



# 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:

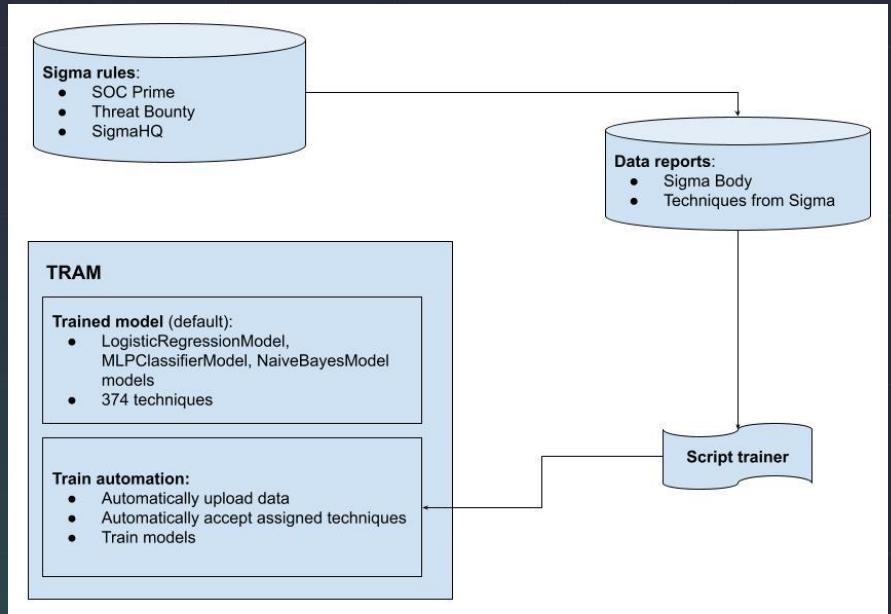
- Trained on 50 techniques
- Is not considered production-ready
- ?? Let's find out!

<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:

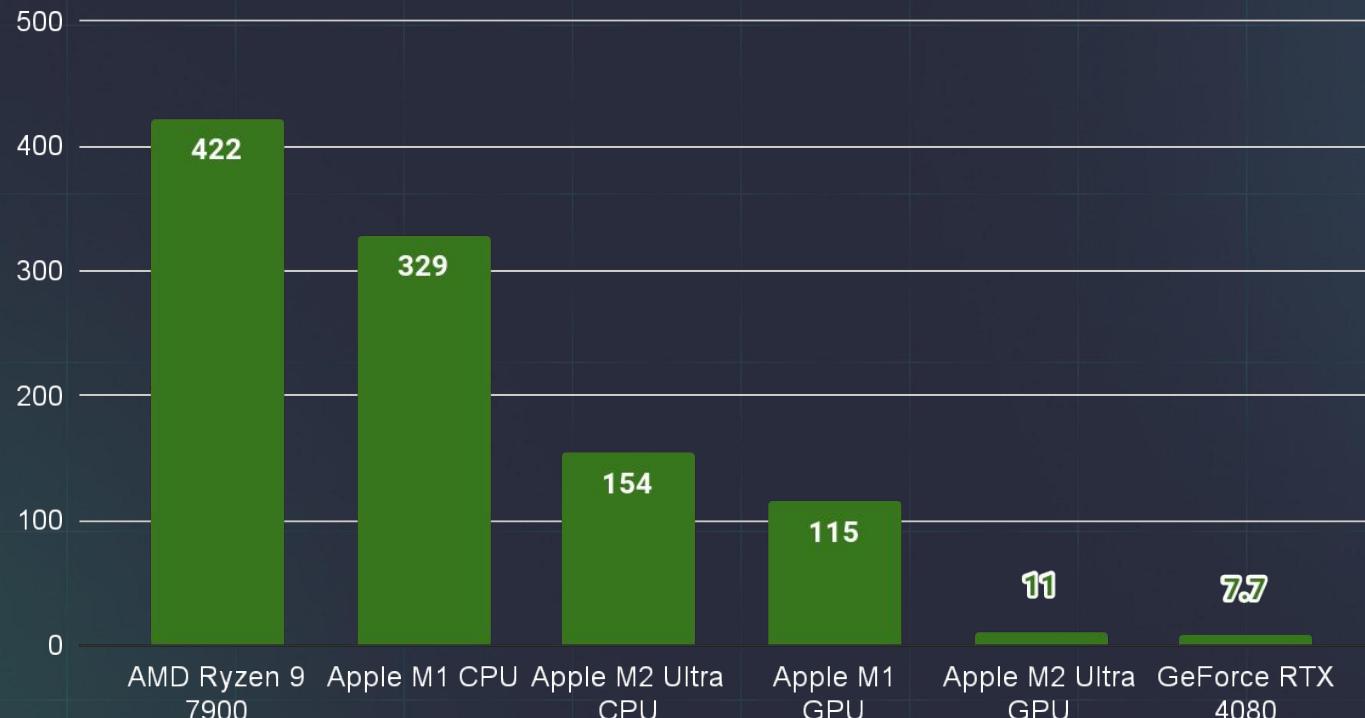
- Apple M2 Ultra
- 24 CPU cores & 70 GPU cores
- 128GB Unified RAM
- 2TB SSD

