A Practical Decision Framework for Implementing Evasion-Resilient Host-Based Analytics

Dr. Joe Mikhail

Brandon Werner

The MITRE Corporation

FloCon2020: January 2020

Overview

- **Research Questions**

  1. Can a framework be developed for non-data scientists to determine whether a given adversary technique is *best detected* with a heuristic analytic or a machine learning (ML) analytic?

     A. Where can I find good host-based ML data?

- **Definitions**

  - Heuristic Analytic: Analytic that uses rules, estimates or educated guesses to find a satisfactory solution to a specific issue.

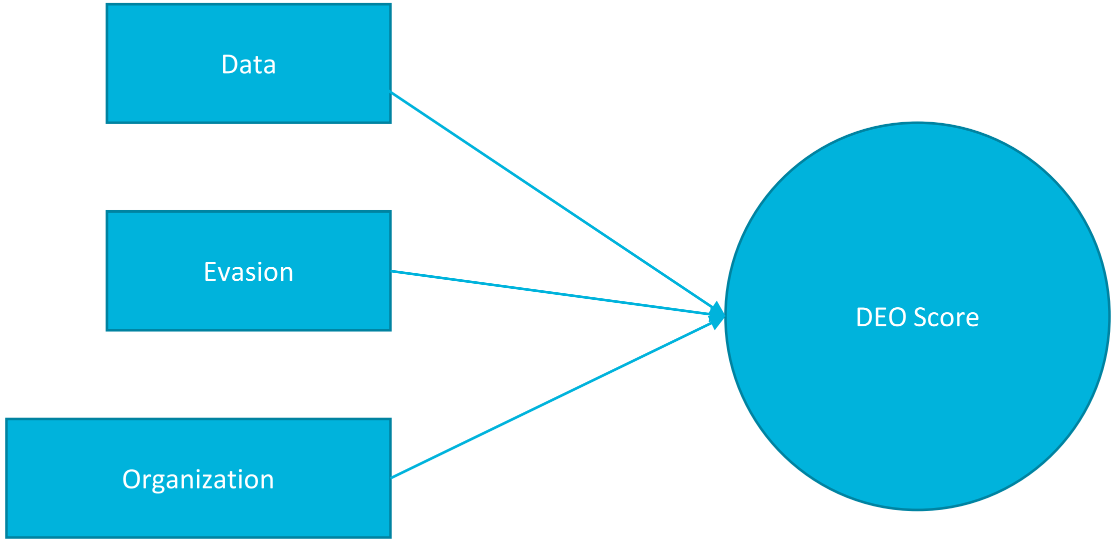
    - Not guaranteed to be optimal, perfect or rational, but sufficient for reaching an immediate, short-term goal

  - ML Analytic: ML analytics discover patterns in data, and construct mathematical models using these discoveries

    - Example: Neural network to detect malicious powershell

Data-Evasion-Organization (DEO) Framework

- **The proposed framework is comprised of a set of weighted criteria to evaluate data, evasion, and organizational factors in order to provide an analytic recommendation based on the DEO Score.**
  - Data: How well the data supports the analytic.
  - Evasion: How versatile the analytic needs to be.
  - Organization: How well the organization supports analytic development.
- **Weighting was assigned by applying framework to multiple use cases -> trial and error.**



Given categorical weights for data, evasion, and organization:
$W_D = 1, W_E = 1.5, W_O = 1,$

And scoring for each category:

$$S_{D,} S_E, S_O$$

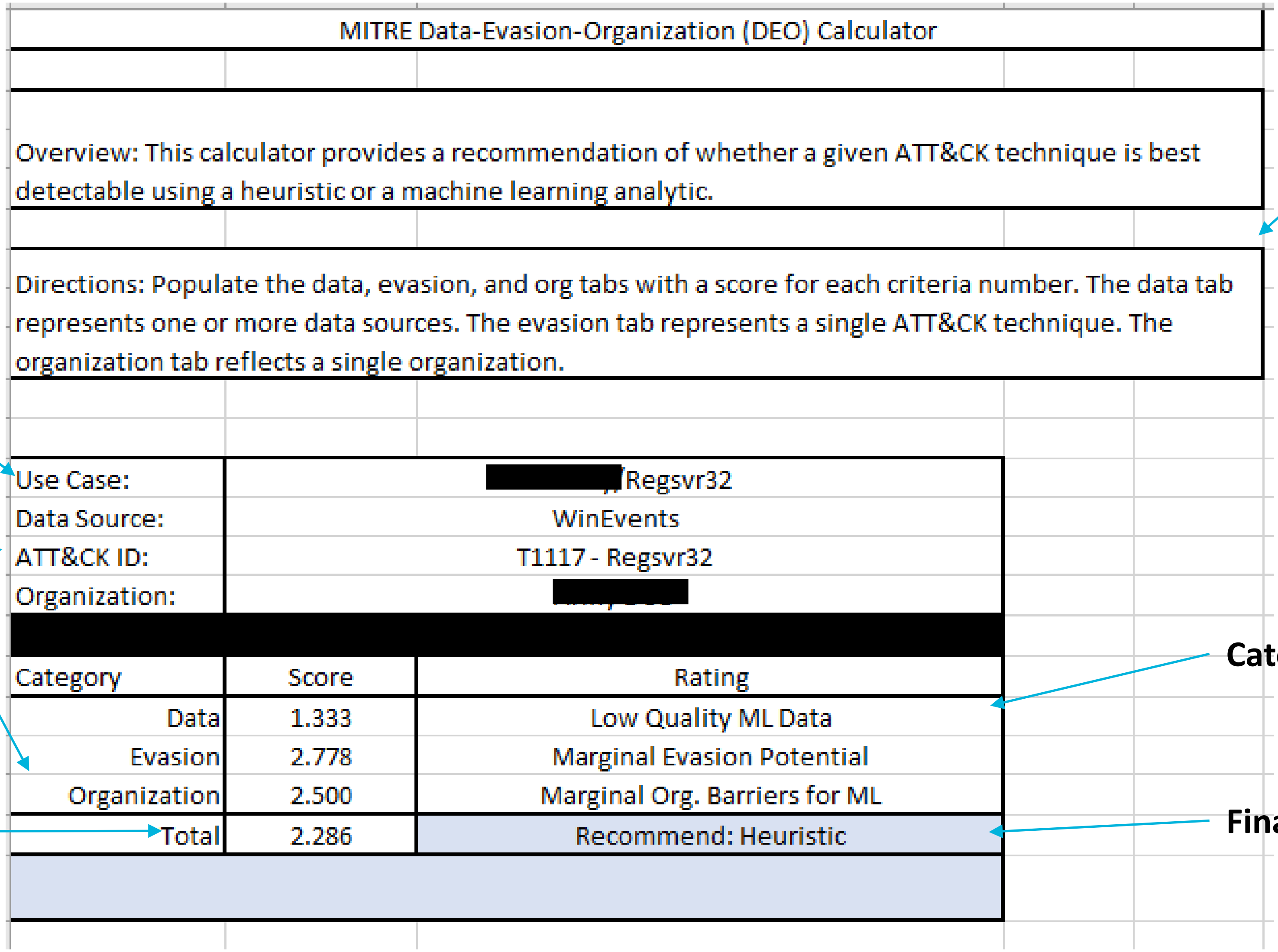
For the weighted total:

$$W_T = W_D + W_E + W_O$$

The final DEO score, $S_{DEO} = W_D S_D + W_E S_E + W_O S_O$

Output:
0<S_{DEO}<2.5: Heuristic
2.5<S_{DEO}<5: ML Model

Data-Evasion-Organization (DEO) Framework



MITRE Data-Evasion-Organization (DEO) Calculator

Overview: This calculator provides a recommendation of whether a given ATT&CK technique is best detectable using a heuristic or a machine learning analytic.

Directions: Populate the data, evasion, and org tabs with a score for each criteria number. The data tab represents one or more data sources. The evasion tab represents a single ATT&CK technique. The organization tab reflects a single organization.

