ShieldFS: A Self-healing, Ransomware-aware Filesystem

Andrea Continella, Alessandro Guagnelli, Giovanni Zingaro, Giulio De Pasquale, Alessandro Barenghi, Stefano Zanero, Federico Maggi

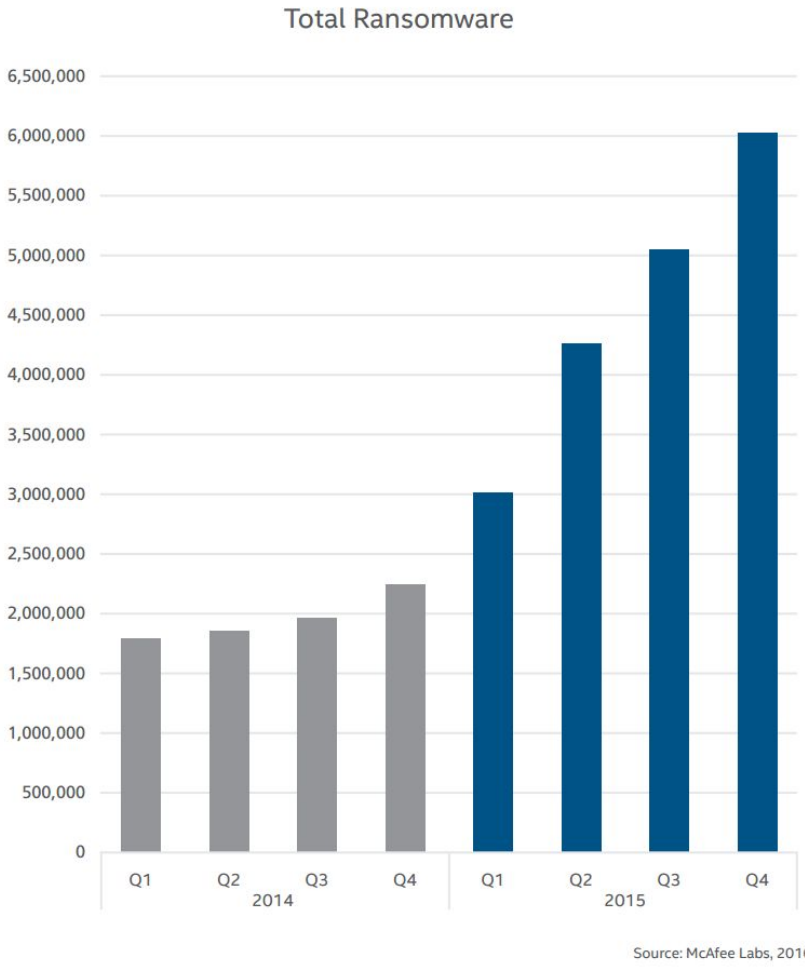
Politecnico di Milano

12-08-2016

Key Takeaways

- The way ransomware interacts with the filesystem is significantly **different** in comparison to benign applications
- We can detect ransomware behaviors by monitoring the **filesystem activity** and the usage of **crypto** primitives
- Mere **detection** is **insufficient**
  - Stopping a suspicious process may **be too late**
  - We need to **protect users' data**, reverting the effects of ransomware attacks.

2016 the "year of extortion"



Source: McAfee Labs, 2016.

CRYPTOWALL RANSOMWARE COST USERS $325 MILLION IN 2015

by NewsEditor on November 2nd, 2015 in Industry and Security News.

Public Service Announcement

FEDERAL BUREAU OF INVESTIGATION

June 23, 2015

Alert Number
I-062315-PSA

CRIMINALS CONTINUE TO DEFRAUD AND EXTORT FUNDS FROM VICTIMS USING CRYPTOWALL RANSOMWARE SCHEMES

Ransomware Hackers Blackmail U.S. Police Departments

Chris Francescani
Tuesday, 26 Apr 2016 | 10:30 AM ET
NBC NEWS

Hollywood hospital pays $17,000 in bitcoin to hackers; FBI investigating

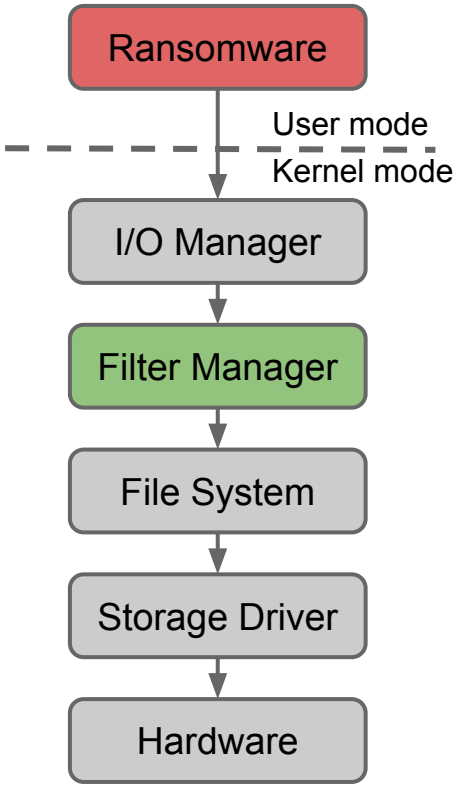
How to Deal With Ransomware?

- Is a classical antivirus enough?
  - Unfortunately no
  - Signatures must be updated
  - Executables are obfuscated and encrypted
- Why don't we monitor Crypto API calls?
  - Malware implement own crypto functions or use libraries
- The OS should be able to detect malicious ransomware
  - Look at the **Filesystem's activity**!

[1] A.Kharraz, W. Robertson, D. Balzarotti, L. Bilge, *E. Kirda, Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks*, DIMVA 2015
[2] A. Kharaz, S. Arshad, W. Robertson, E. Kirda, *UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware*, USENIX Sec 2016
[3] N.Scaife, H. Carter, P. Traynor, K. Butler, *CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data*, ICDCS 2016

FS Activity Monitor

- Develop a Windows Kernel module to monitor and log the file system activity
  - Windows Minifilter Driver
  - Log IRPs (I/O Request Packets)
- Run ransomware samples and collect data about the activity of the FS during infections
- Distribute IRPLogger to 11 clean machines
  - Anonymized data about the activity of the FS during “normal” clean executions
    - 1 months worth of data
    - ~1.7 billion IRPs
    - 2,245 distinct applications



