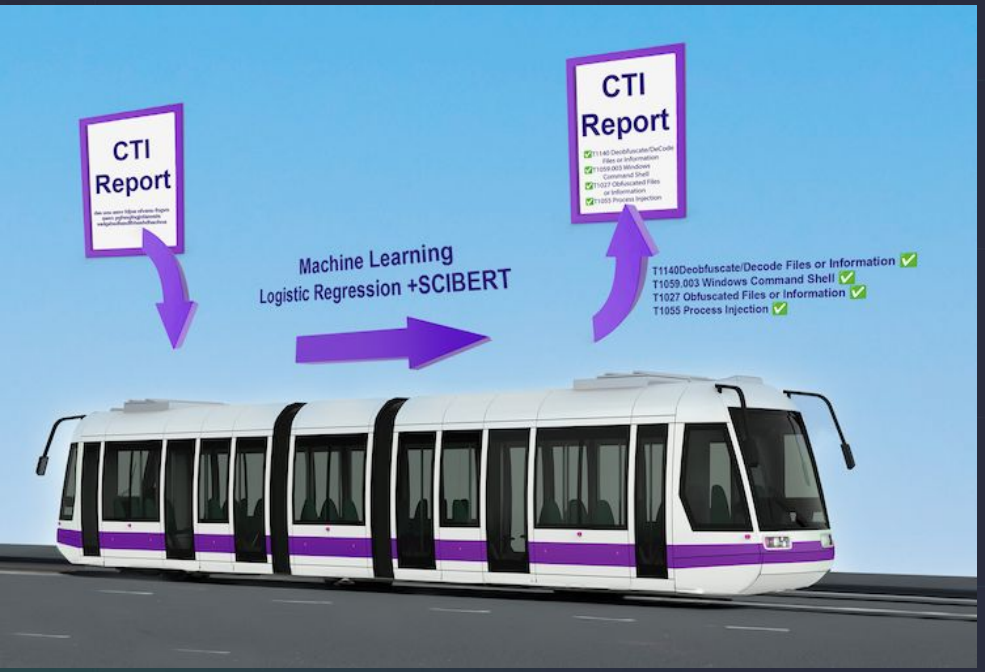
12th EU MITRE ATT&CK® Community Workshop

TRAM LLM tuning

by Andrii Bezverkhyi | founder & CEO @ SOC Prime,
inventor of uncoder.io

What is TRAM? don’t hurt me, don’t hurt me no more)



Threat Report ATT&CK Mapper (TRAM) is an open-source platform designed to reduce cost and increase the effectiveness of integrating ATT&CK across the CTI community.

It does this by automating the mapping of cyber threat intelligence (CTI) reports to MITRE ATT&CK®

TRAM Stock limits:

- Trained on 50 techniques
- Is not considered production-ready
- ?? Let’s find out!

