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

Dr. Joe Mikhail Brandon Werner The MITRE Corporation





© 2019 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited, Case Number 19-3685.

#### **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?

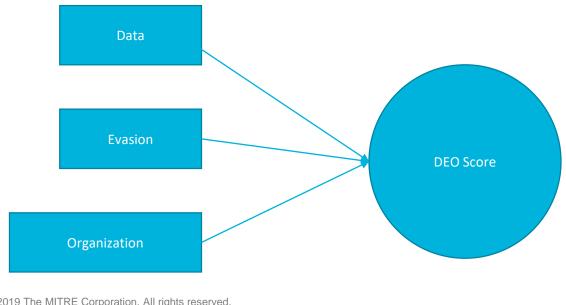
#### Definitions

- 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.



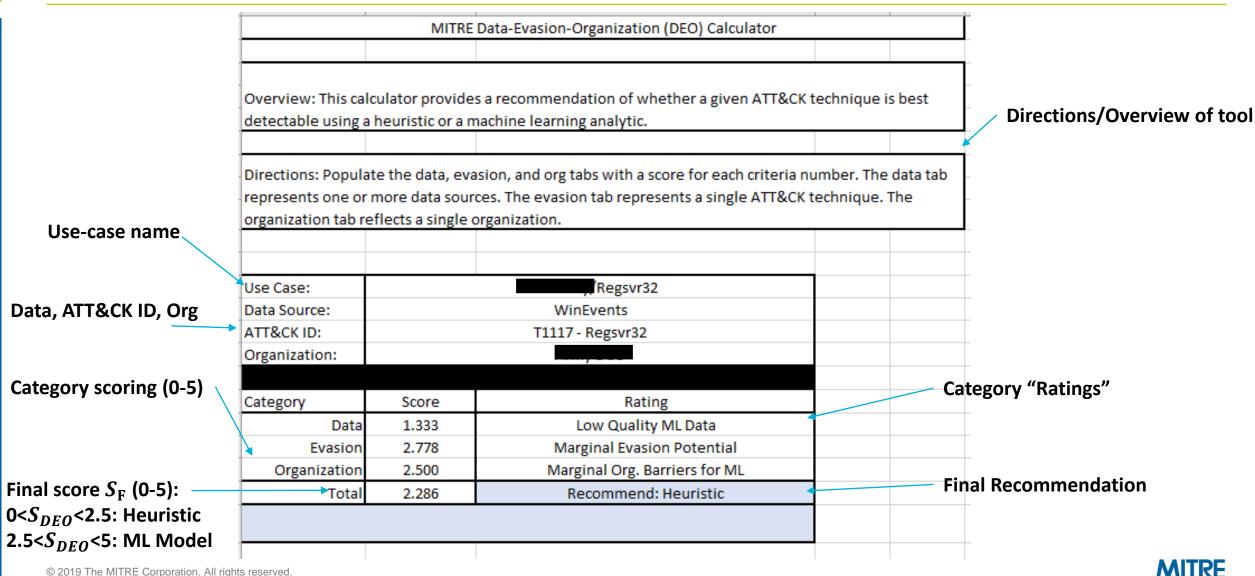
Given categorical weights for data, evasion, and organization:  $W_D = 1, W_F = 1.5, W_O = 1,$ And scoring for each category:  $S_D S_E, S_O$ For the weighted total:  $W_T = W_D + W_F + W_O$ The final DEO score,  $S_{DFO} = W_D S_D + W_F S_F + W_O S_O$ **Output:**  $0 < S_{DEO} < 2.5$ : Heuristic