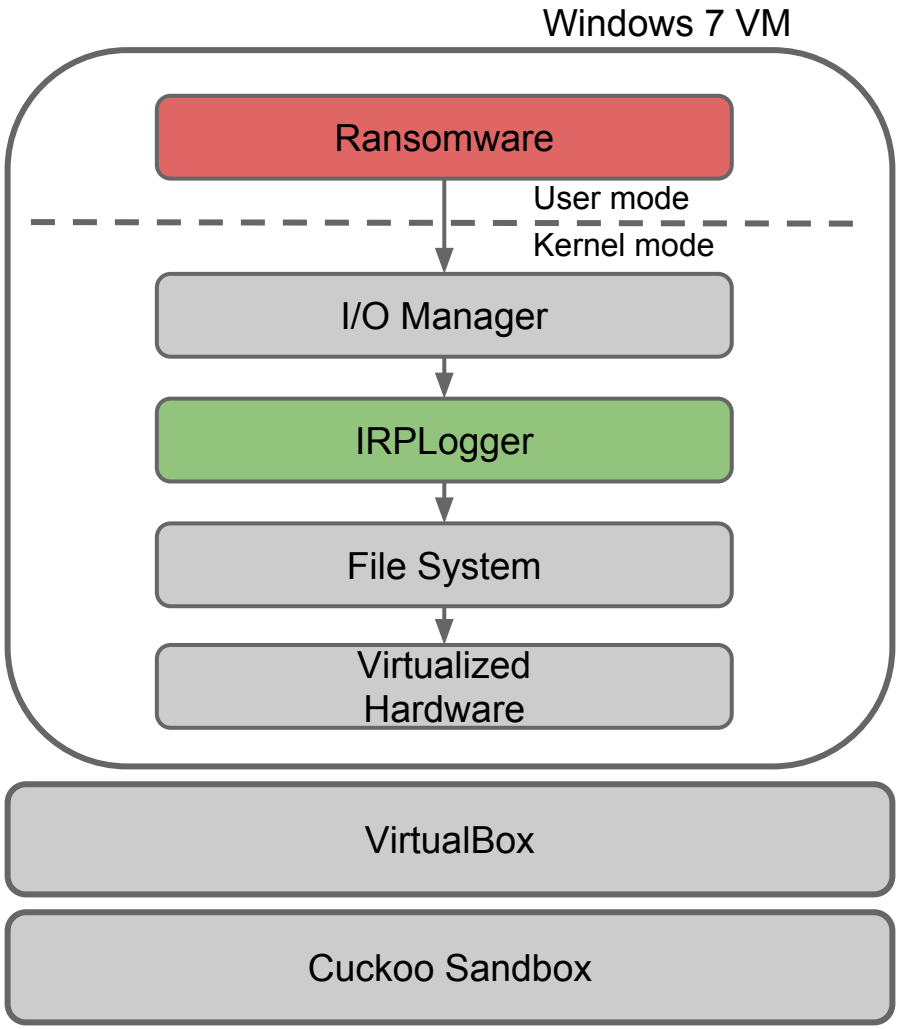
Filter Manager APIs

```
CONST FLT_OPERATION_REGISTRATION Callbacks[] = {
    { IRP_MJ_CREATE,
      0,
      PreCreateOperationCallback,
      PostCreateOperationCallback },

    { IRP_MJ_CLOSE,
      0,
      PreCloseOperationCallback,
      PostCloseOperationCallback },

    { IRP_MJ_READ,
      0,
      PreReadOperationCallback,
      PostReadOperationCallback },

    { IRP_MJ_WRITE,
      0,
      PreWriteOperationCallback,
      PostWriteOperationCallback },
}

FltRegisterFilter( DriverObject,
                   &FilterRegistration,
                   &Filter );
```

Statistics of the collected data

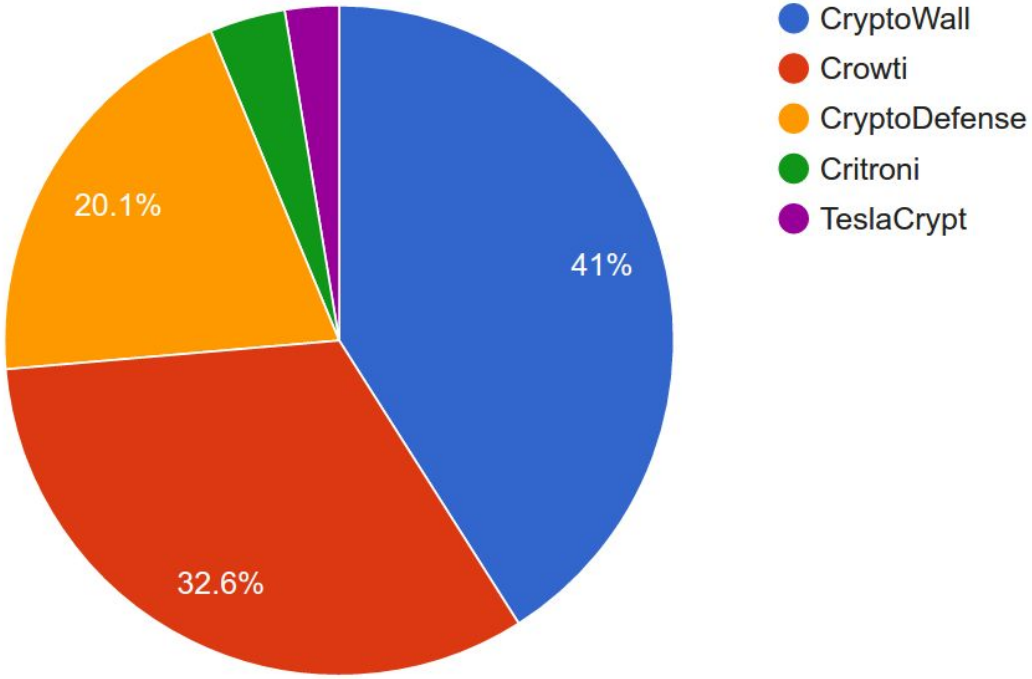
User	Win. ver.	Usage	Data [GB]	#IRPs Mln.	#Procs Mln.	Apps	Period [hrs]	Data Rate [MB/min]
1	10	dev	3.4	230.8	16.60	317	34	7.85
2	8.1	home	2.4	132.1	9.67	132	87	2.04
3	10	office	0.9	54.2	5.56	225	17	0.83
4	7	home	4.7	279.9	18.70	255	122	5.18
5	7	home	2.2	138.1	5.04	141	47	4.10
6	10	dev	1.8	100.4	10.30	225	35	2.42
7	8.1	dev	0.8	49.0	3.28	166	8	5.62
8	8.1	home	0.8	43.9	6.33	148	32	2.16
9	8.1	home	7.7	501.8	24.20	314	215	3.21
10	7	home	0.9	57.6	2.63	151	18	4.60
11	7	office	2.6	175.2	4.69	171	28	8.51
		Total	28.2	1,763.0	107.00	2245	643	-

