Implementation of AI/ML in Threat Hunting

Jess Garcia

Founder & CEO of One eSecurity

@j3ssgarcia

www.one-esecurity.com | www.ds4n6.io

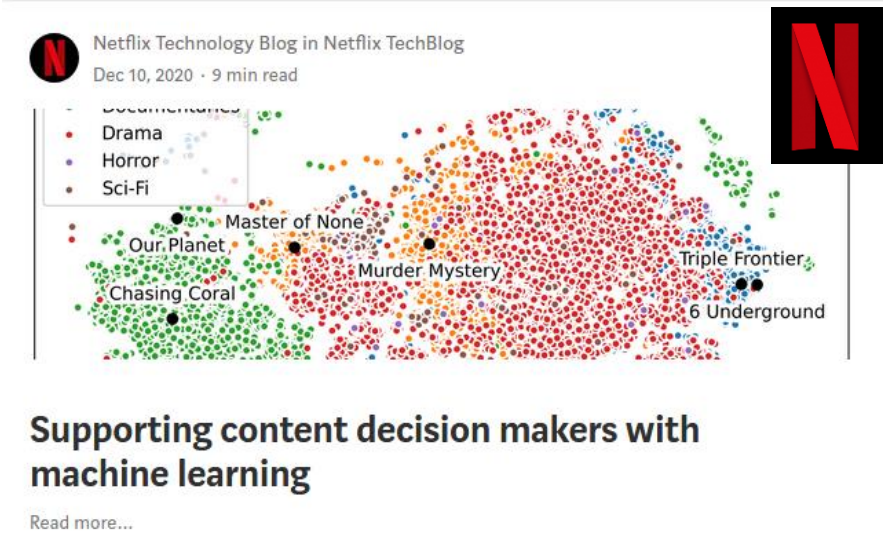
WHO AM I

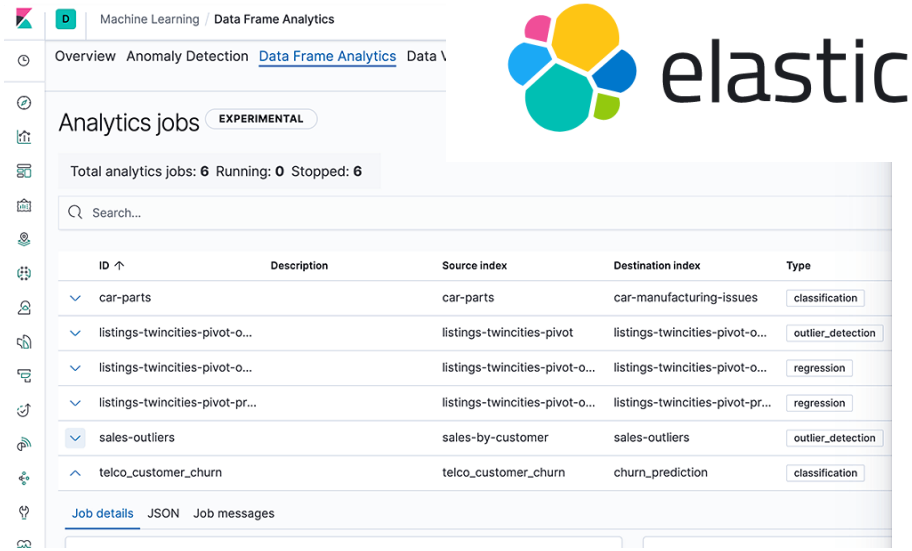
Jess Garcia
@j3ssgarcia

- Founder and CEO of One eSecurity
  +25 years of experience in CybSec / DFIR
- Global DFIR Lead for 15 years
  one-esecurity.com
- DS4N6 Project Lead since 2020
  www.ds4n6.io
- SANS Institute Senior Instructor for 20 years

ML Today

So many technologies and vendors use Machine Learning





elastic

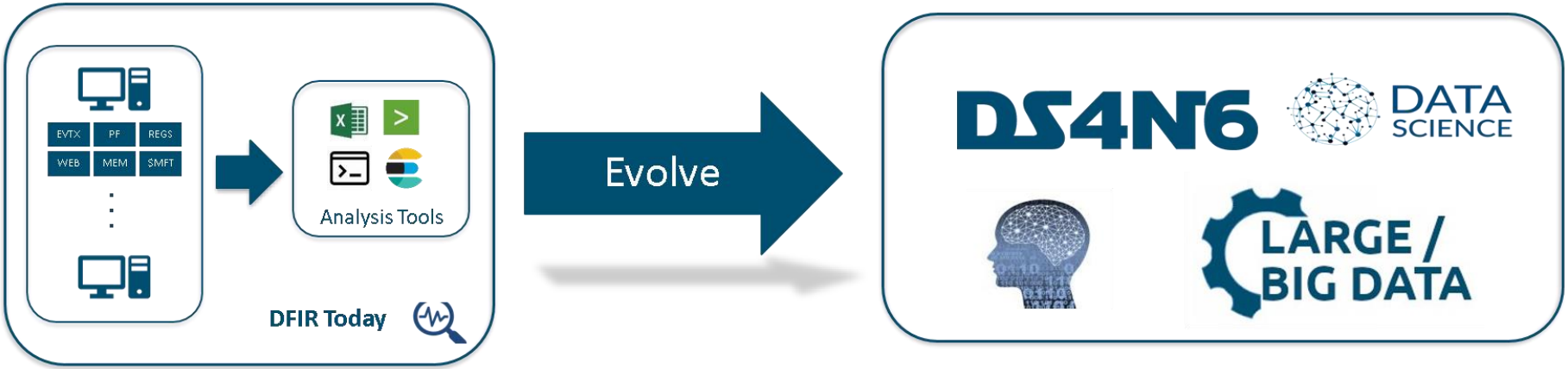
But we know so little about how to use it …

… And in the DFIR Real World?

Why don’t we have anything available to DFIR in terms of Machine Learning?

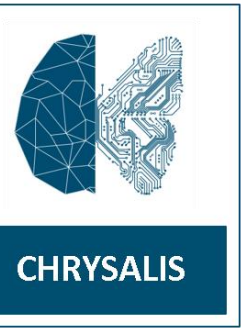
Our Mission

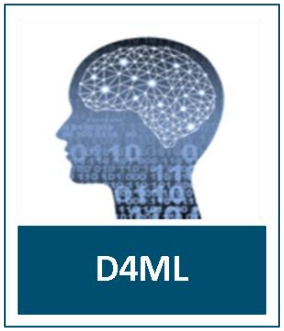
Our DS4N6 project aims to bring Data Science and Artificial Intelligence to the footprints of the average Forensicator, transforming you into an AI-enhanced forensic/threat hunter, incorporating AI into your DFIR arsenal, and furthering advancements in the field.

