a10

9ad8473148e994981454b3b04370d1ec

elf

132.96 KB
Size

2019-10-07 23:29:39 UTC
1 year ago

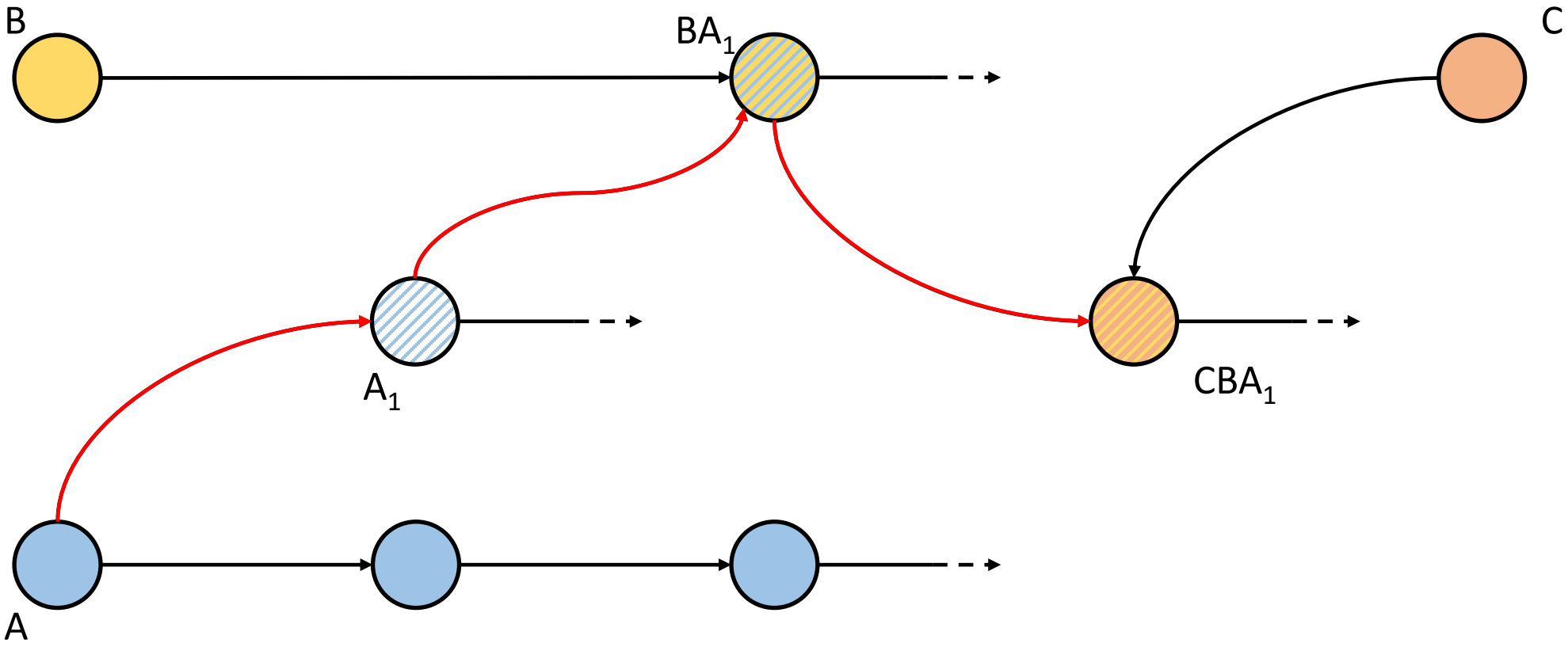
ELF

DETECTION | DETAILS | COMMUNITY 1

Ad-Aware	Backdoor.Linux.Agent.AD	AegisLab	Trojan.Linux.Mirai.4!c
AhnLab-V3	Linux/Mirai.Gen2	ALYac	Backdoor.Linux.Agent.AD
Antiy-AVL	Trojan[Backdoor]/Linux.Mirai.b	Arcabit	Backdoor.Linux.Agent.AD
Avast	ELF:Mirai-A [Trj]	Avast-Mobile	ELF:Mirai-AJO [Trj]