Analysis Environment



Training Dataset

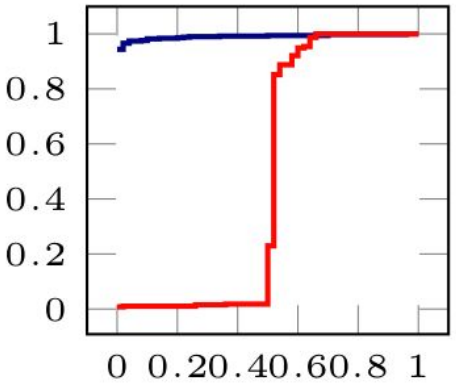
- 383 samples of 5 different families from VirusTotal



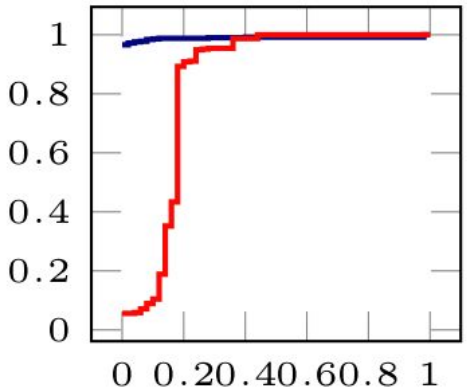
Ransomware vs Benign programs

Cumulative Distribution Functions

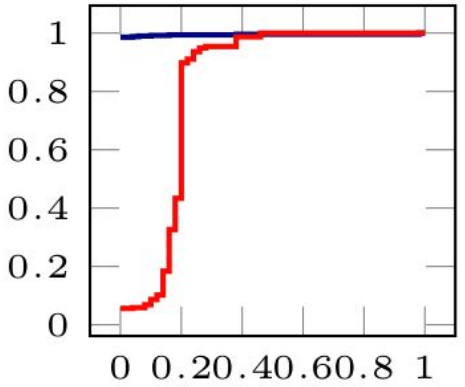
Benign Ransomware



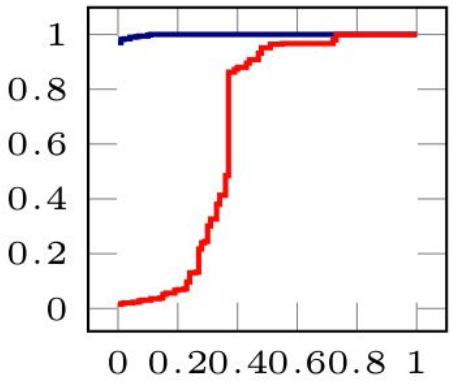
(1) #Folder-listing



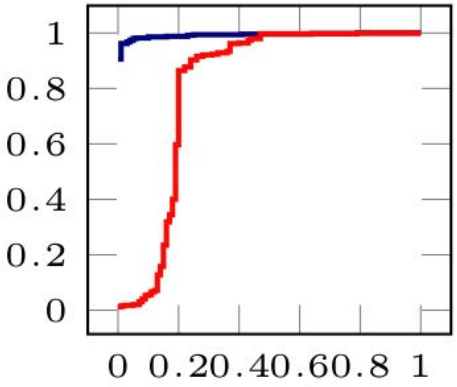
(2) #Files-Read



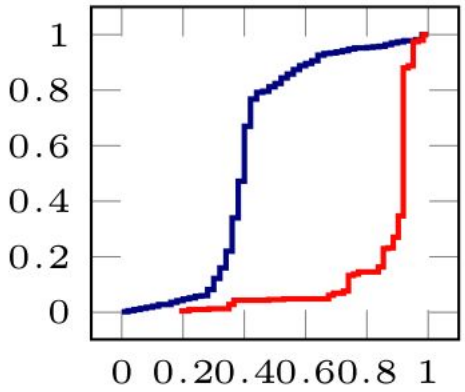
(3) #Files-Written



(4) #Files-Renamed



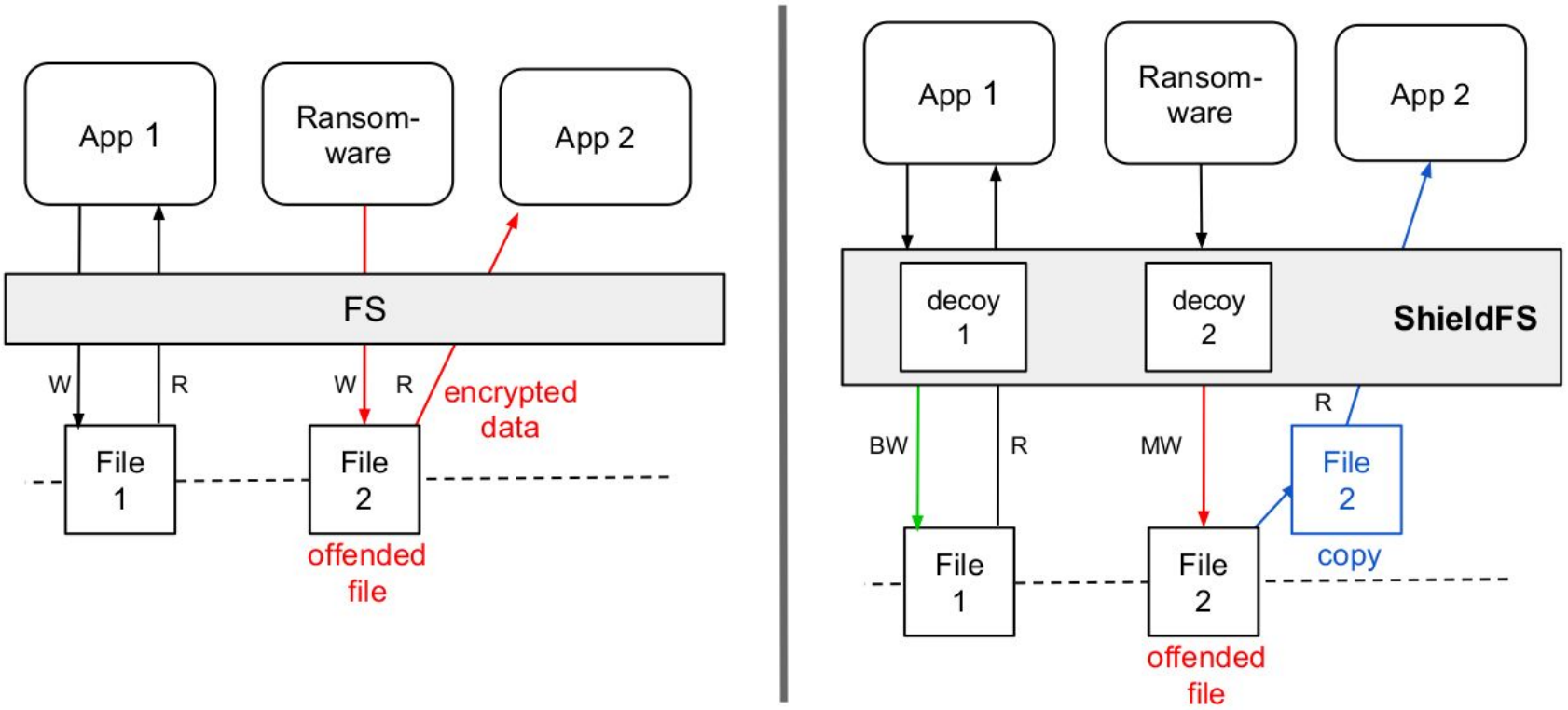
(5) File type coverage



(6) Write-Entropy

ShieldFS
Self-healing Filesystem

ShieldFS: Approach

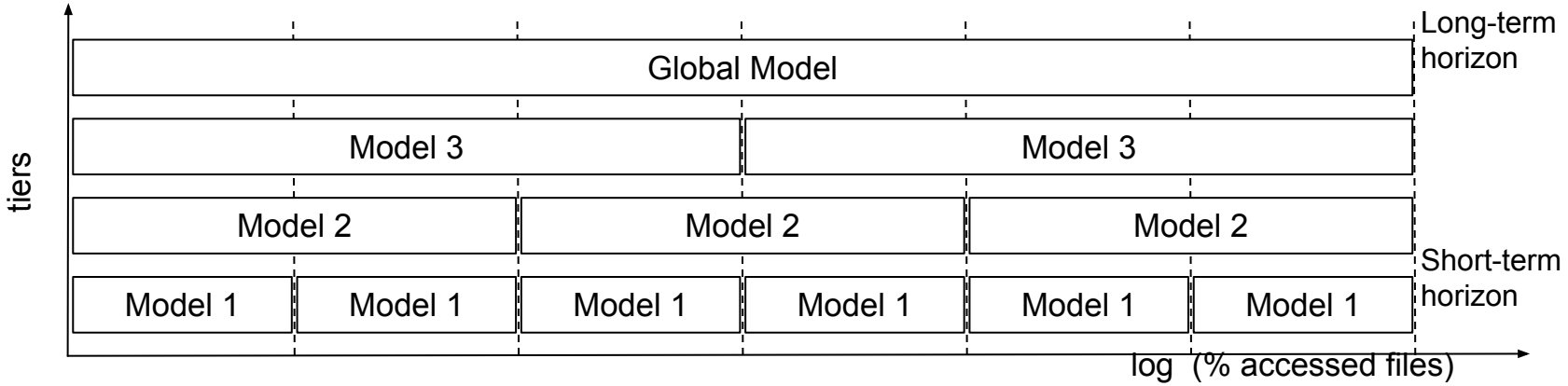


Detection Models