Use Case:	[redacted]/Regsvr32
Data Source:	WinEvents
ATT&CK ID:	T1117 - Regsvr32
Organization:	[redacted]

[redacted]

Category	Score	Rating
Data	1.333	Low Quality ML Data
Evasion	2.778	Marginal Evasion Potential
Organization	2.500	Marginal Org. Barriers for ML
Total	2.286	Recommend: Heuristic

Directions/Overview of tool

Use-case name

Data, ATT&CK ID, Org

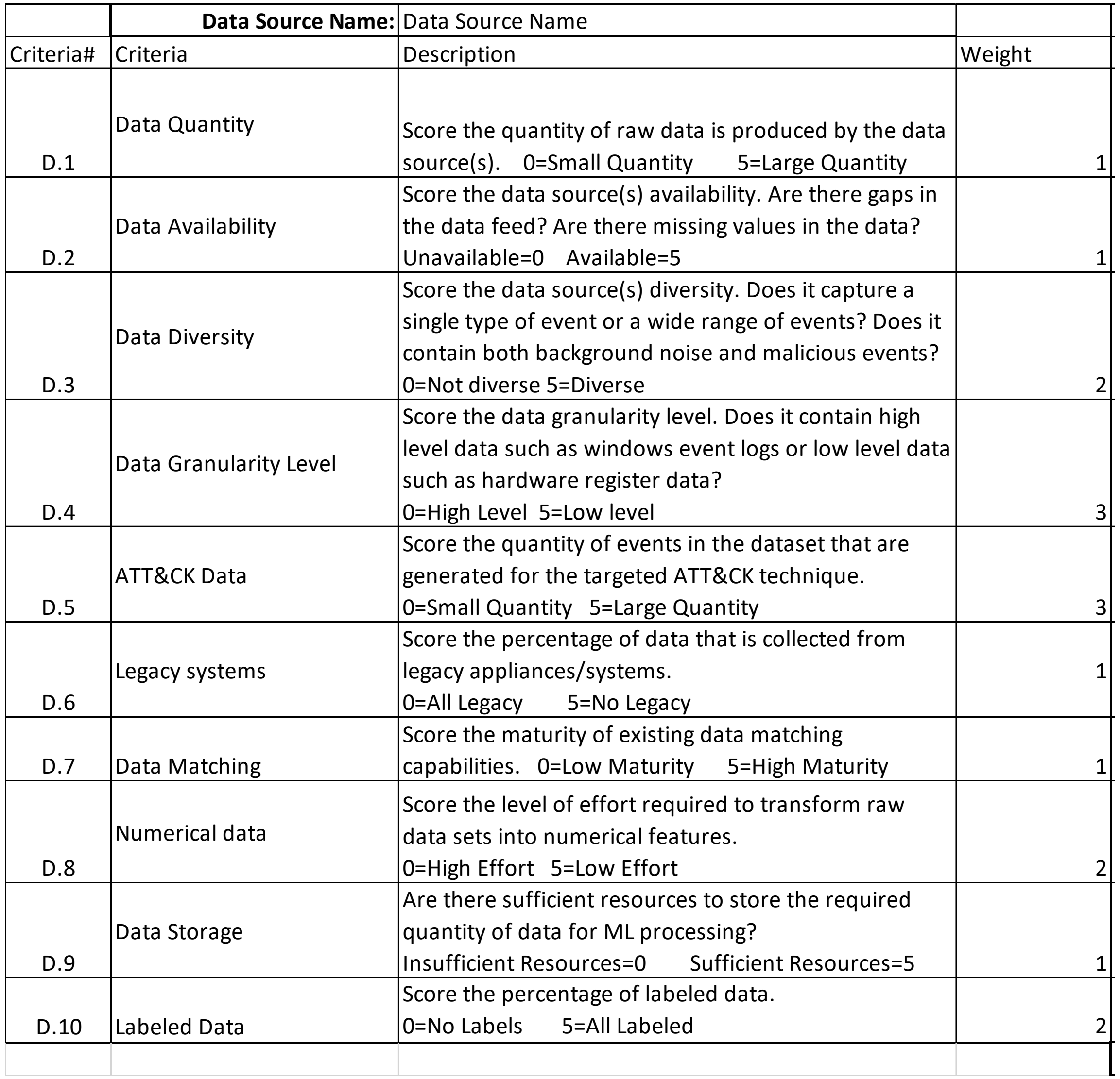
Category scoring (0-5)

Category "Ratings"