https://github.com/center-for-threat-informed-defense/tram/

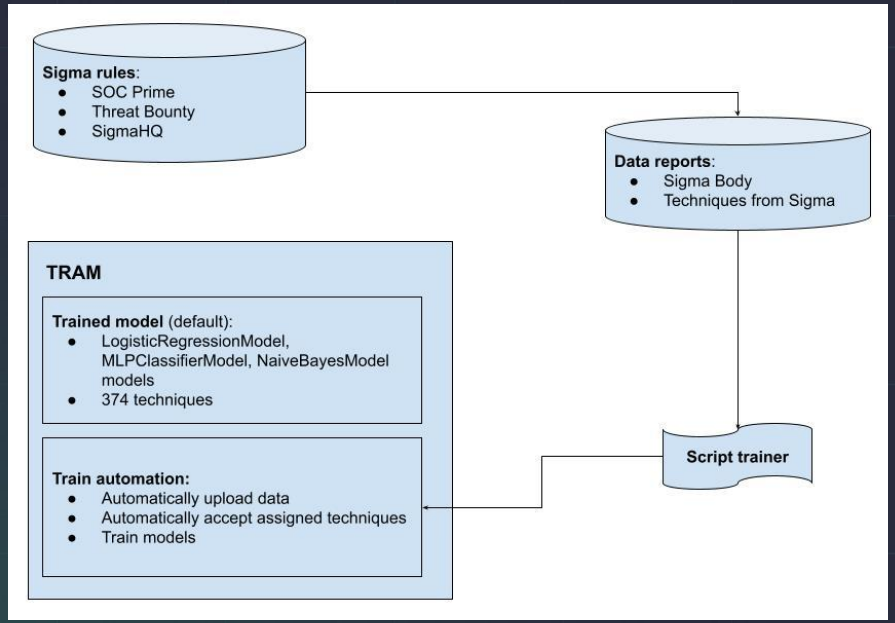
Practical use cases for TRAM

1. Automated analysis and mapping of CTI reports to ATT&CK | STOCK, main goal

2. Improve mapping of Sigma rules to ATT&CK | SOC Prime

3. Suggest ATT&CK tags based on detection rules code in any language | SOC Prime / Roota project

TRAM Stage 1



Inside the TRAM-10K engine, budget $1 per 1 Sigma rule

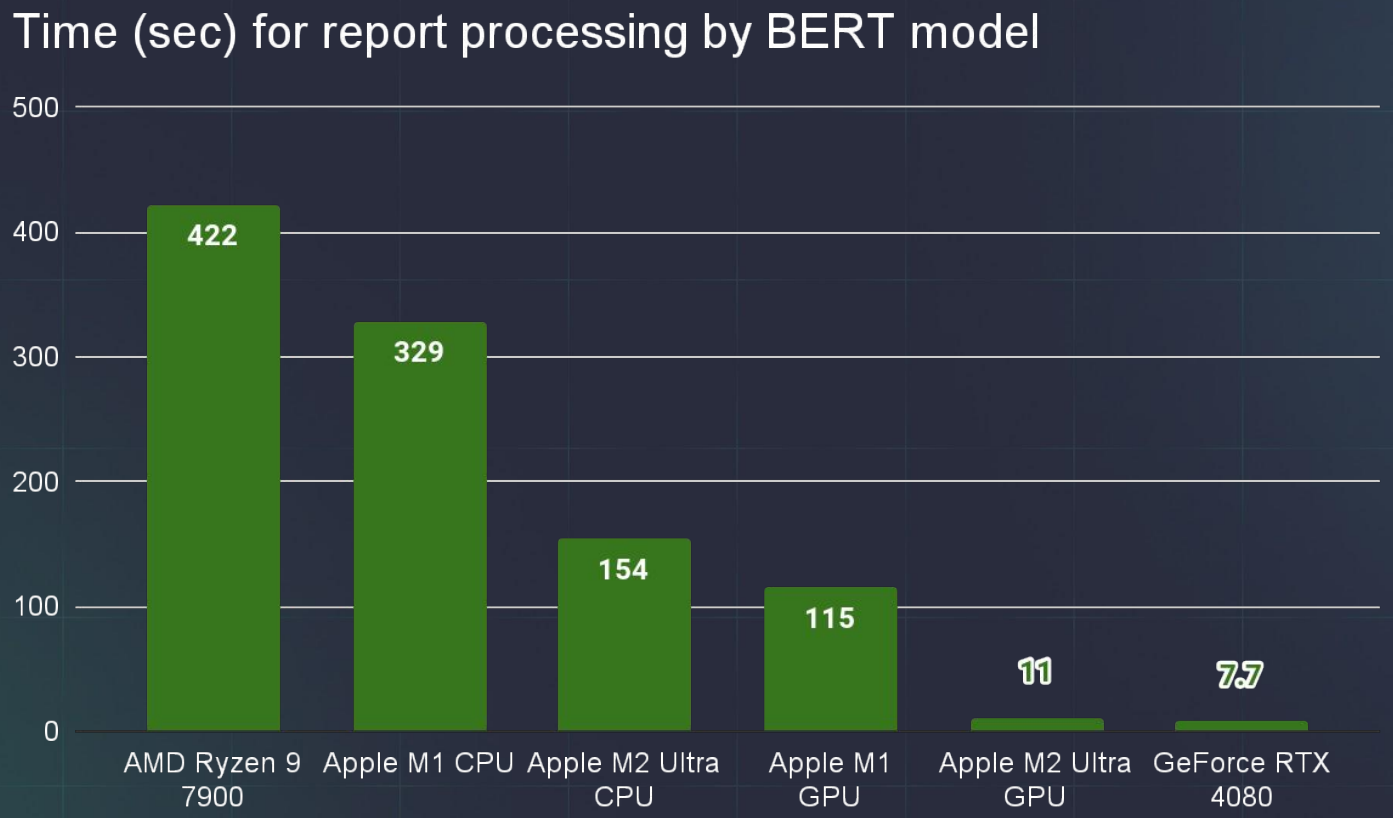
Mac specs:

- Apple M2 Ultra
- 24 CPU cores & 70 GPU cores
- 128GB Unified RAM
- 2TB SSD

PC specs:

- AMD Ryzen 9 7900 12-core
- nVidia RTX 4080/ 9,728 core, 16GB
- 64GB DDR5 RAM EXPO 6000
- Asus ProArt B650 / X670
- 2TB SSD MP600 PRO XT

