RSAConference2022
San Francisco & Digital | June 6 – 9

TRANSFORM

SESSION ID: **HT-W09**

Machine Learning to Ultimately Defeat Advanced Ransomware Threats

Sergey Ulasen
Director of Development, AI
Acronis
linkedin.com/in/sergey-ulasen

Vladimir Strogov
Director of Development, Kernel Team
Acronis
linkedin.com/in/vladimirstrogov

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

©2022 RSA Conference LLC or its affiliates. The RSA Conference logo and other trademarks are proprietary. All rights reserved.

RSAConference2022

Introduction

Zero Day Attacks: Ransomware

Ransomware as the typical example of zero day attacks

- There are many ransomware families.
- Anti-Ransomware defense is problematic due to the following:
  - the samples are often modified.
  - real-time data protection is expensive.
- **Machine Learning** can greatly improve existing data protection.

We are describing injection techniques used by ransomware.

RSAConference2022

Advanced Ransomware Samples

The definition of advanced ransomware.

Shell code injection in well-known good processes.

Ryuk as the most advanced form of ransomware payloads (1)

- The initial stages:
  - Planting several executables in the system, for example using the Zloader botnet.
  - Stopping services, deleting VSS copies, etc.
- The advanced stages:
  - Injecting multiple system and trusted processes.
  - But keeping the system operational: lsass.exe, csrss.exe and explorer.exe are not changed.
  - Detaching the encrypting part from **Ryuk** processes.

Ryuk as the most advanced form of ransomware payloads (2)

- Challenge: abnormal injection detection.
- Important: there are legitimate injection techniques.
- The ML-based solution:
  - Snapshotting of data changes for the thread.
  - Detecting stack anomalies with ML models.
  - Recovering changed data if ransomware is detected.
  - Otherwise discarding the snapshots of data changes.

RSAConference2022

Anatomy of the advanced attack with the shell code injection

Possible scenarios of the attacks

Ransomware Shell code injection with CreateRemoteThread

- The dropper delivers the payload
- The payload injects itself into legitimate processes with
  - OpenProcess
  - VirtualAllocEx
  - WriteProcessMemory writes bufferWithTheEncryptor
  - CreateRemoteThread launches bufferWithTheEncryptor
  - CloseHandle

Ransomware shell code injection with APC

- The dropper delivers the payload
- The payload injects itself into legitimate processes with
  - OpenProcess
  - VirtualAllocEx
  - WriteProcessMemory writes bufferWithTheEncryptor
  - apcRoutine = bufferWithTheEncryptor
  - OpenThread
  - QueueUserApc

Ransomware DLL injection with SetWindowsHookEx

- The dropper delivers the payload
- The payload injects itself into legitimate processes with
  - LoadLibrary (“hook.dll”)
  - Hooker=GetProcAddress(..);
  - SetWindowsHookEx