Final score S_F (0-5):
0<S_{DEO}<2.5: Heuristic
2.5<S_{DEO}<5: ML Model

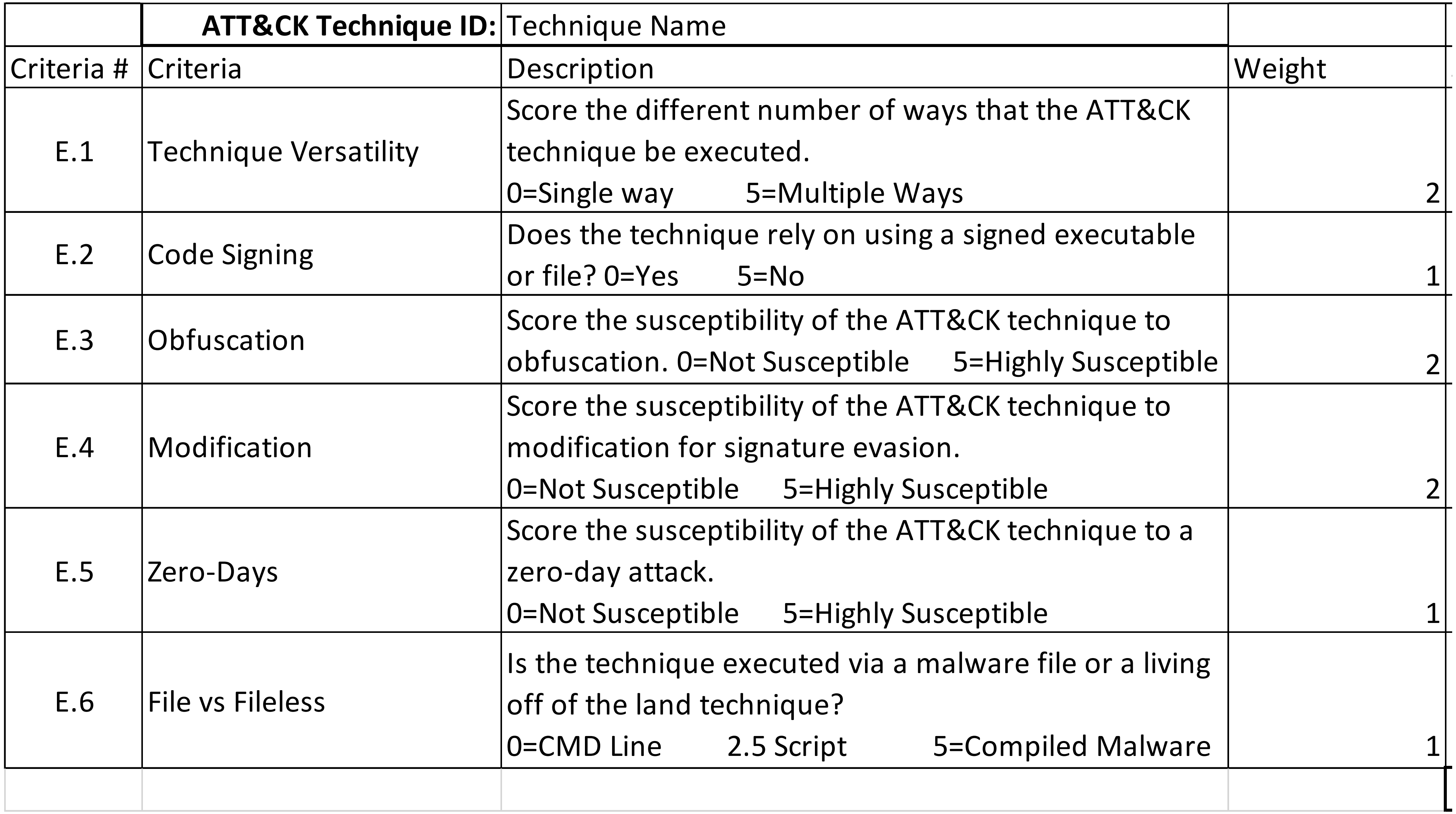
Final Recommendation

Data Scoring Factors



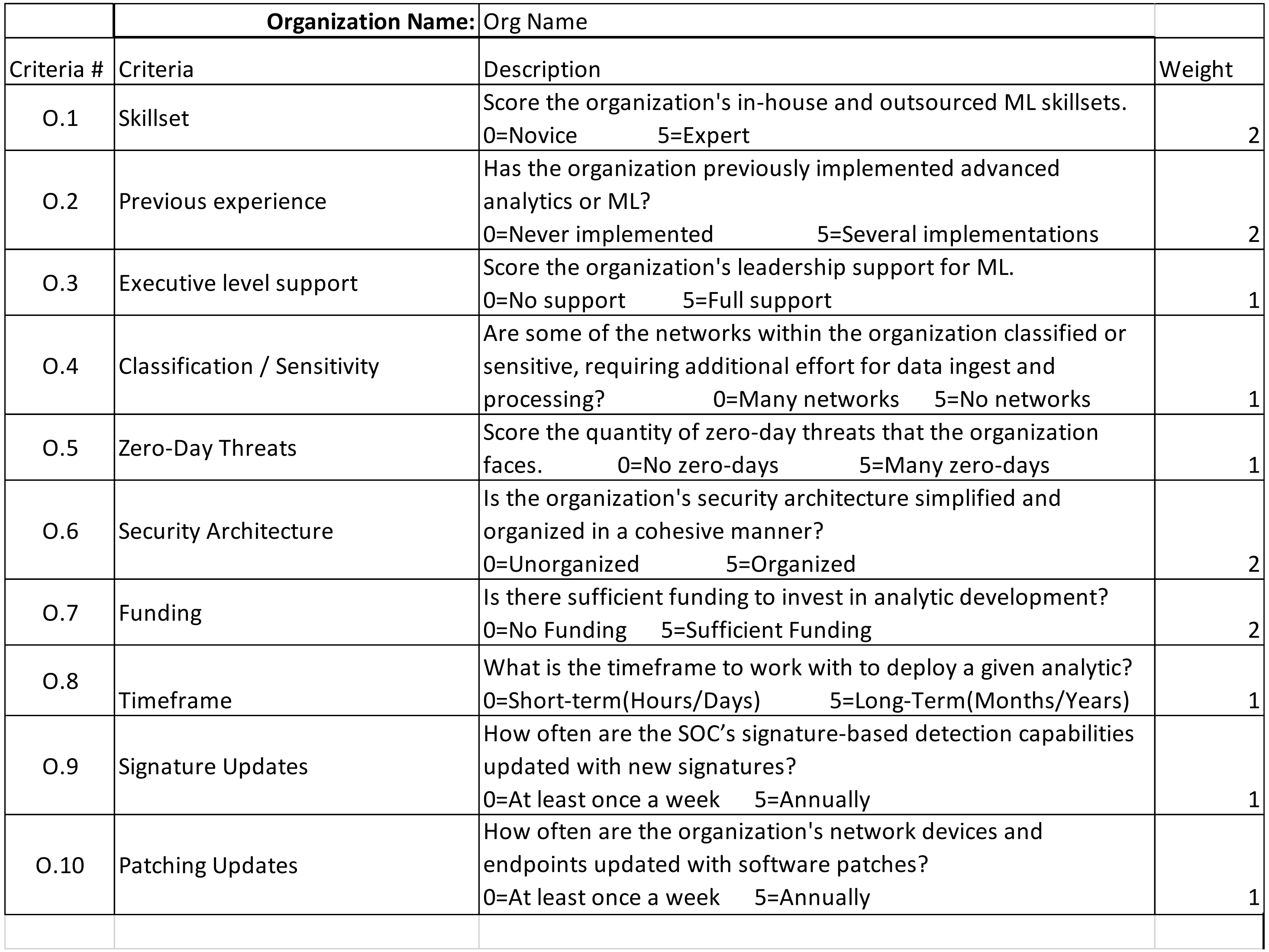
	Data Source Name:	Data Source Name	
Criteria#	Criteria	Description	Weight
D.1	Data Quantity	Score the quantity of raw data is produced by the data source(s). 0=Small Quantity 5=Large Quantity	1
D.2	Data Availability	Score the data source(s) availability. Are there gaps in the data feed? Are there missing values in the data? Unavailable=0 Available=5	1
D.3	Data Diversity	Score the data source(s) diversity. Does it capture a single type of event or a wide range of events? Does it contain both background noise and malicious events? 0=Not diverse 5=Diverse	2
D.4	Data Granularity Level	Score the data granularity level. Does it contain high level data such as windows event logs or low level data such as hardware register data? 0=High Level 5=Low level	3
D.5	ATT&CK Data	Score the quantity of events in the dataset that are generated for the targeted ATT&CK technique. 0=Small Quantity 5=Large Quantity	3
D.6	Legacy systems	Score the percentage of data that is collected from legacy appliances/systems. 0=All Legacy 5=No Legacy	1
D.7	Data Matching	Score the maturity of existing data matching capabilities. 0=Low Maturity 5=High Maturity	1
D.8	Numerical data	Score the level of effort required to transform raw data sets into numerical features. 0=High Effort 5=Low Effort	2
D.9	Data Storage	Are there sufficient resources to store the required quantity of data for ML processing? Insufficient Resources=0 Sufficient Resources=5	1
D.10	Labeled Data	Score the percentage of labeled data. 0=No Labels 5=All Labeled	2

Evasion Scoring Factors



	ATT&CK Technique ID:	Technique Name	
Criteria #	Criteria	Description	Weight
E.1	Technique Versatility	Score the different number of ways that the ATT&CK technique be executed. 0=Single way 5=Multiple Ways	2
E.2	Code Signing	Does the technique rely on using a signed executable or file? 0=Yes 5=No	1
E.3	Obfuscation	Score the susceptibility of the ATT&CK technique to obfuscation. 0=Not Susceptible 5=Highly Susceptible	2
E.4	Modification	Score the susceptibility of the ATT&CK technique to modification for signature evasion. 0=Not Susceptible 5=Highly Susceptible	2
E.5	Zero-Days	Score the susceptibility of the ATT&CK technique to a zero-day attack. 0=Not Susceptible 5=Highly Susceptible	1
E.6	File vs Fileless	Is the technique executed via a malware file or a living off of the land technique? 0=CMD Line 2.5 Script 5=Compiled Malware	1

Organization Scoring Factors

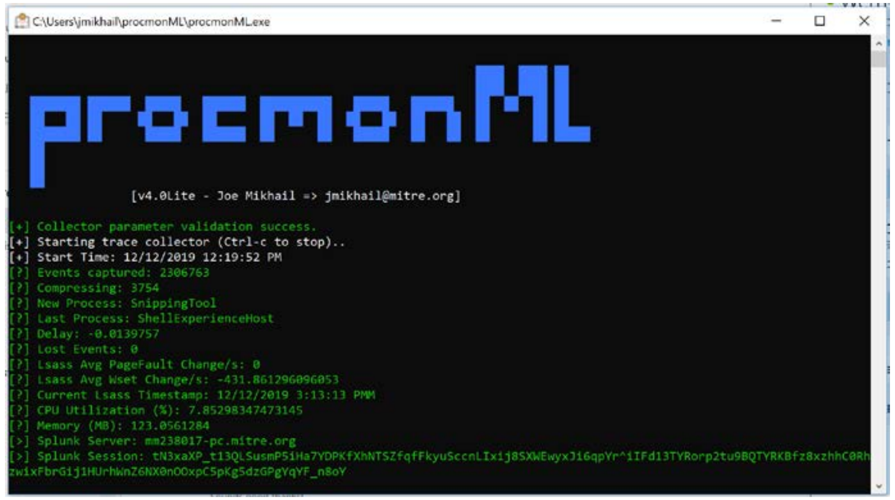


	Organization Name:	Org Name	
Criteria #	Criteria	Description	Weight
O.1	Skillset	Score the organization's in-house and outsourced ML skillsets. 0=Novice 5=Expert	2
O.2	Previous experience	Has the organization previously implemented advanced analytics or ML? 0=Never implemented 5=Several implementations	2
O.3	Executive level support	Score the organization's leadership support for ML. 0=No support 5=Full support	1
O.4	Classification / Sensitivity	Are some of the networks within the organization classified or sensitive, requiring additional effort for data ingest and processing? 0=Many networks 5=No networks	1
O.5	Zero-Day Threats	Score the quantity of zero-day threats that the organization faces. 0=No zero-days 5=Many zero-days	1
O.6	Security Architecture	Is the organization's security architecture simplified and organized in a cohesive manner? 0=Unorganized 5=Organized	2
O.7	Funding	Is there sufficient funding to invest in analytic development? 0=No Funding 5=Sufficient Funding	2
O.8	Timeframe	What is the timeframe to work with to deploy a given analytic? 0=Short-term(Hours/Days) 5=Long-Term(Months/Years)	1
O.9	Signature Updates	How often are the SOC’s signature-based detection capabilities updated with new signatures? 0=At least once a week 5=Annually	1
O.10	Patching Updates	How often are the organization's network devices and endpoints updated with software patches? 0=At least once a week 5=Annually	1

procmonML: The search for ML-friendly host-based data

- **procmonML is a [prototype] tool that generates & utilizes labeled host-based process data in a condensed ML-ready format to detect malicious host-based behavior.**
  - Objective 1: Limit data volume while retaining important information
  - Objective 2: Avoid need for computationally expensive ML models
  - Objective 3: Generate labeled data based on individual ATT&CK techniques
- **Components**
  - Host-based sensor (c# or powershell)
  - Machine Learning training/testing tool (scikit-learn).
    - Skope-Rules to generate Splunk analytics