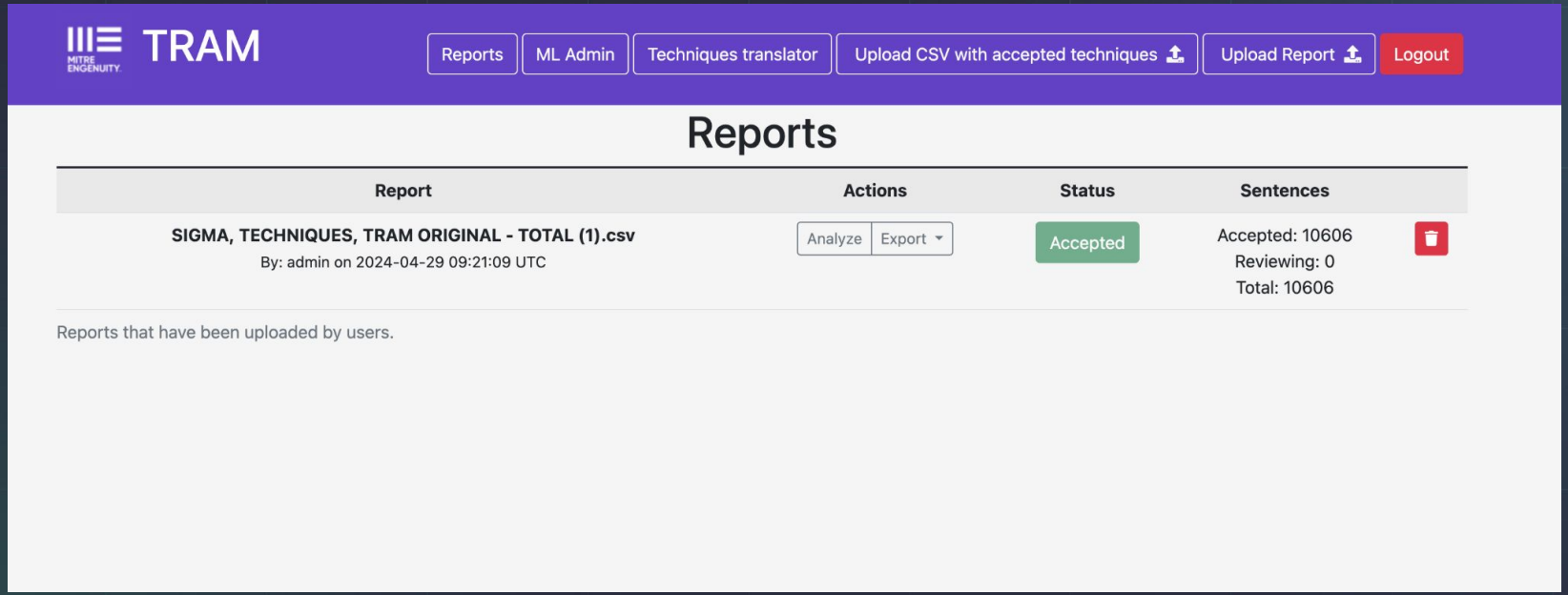
Benchmarks, single report



Report analyzed: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-347a

TRAM Stage 1 tune

Buttons to train it with the SOC Prime ruleset based off Sigma code + reference URL (CTI):

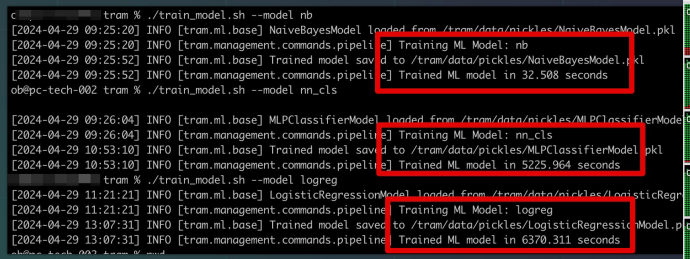


Benchmarks, training time of different TRAM ML models

Dataset of Rules: 1. 10k Sigma Rules from SOC Prime Threat Detection Marketplace
2. 10k Sigma + 30k queries: added queries from 3 top SIEM translated from Sigma rules

Hardware: MacBook Pro, Apple M1 Pro (10 cores), 32Gb RAM

Model	10k Sigma Rules, Docker	10k Sigma Rules, OS deployment	10k Sigma + 30k queries, Docker	10k Sigma + 30k queries, OS deployment
NaiveBayes	32s	1s	1m 54s	3s / 2s @ M2
MLPClassifier	1h 27m	6m 42s	5h 51m	1h 13m / 1h 25m @ M2
LogisticRegression	1h 47m	1h 12m	6h 54m	2h 26m / 184s* @ M2