- We propose a set of custom classifiers trained on the filesystem activity features
- One set of models, called **process centric**, each trained on the processes individually
- A second model, called **system centric**, trained by considering all the IRP logs as coming from a single, large “process” (i.e., the whole system)
- ShieldFS **adapts** these **models** to the filesystem usage habits observed on the protected system

Multi-tier Incremental Models

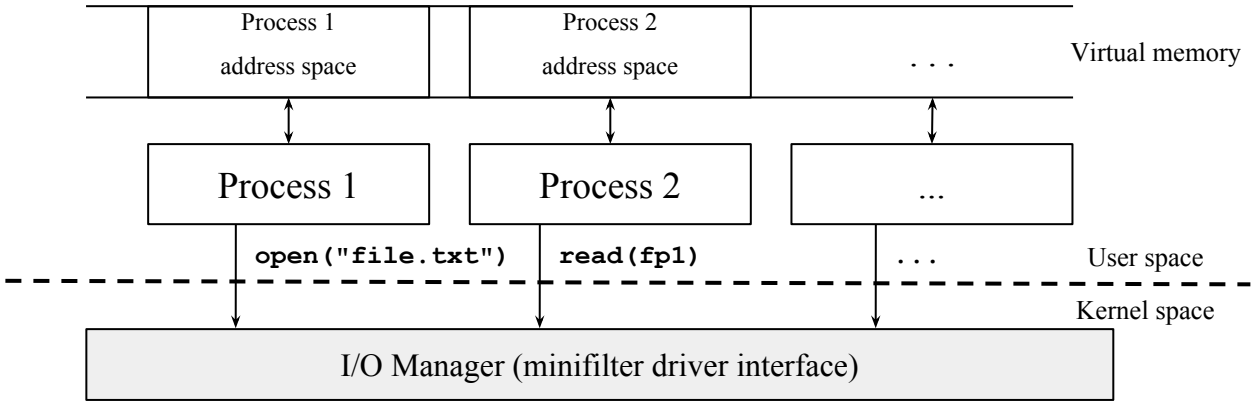
- Split the data in intervals, or *ticks*, defined by the fraction of files accessed by the monitored process
- Multi-tier incremental approach
  - Global Model takes care of typical ransomware
  - Model *i* handles code injection cases