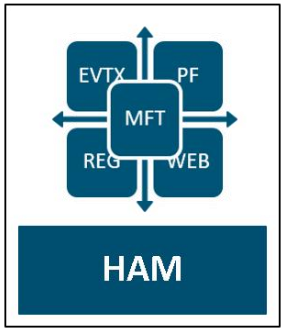


DS4N6: The road so far SINCE 2020

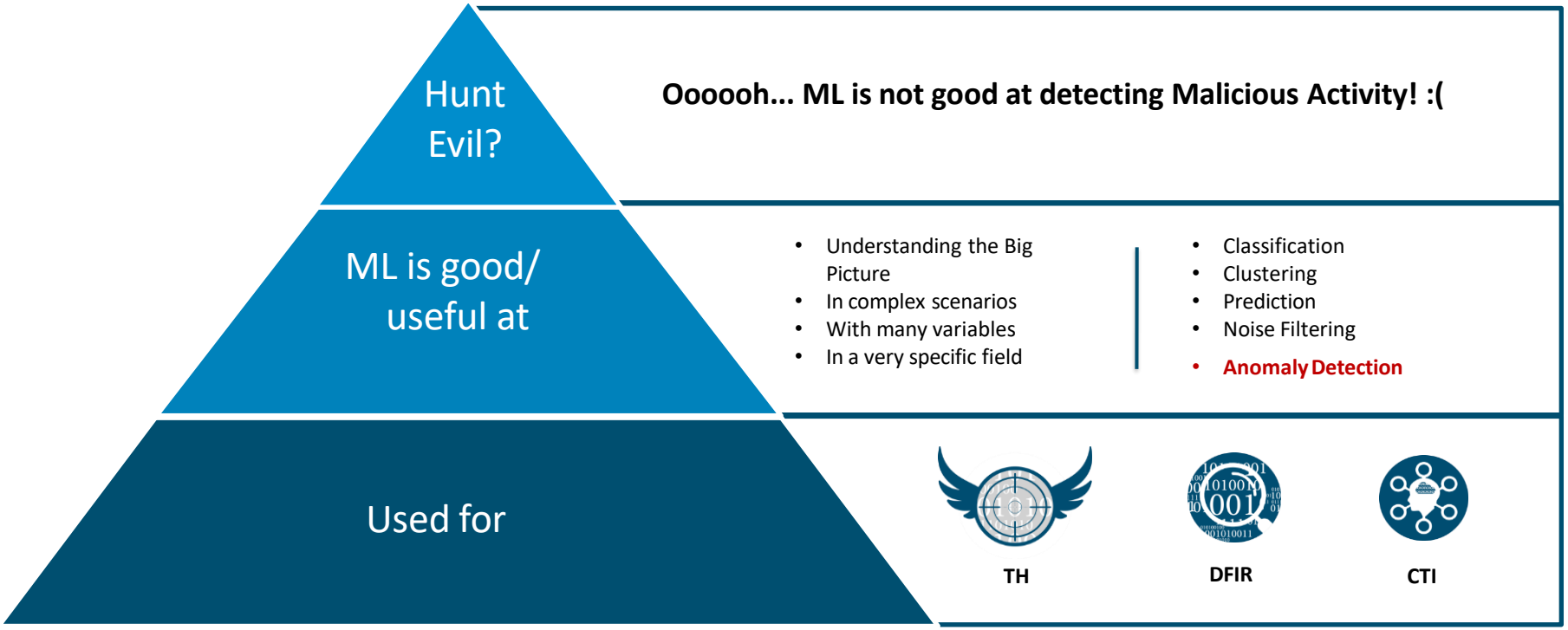
DS4N6



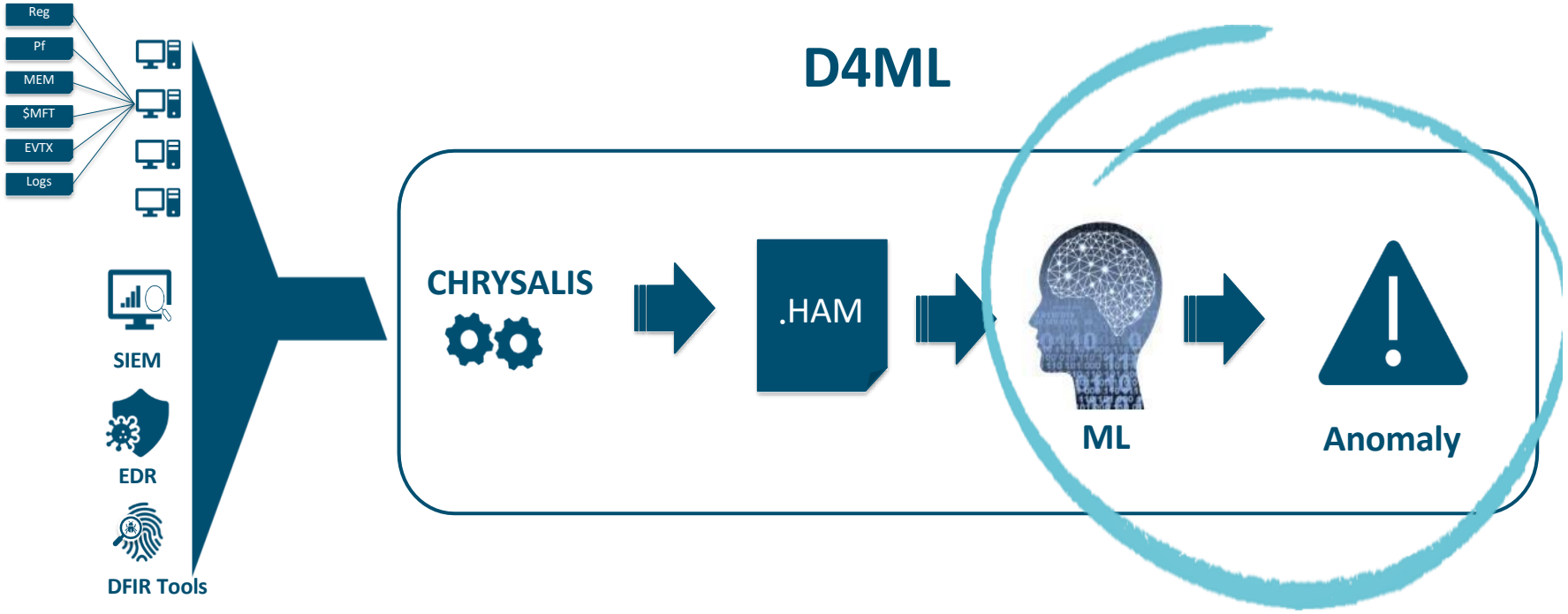




ML for DFIR


Hunt Evil?
Oooooh... ML is not good at detecting Malicious Activity! :(
ML is good/ useful at
Understanding the Big Picture
In complex scenarios
With many variables
In a very specific field
Classification
Clustering
Prediction
Noise Filtering
Anomaly Detection
Used for
TH
DFIR
CTI