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

Time (sec) for report processing by BERT model



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):

The screenshot shows the TRAM application interface. At the top, there is a purple header bar with the MITRE ENGENIUTY logo and the word "TRAM". Below the header, there is a navigation bar with buttons for "Reports", "ML Admin", "Techniques translator", "Upload CSV with accepted techniques" (with an upload icon), "Upload Report" (with an upload icon), and "Logout". The main content area is titled "Reports". A table displays a single report entry:

Report	Actions	Status	Sentences
<b>SIGMA, TECHNIQUES, TRAM ORIGINAL - TOTAL (1).csv</b> By: admin on 2024-04-29 09:21:09 UTC	Analyze Export ▾	Accepted	Accepted: 10606 Reviewing: 0 Total: 10606

Below the table, a message states: "Reports that have been uploaded by users."

# Benchmarks, training time of different TRAM ML models

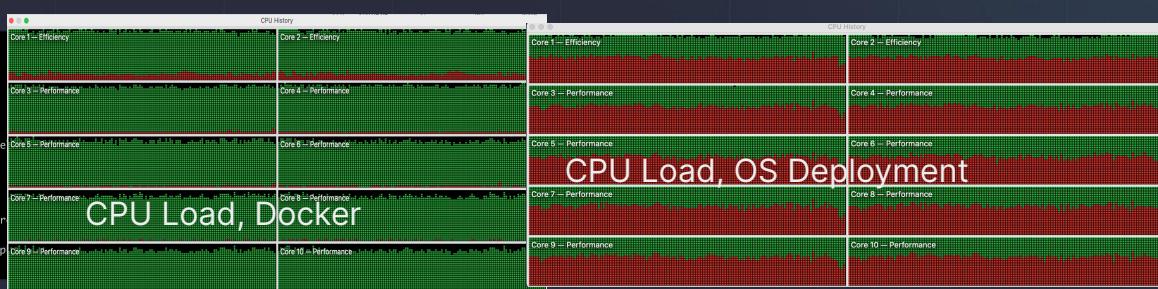
**Dataset of Rules:**

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

```
c:\> tram % ./train_model.sh --model nb
[2024-04-29 09:25:20] INFO [tram.ml.base] NaiveBayesModel loaded from /tram/data/pickles/NaiveBayesModel.pkl
[2024-04-29 09:25:20] INFO [tram.management.commands.pipeline] Training ML Model: nb
[2024-04-29 09:25:52] INFO [tram.ml.base] Trained model saved to /tram/data/pickles/NaiveBayesModel.pkl
[2024-04-29 09:25:52] INFO [tram.management.commands.pipeline] Trained ML model in 32.508 seconds
ob6pc-tech-002 tram % ./train_model.sh --model nn_cls
[2024-04-29 09:26:04] INFO [tram.ml.base] MLPClassifierModel loaded from /tram/data/pickles/MLPClassifierModel.pkl
[2024-04-29 09:26:04] INFO [tram.management.commands.pipeline] Training ML Model: nn_cls
[2024-04-29 09:26:04] INFO [tram.ml.base] Trained model saved to /tram/data/pickles/MLPClassifierModel.pkl
[2024-04-29 09:26:04] INFO [tram.management.commands.pipeline] Trained ML model in 525.964 seconds
[2024-04-29 11:21:21] INFO [tram.ml.base] LogisticRegressionModel loaded from /tram/data/pickles/LogisticRegressionModel.pkl
[2024-04-29 11:21:21] INFO [tram.management.commands.pipeline] Training ML Model: logreg
[2024-04-29 11:21:21] INFO [tram.ml.base] Trained model saved to /tram/data/pickles/LogisticRegressionModel.pkl
[2024-04-29 11:21:21] INFO [tram.management.commands.pipeline] Trained ML model in 6370.311 seconds
ob6pc-tech-002 tram %
```