RSAConference2022

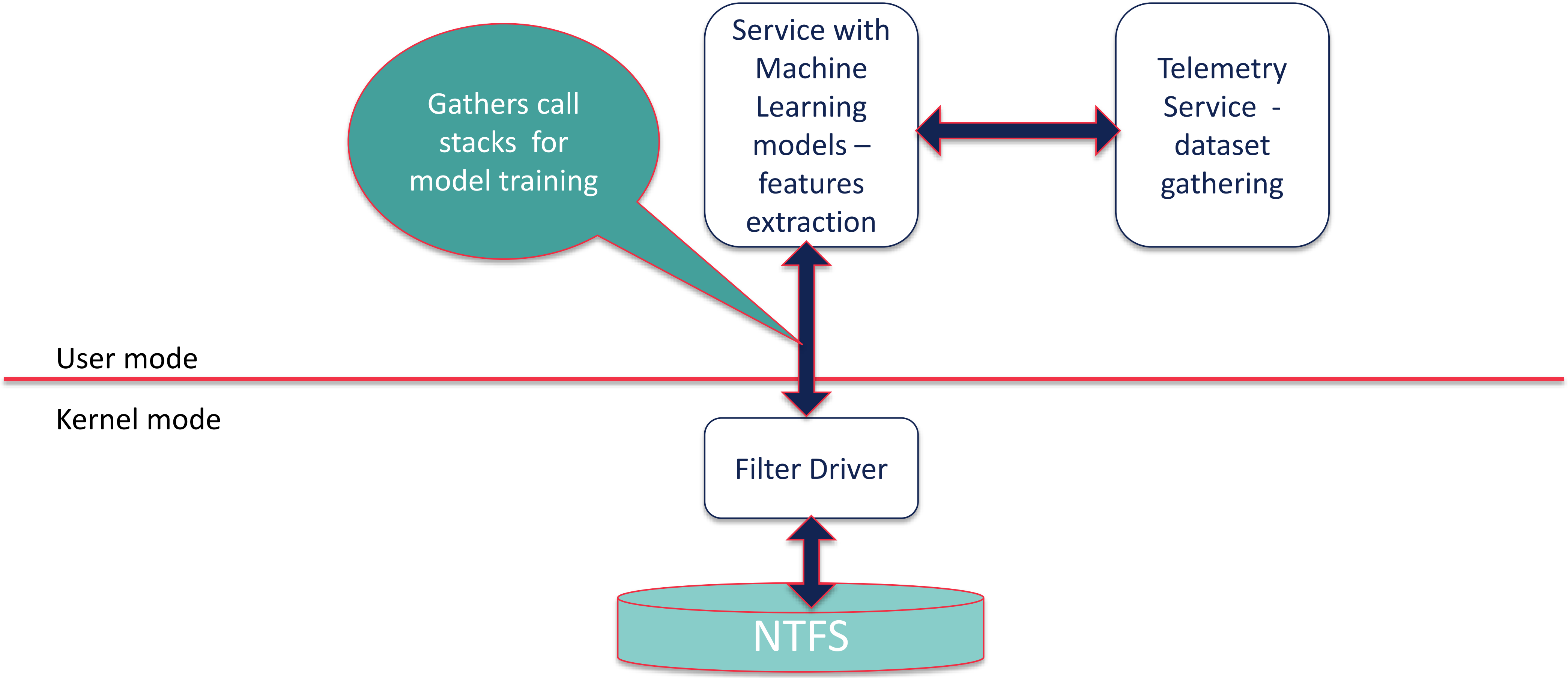
Architecture of the Anti-Ransomware Solution

Windows File system filter driver, advanced call stack analyzer, Machine Learning system