Accuracy Example

https://tdm.socprime.com/tdm/info/Qi4z2rDZGU8t/socprime/#sigma

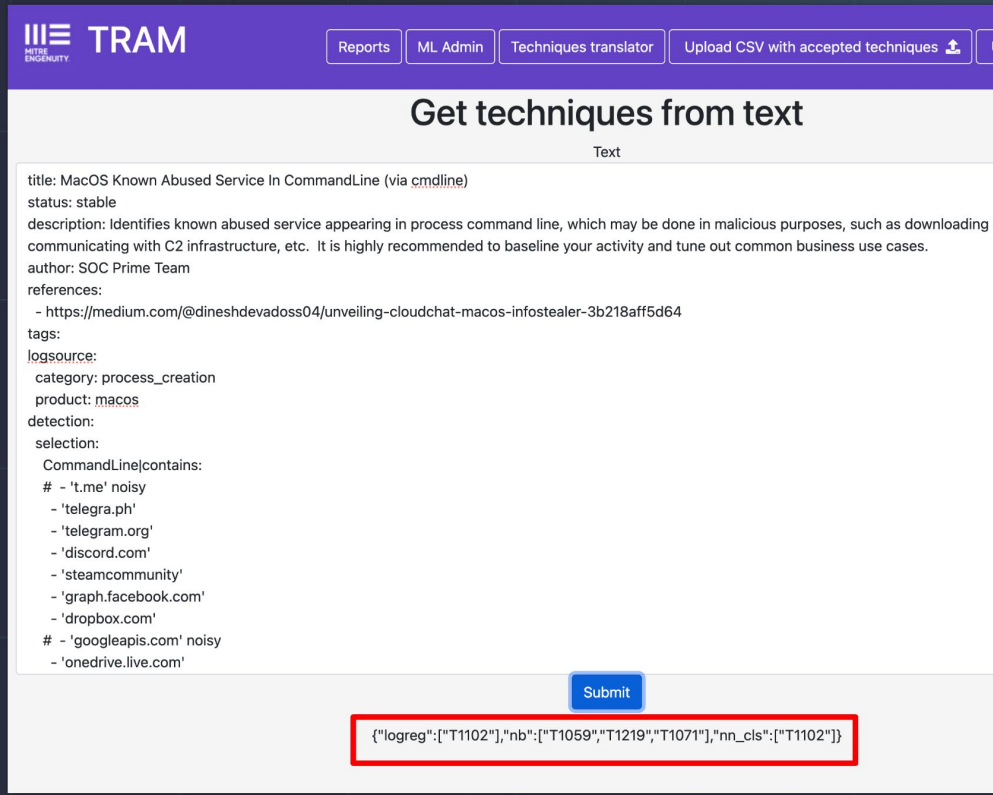
Rule author tags this rule with **T1102 - Web Service**

Trained TRAM suggested next techniques by different data models:

- NaiveBayesModel - **T1059**, **T1219**, **T1071**
- MLPClassifierModel - **T1102**
- LogisticRegressionModel - **T1102**

As we see suggested techniques:

- **T1102 - Web Service** - is 100% correct for the rule
- **T1059 - Command and Scripting Interpreter** - can be added to the rule
- **T1219 - Remote Access Software** - more likely false positive in this case
- **T1071 - Application Layer Protocol** - can be added to the rule



Accuracy Example

TRAM trained with 10,000 Sigma rules and 30,000 translations

https://tdm.socprime.com/tdm/info/T7dfD9zxxjEY/socprime/#sigma

```
title: Suspicious RDP Max Time Was Set (via
registry_event)

description: Identifies RDP maximum time being set to
suspicious value, which may be done by adversaries in
order to prevent users from connecting remotely to the
endpoint.
```

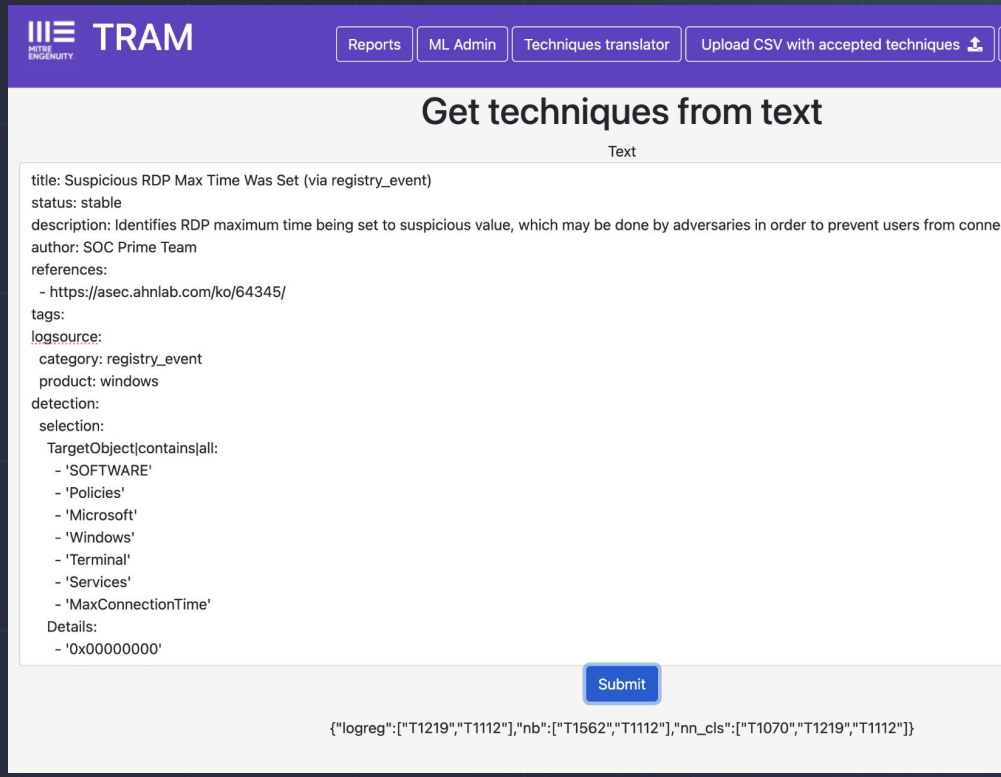
Rule author tags this rule with **T1112 - Modify Registry**