Methodology Overview



TODAY BIG QUESTION

Would we be able to detect

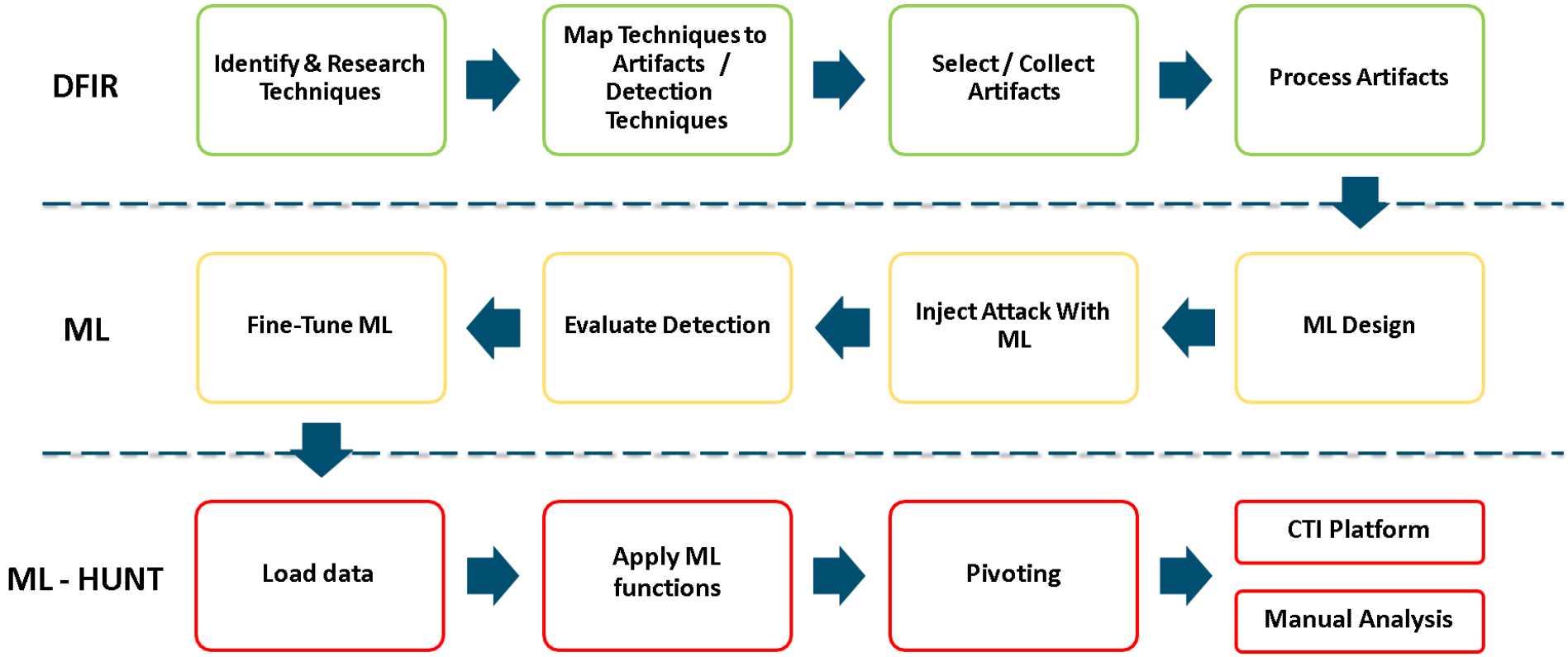
AN UNKNOWN ATTACK

e.g. CONTI

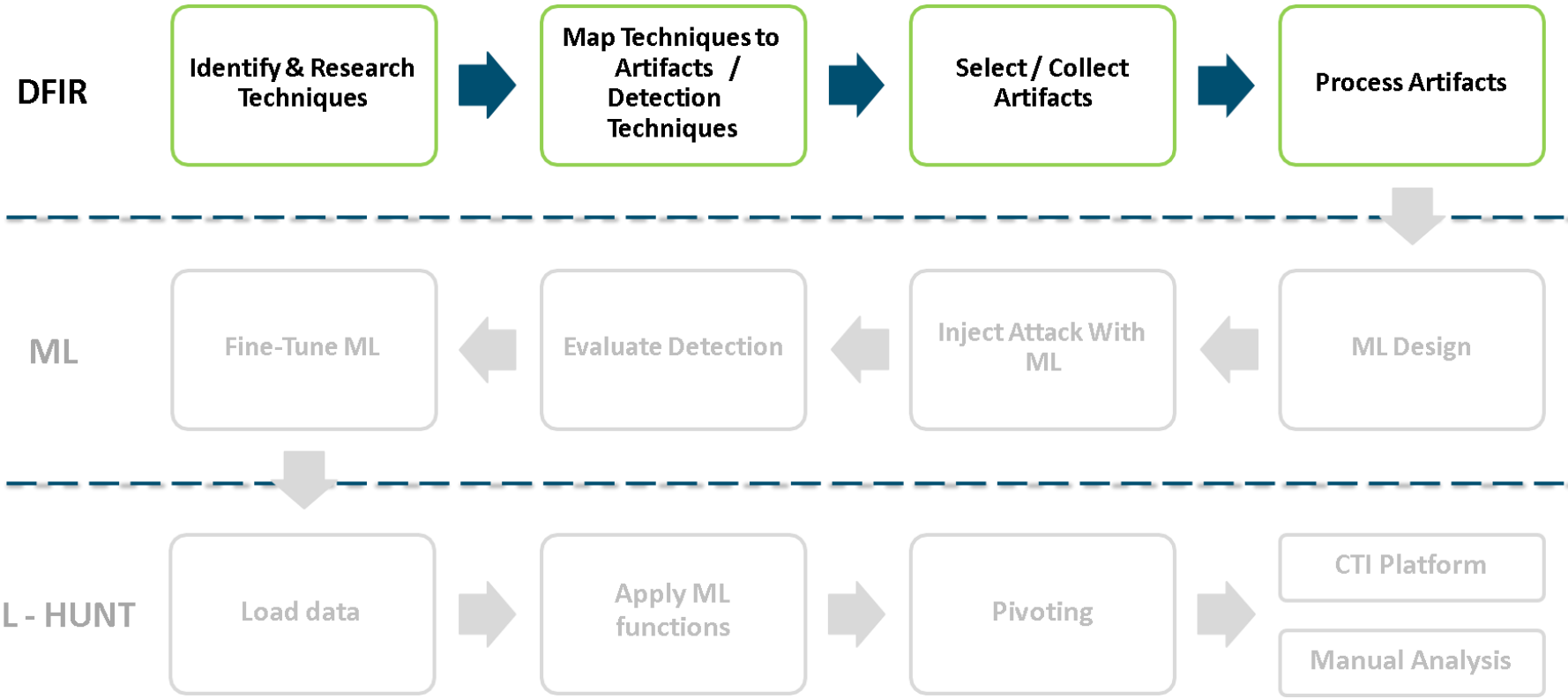
Without **IOCs**

Just by **Multi-Artifact ML Anomaly Analysis**

D4ML Methodology Workflow



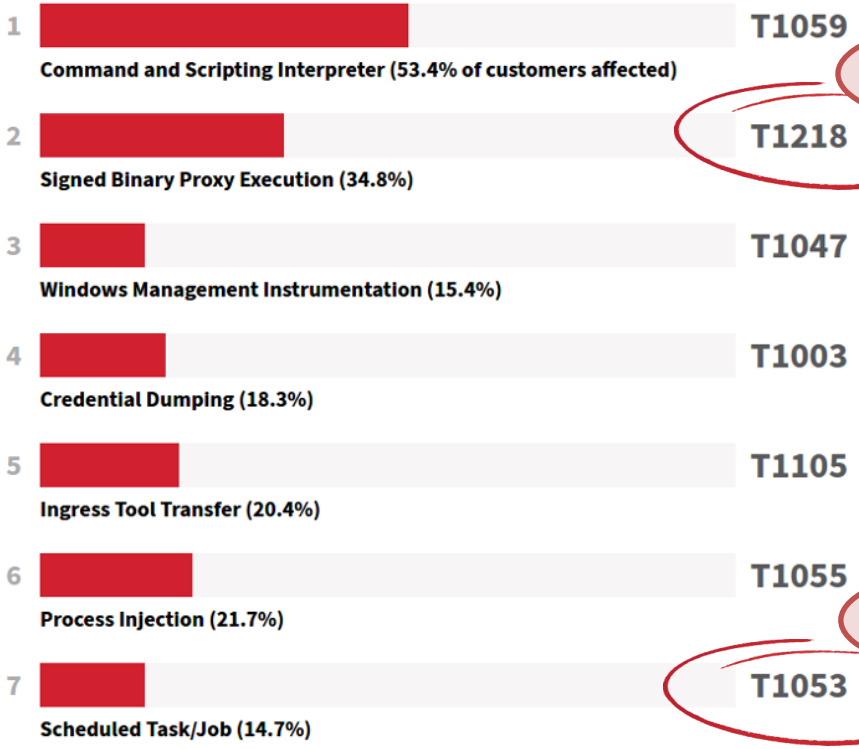
D4ML Methodology Workflow



Select techniques

DFIR | ML | HUNT

red canary



TOP 2

TOP 7

cobaltstrike

MITRE ATT&CK™

TA0001: Initial Access
T1078.003: Malicious Logons

TA0003: Persistence
T1053.005: Scheduled Tasks

TA0005: Defense Evasion
T1218: System Binary Proxy Execution

Research Techniques T1053.005

DFIR | ML | HUNT

Remote Scheduled Task Creation

Metadata

Author	Roberto Rodriguez @Cyb3rWard0g
Creation Date	2020/12/19
Modification Date	2020/12/19
Tactics	['TA0002','TA0003','TA0004','TA0008']
Techniques	['T1053.005']
Tags	None


```
{"SourceName":"Microsoft-Windows-Security-Auditing",
"ProviderGuid":"{54849625-5478-4994-a5ba-3e3b0328c30d}",
"Level":"0",
"Keywords":"0x8020000000000000",
"Channel":"Security",
"Hostname":"WORKSTATION5.theshire.local",
"TimeCreated":"2020-12-19T06:59:28.410Z",
"@timestamp":"2020-12-19T06:59:28.410Z",
"EventID":5158,
"Message":"The Windows Filtering Platform has permitted a bind to a local port.\r
\n\r\nApplication Information:\r\n\tProcess ID:\t\t3108\r\n\tApplication Name:\t\
\device\\harddiskvolume2\\windowsazure\\guestagent_2.7.41491.1005_2020-11-30_0950
05\\guestagent\\windowsazureguestagent.exe\r\n\r\nNetwork Information:\r\n\tSourc
e Address:\t\t0.0.0.0\r\n\tSource Port:\t\t56581\r\n\tProtocol:\t\t6\r\n\r\nFilte
r Information:\r\n\tFilter Run-Time ID:\t0\r\n\tLayer Name:\t\tResource Assignmen
t\r\n\tLayer Run-Time ID:\t36",
"Task":"12810",
"ProcessId":"3108",
"Application":"\\device\\harddiskvolume2\\windowsazure\\guestagent_2.7.41491.1005
_2020-11-30_095005\\guestagent\\windowsazureguestagent.exe",
"SourceAddress":"0.0.0.0",
"SourcePort":"56581",
"Protocol":"6",
"FilterRTID":"0",
"LayerName":"%14608",
"LayerRTID":"36"}
```


Adversary View

```
PS C:\windows\system32> C:\Windows\system32\cmd.exe /C schtasks /create /F /tn "\Microsoft\Windows\
SUCCESS: The scheduled task "\Microsoft\Windows\SoftwareProtectionPlatform\EventCacheManager" has s
PS C:\windows\system32>
```

Datasets Downloads

Dataset Type	Link
Host	https://raw.githubusercontent.com/OTRF/mordor/master/datasets/small/windows/lateral_movement/host/schtask_create.zip
Network	https://raw.githubusercontent.com/OTRF/mordor/master/datasets/small/windows/lateral_movement/network/schtask_create.zip

The Mordor Project

https://mordordatasets.com

Task Scheduler Service

Roberto Rodriguez @Cyb3rWard0g

The Task Scheduler service allows you to perform automated tasks on a chosen computer. With this service, you can schedule any program to run at a convenient time for you or when a specific event occurs.

Task Scheduler Service Remote Protocol

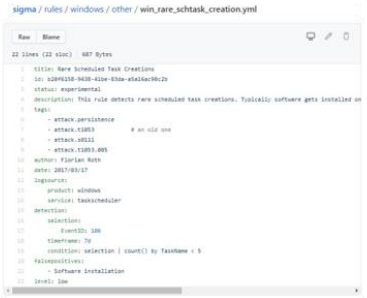
- RPC Server
  - **Name:** ITaskSchedulerService
  - **UUID:** 86d35949-83c9-4044-b424-db363231fd0c
  - **FilePath:** C:\Windows\System32\schedsvc.dll