2.5<*S*<sub>DE0</sub><5: ML Model

© 2019 The MITRE Corporation. All rights reserved



### **Data-Evasion-Organization (DEO) Framework**



### **Data Scoring Factors**

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



### **Evasion Scoring Factors**

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

### **Organization Scoring Factors**

	Organization Na	me: Org Name	_
Criteria #	Criteria	Description	Weight
0.1	Chillest	Score the organization's in-house and outsourced ML skillsets.	
0.1	Skillset	0=Novice 5=Expert	
		Has the organization previously implemented advanced	
0.2	Previous experience	analytics or ML?	
		0=Never implemented 5=Several implementations	
0.3	Executive level support	Score the organization's leadership support for ML.	
0.5		0=No support 5=Full support	
		Are some of the networks within the organization classified or	
0.4	Classification / Sensitivity	sensitive, requiring additional effort for data ingest and	
		processing? 0=Many networks 5=No networks	
0.5	Zero-Day Threats	Score the quantity of zero-day threats that the organization	
0.5		faces. 0=No zero-days 5=Many zero-days	:
		Is the organization's security architecture simplified and	
0.6	Security Architecture	organized in a cohesive manner?	
		0=Unorganized 5=Organized	
0.7	Funding	Is there sufficient funding to invest in analytic development?	
0.7		0=No Funding 5=Sufficient Funding	
0.8		What is the timeframe to work with to deploy a given analytic?	
0.8	Timeframe	0=Short-term(Hours/Days) 5=Long-Term(Months/Years)	
		How often are the SOC's signature-based detection capabilities	
0.9	Signature Updates	updated with new signatures?	
		0=At least once a week 5=Annually	
		How often are the organization's network devices and	
0.10	Patching Updates	endpoints updated with software patches?	
		0=At least once a week 5=Annually	:



### 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

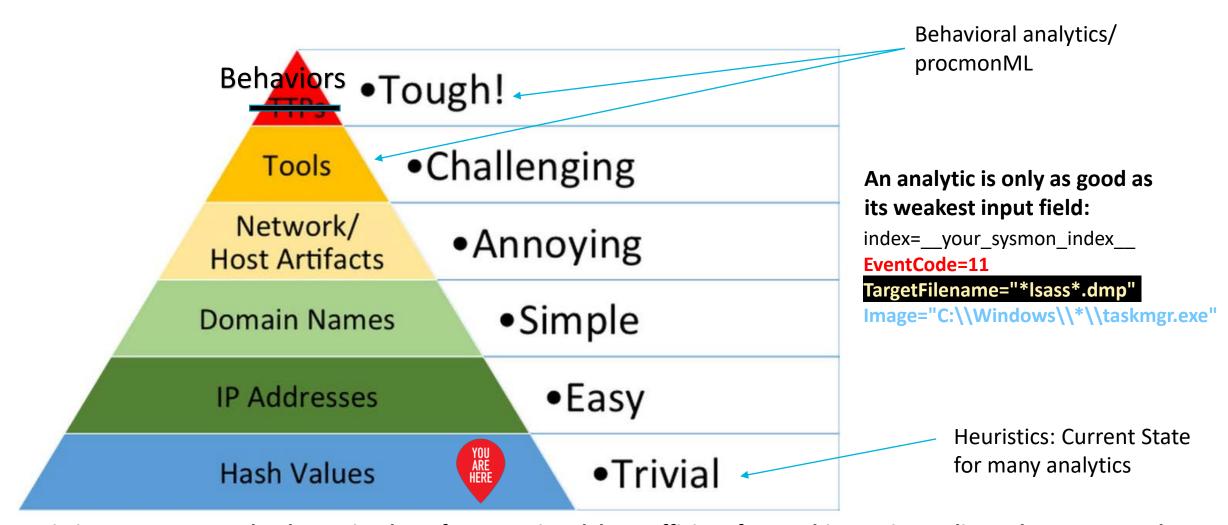


Why ML for host-based detection?

- 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**



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

© 2019 The MITRE Corporation. All rights reserved.