# 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

The screenshot shows the TRAM interface with a purple header containing the logo and the word "TRAM". Below the header are four buttons: "Reports", "ML Admin", "Techniques translator", and "Upload CSV with accepted techniques". The main content area has a title "Get techniques from text" and a "Text" input field containing the following JSON data:

```
title: MacOS Known Abused Service In CommandLine (via cmdline)
status: stable
description: Identifies known abused service appearing in process command line, which may be done in malicious purposes, such as downloading communicating with C2 infrastructure, etc. It is highly recommended to baseline your activity and tune out common business use cases.
author: SOC Prime Team
references:
tags:
logsource:
category: process_creation
product: macos
detection:
selection:
CommandLine|contains:
# - 't.me' noisy
- 'telegra.ph'
- 'telegram.org'
- 'discord.com'
- 'steamcommunity'
- 'graph.facebook.com'
- 'dropbox.com'
# - 'googleapis.com' noisy
- 'onedrive.live.com'
```

At the bottom right of the content area is a blue "Submit" button. Below the content area, a red box highlights the JSON data in the "Text" input field:

```
{"logreg":["T1102"], "nb":["T1059","T1219","T1071"], "nn_cls":["T1102"]}
```

# 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

The screenshot shows the TRAM web application interface. At the top, there's a purple header bar with the TRAM logo (MITRE ENGENIERT) and navigation links for Reports, ML Admin, Techniques translator, and Upload CSV with accepted techniques. Below the header, a card displays a Sigma rule with the following details:

- title:** Suspicious RDP Max Time Was Set (via registry\_event)
- status:** stable
- 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.
- author:** SOC Prime Team
- references:**
  - <https://asec.ahlab.com/ko/64345/>
- tags:**
- logsource:**
- category:** registry\_event
- product:** windows
- detection:**
- selection:**
  - TargetObject[containsAll]:
    - 'SOFTWARE'
    - 'Policies'
    - 'Microsoft'
    - 'Windows'
    - 'Terminal'
    - 'Services'
    - 'MaxConnectionTime'
- Details:**
  - '0x00000000'

At the bottom right of the card is a blue "Submit" button. Below the card, a search bar contains the JSON object: {"logreg": ["T1219", "T1112"], "nb": ["T1562", "T1112"], "nn\_cls": ["T1070", "T1219", "T1112"]}