  https://github.com/scikit-learn-contrib/skope-rules

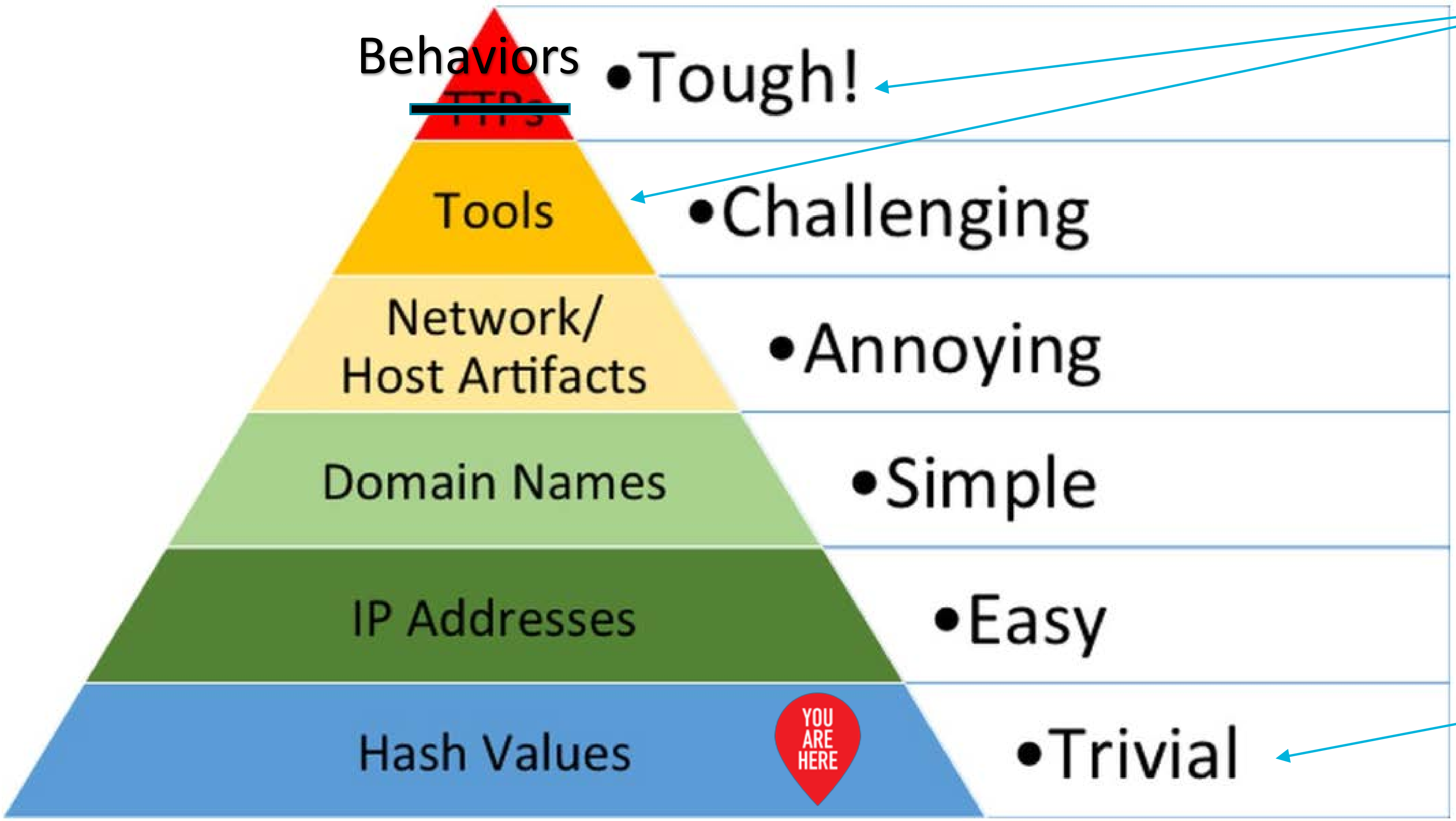


```
            [v4.0Lite - Joe Mikhail => jmikhail@mitre.org]

[+] Collector parameter validation success.
[+] Starting trace collector (Ctrl-c to stop)..
[+] Start Time: 12/12/2019 12:19:52 PM
[?] Events captured: 2306763
[?] Compressing: 3754
[?] New Process: SnippingTool
[?] Last Process: ShellExperienceHost
[?] Delay: -0.0139757
[?] Lost Events: 0
[?] Lsass Avg PageFault Change/s: 0
[?] Lsass Avg Wset Change/s: -431.861296096053
[?] Current Lsass Timestamp: 12/12/2019 3:13:13 PMM
[?] CPU Utilization (%): 7.85298347473145
[?] Memory (MB): 123.0561284
[>] Splunk Server: mm238017-pc.mitre.org
[>] Splunk Session: tN3xaXP_t13QLSusmP5iHa7YDPKfXhNTSZfqfFkyuSccnLIxij8SXWEwyxJ16qpYr^iIFd13TYRorp2tu9BQTYRKBfz8xzhhC0Rh
zwixFbrGij1HUrhWnZ6NX0nOOxpC5pKg5dzGPgYqYF_n8oY
```

Why ML for host-based detection?

1. Many heuristic analytics rely on string matching – Easily evaded.
2. ML analytics increase the adversary workload needed to evade analytics.

Pyramid of Pain: Heuristic vs. Behavioral Analytics



Behavioral analytics/
procmonML

An analytic is only as good as its weakest input field:

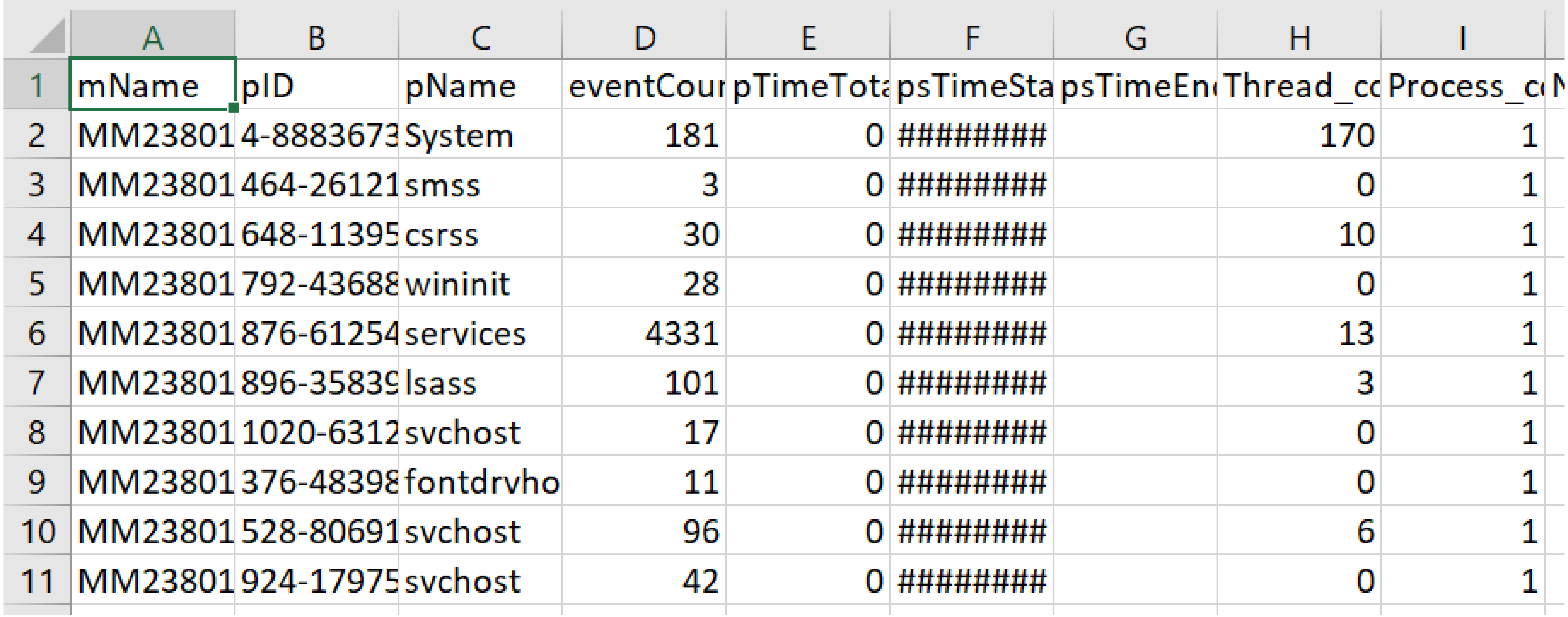
```
index=__your_sysmon_index__
EventCode=11
TargetFilename="*lsass*.dmp"
Image="C:\\Windows\\*\\taskmgr.exe"
```

Heuristics: Current State for many analytics

Heuristic: not guaranteed to be optimal, perfect or rational, but sufficient for reaching an immediate, short-term goal.

procmonML Data Organization

No PII!