MITRE

### procmonML Data Organization

# No PII!

	А	В	С	D	Е	F	G	Н	I
1	mName	pID	pName	eventCour	pTimeTota	psTimeSta	psTimeEn	Thread_co	Process_c(N
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

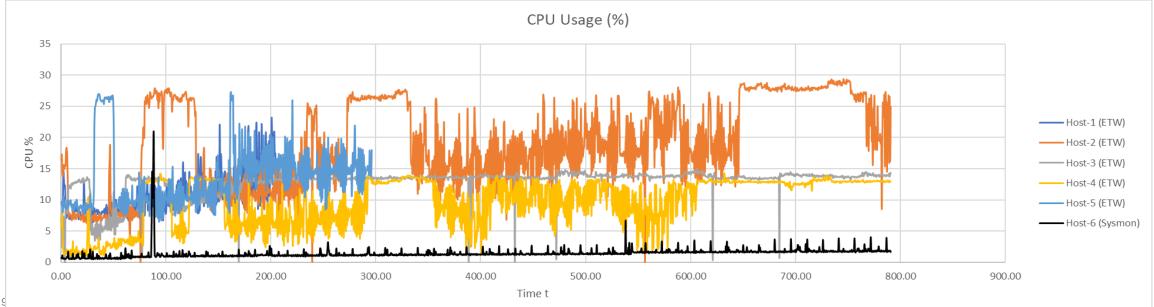
#### Windows ETW:

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

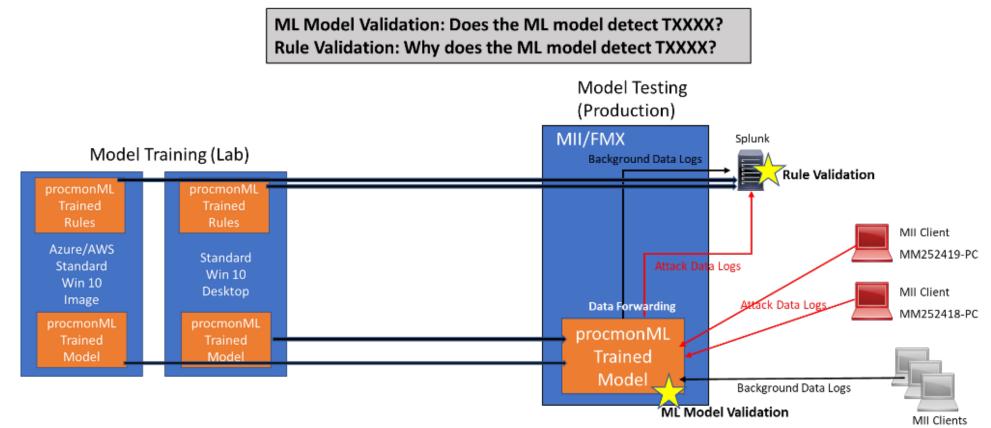
#### Sysmon:

 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



- 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

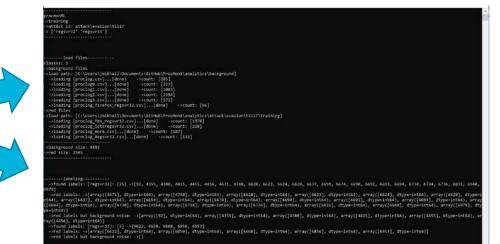
#### Background process monitoring data

048-5979284 SearchProtocolHost 2352-906877 SearchFriterHost	16426		71 3.007692 4		757 5.38668 27.2171	196 13 28021 27 57392	ize_max size_mea size_stdc siz 47 1.461538 4.280332		85714 0.4879				1 0	1 0.303822		1 NaN		45 1 5 1634	
	998	55 PROBADDO ADDRESS	56 4.96875 1		49 5.292508 10.23127	33 4.205882 7.896219	14 1.041667 2.29781	1.0.4	Baran Windra	3 9				3 0.431637		A rears		4 0.545455 1.0564	
1892-961966 RuntimeBroker	1106	35 ADDREEDS ADDREEDS	32 2.982143 5		39 3.54321 6.521214	25 3.056604 5.527606	13 0.778761 1.602089			0 0				5 0.57735	0	0		4 0.483871 0.8894	
4344-526532 suchost	1067	A REALIZED DARRAND	20 4 204545 4		52 5.12766 9.513416	13 8.870968 10.95062	9 1.079365 1.869117		0 NaN					0.599145	0		-	8 0.612903 1.4754	
	2067	C personal management						0	o reare	0					0	0	0		
496-7175819 splunk-MonitorNoHandle	7795		1 0.352941 0		28 2.62069 5.505428	35 4 109375 7 536403	3508 96.71233 573 5565		0	0 0				3 0.816497 5 0.46291			- <u>v</u>	3 0.35 0.7696	
2908-549081 splunk-admon	9001	0 88888888 88888888	2 0.714286 5 0.642857 1		24 2.387097 4.785069 28 2.854545 5.612216	36 4.637931 8.835154	3508 98.07042 573.3087 3608 105.1739 606.5107	0	0	0.0		0		15 0.46291 17 0.377965	0	.0		3 0.447368 0.7604	
412-3845077 splunk-netmon		O separate accords				36 5.111111 8.807515		0	0	0 0	0	0			0	0	0	3 0.473684 0.8651	
464-3376957 splurk-powershell	7784	O RESERVE RESERVED		632456	28 2.5 5.173269	36 4.457627 8.508256	5508 102 2319 589 7057	0	0	0 0		0	2 0.	15 0,707107	0	0	0	5 0.486486 1.1211	
748-6247744 splunk-powershell	7973	O REARESS RESERVED		632456	28 2.442623 5.17212	35 4.672727 8.711539	3608 102 2113 598 0402	8	9	0 0		9		0 0	6		8	4 0.545455 1.0028	
3668-729672 splunk-regmon	9440	O REALESS RECEIPT	1 0.333333 0		28 1.139736 3.853181	109 16 58242 25 45631	5608 45.28662 403.8361	0	0	0 0	0	0	0	0 0	0	0	0	3 0.444444 0.8088	
3604-632617 splunk-winprintmon	8202	O ABDROOM BARREDO	2 0.352941 0		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	3 0.394737 0.8233	
4688-692164 splunk-MonitorNoHandle	7586	O REALIZED ADDRESS OF	1 0.4375 0		10 0.75 1.551632	38 0.905325 3.822307	6784 129,6981 931,6219	0	0	0 0	0	0		0.377965	0	0	0	3 0.421053 0.7580	
5296-921541 splunk-admon	7796	O BREESES REPRESE	1 0.353355 0		24 2.5 4.841558	37 4.45 8.50443	3508 99.4507 581.4539	0	0	0 0	0	0		5 0.707107	0	0	0	3 0.410256 0.7510	
4056-828360 splunk-netmon	7817	C BREARDER REPRESENT	3 0.4375 0		25 2.533333 5.146882	35 4.52459 8.755201	3508 96.78082 575.5454	0	0	0 0	0	0		57 0.377965	0	0	0	5 0.4 0.7443	
744-2547906 splunk-powershell	7995	0 ######## #########		1.5484	28 2.888889 5.655742	35 4.767857 8.625525	3608 110.1061 619.8947	0	0	0 0	0	0		15 0.46291	0	0	0	3 0.571429 0.9166	
6656-801236 splunk-powershell	7786	O second account	1 0.333333 0		28 2.684211 5.375732	35 4.583333 7.966225	3508 106.9697 602.7268	0	0	0 0	0	0		5 0.46291	0	0	0	4 0.571429 1.0925	