AVG	ELF:Mirai-A [Trj]	Avira (no cloud)	LINUX/Mirai.bonb
BitDefender	Backdoor.Linux.Agent.AD	CAT-QuickHeal	Trojan.Linux.IoTReaper
ClamAV	Unix.Trojan.IoTReaper-6355326-0	Comodo	Malware@#253dt8xqskql3
DrWeb	Linux.IotReaper.7	Emsisoft	Backdoor.Linux.Agent.AD (B)
eScan	Backdoor.Linux.Agent.AD	ESET-NOD32	Linux/Iotreaper.D
F-Secure	Malware.LINUX/Mirai.bonb	FireEye	Backdoor.Linux.Agent.AD

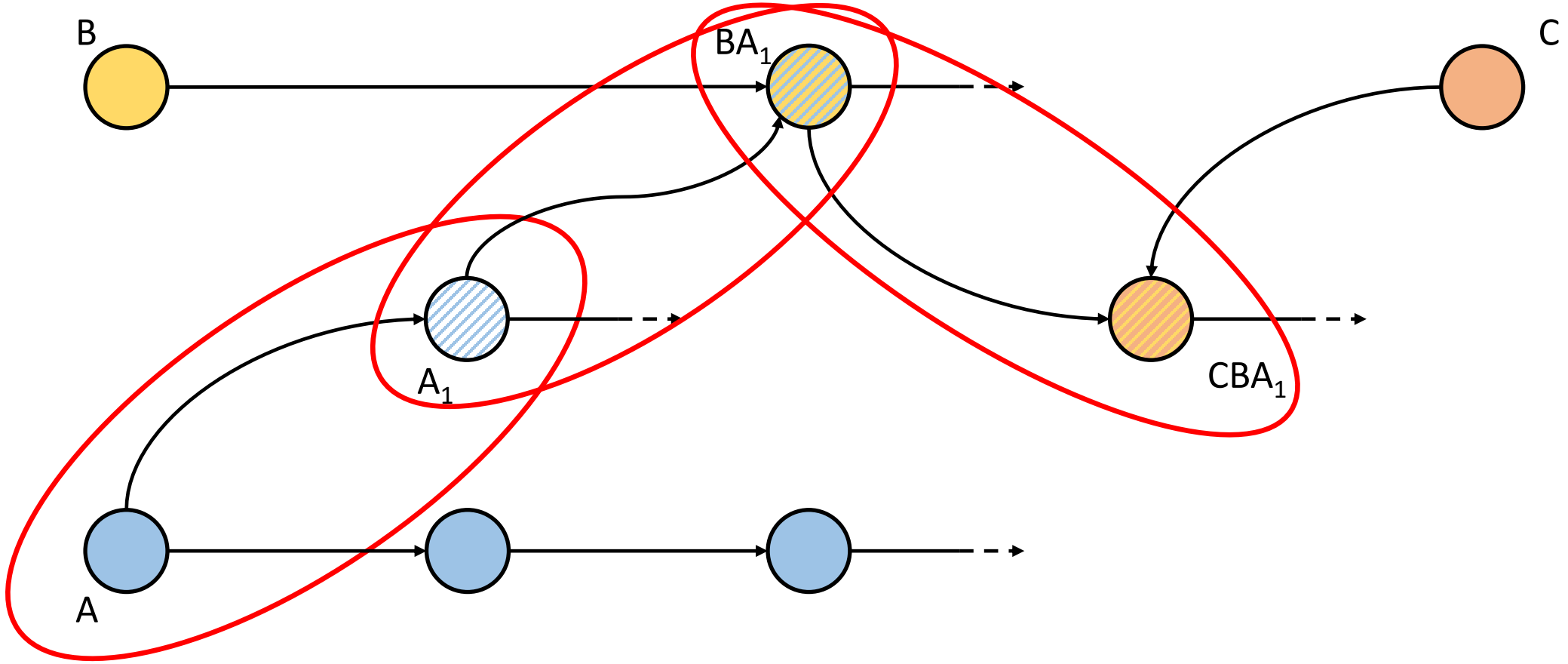
Inter- and intra-family variety

- Code reuse?