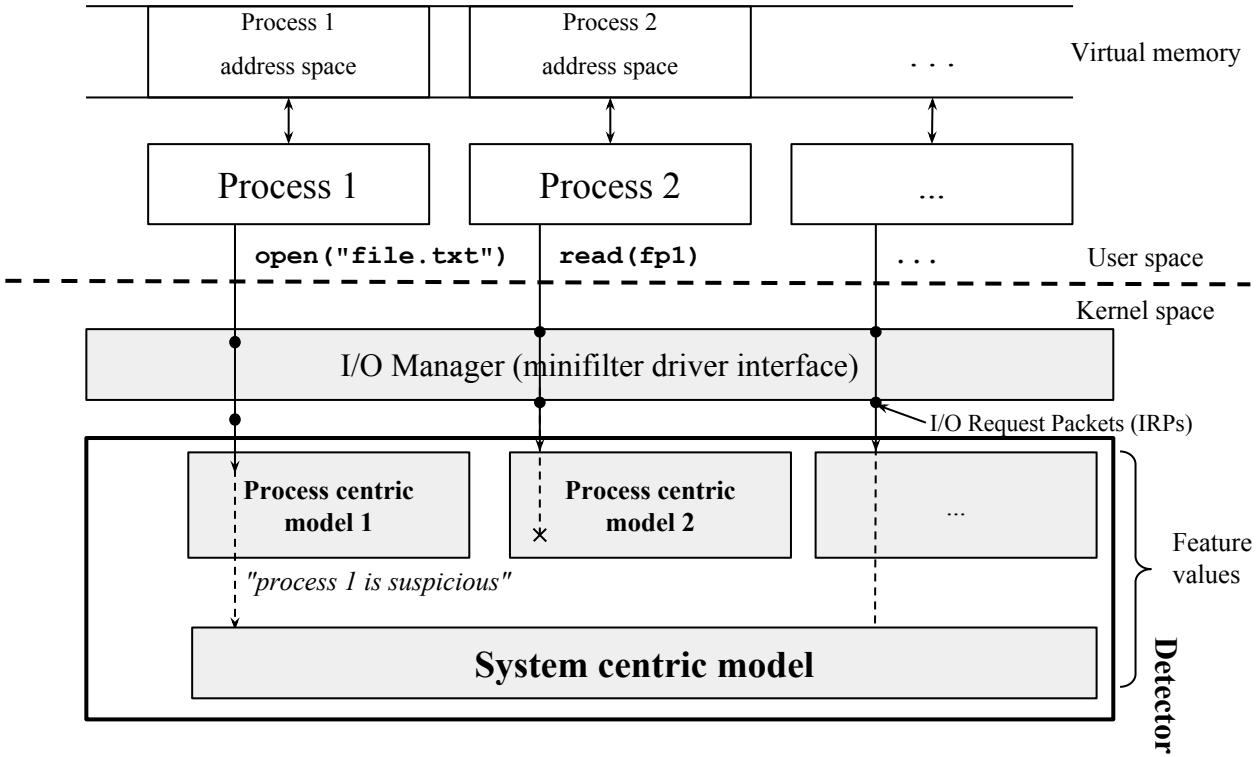
CryptoFinder

- Block ciphers expand the key in a sequence of values, known as the **key schedule**, used during each round
- The key schedule is **deterministic** and known!
- It is materialized **in memory** during all the encryption procedure
- Look for valid schedule to detect usage of crypto!