5940-898464 splunk-winevtlog	7776	O second approach	8 0.882353 1		24 2.666667 4.979721	37 4.890909 8.506365	5473 96.56986 565.2848	0	0	0 0	0	0		57 0.377965	0	0	0	3 0.459459 0.7671	
2676-470945 splunk-regmon	12034	O SOUTHER BECOMES	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 1.0185	3545
4708-941624 splunk-winprintmon	8224	O SERRERA SERRERS O	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 0.8691	19918
524-2433549 splunk-MonitorNoHandle	7574	0 ######## #########	3 0.769231 0		28 2.559322 5.430664	35 3.969231 8.023343	3408 102.2587 581.2482	0	0	0 0	0	0		13 0.816497	0	0	0	3 0.333333 0.8054	
508-3161153 splunk-admon	7589	O DESERVE ADDRESSES	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.	15 0.46291	0	0	0	3 0.432432 0.7653	6236
7256-429338 splunk-netmon	7811	O BREERER REFERER	1 0.277778 0	460889	28 2.854545 3.635266	36 4.396825 8.172613	3508 105.3433 598.2914	0	0	0 0	0	0	1 0.1428	37 0.377965	0	0	0	3 0.414634 0.8054	15465
128-7206758 splunk-powershell	7783	O REPORTE RECORDED	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.	15 0.46291	0	0	0	3 0.486486 0.9313	1594
4636-578331 splunk-powershell	7771	O SECOND ASSESSED O	1 0.411765	0.5073	28 2.75 5.527617	36 4.096774 8.565485	3508 110.2815 611.9182	0	0	0 0	0	0	1 0.	15 0.46291	0	0	0	4 0.457143 0.9804	10482
56-4166404 splunk-regmon	9260	O SPERSOR ASSESSED	1 05 0	1.516398	28 1.093525 3.68682	109 17.27273 25.53192	3508 43.3681 385.3957	0	0	0 0	0	0	1 0.	15 0.46291	0	0	0	4 0.636364 1.1406	0674
4744-715352 splunk-winpringmon	\$210	O sessess assesses	3 0.533333 0	833809	25 2.5125 4.849726	35 2.104762 5.791113	3718 101.0155 504 3438	ő	0	0 0	6	0	0	0 0	0	0	0	12 0.72549 1.8008	:0871
5544-633192 splunk-MonitorNoHandle	7765	O ADDRESS ADDRESS (	1 0.769291 0	1.438529	28 2.62069 5.565645	35 3.909091 7.763281	2801 41,93373 274 3251	0	0	0 0	0	0	1 0.1428	7 0.377965	0	8	0	4 0.444444 0.9984	18411
2556-615300 splunk-admon	7814	O presses assesses O	1 0.411765	0.5073	24 2.5 4.820507	37 4.106061 7.963626	3509 91.7013 358.7585	0	D	0 0	0	0	1 0.	15 0.46291	0	0	0	3 0.461538 0.7890	19605
020-6447914 splunk-netmon	7819	O RESERVED REPRESENT	1 0,277778 0	460889	25 2.47541 4.894237	37 4.089552 7.959617	\$508 107.0909 602.7077	0	0	0 0	0	0	1 0 1428	7 0.377965	0	0	0	3 0.580952 0.7835	13578
5728-428581 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	8	1 0.	15 0.46291	0	0	0	4 0.292683 0.8439	13917
316-6291023 splunk-powershell	7773	O	2 0.5 0	.432456	28 2.818182 5.699548	15 4.465517 8.308135	3508 105 3154 598 2958	0	0	0 0	0	0	1 0.	5 0.46291	0	0	0	3 0.545455 0.84	\$601
1704-694228 splunk-winevtice	7691	O ABBRRAR ABBRRARY	8 11	932184	24 2.666667 5.282496	37 4,785714 8,679578	3408 100.7681 572.7945	0	0	0 0	0	0	1 0 1428	7 0.177965	0	0	0	3 0.473684 0.763	16182
4828-461628 splunk-regmon	12237	0	1 0.411765	0.5075	25 0.542553 2.624813	125 30.71318 31.4074	3608 23 285 1355	0	0	0 0	0	0	1 0	5 0.46291	0	0	0	4 0.421053 0.9193	9213
6200-480606 splunk-wingrintmon	8200	O apparents apparent O	2 0.352941 0		25 2.701754 5.551728	57 4.732148 9.187919	3718 115.303 639.0056	0	0	0 0	0	0	0	0 0	0	0	0	6 0.558824 1.210	
	(+)							10	4									1	1