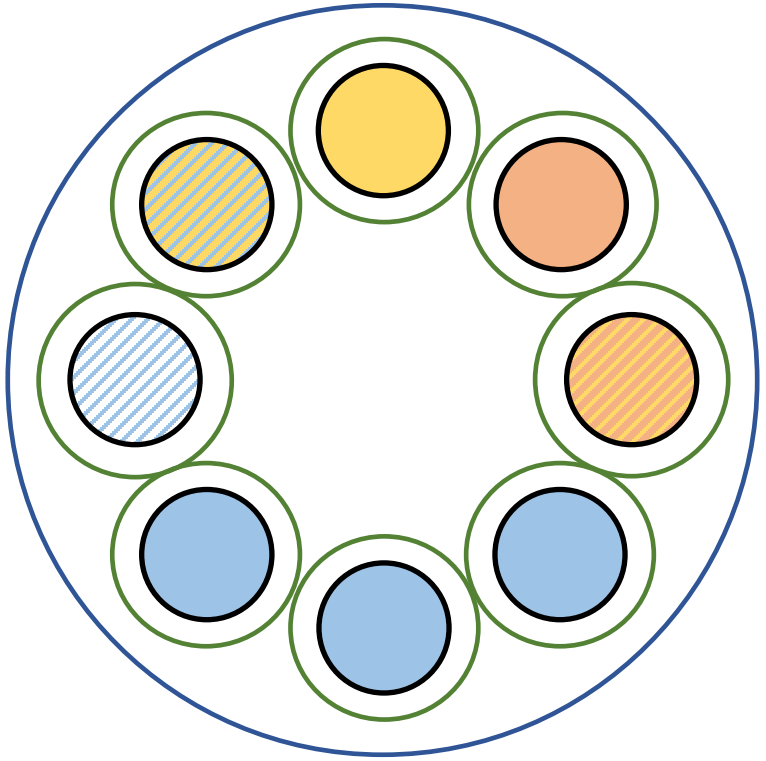
Classification of variants

- Which is which?



Features-based clustering

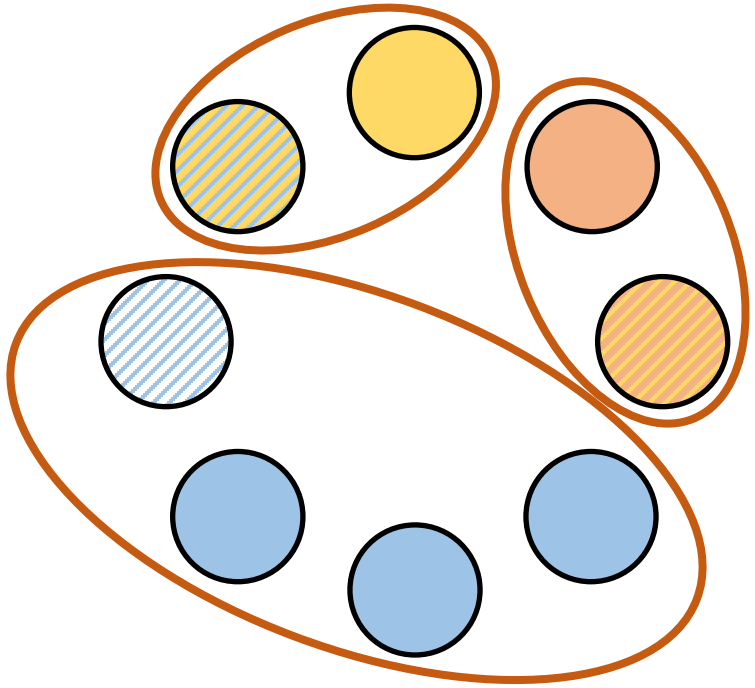
- Static features
  - ELF, disassembly, strings
  - Too fine-grained
- Dynamic features
  - Runtime behavior, network
  - Too coarse-grained