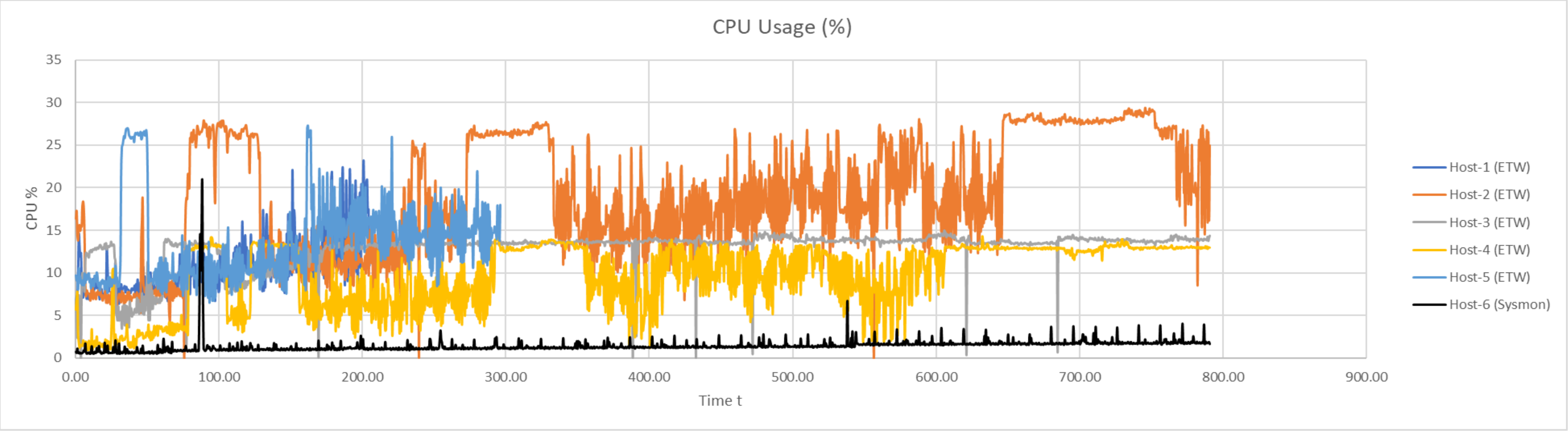
	A	B	C	D	E	F	G	H	I
1	mName	pID	pName	eventCour	pTimeTota	psTimeSta	psTimeEn	Thread_cc	Process_c
2	MM23801	4-8883673	System	181	0	#########		170	1
3	MM23801	464-26121	smss	3	0	#########		0	1
4	MM23801	648-11395	csrss	30	0	#########		10	1
5	MM23801	792-43688	wininit	28	0	#########		0	1
6	MM23801	876-61254	services	4331	0	#########		13	1
7	MM23801	896-35839	lsass	101	0	#########		3	1
8	MM23801	1020-6312	svchost	17	0	#########		0	1
9	MM23801	376-48398	fontdrvho	11	0	#########		0	1
10	MM23801	528-80691	svchost	96	0	#########		6	1
11	MM23801	924-17975	svchost	42	0	#########		0	1

The Big Tradeoff: Feature Processing vs. Event Consumption

procmonML Data Sources Investigated

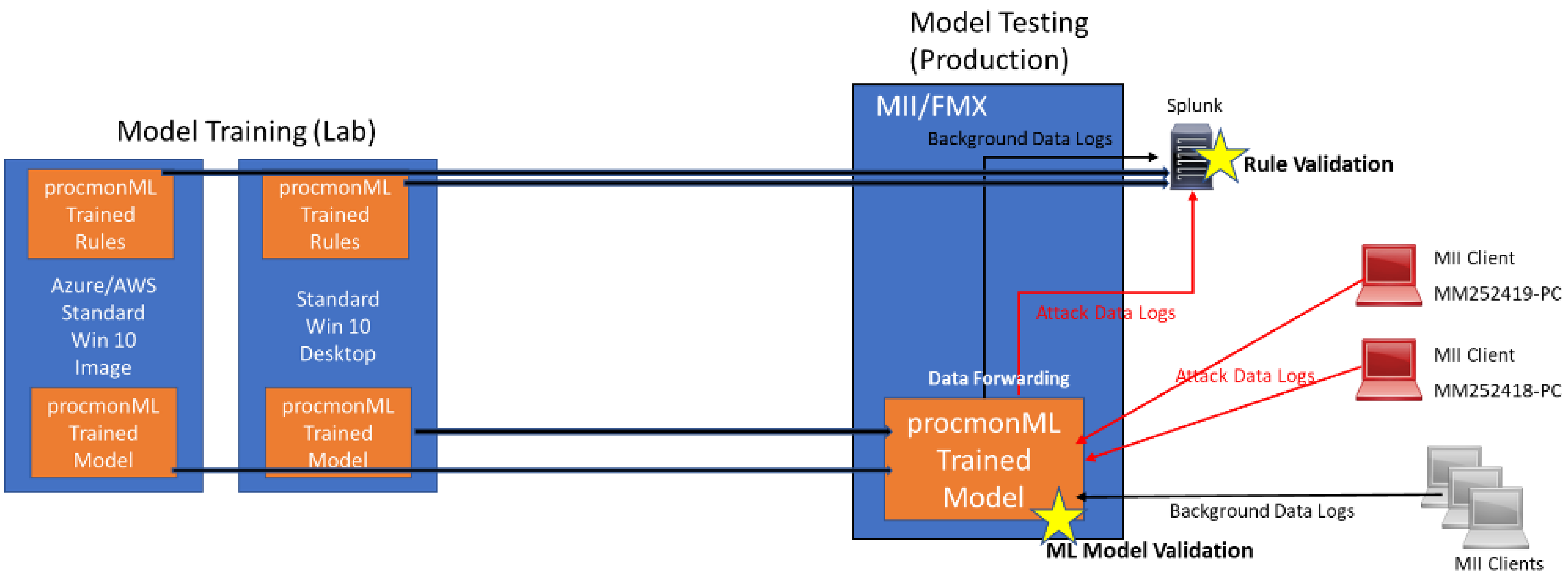
- **Windows ETW:**
  - Threads, Processes, Registry, Module Loads, Network
  - Timeseries data: Sequential events
  - Timeseries data: Module Load Sizes, Registry Depth

- **Sysmon:**
  - Event 1 (Process), Event 3 (Network), Event 5 (Process), Event 7 (Module Loads), Event 8 (Remote Thread), Event 9 (Raw Disk Access), Event10 (Lsass Access), Event 11 (File Created) - SwiftOnSec, Event 12-14 Registry – SwiftOnSec, Event 15 (FileCreateStream), Event 17/18 – Pipe Connect, Event 22 (DNS) – SwiftOnSec
  - Timeseries data: Module Load Sizes, Registry Depth



procmonML Experimental Setup

ML Model Validation: Does the ML model detect TXXXX?
Rule Validation: Why does the ML model detect TXXXX?

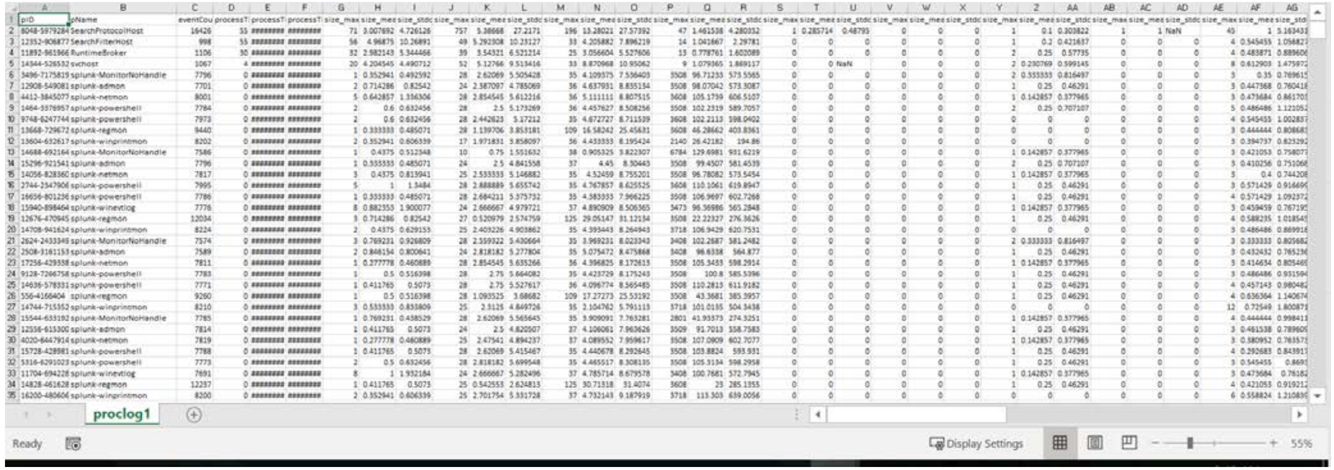


1. Collect Background/Attack Data
2. Train Model on Background/Attack Data
3. Develop Rules from Trained Model
4. Transfer Trained Model to Production