# 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:**

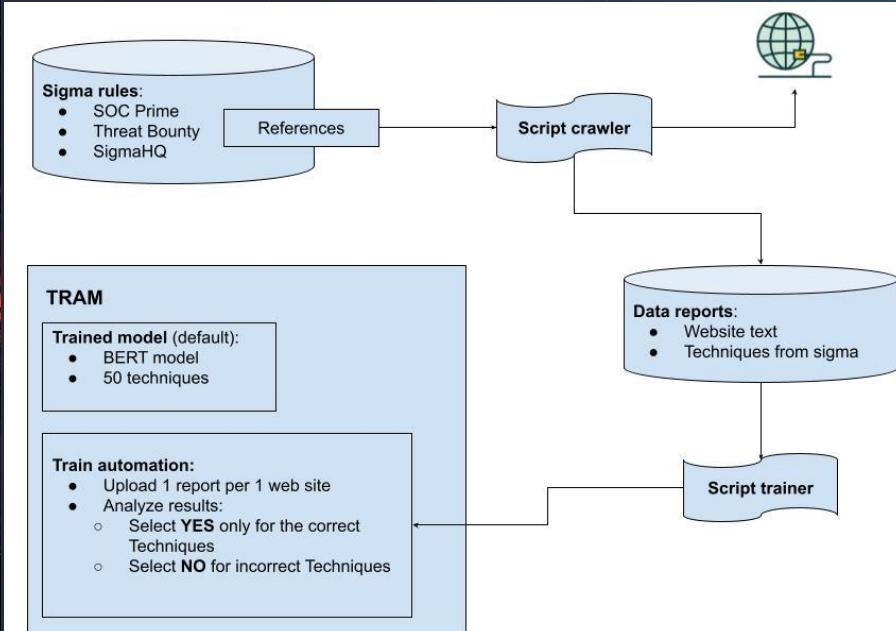
logreg: T1112 - Modify Registry

nb: T1112 - Modify Registry

nn\_cls: T1112 - Modify Registry

The screenshot shows the TRAM interface with a purple header. In the top right, there are buttons for 'Reports', 'ML Admin', 'Techniques translator', and two 'Upload CSV' buttons. Below the header, the text 'Get techniques from text' is displayed, followed by a 'Text' button. A code snippet is shown in a text area: (windows.EventData.TargetObject="\*SOFTWARE\*") (windows.EventData.TargetObject="\*Policies\*") (windows.EventData.TargetObject="\*Microsoft\*") (windows.EventData.TargetObject="\*Windows\*") (windows.EventData.TargetObject="\*Terminal\*") (windows.EventData.TargetObject="\*Services\*") (windows.EventData.TargetObject="\*MaxConnectionTime\*") windows.EventData.Details="0x00000000". At the bottom right of the interface is a blue 'Submit' button, and below it is the JSON output: {"logreg":["T1112"], "nb":["T1112"], "nn\_cls":["T1112"]}

# TRAM Stage 2



# TRAM Stage 2: 5x larger training dataset



[Reports](#) [ML Admin](#) [Techniques translator](#) [Upload CSV with accepted techniques](#) [Upload Report](#) [Logout](#)

## ML Admin

### Sentences Per Technique

#	Technique ID	Technique Name	Accepted Sentences	Pending Sentences	Total Sentences
1	<a href="#">T1027</a>	Obfuscated Files or Information	<a href="#">1043</a>	<a href="#">0</a>	<a href="#">1043</a>
2	<a href="#">T1105</a>	Ingress Tool Transfer	<a href="#">657</a>	<a href="#">0</a>	<a href="#">657</a>
3	<a href="#">T1140</a>	Deobfuscate/Decode Files or Information	<a href="#">611</a>	<a href="#">0</a>	<a href="#">611</a>
4	<a href="#">T1059.003</a>	Windows Command Shell	<a href="#">608</a>	<a href="#">0</a>	<a href="#">608</a>
5	<a href="#">T1082</a>	System Information Discovery	<a href="#">487</a>	<a href="#">0</a>	<a href="#">487</a>
6	<a href="#">T1071.001</a>	Web Protocols	<a href="#">405</a>	<a href="#">0</a>	<a href="#">405</a>
7	<a href="#">T1083</a>	File and Directory Discovery	<a href="#">393</a>	<a href="#">0</a>	<a href="#">393</a>
8	<a href="#">T1055</a>	Process Injection	<a href="#">374</a>	<a href="#">0</a>	<a href="#">374</a>
9	<a href="#">T1016</a>	System Network Configuration Discovery	<a href="#">335</a>	<a href="#">0</a>	<a href="#">335</a>
10	<a href="#">T1070.004</a>	File Deletion	<a href="#">322</a>	<a href="#">0</a>	<a href="#">322</a>
11	<a href="#">T1057</a>	Process Discovery	<a href="#">320</a>	<a href="#">0</a>	<a href="#">320</a>
12	<a href="#">T1106</a>	Native API	<a href="#">290</a>	<a href="#">0</a>	<a href="#">290</a>