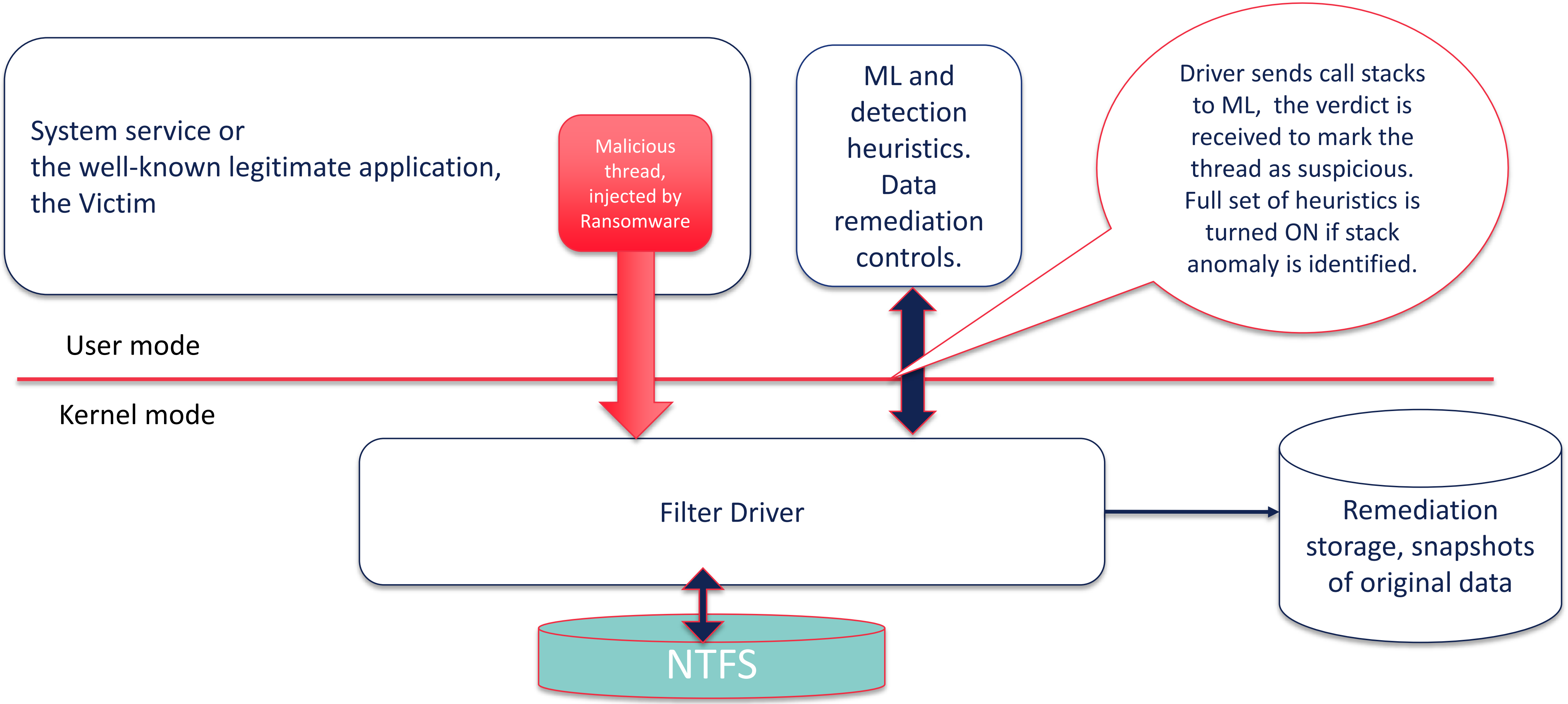
Fighting Advanced Ransomware: Main Steps

- Monitor injections using RtlCaptureStackBackTrace.
- Analyze injections with Machine Learning Model.
- Start data protection per the injection affected process.
- Analyze process behavior.
- When the detection decision is made, recover the encrypted files and terminate hostile injected objects.

System diagram – training mode (malicious or good scenario)



System diagram – production mode



RSAConference2022

Machine Learning based on call stacks

Analysis of injections during execution

Malware Inject Detection By API Call Sequence

Suspicious Example: Create Thread operation

Modules to which return addresses on stack belong

Returned address in the allocated memory doesn't belong to any processes

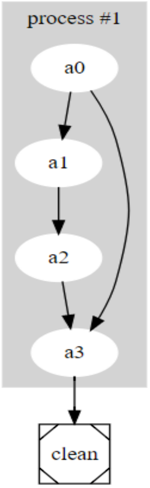
Clean Example: Create Section operation

Just-in-time code compilation: whitelisted

ML model training pipeline

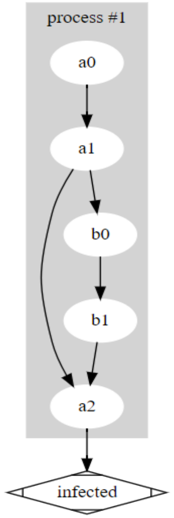
Clean processes

Module	Address	Size
AcronisAccessCli...	0xf70000	0x46000
_hashlib.pyd	0x15b0000	0x49000
_sqlite.pyd	0x10000000	0x98000
python27.dll	0x1e000000	0x23b000
wow64win.dll	0x5cb50000	0x77000
wow64.dll	0x5cbd0000	0x52000
wow64cpu.dll	0x5cc30000	0xa000
MSVCR90.dll	0x73540000	0xa3000
CRYPTBASE.dll	0x740d0000	0xa000
SspiCli.dll	0x740e0000	0x1e000
cfgmgr32.dll	0x74120000	0x36000
KERNELBASE.dll	0x74160000	0x1a1000
ucrtbase.dll	0x744d0000	0xe0000
profapi.dll	0x74610000	0xf000