#### Regsvr32 attack process monitoring data

	Α		В	С	D	E	F	G	Н	1	J	
1	pID	Ŧ	pName 🖅	eventC 💌	process 🔻	process 💌	process 💌	size_m 💌	size_m 💌	size_st 💌	size_m 💌 s	si
34	16692-4	103(	regsvr32	41	0	****	****	0	0	NaN	0	
38	13608-2	2499	regsvr32	627	0	*****	*****	7	1.7	2.110819	19	
40	5432-49	9026	regsvr32	7161	0	*****	****	14	2.97561	4.071166	600	
35	8952-62	2492	regsvr32	3198	1	****	*****	59	4.705882	8.939363	40	
00												

#### **Model Supervised Training**



MITRE

#### © 2019 The MITRE Corporation. All rights reserved.

## **Behavioral vs Heuristic Analytics**

#### • 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</li>
  - 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

rs •Tough!

lost Artifact

Hash Values

Challenging
 Annoying

•Simple •Easy

Trivial

- ->ImageLoadLongestAbove\_ts [0.02960394775174515]
- ->ImageLoadStddev\_ts [0.03570493301655956]
- ->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 [0.01432916390636319]
- ->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]

14



### **Behavior Analytics in Splunk**

		+								
$- \rightarrow C$ (i) local	host:8000/en-U	5/app/sea	arch/search?earlie	=0&latest=&q=search%20index%3D"procmonml		dCAbove_ts%20>	%2015.5%20	DAN	☆ {=}	
splunk>enterprise	App: Search	& Reporti	ing ▼	i Administrator 🔻	2 Messages ▼	Settings -	Activity -	Help 🔻	Find	<u>م</u>
Search Analytics	Datasets	Reports	Alerts D	hboards				>	Search &	Reporting
New Search								5	Save As 🔻	Close
index="procmonml_mii	' ImageLoadCAb	ove_ts >	15.5 AND Image	adCBelow_ts > 55.5 AND pChildCount > 0.5 AND	) pEventCount <= 9	0.5 AND pTotalT	ime <= 19.0	1	All time	• Q
/ 1 event (before 1/7/20 8	8:49:36.000 AM)	No E	Event Sampling 🔻		L	ob 🕶 🔢 🔳	ē	Ŧ	<b>₽</b> Verbos	se Mode 🔻
Events (1) Patterns	Statistics	Visualiz	zation							
Format Timeline 🔻	– Zoom Out	+ Zooi	m to Selection	× Deselect					1 milliseco	nd per colum
		List	<ul> <li>Format</li> </ul>	20 Per Page 🔻						
	:≡ All Fields	i	Time	Event						
< Hide Fields	ECTED FIELDS ost 1 ource 1		<pre>&gt; 1/2/20 MM252419-PC,693afd82-40c7-5e0e-0200-00107daf56cd,2944,Microsoft(C) Register Server,C:\Window 1:46:00.000 PM e,T1117-SCROBJ,79,1,1/2/2020 14:13,1/2/2020 14:13,1,"""C:\windows\system32\regsvr32.exe"" robj.dll",C:\Users\JMIKHAIL\desktopMITRE\JMIKHAIL,"""C:\Windows\System32\WindowsPowerShel ",1,0,0,0,0,0,0,1,77,0,0,0,0,0,0,0,0,0,0,</pre>						2	0
< Hide Fields SELECTED FIELDS 7 host 1 7 source 1 7 sourcetype 1		>		e,T1117-SCROBJ,79,1,1/2/2020 14:13,1/2/2020 robj.dll",C:\Users\JMIKHAIL\desktopMITRE\ ",1,0,0,0,0,0,0,1,77,0,0,0,0,0,0,0,0,0,0,	) 14:13,1,"""C:\w JMIKHAIL,"""C:\W 77,986513.974,22	ndows\system32\ ndows\System32\	regsvr32.ex WindowsPowe	e"" /s / rShell\\	1.0\power	shell.exe

# **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 <u>bwerner@mitre.org</u>



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 <u>www.mitre.org</u>