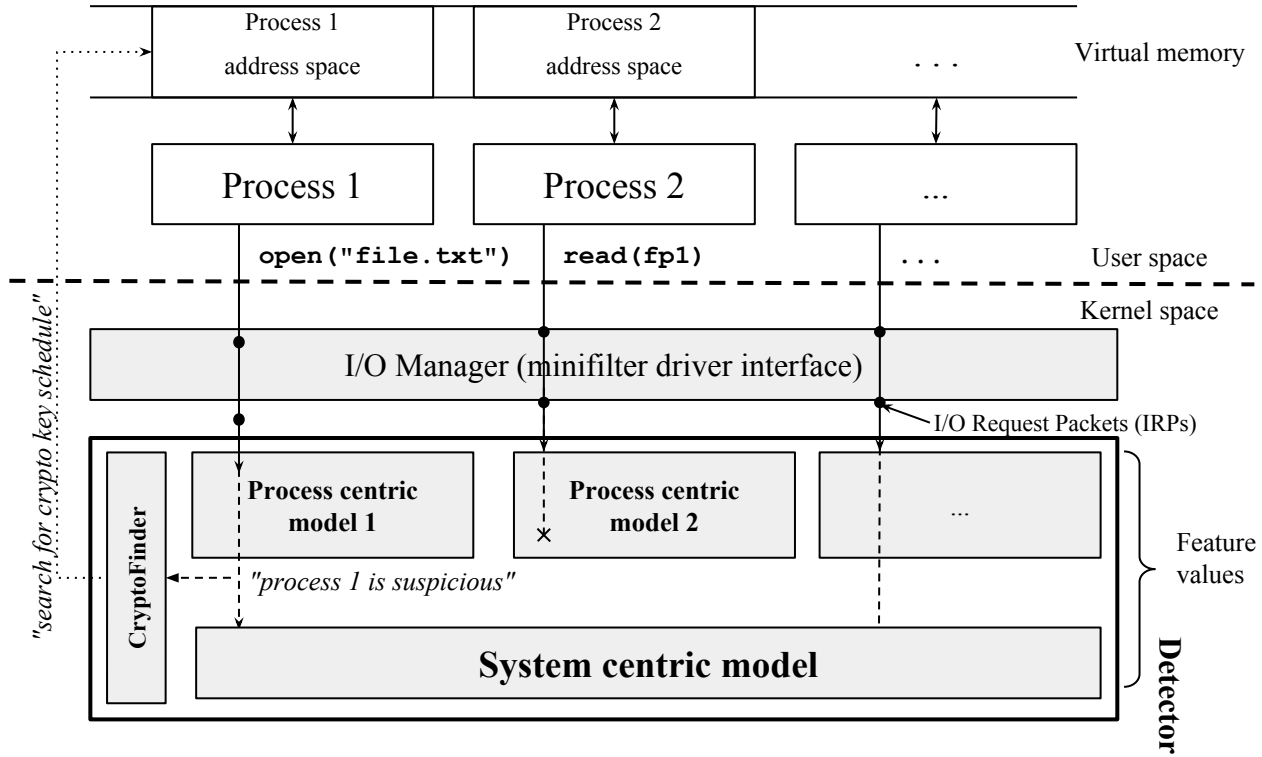
ShieldFS Architecture



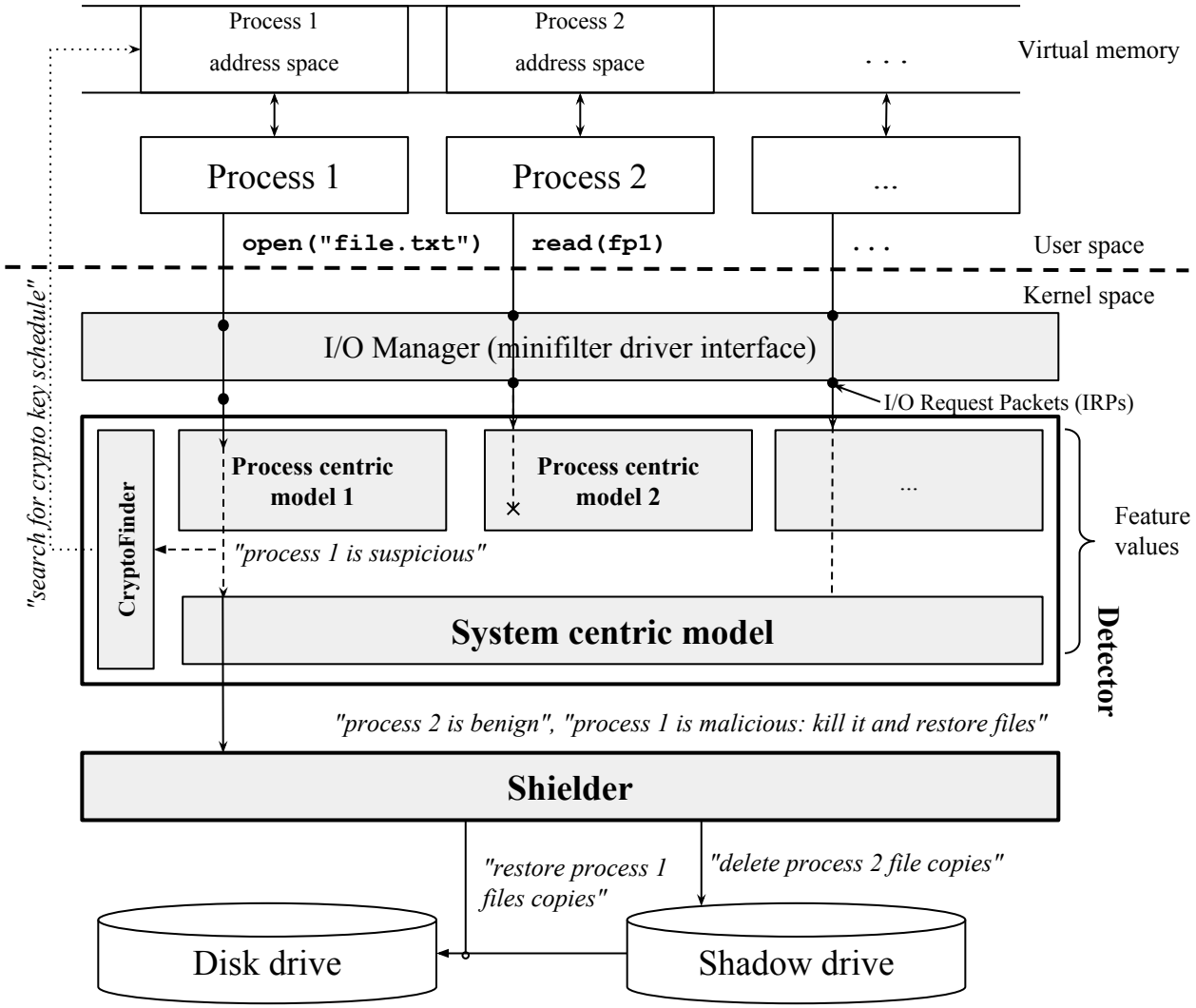
ShieldFS Architecture



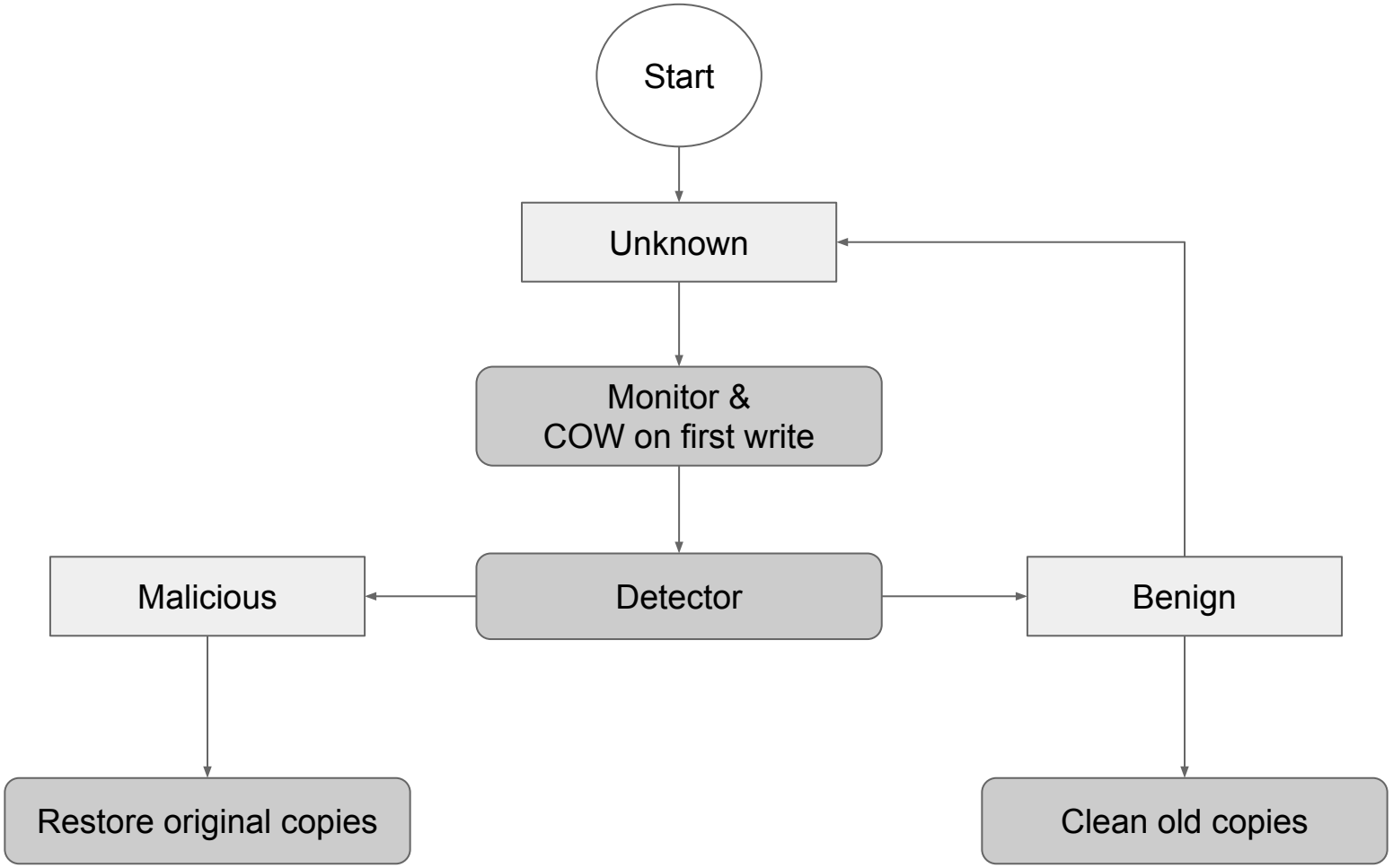
ShieldFS Architecture



ShieldFS Architecture

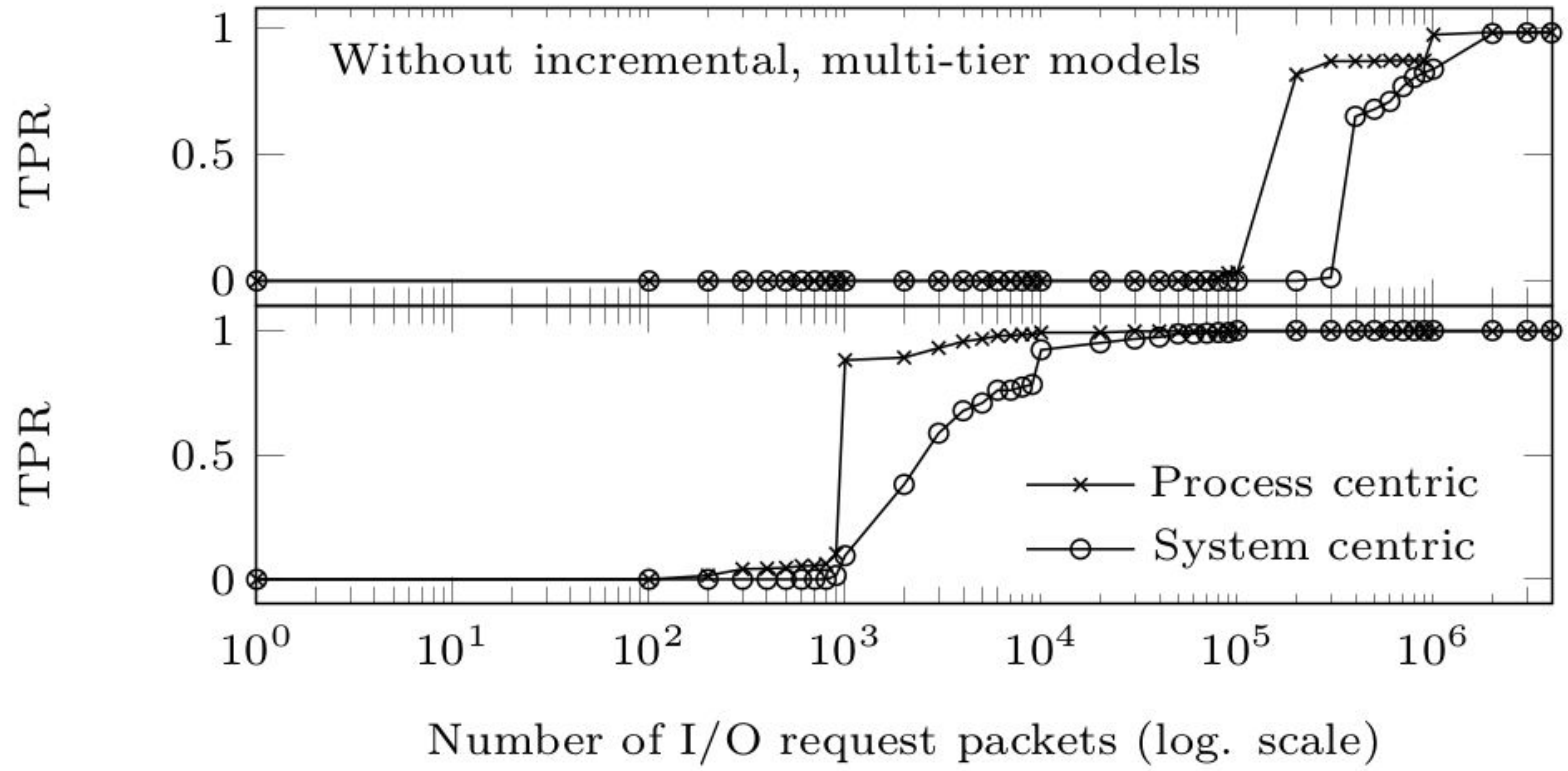


Automatic File Recovery Workflow



Experimental Results

Detection Accuracy