*Check our presentation at LASER Workshop if you want to know more

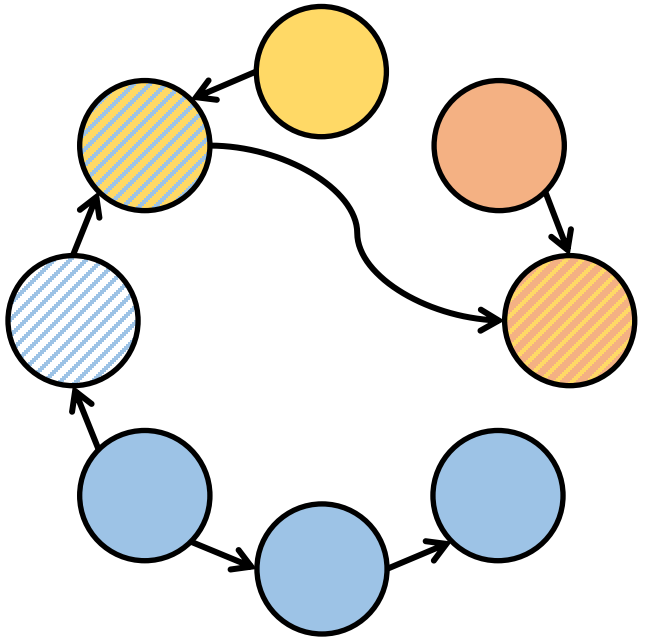
Code-based clustering

- Code does not lie
- Code-level analysis and function similarity
- IoT malware is currently rudimentary
- Complex and time-consuming
  - Dataset of 93k samples over a period of 3.5 years