Microsoft-Windows-TaskScheduler/Operational Logs

- EventID 140: Task Registration Updated
- EventID 100: Task Started
- EventID 201: Action Completed
- EventID 201: Task Completed
- EventID 111: Task Terminated
- EventID 332: Launch request ignored. Instance already running

https://threathunterplaybook.com

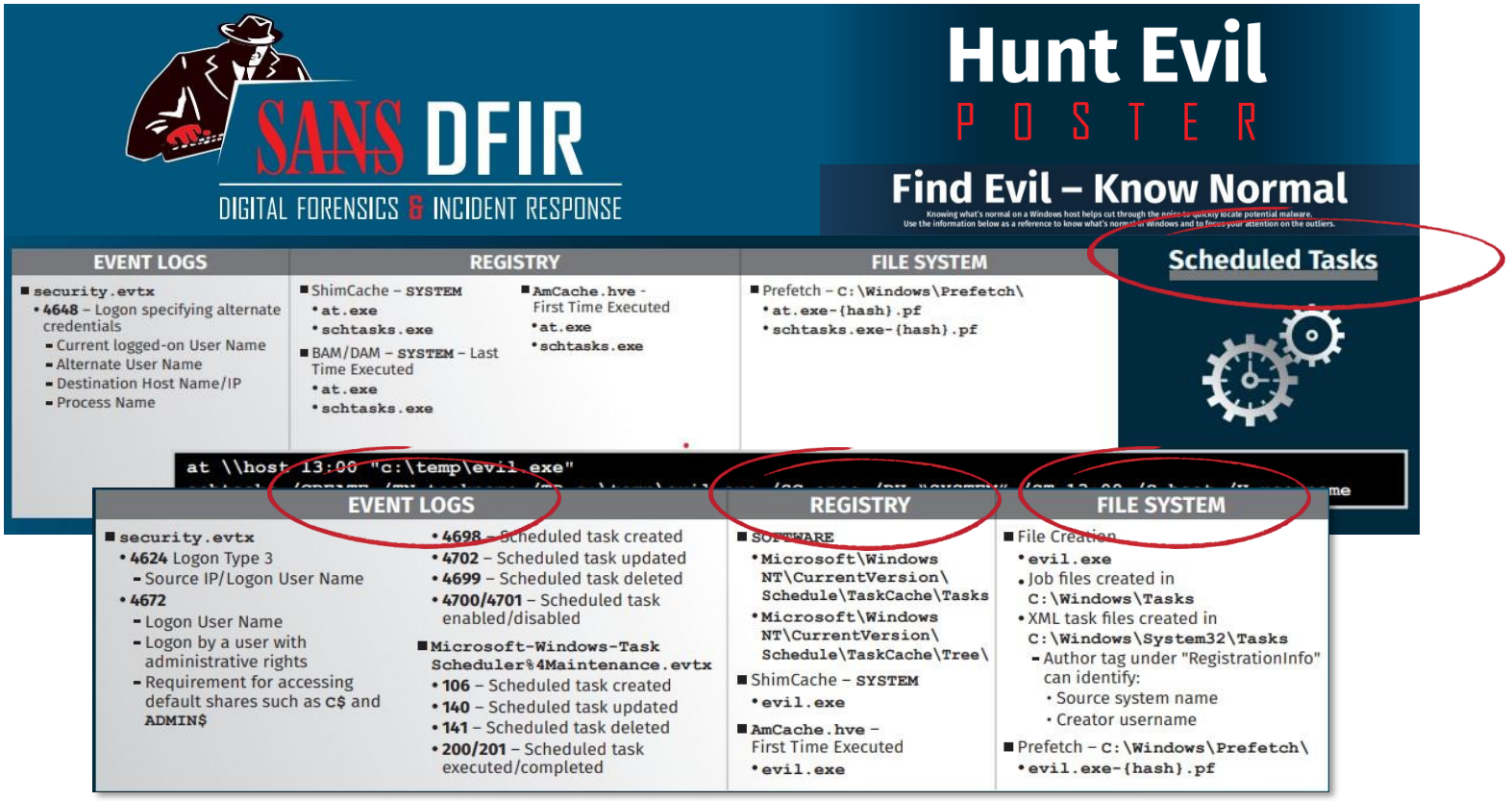
SIGMA



https://github.com/Neo23x0/sigma

Research Techniques T1053.005

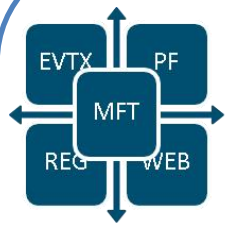
DFIR | ML | HUNT



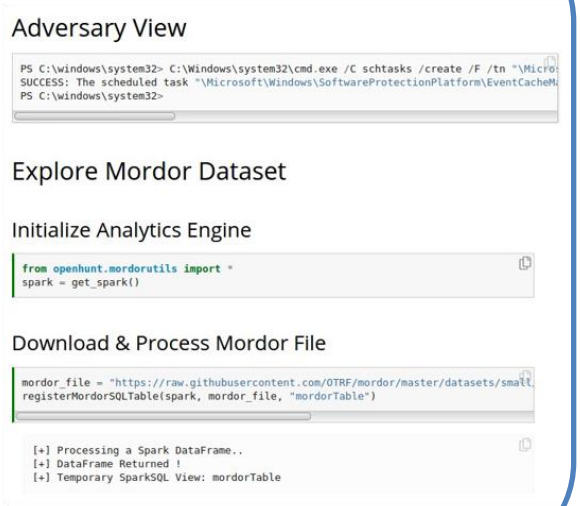
https://www.sans.org/security-resources/posters/dfir/hunt-evil-165

Map Techniques to Artifacts / Detection Techniques

DFIR | ML | HUNT



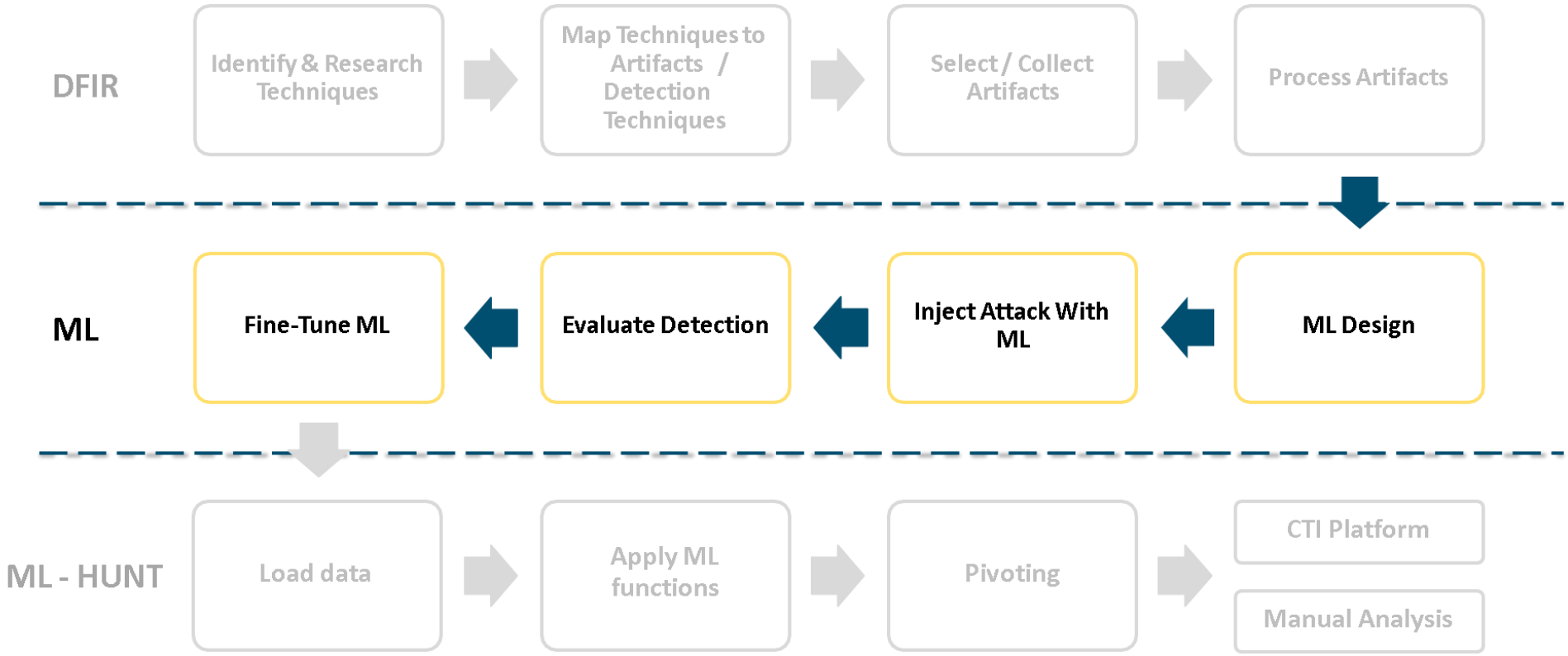

```yaml
title: Persistence and Execution at Scale via GPO Scheduled Task
id: a8f29a7b-b137-4446-80a0-b804272f3da2
description: Detect lateral movement using GPO scheduled task, usually used to deploy ransomware at scale
author: Samir Bousseaden
date: 2019/04/03
references:
    - https://twitter.com/menasec1/status/1106899890377052160
    - https://www.secureworks.com/blog/ransomware-as-a-distraction
tags:
    - attack.persistence
    - attack.lateral_movement
    - attack.t1053          # an old one
    - attack.t1053.005
logsource:
    product: windows
    service: security
    definition: 'The advanced audit policy setting "Object Access > Audit Detailed File Share" must be configured for Success/Failure'
detection:
    selection:
        EventID: 5145
        ShareName: \\*\SYSVOL
        RelativeTargetName: '*ScheduledTasks.xml'
        Accesses: '*WriteData*'
    condition: selection
falsepositives:
    - if the source IP is not localhost then it's super suspicious, better to monitor both local and remote changes to GPO scheduledtasks
level: high
```



Collect / Process Artifacts

DFIR | ML | HUNT

D4ML Methodology Workflow



DS Harmonization (HAM) vs ML Feature Engineering (HML)

DFIR | **ML** | HUNT

Philosophy

- **Flexibility** (many customers, many realities)
- **Open Source**
- Bring artifacts **however you want**
- **Process artifacts** with whatever you want
- **Compatible with all tools** (plaso, kansa, kape, YourCoolTool™)