1. Collect Background/Attack Data
2. Test ML Model on Background/Attack Data
3. Test Rules in Splunk

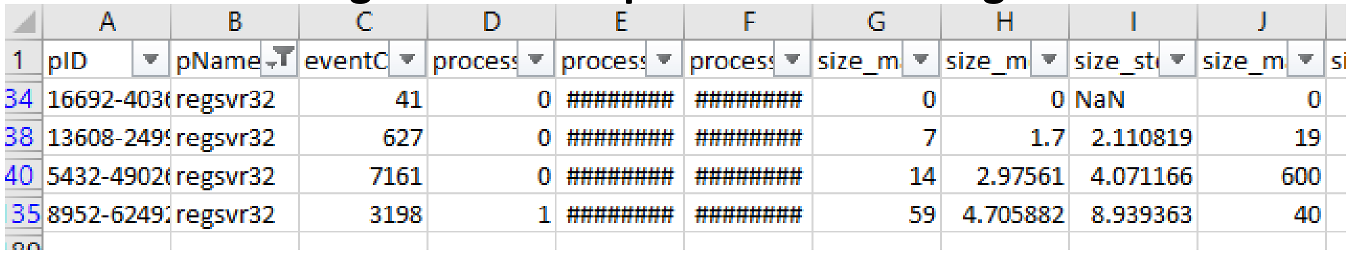
procmonML: T1117 Regsvr32 Training

Background process monitoring data



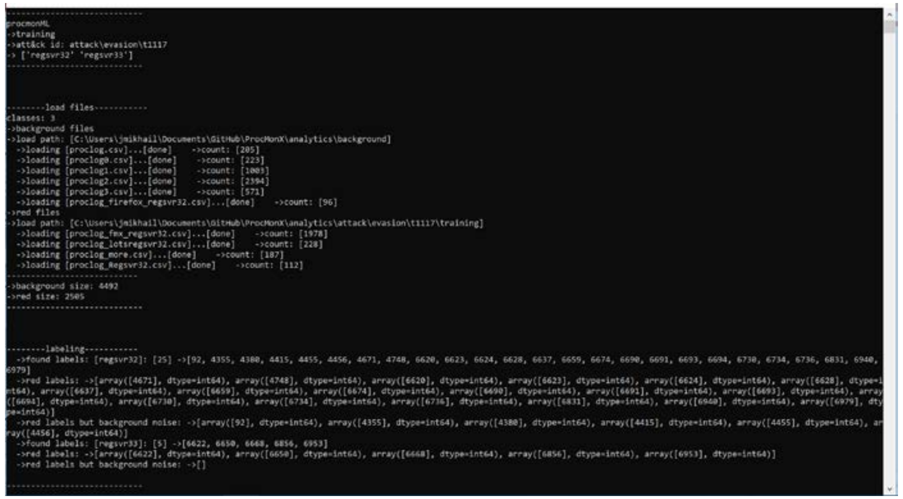
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1	pID	pName	eventCou	processT	processT	processT	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_stdc	size_max	size_mea	size_std
2	8048-5979284	SearchProtocolHost	16426	55	########	########	71	3.007692	4.726126	757	5.38668	27.2171	196	13.28021	27.57392	47	1.461538	4.280352	1	0.285714	0.48795	0	0	0	1	0.1	0.303822	1	1	NaN	45	1	5.16343
3	12352-906877	SearchFilterHost	998	55	########	########	56	4.96875	10.26891	49	5.292308	10.23127	33	4.205882	7.896219	14	1.041667	2.29781	0	0	0	0	0	0	1	0.2	0.421637	0	0	0	4	0.545455	1.05882
4	11892-961966	RuntimeBroker	1106	30	########	########	32	2.982143	5.344466	39	3.54321	6.521214	25	3.056604	5.527606	13	0.778761	1.602089	0	0	0	0	0	0	2	0.25	0.57735	0	0	0	4	0.483871	0.88960
5	14344-526532	svchost	1067	4	########	########	20	4.204545	4.490712	52	5.12766	9.513416	33	8.870968	10.95062	9	1.079365	1.869117	0	0	NaN	0	0	0	2	0.230769	0.599145	0	0	0	8	0.612903	1.47597
6	3496-7175819	splunk-MonitorNoHandle	7796	0	########	########	1	0.352941	0.492592	28	2.62069	5.505428	35	4.109375	7.536403	3508	96.71233	573.5565	0	0	0	0	0	0	2	0.333333	0.816497	0	0	0	3	0.35	0.76961
7	12908-549081	splunk-admon	7701	0	########	########	2	0.714286	0.82542	24	2.387097	4.785069	36	4.637931	8.835154	3508	98.07042	573.3087	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	3	0.447368	0.76041
8	4412-3845077	splunk-netmon	8001	0	########	########	5	0.642857	1.336306	28	2.854545	5.612216	36	5.111111	8.807515	3608	105.1739	606.5107	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.473684	0.86170
9	1464-5576957	splunk-powershell	7784	0	########	########	2	0.6	0.632456	28	2.5	5.173269	36	4.457627	8.508256	3508	102.2319	589.7057	0	0	0	0	0	0	2	0.25	0.707107	0	0	0	5	0.486486	1.12105
10	9748-6247744	splunk-powershell	7973	0	########	########	2	0.6	0.632456	28	2.442623	5.17212	35	4.672727	8.711539	3608	102.2113	598.0402	0	0	0	0	0	0	0	0	0	0	0	0	4	0.545455	1.00283
11	13668-729672	splunk-regmon	9440	0	########	########	1	0.333333	0.485071	28	1.139706	3.853181	109	16.58242	25.45631	3608	46.28662	403.8361	0	0	0	0	0	0	0	0	0	0	0	0	3	0.444444	0.80868
12	13604-632617	splunk-winprintmon	8202	0	########	########	2	0.352941	0.606339	17	1.971831	3.858097	36	4.433333	8.195424	2140	26.42182	194.86	0	0	0	0	0	0	0	0	0	0	0	0	3	0.394737	0.82329
13	14688-692164	splunk-MonitorNoHandle	7586	0	########	########	1	0.4375	0.512348	10	0.75	1.551632	38	0.905325	3.822307	6784	129.6981	931.6219	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.421053	0.75807
14	15296-921541	splunk-admon	7796	0	########	########	1	0.333333	0.485071	24	2.5	4.841558	37	4.45	8.30443	3508	99.4507	581.4539	0	0	0	0	0	0	2	0.25	0.707107	0	0	0	3	0.410256	0.75106
15	14056-828360	splunk-netmon	7817	0	########	########	3	0.4375	0.813941	25	2.533333	5.146882	35	4.52459	8.755201	3508	96.78082	575.5454	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.4	0.74420
16	2744-2347906	splunk-powershell	7995	0	########	########	5	1	1.3484	28	2.888889	5.655742	35	4.767857	8.625525	3608	110.1061	619.8947	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	3	0.571429	0.91669
17	16656-801236	splunk-powershell	7786	0	########	########	1	0.333333	0.485071	28	2.684211	5.375732	35	4.383333	7.966225	3508	106.9697	602.7268	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	4	0.571429	1.09237
18	15940-898464	splunk-winevtlog	7776	0	########	########	8	0.882353	1.900077	24	2.666667	4.979721	37	4.890909	8.506365	3473	96.36986	565.2848	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.459459	0.76719
19	12676-470945	splunk-regmon	12034	0	########	########	3	0.714286	0.82542	27	0.520979	2.574759	125	29.05147	31.12134	3508	22.22327	276.3626	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	4	0.588235	1.01854
20	14708-941624	splunk-winprintmon	8224	0	########	########	2	0.4375	0.629153	25	2.403226	4.903862	35	4.393443	8.264943	3718	106.9429	620.7531	0	0	0	0	0	0	0	0	0	0	0	0	3	0.486486	0.86991
21	2624-2433349	splunk-MonitorNoHandle	7574	0	########	########	3	0.769231	0.926809	28	2.559322	5.430664	35	3.969231	8.023343	3408	102.2687	581.2482	0	0	0	0	0	0	2	0.333333	0.816497	0	0	0	3	0.333333	0.80568
22	2508-3161153	splunk-admon	7589	0	########	########	2	0.846154	0.800641	24	2.818182	5.277804	35	5.075472	8.475868	3408	96.6338	564.877	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	3	0.432432	0.76523
23	17256-429338	splunk-netmon	7811	0	########	########	1	0.277778	0.460889	28	2.854545	5.635266	36	4.396825	8.172613	3508	105.3433	598.2914	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.414634	0.80546
24	9128-7266758	splunk-powershell	7783	0	########	########	1	0.5	0.516398	28	2.75	5.664082	35	4.423729	8.175243	3508	100.8	585.5396	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	3	0.486486	0.93159
25	14636-578331	splunk-powershell	7771	0	########	########	1	0.411765	0.5073	28	2.75	5.527617	36	4.096774	8.565485	3508	110.2813	611.9182	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	4	0.457143	0.98048
26	556-4166404	splunk-regmon	9260	0	########	########	1	0.5	0.516398	28	1.093525	3.68682	109	17.27273	25.53192	3508	43.3681	385.3957	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	4	0.636364	1.14067
27	14744-715352	splunk-winprintmon	8210	0	########	########	3	0.533333	0.833809	25	2.3125	4.849726	35	2.104762	5.791113	3718	101.0135	504.3438	0	0	0	0	0	0	0	0	0	0	0	0	12	0.72549	1.80087
28	15544-633192	splunk-MonitorNoHandle	7785	0	########	########	1	0.769231	0.438529	28	2.62069	5.565645	35	3.909091	7.763281	2801	41.93373	274.3251	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	4	0.444444	0.99841
29	12556-615300	splunk-admon	7814	0	########	########	1	0.411765	0.5073	24	2.5	4.820507	37	4.106061	7.963626	3509	91.7013	558.7583	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	3	0.461538	0.78960
30	4020-6447914	splunk-netmon	7819	0	########	########	1	0.277778	0.460889	25	2.47541	4.894237	37	4.089552	7.959617	3508	107.0909	602.7077	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.380952	0.76357
31	15728-428881	splunk-powershell	7788	0	########	########	1	0.411765	0.5073	28	2.62069	5.415467	35	4.440678	8.292645	3508	103.8824	593.931	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	4	0.292683	0.84391
32	5316-6291023	splunk-powershell	7773	0	########	########	2	0.5	0.632456	28	2.818182	5.699548	35	4.465517	8.308135	3508	105.3134	598.2958	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	3	0.545455	0.869
33	11704-694228	splunk-winevtlog	7691	0	########	########	8	1	1.932184	24	2.666667	5.282496	37	4.785714	8.679578	3408	100.7681	572.7945	0	0	0	0	0	0	1	0.142857	0.377965	0	0	0	3	0.473684	0.7618
34	14828-461628	splunk-regmon	12237	0	########	########	1	0.411765	0.5073	25	0.542553	2.624813	125	30.71318	31.4074	3608	23	285.1355	0	0	0	0	0	0	1	0.25	0.46291	0	0	0	4	0.421053	0.91921
35	16200-480606	splunk-winprintmon	8200	0	########	########	2	0.352941	0.606339	25	2.701754	5.331728	37	4.732143	9.187919	3718	113.303	639.0056	0	0	0	0	0	0	0	0	0	0	0	0	6	0.558824	1.21083