Infected processes

Module	Address	Size
zlib1.dll	0xf70000	0x46000
wshhyperv.dll	0x15b0000	0x49000
mswsock.dll	0x10000000	0x98000
bcryptPrimitives.	0x1e000000	0x23b000
KERNELBASE.dll	0x5cb50000	0x77000
ucrtbase.dll	0x5cbd0000	0x52000
combase.dll	0x5cc30000	0xa000
RPCRT4.dll	0x73540000	0xa3000
ADVAPI32.dll	0x740d0000	0xa000
WS2_32.dll	0x740e0000	0x1e000
sechost.dll	0x74120000	0x36000
KERNEL32.DLL	0x74160000	0x1a1000
ucrtbase.dll	0x744d0000	0xe0000
profapi.dll	0x74610000	0xf000



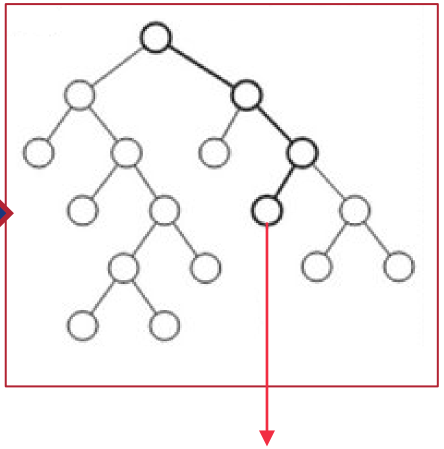
Training dataset

ntkrnlpa.exe,ntdll.dll,KernelBase.dll, ..., **clean**
KernelBase.dll,kernel32.dll,kernel32.dll, ..., **clean**
NetSetupSvc.dll,ELSCore.dll,ELSCore.dll, ...,**clean**
com.docker.9pdb.exe, n/a,cryptsp.dll, ..., **infected**
ntkrnlpa.exe,ntdll.dll,KernelBase.dll, **clean**

...
...
...

ntkrnlpa.exe,ntdll.dll,KernelBase.dll, **clean**
n/a,clr.dll,clr.dll,clr.dll, combase.dll, ..., **clean**