Raw

```
0|/$AttrDef ($FILE_NAME)|4-48-2|r/rr-xr-xr-x|48|0|82|1508354468|1508354468|1508354468|1508354468
0|/$AttrDef|4-128-4|r/rr-xr-xr-x|48|0|2560|1508354468|1508354468|1508354468|1508354468
0|/$BadClus ($FILE_NAME)|8-48-3|r/rr-xr-xr-x|0|0|82|1508354468|1508354468|1508354468|1508354468
0|/$BadClus|8-128-2|r/rr-xr-xr-x|0|0|0|1508354468|1508354468|1508354468|1508354468
0|/$BadClus:$Bad|8-128-1|r/rr-xr-xr-x|0|0|20948447232|1508354468|1508354468|1508354468|1508354468
0|/$Bitmap ($FILE_NAME)|6-48-2|r/rr-xr-xr-x|0|0|80|1508354468|1508354468|1508354468|1508354468
0|/$Bitmap|6-128-4|r/rr-xr-xr-x|0|0|639296|1508354468|1508354468|1508354468|1508354468
0|/$Boot ($FILE_NAME)|7-48-2|r/rr-xr-xr-x|48|0|76|1508354468|1508354468|1508354468|1508354468
0|/$Boot|7-128-1|r/rr-xr-xr-x|48|0|8192|1508354468|1508354468|1508354468|1508354468
0|/$Extend ($FILE_NAME)|11-48-3|d/dr-xr-xr-x|0|0|80|1508354468|1508354468|1508354468|1508354468
0|/$Extend|11-144-4|d/dr-xr-xr-x|0|0|656|1508354468|1508354468|1508354468|1508354468
0|/$Extend/$Deleted ($FILE_NAME)|24-48-1|d/dr-xr-xr-x|0|0|82|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$Deleted|24-144-2|d/dr-xr-xr-x|0|0|48|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$ObjId ($FILE_NAME)|26-48-1|r/rr-xr-xr-x|0|0|78|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$ObjId:$O|26-144-5|r/rr-xr-xr-x|0|0|56|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$Quota ($FILE_NAME)|25-48-1|r/rr-xr-xr-x|0|0|78|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$Quota:$O|25-144-3|r/rr-xr-xr-x|0|0|88|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$Quota:$Q|25-144-2|r/rr-xr-xr-x|0|0|208|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$Reparse ($FILE_NAME)|27-48-1|r/rr-xr-xr-x|0|0|82|1508354469|1508354469|1508354469|1508354469
0|/$Extend/$Reparse:$R|27-144-5|r/rr-xr-xr-x|0|0|56|1508354469|1508354469|1508354469|1508354469
```

HAM

```
D4_DataType_              category
D4_Orchestrator_          category
D4_Tool_                  category
D4_Plugin_                category
D4_Hostname_              category
MTStampEpoch_              float64
MTStamp_                datetime64[ns]
MTStampDate_                object
MTStampTime_                object
MTStampDoW_                 object
ATStampEpoch_              float64
ATStamp_                datetime64[ns]
ATStampDate_                object
ATStampTime_                object
ATStampDoW_                 object
CTStampEpoch_              float64
CTStamp_                datetime64[ns]
CTStampDate_                object
CTStampTime_                object
CTStampDoW_                 object
Size_                        Int64
Meta_                        Int64
Type_                       object
FilePath_                   object
FileName_                   object
FileStem_                   object
FileExtension_              object
ParentPath_                 object
ParentName_                 object
PathSeparator_              object
FilePath-Hash_               Int64
FileName-Hash_               Int64
FileStem-Hash_               Int64
ParentPath-Hash_             Int64
ParentName-Hash_             Int64
```

Feature Selection

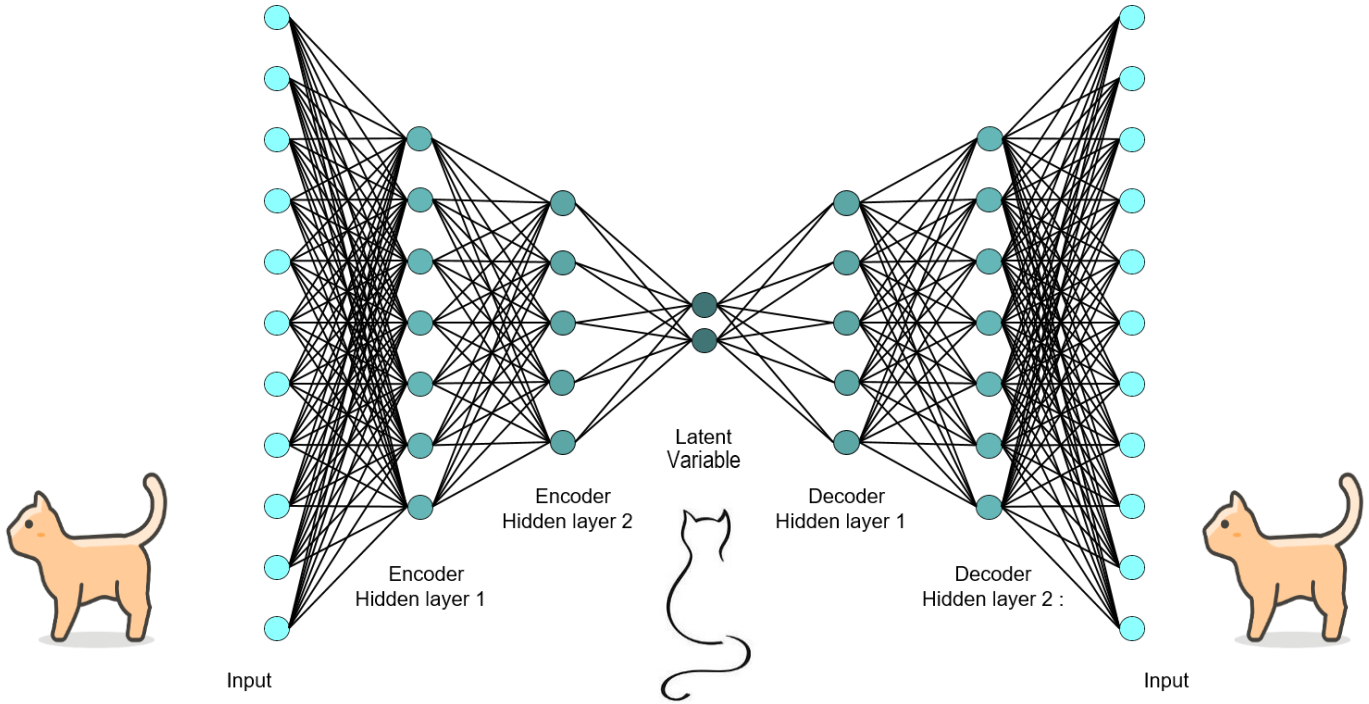
Feature Engineering

HML

```
Hostname_          category
FilePath_            object
MTStampDate_         object
MTStampTime_         object
ATStampDate_         object
ATStampTime_         object
CTStampDate_         object
CTStampTime_         object
Size_                 int64
```