Regsvr32 attack process monitoring data



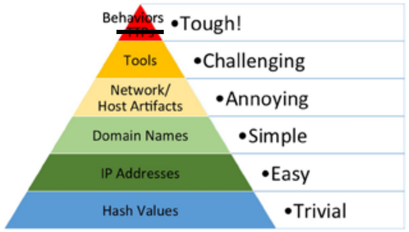
	A	B	C	D	E	F	G	H	I	J	
1	pID	pName	eventC	process	process	process	size_m	size_m	size_st	size_m	si
34	16692-403(	regsvr32	41	0	#########	#########	0	0	NaN	0	
38	13608-249	regsvr32	627	0	#########	#########	7	1.7	2.110819	19	
40	5432-4902(	regsvr32	7161	0	#########	#########	14	2.97561	4.071166	600	
35	8952-6249	regsvr32	3198	1	#########	#########	59	4.705882	8.939363	40	

Model Supervised Training



```
----------------------------
procmonML
->training
->att&ck id: attack\evasion\t1117
-> ['regsvr32' 'regsvr33']
----------------------------



--------load files-----------
classes: 3
->background files
->load path: [C:\Users\jmikhail\Documents\GitHub\ProcMonX\analytics\background]
  ->loading [proclog.csv]...[done]    ->count: [205]
  ->loading [proclog0.csv]...[done]    ->count: [223]
  ->loading [proclog1.csv]...[done]    ->count: [1003]
  ->loading [proclog2.csv]...[done]    ->count: [2394]
  ->loading [proclog3.csv]...[done]    ->count: [571]
  ->loading [proclog_firefox_regsvr32.csv]...[done]    ->count: [96]
->red files
->load path: [C:\Users\jmikhail\Documents\GitHub\ProcMonX\analytics\attack\evasion\t1117\training]
  ->loading [proclog_fmx_regsvr32.csv]...[done]    ->count: [1978]
  ->loading [proclog_lotsregsvr32.csv]...[done]    ->count: [228]
  ->loading [proclog_more.csv]...[done]    ->count: [187]
  ->loading [proclog_Regsvr32.csv]...[done]    ->count: [112]
---------------------------
->background size: 4492
->red size: 2505
----------------------------



--------labeling-----------
  ->found labels: [regsvr32]: [25] ->[92, 4355, 4380, 4415, 4455, 4456, 4671, 4748, 6620, 6623, 6624, 6628, 6637, 6659, 6674, 6690, 6691, 6693, 6694, 6730, 6734, 6736, 6831, 6940,
6979]
  ->red labels: ->[array([4671], dtype=int64), array([4748], dtype=int64), array([6620], dtype=int64), array([6623], dtype=int64), array([6624], dtype=int64), array([6628], dtype=i
nt64), array([6637], dtype=int64), array([6659], dtype=int64), array([6674], dtype=int64), array([6690], dtype=int64), array([6691], dtype=int64), array([6693], dtype=int64), array
([6694], dtype=int64), array([6730], dtype=int64), array([6734], dtype=int64), array([6736], dtype=int64), array([6831], dtype=int64), array([6940], dtype=int64), array([6979], dty
pe=int64)]
  ->red labels but background noise: ->[array([92], dtype=int64), array([4355], dtype=int64), array([4380], dtype=int64), array([4415], dtype=int64), array([4455], dtype=int64), ar
ray([4456], dtype=int64)]
  ->found labels: [regsvr33]: [5] ->[6622, 6650, 6668, 6856, 6953]
  ->red labels: ->[array([6622], dtype=int64), array([6650], dtype=int64), array([6668], dtype=int64), array([6856], dtype=int64), array([6953], dtype=int64)]
  ->red labels but background noise: ->[]

----------------------------
```

Behavioral vs Heuristic Analytics

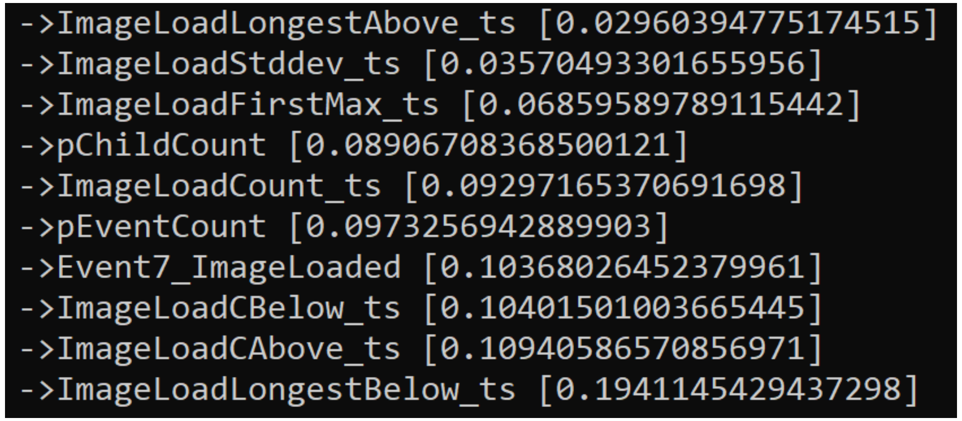


Behaviors •Tough!
Tools •Challenging
Network/ Host Artifacts •Annoying
Domain Names •Simple
IP Addresses •Easy
Hash Values •Trivial

- **T1117/Regsvr32**
  - **Heuristic:** index=__your_sysmon_data__ EventCode=1 regsvr32.exe | search ParentImage="*regsvr32.exe" AND Image!="*regsvr32.exe*“
  - **Behavior:** ImageLoadCAbove_ts > 15.5 AND ImageLoadCBelow_ts > 55.5 AND pChildCount > 0.5 AND pEventCount <= 90.5 AND pTotalTime <= 19.0
    - Generated from Skope-Rules