Decision making

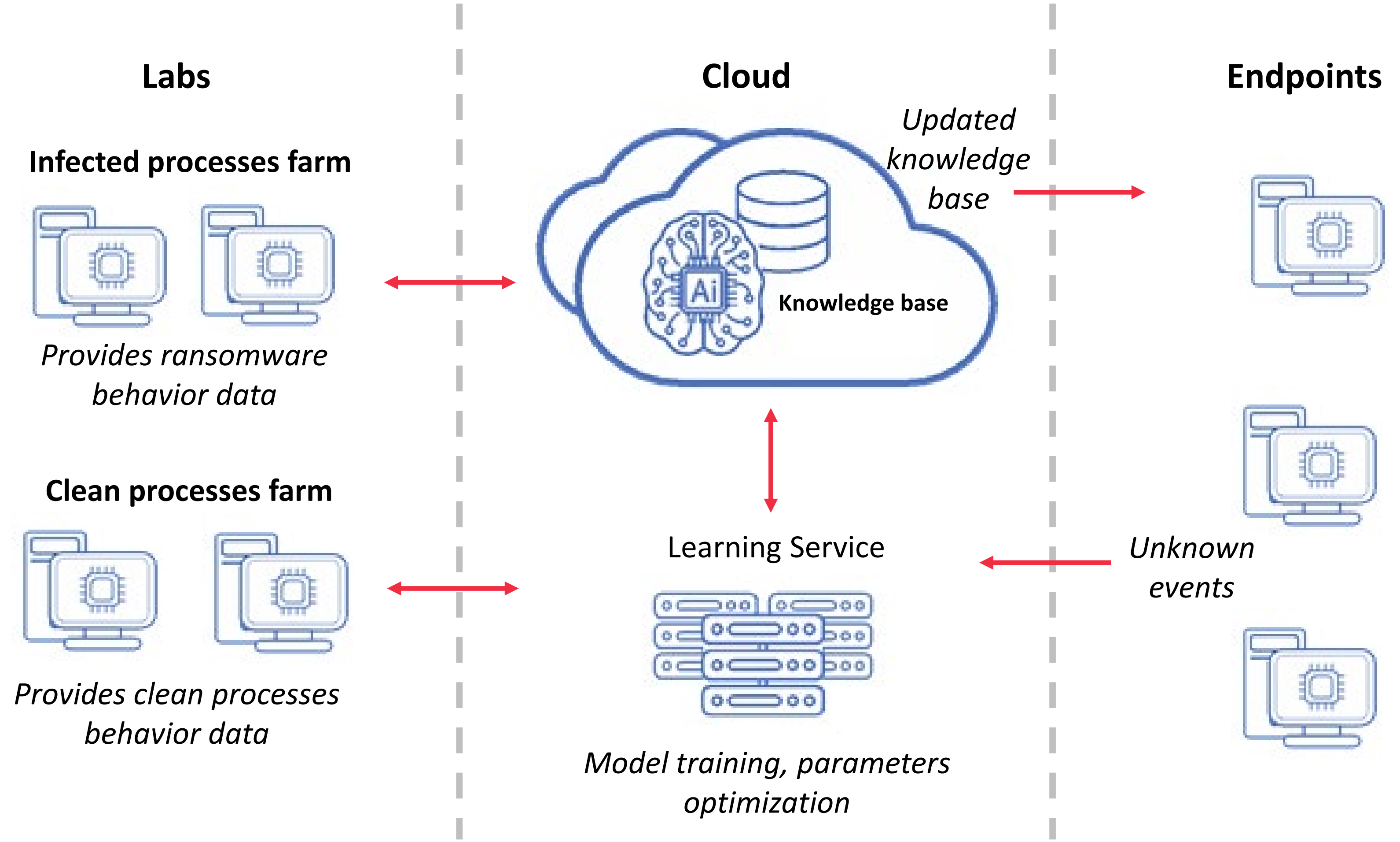


Infected process

ML model determines if the process is clean or not

Trace operation (open, i/o operations), trace normalization for ML, algorithm, remove process ID and other non needed parameters

Learning Infrastructure



Analysis of injections: Models comparison

Samples database: 850M records, **23M** is unique

New samples: 1-2M per day

Stacktrace Analyzer 1.0:

Model: Random Forest

Model details:

- Input – fixed number of frames
- Output – clean/suspicious
- Size – **8M**

Test results:

- Accuracy – 0.96
- Execution Time: 10-20 ms

Stacktrace Analyzer 2.0:

Model: Gradient Boosting Tree

Model details:

- Input – deduplicated frames
- Output – clean/suspicious
- Size – **900K**

Test results:

- Accuracy – 0.98
- Execution Time: 1-5 ms

DEMO

- We launch the Real-world ransomware and demonstrate how the injection is detected and malicious file data modifications are rolled back:
  - The video that demonstrates how the injected stacks are detected https://drive.google.com/file/d/1KKptRRvGEy0ri-2DsdV8U1N203Qh9Eg5/view?usp=sharing
  - The video that shows the post-mortem analysis of files encryption and recovery https://drive.google.com/file/d/1o68zFgRioNEgteaMhhgMXKbEq4pWA3Ti/view?usp=sharing

RSAConference2022

Dealing with false positives of the call stack anomaly detection

Find methods to reduce false positives, connect with other methodologies and detections

How to Reduce False Positives

- The knowledge of injection source helps to reduce false positives.
- Sensors: file system mini-filter callbacks, user mode or hypervisor assisted hooking.
- Validation: whitelisted services or behavior models.

RSAConference2022

Summary

How to get better anti-ransomware protection

The advanced anti-ransomware defense steps

- Detect abnormal injected call stacks.
- Start data protection in real time.
- Do data protection with high granularity.
- Track the behavior of the injected code.
- Make the final verdict and remediation actions.

Enhance anti-ransomware defense with ML

- Gather all types of injections routinely.
- Develop the model training infrastructure.
- Start with simple models like Random Forest.
- Update your model regularly.
- Automate the data annotation process.
- Apply ML to behavior analysis.