[Reports](#) [ML Admin](#) [Techniques translator](#) [Upload CSV with accepted techniques](#) [Upload Report](#) [Logout](#)

## ML Admin

### Sentences Per Technique

#	Technique ID	Technique Name	Accepted Sentences	Pending Sentences	Total Sentences
1	<a href="#">T1059</a>	Command and Scripting Interpreter	<a href="#">5239</a>	<a href="#">0</a>	<a href="#">5239</a>
2	<a href="#">T1190</a>	Exploit Public-Facing Application	<a href="#">2468</a>	<a href="#">0</a>	<a href="#">2468</a>
3	<a href="#">T1218</a>	System Binary Proxy Execution	<a href="#">2312</a>	<a href="#">0</a>	<a href="#">2312</a>
4	<a href="#">T1112</a>	Modify Registry	<a href="#">2295</a>	<a href="#">0</a>	<a href="#">2295</a>
5	<a href="#">T1059.001</a>	PowerShell	<a href="#">2097</a>	<a href="#">0</a>	<a href="#">2097</a>
6	<a href="#">T1204</a>	User Execution	<a href="#">1957</a>	<a href="#">0</a>	<a href="#">1957</a>
7	<a href="#">T1059.003</a>	Windows Command Shell	<a href="#">1717</a>	<a href="#">0</a>	<a href="#">1717</a>
8	<a href="#">T1547</a>	Boot or Logon Autostart Execution	<a href="#">1638</a>	<a href="#">0</a>	<a href="#">1638</a>
9	<a href="#">T1547.001</a>	Registry Run Keys / Startup Folder	<a href="#">1627</a>	<a href="#">0</a>	<a href="#">1627</a>
10	<a href="#">T1053</a>	Scheduled Task/Job	<a href="#">1609</a>	<a href="#">0</a>	<a href="#">1609</a>
11	<a href="#">T1027</a>	Obfuscated Files or Information	<a href="#">1549</a>	<a href="#">0</a>	<a href="#">1549</a>
12	<a href="#">T1562</a>	Impair Defenses	<a href="#">1529</a>	<a href="#">0</a>	<a href="#">1529</a>

### ML Models

Model Name	Last Trained	Accuracy (f1 Score)	Trained Techniques
<a href="#">MLPClassifierModel</a>	05/12/2024 23:22:56 UTC	71.74%	466
<a href="#">LogisticRegressionModel</a>	05/13/2024 11:33:43 UTC	58.01%	466
<a href="#">NaiveBayesModel</a>	05/13/2024 11:29:25 UTC	34.47%	466
<a href="#">DummyModel</a>	Never trained	0.0%	0
<a href="#">BERTClassifierModel</a>	Never trained	0.0%	0

### ML Settings

Manage

Setting	Value	Note
ML_ACCEPT_THRESHOLD	4	Exclude Attack Techniques with less than the specified number of accepted sentences
ML_CONFIDENCE_THRESHOLD	25	Do not proposed Attack Techniques if the confidence is below the threshold