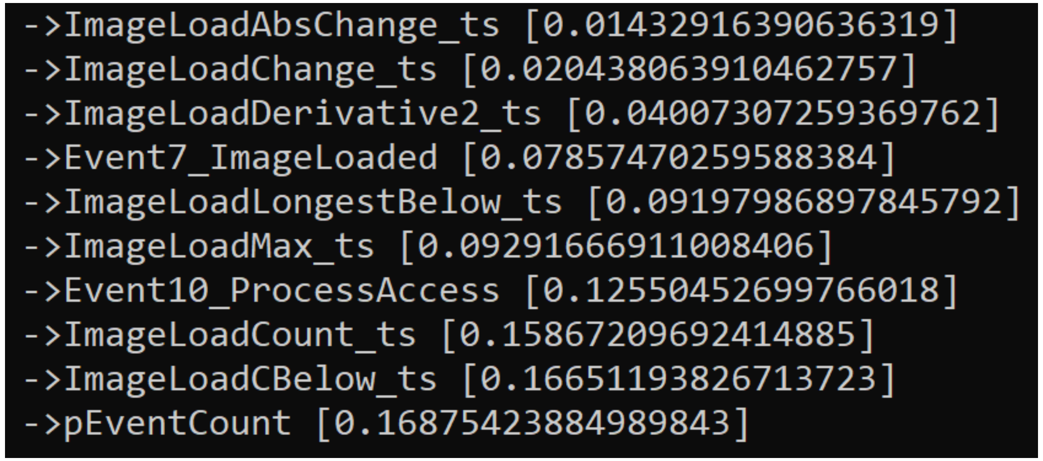
- **T1003/Lsass Memory Dumping via Task Manager**
  - **Heuristic**: index=__your_sysmon_index__ EventCode=11 TargetFilename="*lsass*.dmp" Image="C:\\Windows\\*\\taskmgr.exe“
  - **Behavior:** Event10_ProcessAccess > 26.0 AND ImageLoadCount_ts > 72.5 AND ImageLoadMax_ts > 27887596.0
    - Generated from Skope-Rules

T1117 Random Forest: Top 10 Important Features



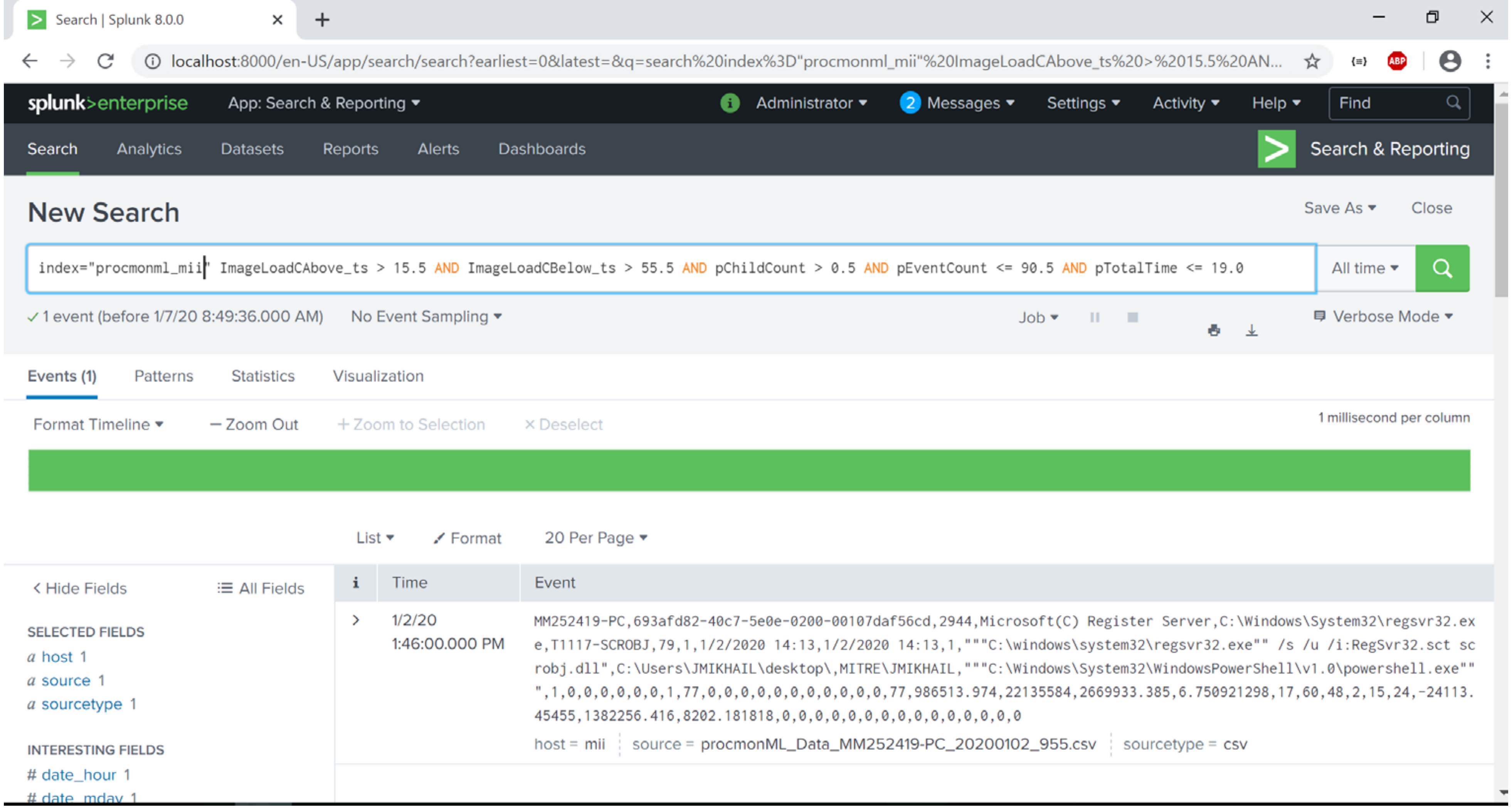
```
->ImageLoadLongestAbove_ts [0.029603947751745l5]
->ImageLoadStddev_ts [0.035704933016559561]
->ImageLoadFirstMax_ts [0.06859589789115442]
->pChildCount [0.08906708368500121]
->ImageLoadCount_ts [0.09297165370691698]
->pEventCount [0.0973256942889903]
->Event7_ImageLoaded [0.10368026452379961]
->ImageLoadCBelow_ts [0.10401501003665445]
->ImageLoadCAbove_ts [0.10940586570856971]
->ImageLoadLongestBelow_ts [0.1941145429437298]
```

T1003/Task Manager Random Forest:
Top 10 Important Features



```
->ImageLoadAbsChange_ts [0.014329163906363l9]
->ImageLoadChange_ts [0.020438063910462757]
->ImageLoadDerivative2_ts [0.04007307259369762]
->Event7_ImageLoaded [0.07857470259588384]
->ImageLoadLongestBelow_ts [0.09197986897845792]
->ImageLoadMax_ts [0.09291666911008406]
->Event10_ProcessAccess [0.12550452699766018]
->ImageLoadCount_ts [0.15867209692414885]
->ImageLoadCBelow_ts [0.16651193826713723]
->pEventCount [0.16875423884989843]
```

Behavior Analytics in Splunk



```
index="procmonml_mii" ImageLoadCAbove_ts > 15.5 AND ImageLoadCBelow_ts > 55.5 AND pChildCount > 0.5 AND pEventCount <= 90.5 AND pTotalTime <= 19.0
```

✓ 1 event (before 1/7/20 8:49:36.000 AM) No Event Sampling

1/2/20 1:46:00.000 PM

```
MM252419-PC,693afd82-40c7-5e0e-0200-00107daf56cd,2944,Microsoft(C) Register Server,C:\Windows\System32\regsvr32.ex
e,T1117-SCROBJ,79,1,1/2/2020 14:13,1/2/2020 14:13,1,"""C:\windows\system32\regsvr32.exe"" /s /u /i:RegSvr32.sct sc
robj.dll",C:\Users\JMIKHAIL\desktop\,MITRE\JMIKHAIL,"""C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe""
",1,0,0,0,0,0,0,1,77,0,0,0,0,0,0,0,0,0,0,0,77,986513.974,22135584,2669933.385,6.750921298,17,60,48,2,15,24,-24113.
45455,1382256.416,8202.181818,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

host = mii | source = procmonML_Data_MM252419-PC_20200102_955.csv | sourcetype = csv

Closing Thoughts

- **The susceptibility of a given technique to evasion (as characterized by slide 6) should be one of the primary factors of whether to implement a machine learning analytic or a heuristic analytic**
  - Data and organization factors are key underlying components
- **Analytics relying on primarily string/signature-based data sources are too easy to evade**
- **Process monitoring offers data about the behavior of a process – much more difficult to evade**
  - Inherently higher dimensional data requiring more complex analytics
  - Process monitoring data can be condensed on the endpoint to reduce data quantity
- **Adversaries will try to evade ML models – but this increases their work factor!**

- **Contact Info**
  - Joe Mikhail jmikhail@mitre.org
  - Brandon Werner bwerner@mitre.org

MITRE's mission-driven teams are dedicated to solving problems for a safer world. Through our federally funded R&D centers and public-private partnerships, we work across government to tackle challenges to the safety, stability, and well-being of our nation.

Learn more www.mitre.org