AutoEncoders for Anomaly Detection

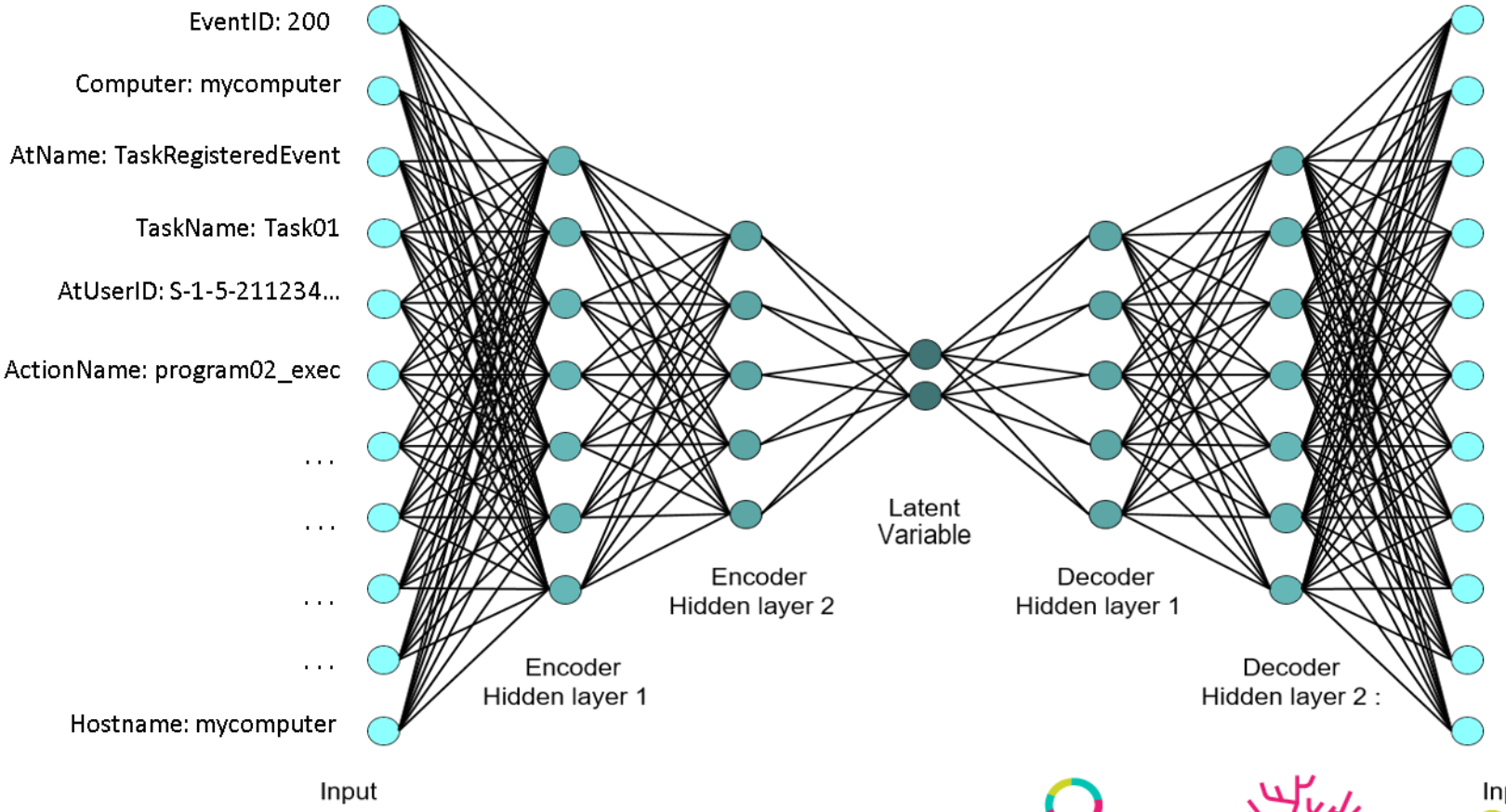
DFIR | **ML** | HUNT



Source:
https://towardsdatascience.com/extreme-rare-event-classification-using-autoencoders-in-keras-a565b386f098

AutoEncoders for Anomaly Detection

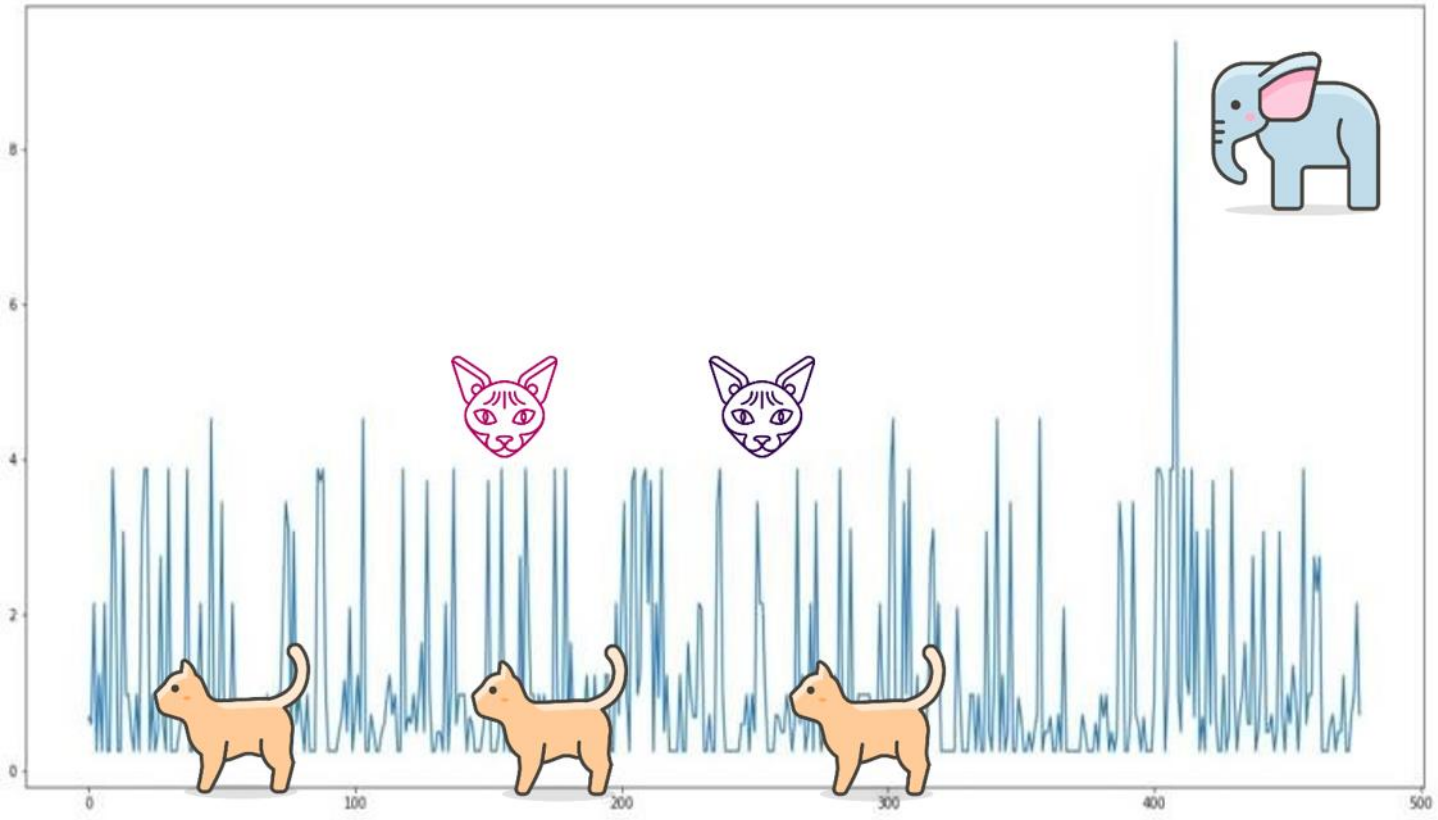
DFIR | ML | HUNT



Standard (Vanilla) AutoEncoders

DFIR | ML | HUNT

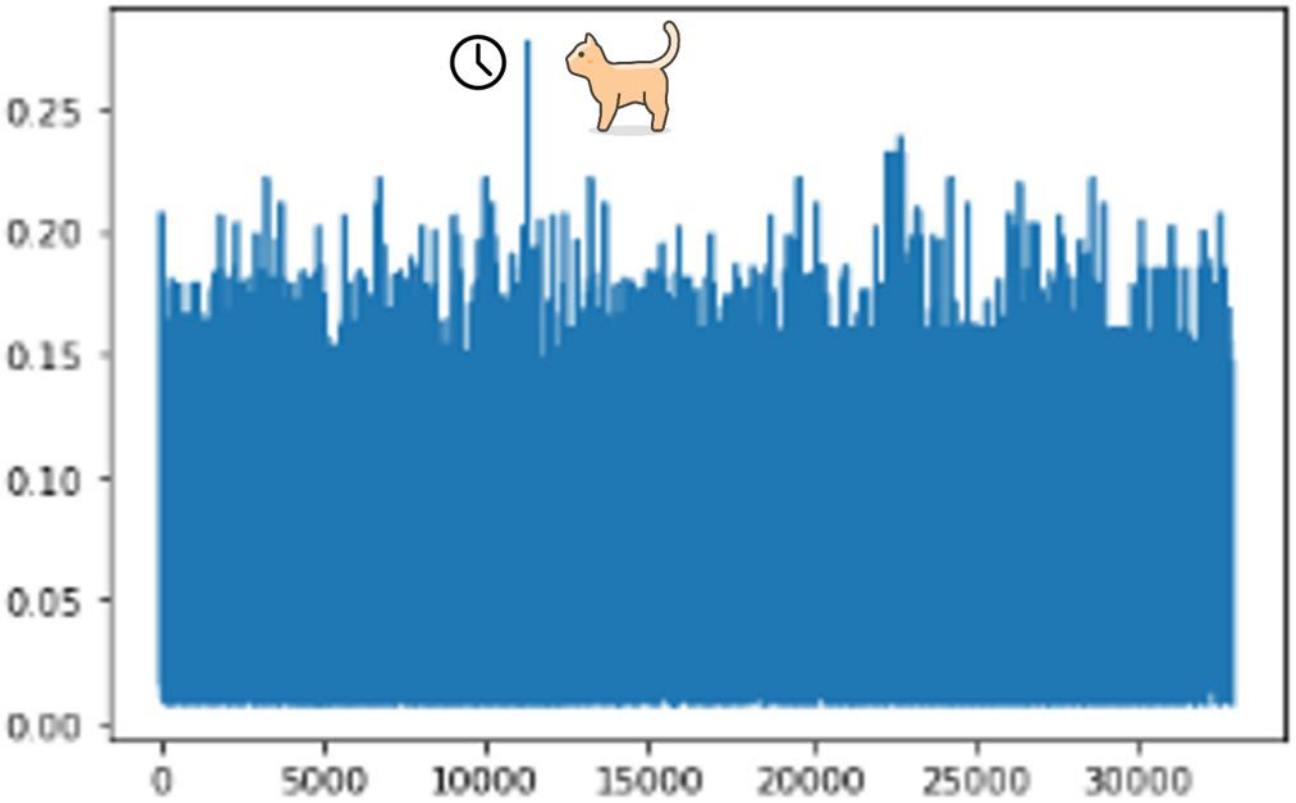
The elephant VS the cat



LSTM AutoEncoder

DFIR | ML | HUNT

The right cat at the wrong time



Inject Attack With ADAM – The DS4N6 ADversAry eMulator

DFIR | ML | HUNT


```
title: Sheduled Task Jobs
name: sheduled_task_jobs
id: 52753ea4-b3a0-4365-910d-36cff487b789
status: experimental
description: Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code.
references:
    - https://attack.mitre.org/techniques/T1053/
tags:
    - attack.persistence
    - attack.t1053
    - attack.t1053.005
date: 2021/02/02
author: Jess Garcia
source:
    - type:
        datatype: evtx-ham
        dfname: tskopevtxdf
        objtype: df
        copycond: 'EventID_ == 200'
        columns: [EventID_, Computer_,     AtName_,       TaskName_,          AtUserID_,      ActionName_,      ResultCode_,      UserNC_,   Hostname_]
        row: [    "200",    "%%computer%%", "%%atname%%", "\\%%taskname%%", "%%atuserid%%", "%%actionname%%", "%%resultcode%%", "%%usernc%%", "%%computer%%"]
        resequencecol: 'EventRecordID'
        sort: 'EventRecordID'
    - type:
        datatype: flist-ham
        dfname: tskflistdf
        objtype: df
        copycond: any
        columns: [Hostname_,      FilePath_,      MTStampDate_,      ATStampDate_,      CTStampDate_,      Size_]
        row:     ["%%hostname%%", "%%filepath%%", "%%mtstampdate%%", "%%atstampdate%%", "%%ctstampdate%%", "%%size%%" ]
        sort: 'MTStampDate_'
~
```