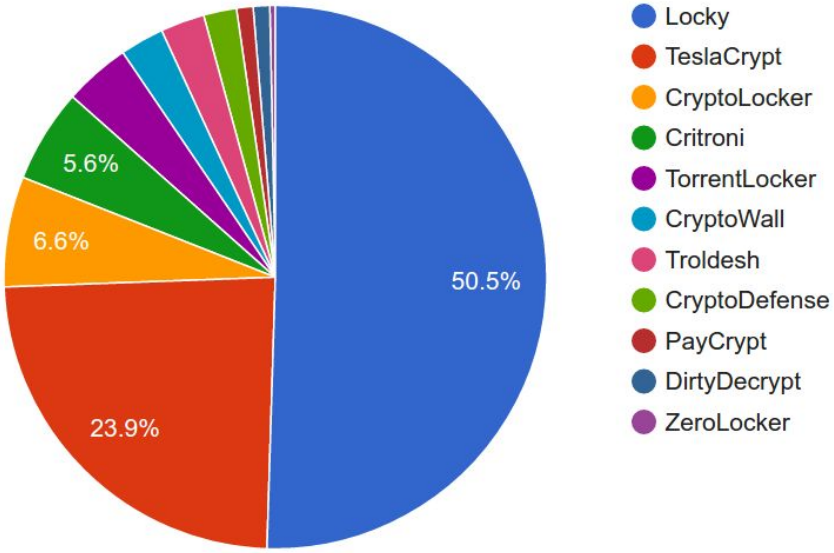
False Positive Evaluation

User	False positive rate [%]		
Machine	Process	System	**Outcome**
1	0.53	23.26	**0.27**
2	0.00	0.00	**0.00**
3	0.00	0.00	**0.00**
4	0.00	1.20	**0.00**
5	0.22	45.45	**0.15**
6	0.00	4.76	**0.00**
7	0.00	88.89	**0.00**
8	0.00	0.00	**0.00**
9	0.00	0.00	**0.00**
10	0.00	0.00	**0.00**
11	0.00	0.00	**0.00**