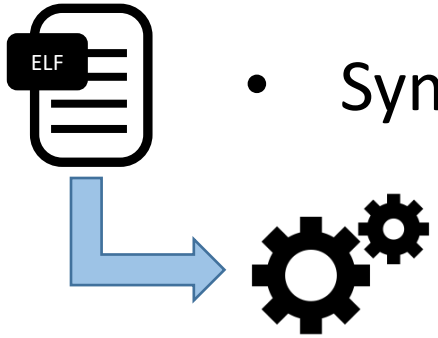


Lineage graph

- Track IoT malware evolution
- Code reuse
- Variants
- Outliers and AV errors

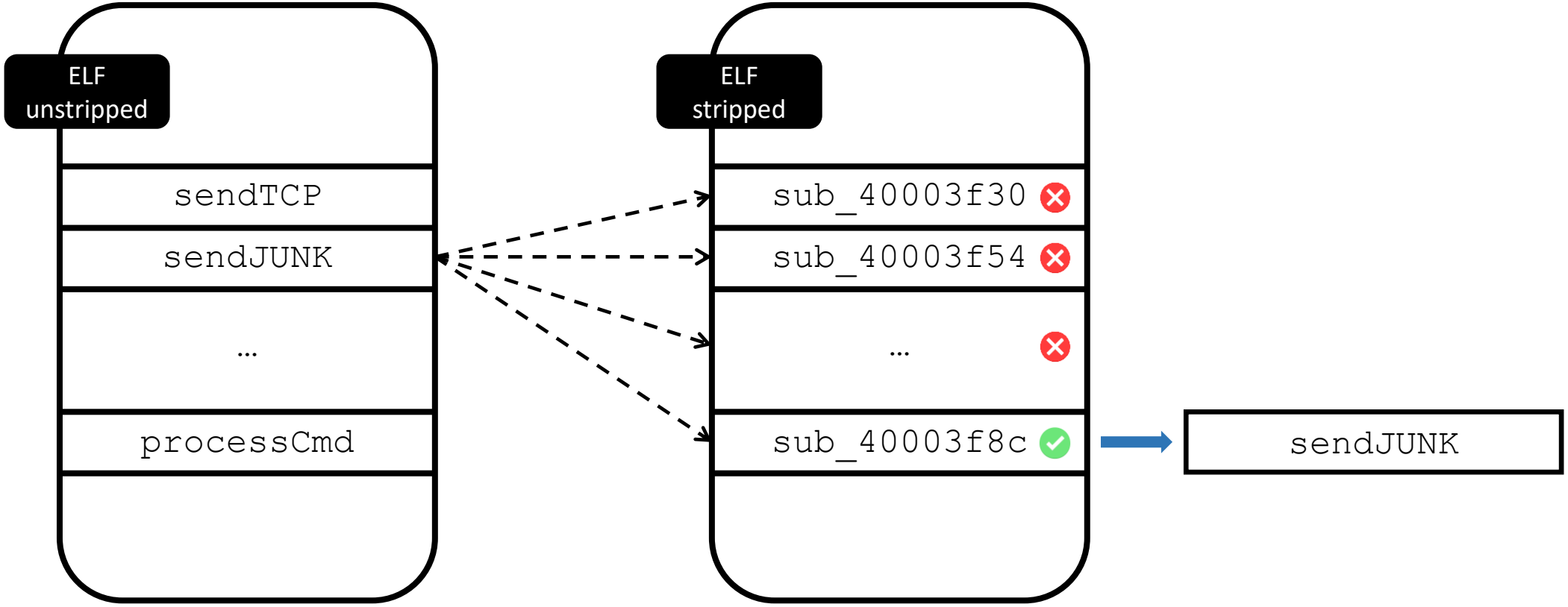


Analysis workflow

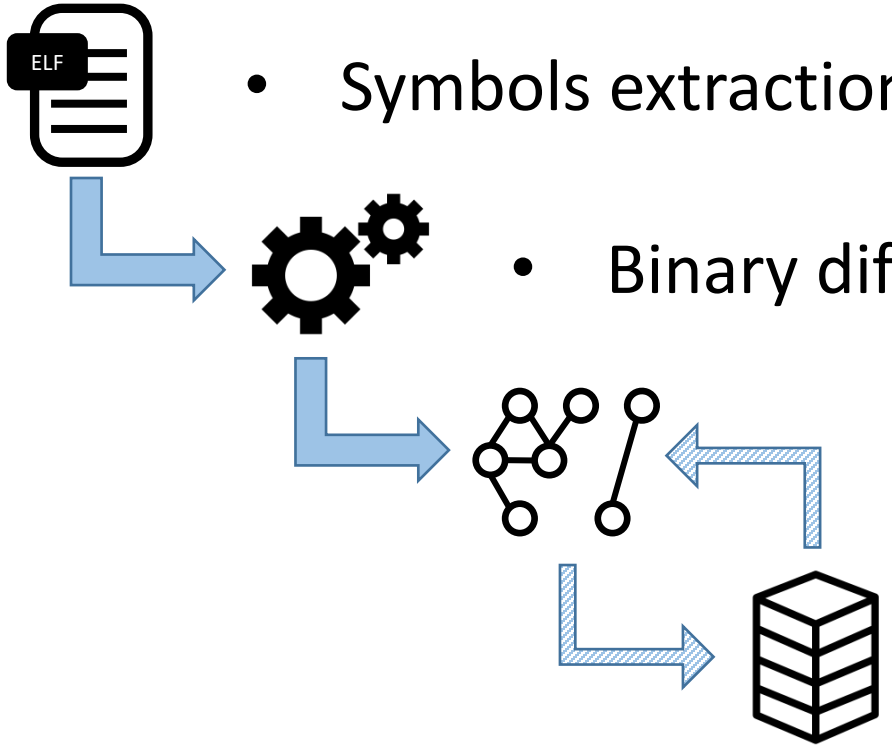