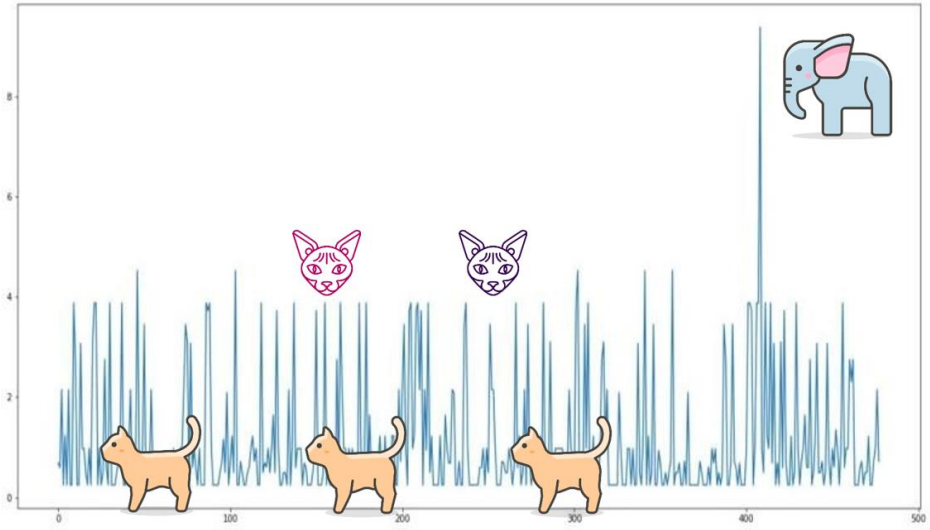
```
* Executing rule: scheduled_task_jobs ...

Title: Sheduled Task Jobs
Status: experimental
Description: Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code.
Date: 2021/02/02

+ Building tskflistdf
- Injection Done.
(11332, 6)
```

	Hostname_	FilePath_	MTStampDate_	ATStampDate_	CTStampDate_	Size_
11331	victimcomputer	\EvilTask	2021-02-01	2021-02-01	2021-02-01	66666666666