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

```

tram - Python - tram pipeline train --model bert - 50it
Last login: Sun May 12 20:47:30 on ttys002
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208850.
pc-sell-003:tram abys$ source venv/bin/activate
(venv) pc-sell-003:tram abys$ tram pipeline train --model bert
Using device: mps
Number of classes: 50
Model loaded from /Users/abys/Downloads/tram/data/ml-models/bert_model
[2024-05-12 18:49:50] INFO [tram.ml.base] BERTClassifierModel loaded from __init__
Training size before filtering by techniques: 11130
/Users/abys/Downloads/tram/venv/lib/python3.11/site-packages/sklearn/base.py:432: UserWarning:
    X has feature names, but OneHotEncoder was fitted without feature names
      warnings.warn(
Training size: 11130
1113it [05:40, 3.27it/s]
epoch 1 loss: 0.01476642857841237
1113it [05:41, 3.26it/s]
epoch 2 loss: 0.016941910917271992
1113it [05:40, 3.27it/s]
epoch 3 loss: 0.011741957399374508
1069it [05:27, 3.26it/s]

C:\Windows\system32\cmd: x + v
(venv) C:\tram-dev2\venv\Scripts>tram pipeline train --model bert
Using device: cuda
Number of classes: 50
Model loaded from C:\tram-dev2\data\ml-models/bert_model
[2024-05-13 16:45:01] INFO [tram.ml.base] BERTClassifierModel loaded from __init__
[2024-05-13 16:45:01] INFO [tram.management.commands.pipeline] Training ML Model: bert
Training size before filtering by techniques: 64838
::\tram-dev2\venv\Lib\site-packages\sklearn\base.py:432: UserWarning: X has feature names, but
OneHotEncoder was fitted without feature names
  warnings.warn(
Training size: 28794
1880it [08:14, 5.82it/s]
epoch 1 loss: 0.01891852554154967
1880it [08:15, 5.81it/s]
epoch 2 loss: 0.021102562638184788
1880it [08:15, 5.81it/s]
epoch 3 loss: 0.022934939430856496
1880it [08:14, 5.82it/s]
epoch 4 loss: 0.020169726565008102
1880it [08:14, 5.83it/s]
epoch 5 loss: 0.022202110073905917
1880it [08:14, 5.82it/s]
epoch 6 loss: 0.023362193396820254
Training size before filtering by techniques: 64838
Testing amount: 51870
Testing amount: 12968

```

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:

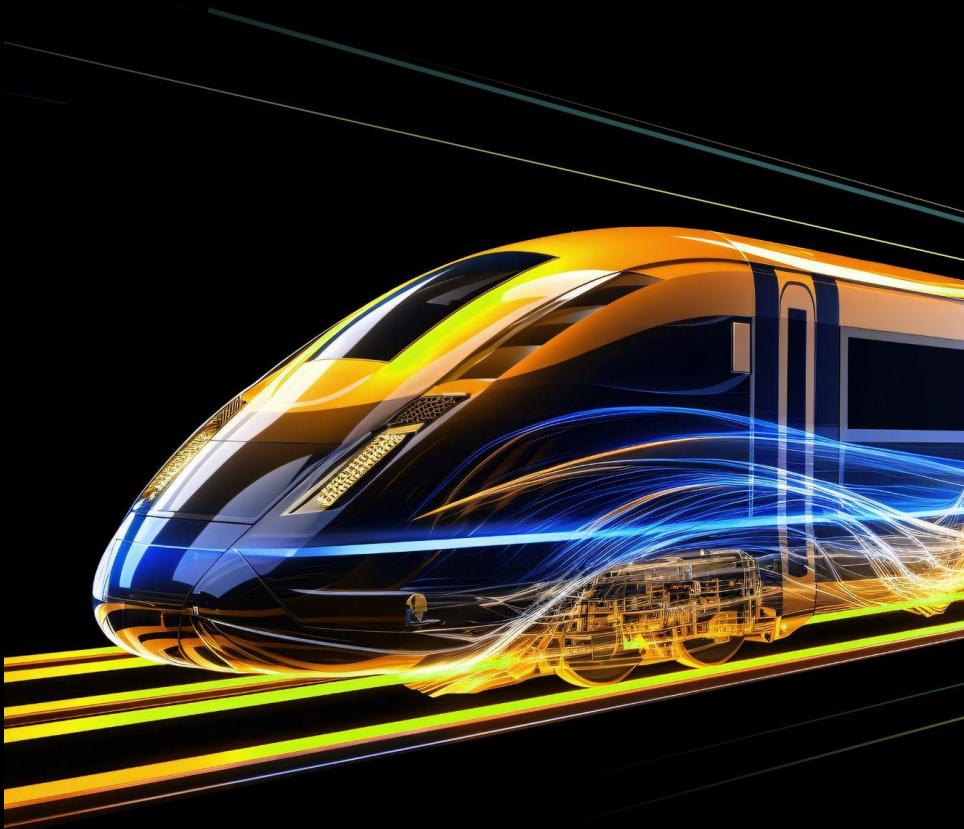
Going beyond 50 techniques

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>