- Symbols extraction
- Binary diffing and symbols propagation

Symbols propagation

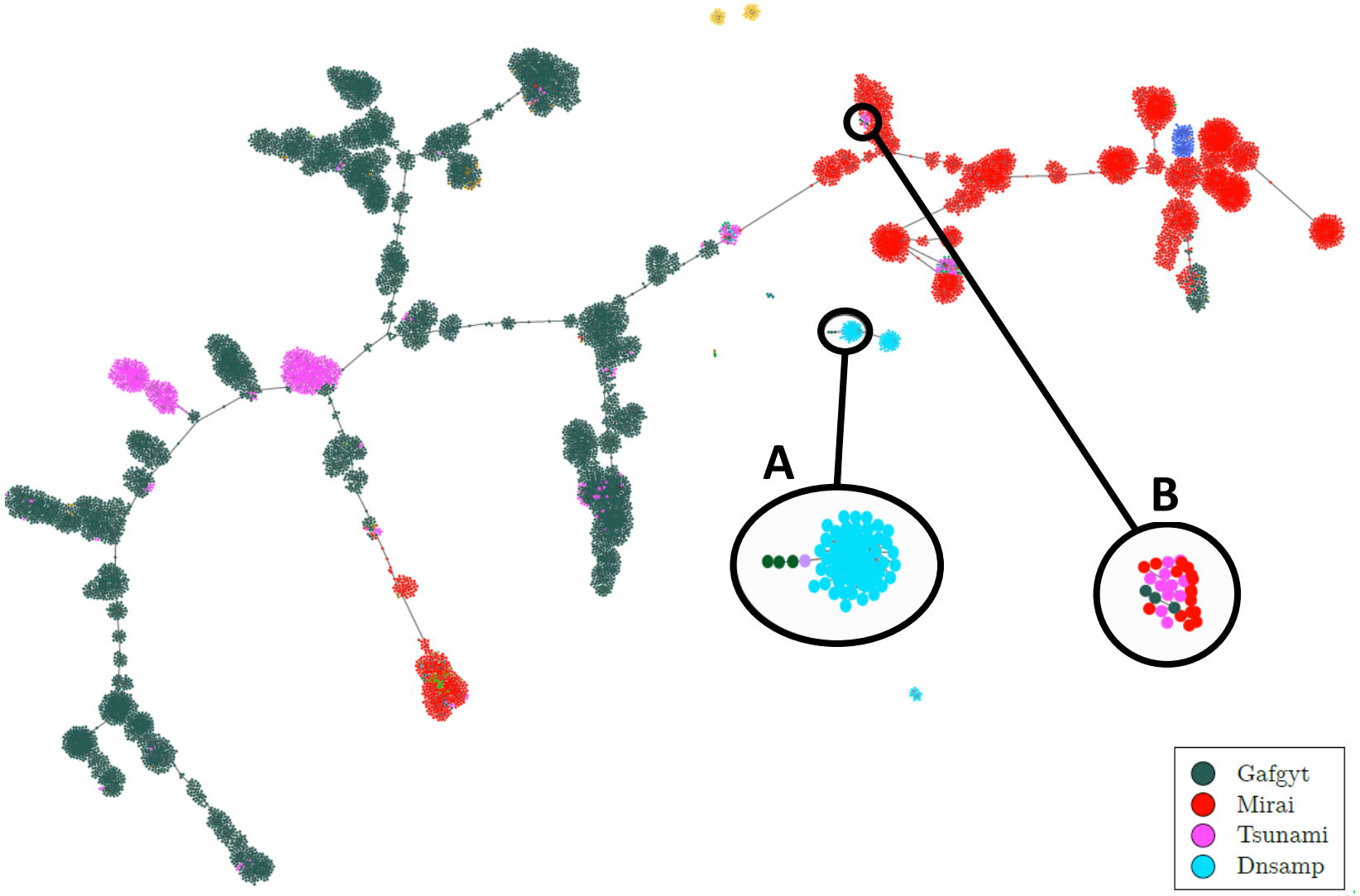


Analysis workflow



- Symbols extraction
- Binary diffing and symbols propagation
- Similarity graph
- Source code collection