TRAM suggestions for Sigma:

logreg: T1112 - Modify Registry, T1219 - Remote Access Software

nb: T1562 - Exploit Public-Facing Application, T1112 - Modify Registry

nn_cls: T1070 - Indicator Removal, T1219 - Remote Access Software, T1112 - Modify Registry



Accuracy Example

TRAM trained with 10,000 Sigma rules and 30,000 translations

https://tdm.socprime.com/tdm/info/T7dfD9zxxjEY/socprime/#sigma

```
title: Suspicious RDP Max Time Was Set (via registry_event)

description: Identifies RDP maximum time being set to
suspicious value, which may be done by adversaries in order
to prevent users from connecting remotely to the endpoint.
```

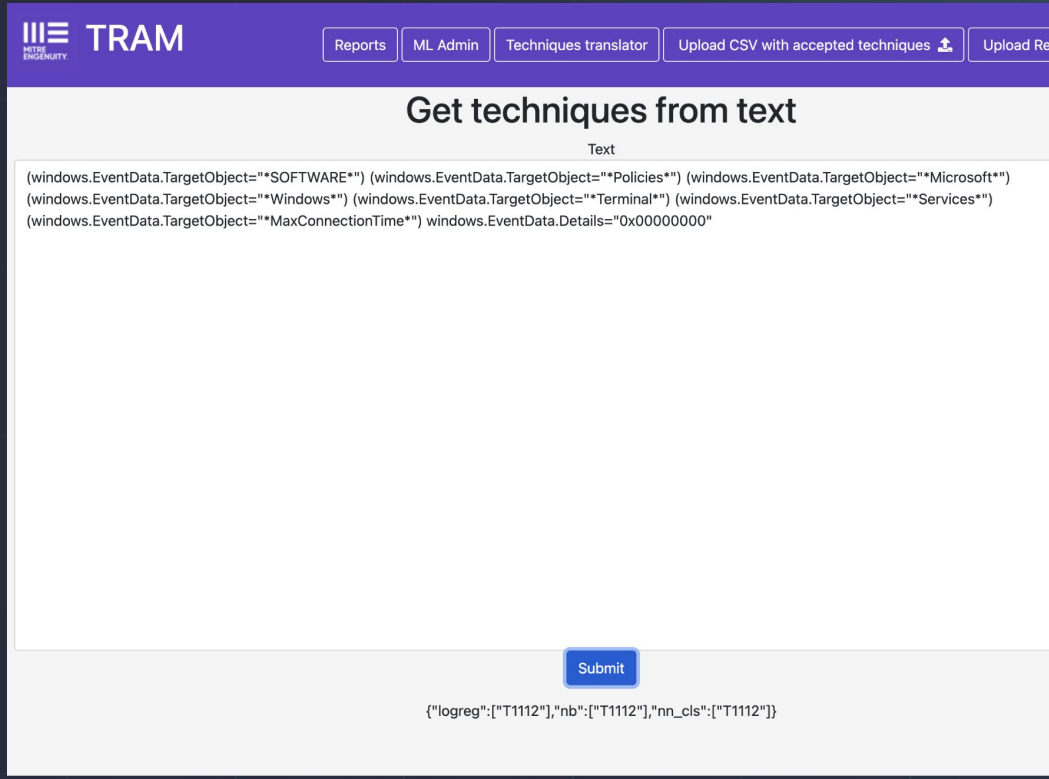
Rule author tags this rule with **T1112 - Modify Registry**

TRAM suggestions for translate queries:

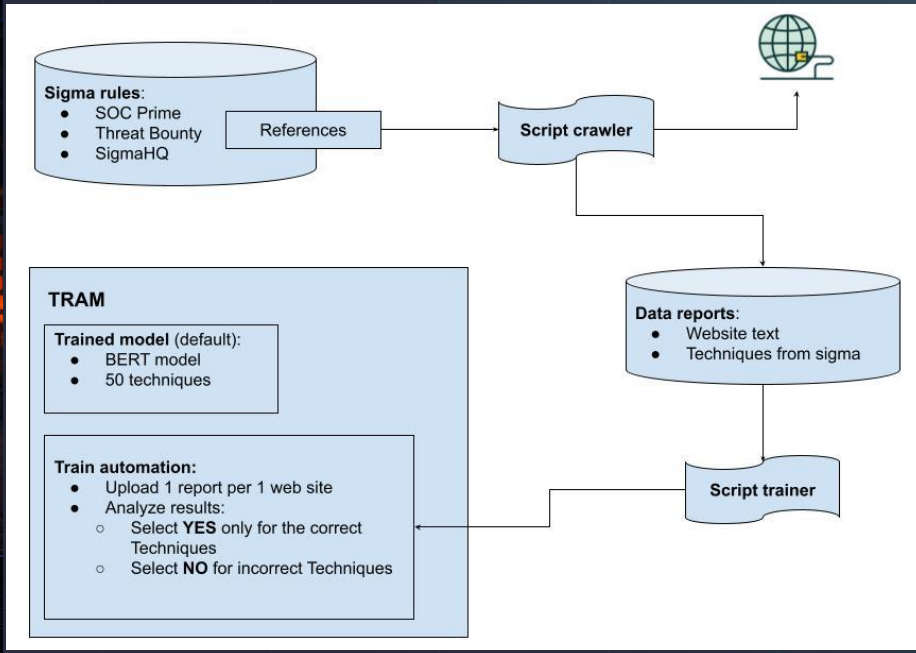
logreg: T1112 - Modify Registry

nb: T1112 - Modify Registry

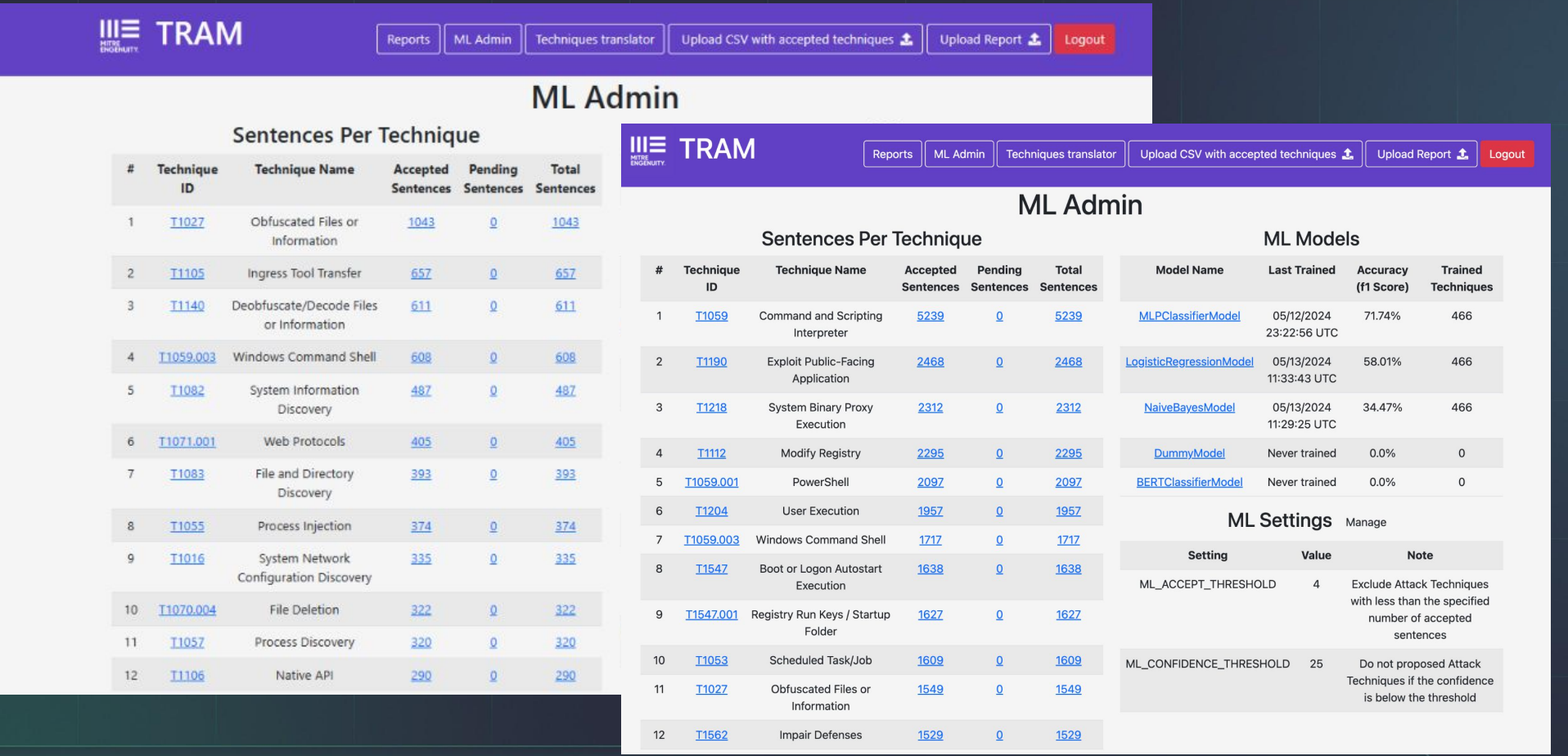
nn_cls: T1112 - Modify Registry



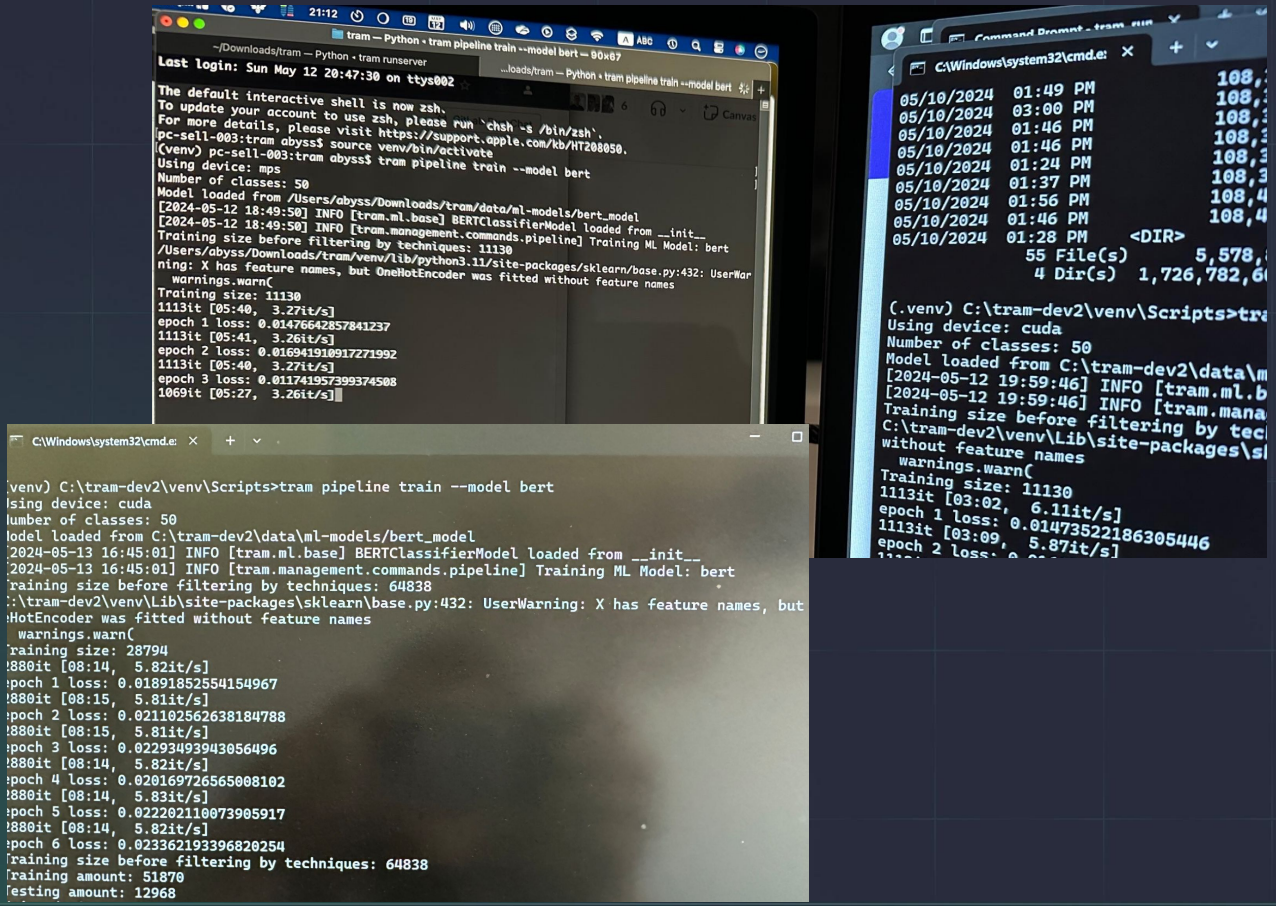
TRAM Stage 2



TRAM Stage 2: 5x larger training dataset



M2 Ultra GPU left vs RTX4080 right (R9 7900 8x pcie)



BERT inference training with stock+10k sigma +30k rules

64,838 training size

Accepting only 50 techniques

1h 33m @ M2 Ultra ‘mps’