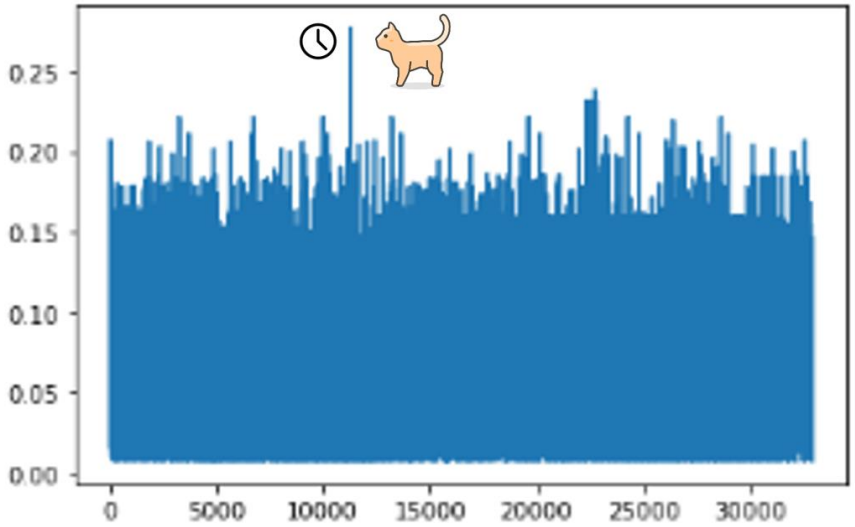
Evaluate Detection

DFIR | **ML** | HUNT

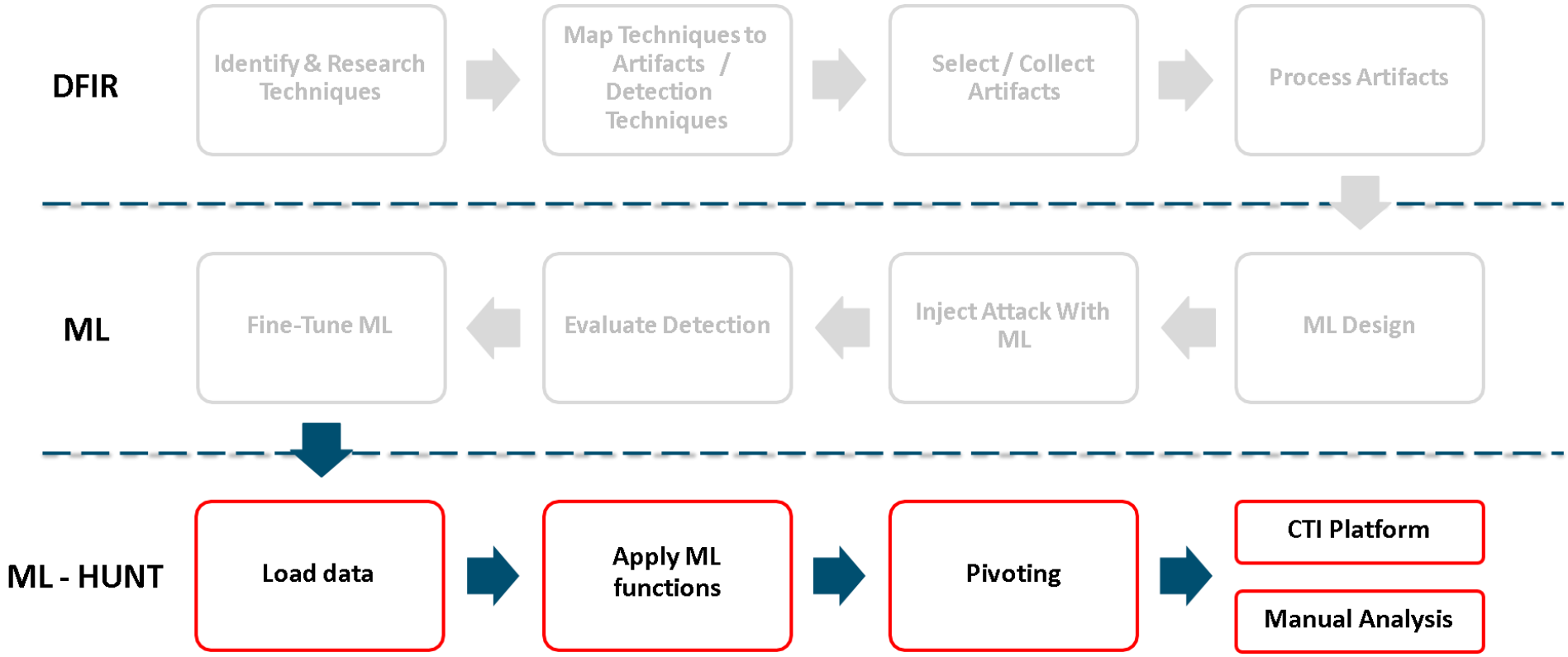
Simple AutoEncoder



Long Short-Term Memory AutoEncoder

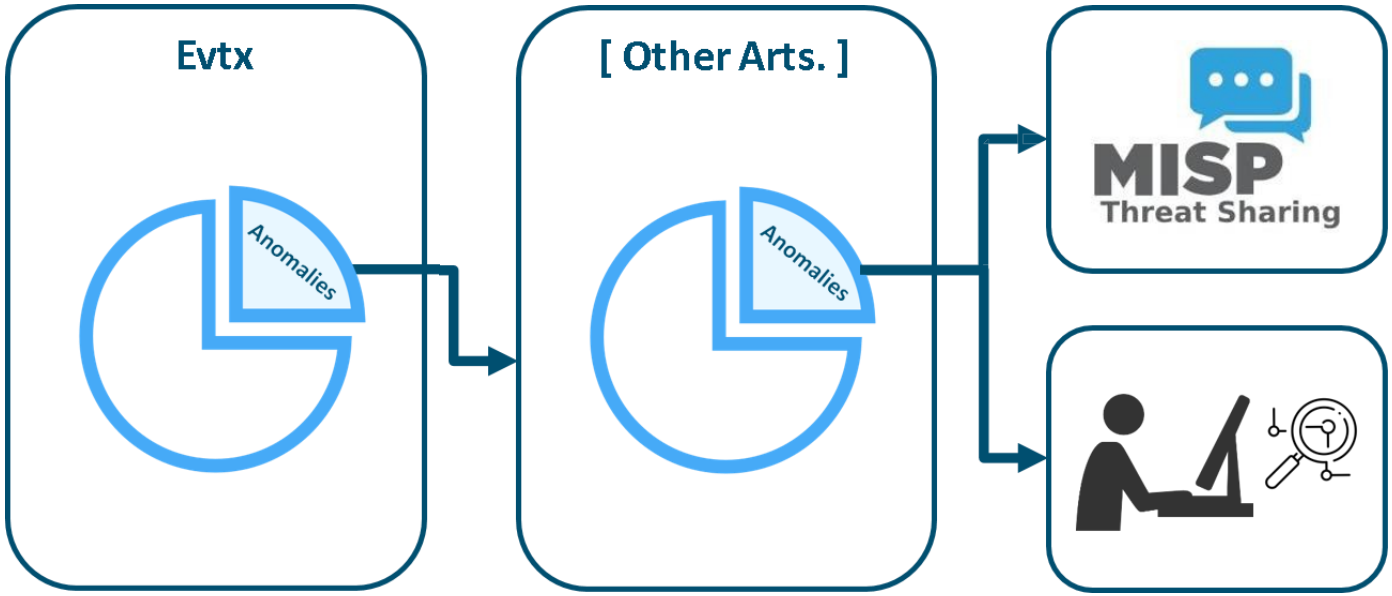


D4ML Methodology Workflow

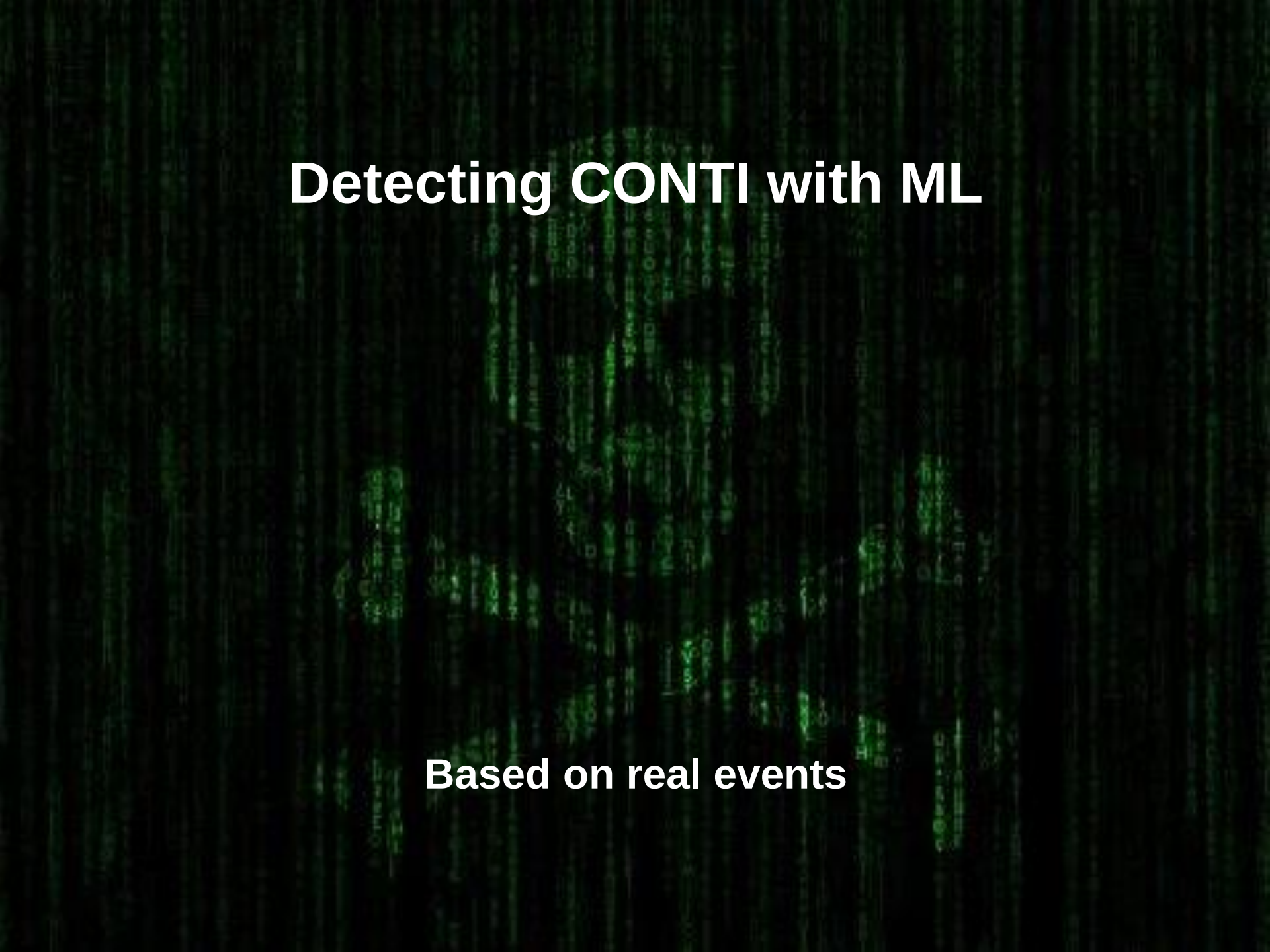


Optional Pivoting and MISP/Manual analysis

DFIR ML **HUNT**



Detecting CONTI with ML



Based on real events

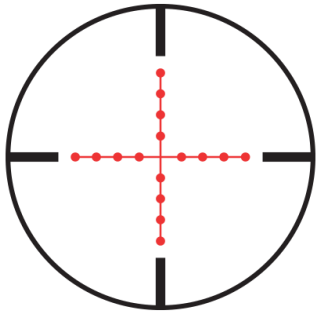
The Ransomware Attack

Global Company

The attack could spread

CONTI

TOP Threat Actor from Russia using Cobalt Strike



Worldwide Scope

5k Servers + 350 DCs + 12k Laptops

III Encuentro ENS
'Ransomware: lecciones desde las trincheras ':
one-esecurity.com/ens21

IV Encuentro ENS
'Desde las trincheras: Threat Hunting – Cazando y luchando contra los adversarios ':
one-esecurity.com/ens22

II Jornadas STIC Capítulo Colombia
'Desde las trincheras: Sobreviviendo a una intrusión avanzada de ransomware'
'Sobreviviendo a una intrusión avanzada de ransomware: Claves de éxito'
one-esecurity.com/sticco22

The Breach. Day 0

SOC Alert!

Pre-Ransomware tools found

5 infected hosts

5 days since intrusion

Possibly spread

Detecting The Enemy

We will detect the intrusion in different phases

MITRE ATT&CK™ NEW

Detecting Cobalt Strike with prefetch

TA0001: Initial Access
T1078.003: Malicious Logons

ID: T1078.003

Sub-technique of: T1078

① Tactics: Defense Evasion, Persistence, Privilege Escalation, Initial Access

① Platforms: Containers, Linux, Windows, macOS

① Permissions Required: Administrator, User

Version: 1.2

Created: 13 March 2020

Last Modified: 18 October 2021

TA0003: Persistence
T1053.005: Scheduled

ID: T1053.005

Sub-technique of: T1053

① Tactics: Execution, Persistence, Privilege Escalation

① Platforms: Windows

① Permissions Required: Administrator

① Supports Remote: Yes

Contributors: Andrew Northern, @ex_raritas; Bryan Campbell, @bry_campbell; Selena Larson, @selenalarson; Zachary Abzug, @ZackDoesML

Version: 1.1

Created: 27 November 2019

Last Modified: 14 April 2022

TA0005: Defense Evasion
T1218: System Binary Proxy

ID: T1218

Sub-techniques: T1218.001, T1218.002, T1218.003, T1218.004, T1218.005, T1218.007, T1218.008, T1218.009, T1218.010, T1218.011, T1218.012, T1218.013, T1218.014

① Tactic: Defense Evasion

① Platforms: Linux, Windows, macOS

① Defense Bypassed: Anti-virus, Application control, Digital Certificate Validation

Contributors: Hans Christoffer Gaardløs; Nishan Maharjan, @loki248; Praetorian; Wes Hurd

Version: 3.0

Created: 18 April 2018

Last Modified: 18 April 2022

New Lethal Forensicator Technique!


DEMO
TIME
(one)
eSecurity

Summary

Everything you've seen here **is open source**
The processing is based on open source tools (Plaso, etc.)

CHRYSALIS is open source and allows you to apply this methodology **to multiple types of artifacts**

All the analysis you've seen has been performed on **Real World Data** from a partnering **Fortune 500 company**. **SO IT WORKS!**

We want **ML** to become **one more tool in your arsenal**

All the details about this talk:

ds4n6.io/umacv22

THANK YOU

Jess Garcia

@j3ssgarcia

DS4N6

ds4n6.io

@ds4n6_io

DS4N6

In collaboration with:

- Beatriz Padilla
- David Contreras
- Luis Cortes Ferre

(one) eSecurity

one-esecurity.com

One_eSecurity

One eSecurity