Code reuse

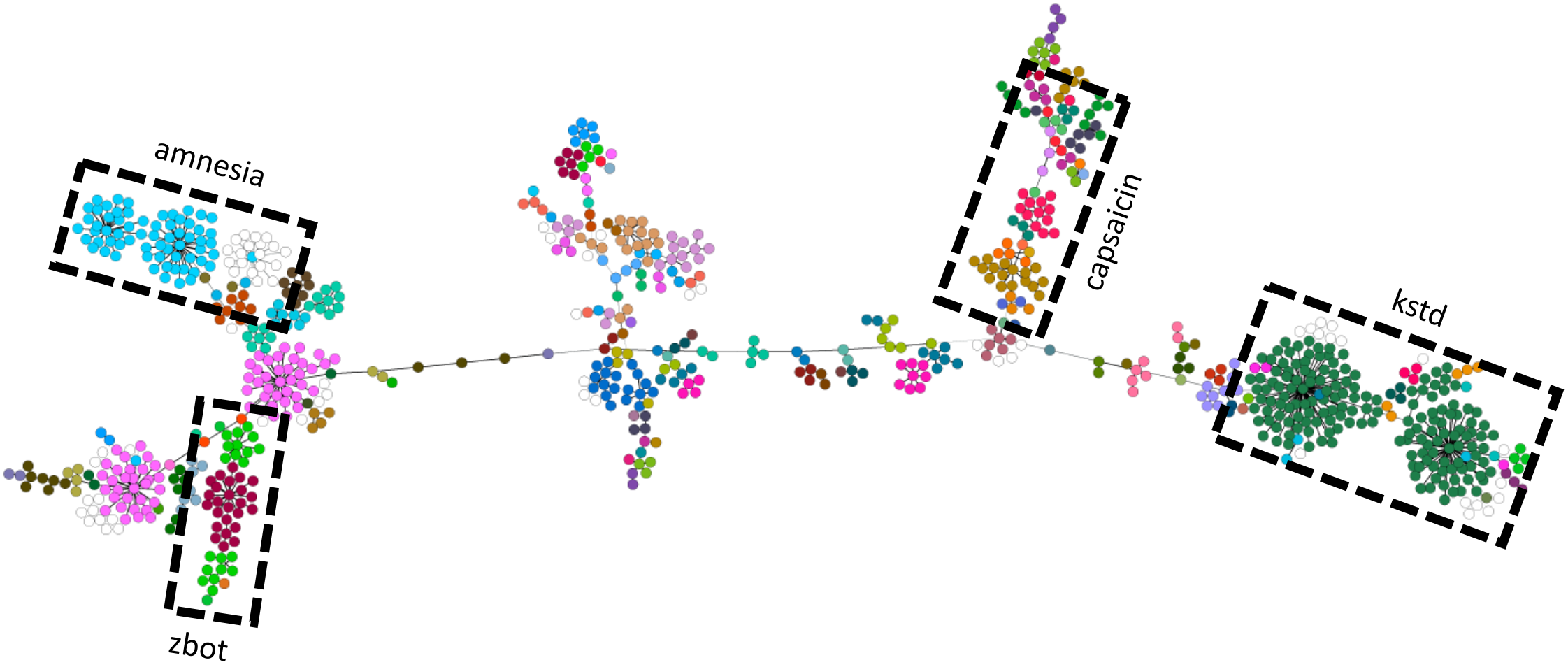


Outliers and AV errors

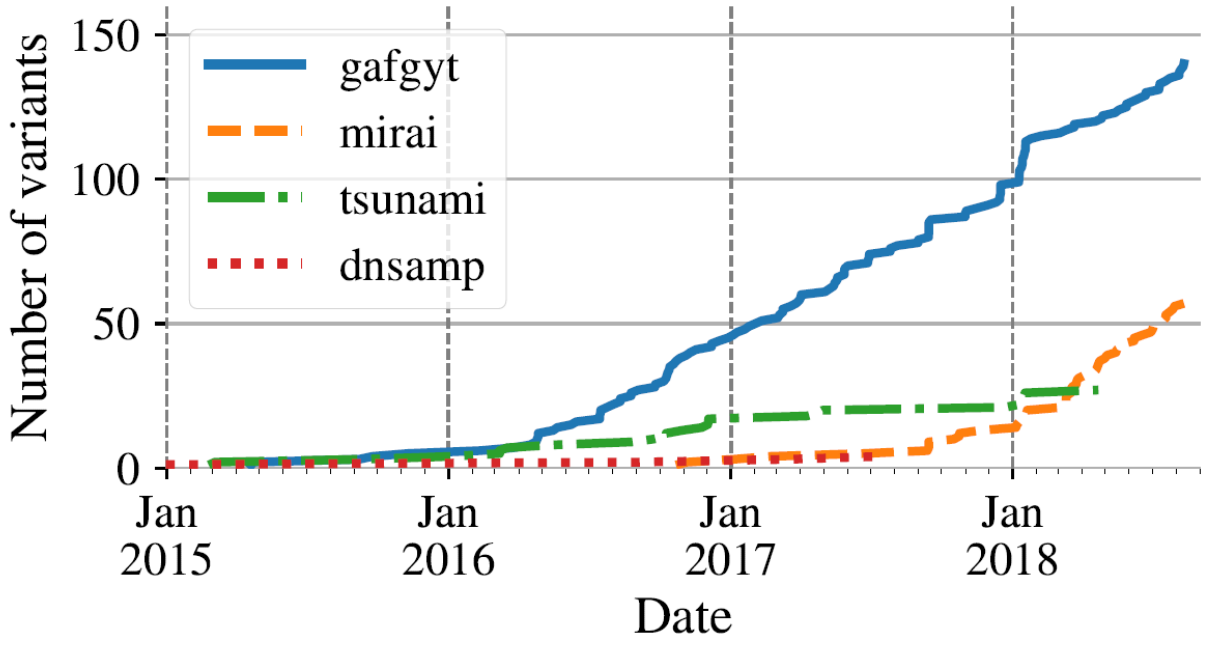
- High code reuse can confuse AVs
- Inconsistency in labelling
- Simple packers (UPX) can cause troubles

Architecture	Number of samples		
	Wrong label	Without label	Total
ARM 32-bit	19	9	28
MIPS I	25	41	66
PowerPC	1	4	5
SPARC	2	0	2
Hitachi SH	7	0	7
Motorola 6800	8	2	10
Total	62	56	118

Variants



Variants over time



Conclusions

- Largest study to date over 93k IoT malware
- High fragmentation makes characterization harder
- Assigning a name to a given sample is difficult
- Small modifications and code reuse are enough to derail AVs

- Static analysis is *currently* more accurate then dynamic reports

The Tangled Genealogy of IoT Malware

Emanuele Cozzi[1], Pierre-Antoine Vervier, Matteo Dell'Amico[1], Yun Shen[2], Leyla Bilge[2] and Davide Balzarotti[1]

Thank you!

https://github.com/eurecom-s3/tangled_iot

cozzi@eurecom.fr

@invano

Annual Computer Security Applications Conference (ACSAC) 2020