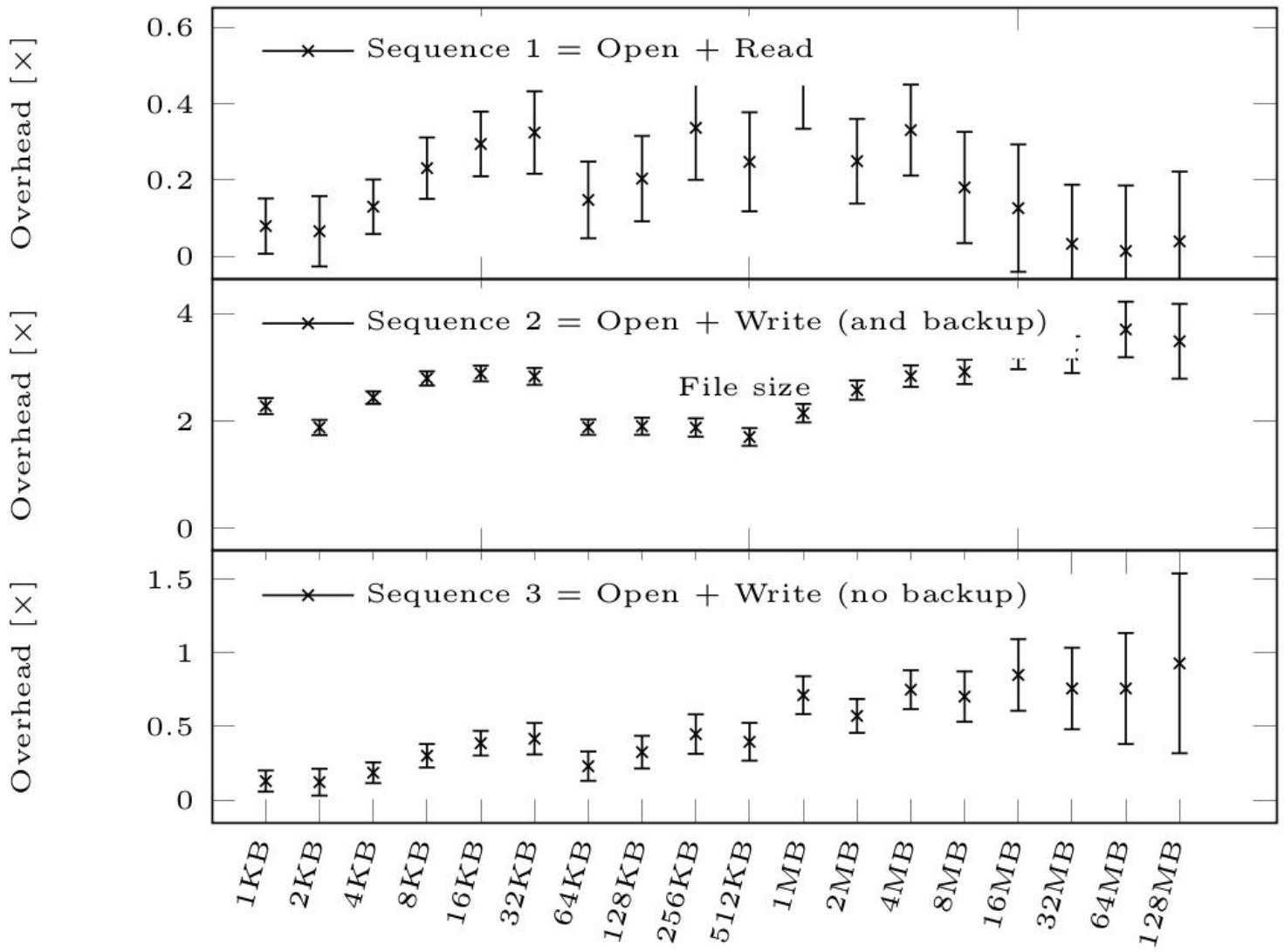
FPR with One-machine-off Cross Validation

Detection and Recovery Capabilities

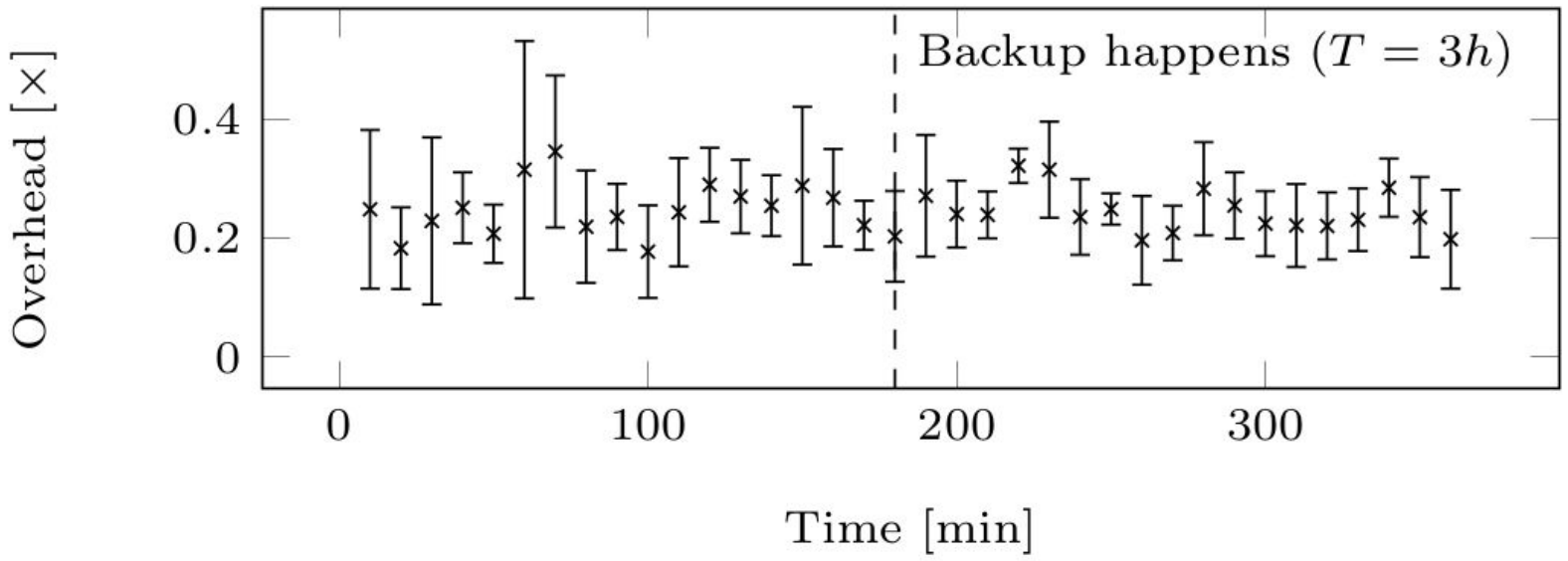
- 305 unseen samples (from VT) of 11 different ransomware families
  - 7 new families, not present in the training dataset
- Files protected: always **100%**
  - Even in case of missed detection
- Detection rate: 298/305, **97.70%**



System Overhead



Perceived Overhead



Storage Overhead

User	Period [hrs]	Storage Required Max [GB]	Storage Required Avg. [GB]	Storage Overhead **Max [%]**	Storage Overhead **Avg [%]**	Max Cost [USD]
1	34	14.73	0.63	**4.29**	**0.18**	44.2¢
2	87	0.62	0.19	**0.95**	**0.29**	1.86¢
4	122	9.11	0.73	**8.53**	**0.68**	27.3¢
5	47	2.41	0.56	**5.49**	**1.29**	7.23¢
7	8	1.00	0.39	**3.35**	**1.28**	3.00¢

Limitations & Future work

- Susceptibility to targeted evasion
  - Mimicry attacks
  - Multiprocess Malware
- Cryptographic primitives detection evasion
  - Intel AES-NI extensions
  - Support other ciphers
- Impact on the performance
  - Perform the COW at the block disk level

Conclusions

- Ransomware **significantly differs** from benign software from the filesystem's viewpoint
  - first, large-scale data collection of IRPs generated by benign applications
- ShieldFS creates **generic models** to identify ransomware behaviors
  - Filesystem activity
  - Use of symmetric crypto primitives
- Pure detection is not enough
  - ShieldFS applies detection in a self-healing virtual FS able to transparently **revert the effects** of ransomware attacks, once detected

Thank you!
Questions?

andrea.continella@polimi.it

@_conand

http://shieldfs.necst.it/

NECST laboratory