~~21m @ RTX 4080 OC crashed~~

51m @ RTX 4080 ‘cuda’ stock

177% faster with RTX

We’re figuring out how to train it beyond 50 techniques without sacrificing accuracy

Roota spec update for TRAM tag exchange

```
name: Possible Credential Dumping Using Comsvcs.dll (via cmdline)
details: Adversaries can use built-in library comsvcs.dll to dump credentials on a compromised host.
author: SOC Prime Team
type: query
class: behaviour
mitre-attack:
    - t1003.001
    - t1136.003
tram-tags:
    NaiveBayes:
        - t1136.003
    MLPClassifier:
        - t1003.001
    LogisticRegression:
        - t1003
        - t1003.005
detection:
    language: splunk-spl-query # elastic-lucene-query, logscale-lql-query, mde-kql-query
    body: index=* ((((process="*comsvcs*") AND (process="*MiniDump*")) OR ((process="*comsvcs*") AND
(process="*#24*"))) OR ((process="*comsvcs*") AND (process="*full*")))
```

https://github.com/UncoderIO/RootA/blob/main/README.md#full-roota-rule-example

TRAM Stage 3:

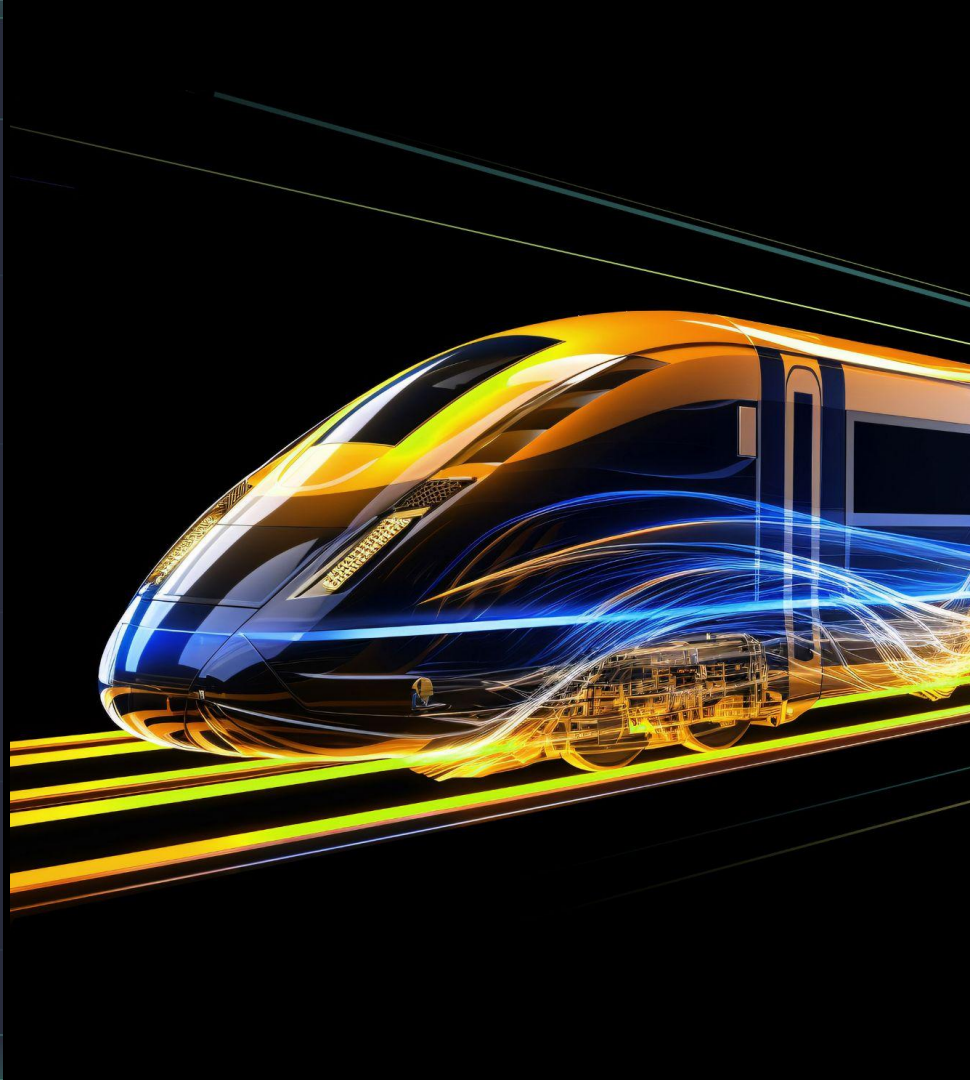
Going beyond 50 techniques

More Apple silicon for CPU models

GPU for BERT and other LLM

Liquid cooling

Integration with Uncoder



THANK YOU!

Useful links

https://github.com/center-for-threat-informed-defense/tram/wiki/Developers

https://uncoder.io and https://github.com/UncoderIO

https://roota.io

https://socprime.com