ENDGAME.

ProblemChild

Discovering anomalous patterns based on parent-child process relationships

Bobby Filar (@filar)
Director of Data Science

Problem.

Petya Ransomware

- June 2017 Ransomware variant Petya hit orgs using an SMB vulnerability,
- Post-exploitation was a series of benign actions for cred dumping, self-execution, scheduling tasks, and wiping logs.
- Used available MSFT tools like WMIC and schtasks.exe.

Living Off The Land

- *FIST! FIST! FIST! It's all in the wrist: Remote Exec* by **grugq** (July 2004)

- Living Off The Land
  - **Incursion** – initial access vector
  - **Persistence** – post-compromise actions
  - **Payload** – dual-use tools (e.g. psexec)

- LOLBAS project details binaries, and libraries used for "Living Off The Land"

Response.

MITRE ATT&CK™ Framework

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data Staged	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe

- Knowledge base that organizes behaviors (**techniques**) by objectives (**tactics**)
- Most techniques are used by multiple groups and red teams

EQL: Event Query Language

- Simple and concise language for threat researchers
- Schema-independent and OS-agnostic
- Real-time detection with stream processing
- Supports multi-event behaviors, stacking and sifting through data

Example Detector: Initial Access & Execution

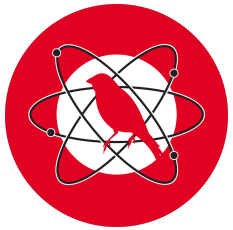
```
process where
  parent_process_name in ("winword.exe", "excel.exe", "powerpnt.exe")
  and process_name in ("powershell.exe", "cscript.exe",
                       "wscript.exe", "cmd.exe")
```

- **Technique** Spearphishing Attachment (T1193)
  PowerShell (T1086)
- **Detection** Scriptable child processes of Office products
  - PowerShell, VB script, cmd.exe

EQL

Red vs Blue Frameworks

- Red Team Frameworks
  - Atomic Red Team
  - Red Team Automation
  - Caldera Adversary Emulation
  - Metta
- Blue Team Frameworks
  - AtomicBlue
  - Cyber Analytics Repository (CAR)
  - MSFT ATP Queries



MITRE

ENDGAME.

Uber

Microsoft

Great sources for features and labeled data!

Machine Learning + Threat Research

- **SOMETIMES SECURITY PRODUCTS FAIL**
  (yes... even ML/AI/Deep Learning Magic)
  - These tactics bypass static signatures & NGAV
- Living off the land → Rules Engines
- **GOAL:** Apply ML to help researchers write detectors
  - Attempt to reduce/rank event data

ProblemChild Approach

1. Data Ingest
2. Detecting Anomalous Events
3. Prevalence Service
4. Output

01 Data Ingest.

Data Ingest

- Data from multiple sources
  - Winlogbeat
  - Sysmon
  - osquery
  - Endpoint sensors
- Target process creation events
  - Focus on parent-child process chains

```
{'timestamp_utc': '2018-08-15 01:14:44Z',
 'pid': 4856,
 'signature_status': 'trusted',
 'serial_event_id': 240227,
 'signature_signer': 'Microsoft Windows',
 'event_subtype_full': 'creation_event',
 'command_line': 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding',
 'ppid': 892,
 'sha256': 'c8533bb3b6088efb1d641b76fc7583c6bb7a...',
 'user_name': 'SYSTEM',
 'process_path': 'C:\\Windows\\System32\\wbem\\WmiPrvSE.exe',
 'user_sid': 'S-1-5-18',
 'timestamp': 131787692844718134,
 'process_name': 'WmiPrvSE.exe',
 'authentication_id': 999,
 'original_file_name': 'Wmiprvse.exe',
 'md5': 'a782a4ed336750d10b3caf776afe8e70',
 'sha1': 'bdab221ccef7acd7a027447725de8ffeaebef22c',
 'event_type_full': 'process_event',
 'opcode': 1,
 'user_domain': 'NT AUTHORITY'}
```

Data Normalization & Feature Engineering

How do we go from this...


```
{'timestamp_utc': '2018-08-15 01:14:44Z',
 'pid': 4856,
 'signature_status': 'trusted',
 'serial_event_id': 240227,
 'signature_signer': 'Microsoft Windows',
 'event_subtype_full': 'creation_event',
 'command_line': 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding',
 'ppid': 892,
 'sha256': 'c8533bb3b6088efb1d641b76fc7583c6bb7a...',
 'user_name': 'SYSTEM',
 'process_path': 'C:\\Windows\\System32\\wbem\\WmiPrvSE.exe',
 'user_sid': 'S-1-5-18',
 'timestamp': 131787692844718134,
 'process_name': 'WmiPrvSE.exe',
 'authentication_id': 999,
 'original_file_name': 'Wmiprvse.exe’,
 'md5': 'a782a4ed336750d10b3caf776afe8e70',
 'sha1': 'bdab221ccef7acd7a027447725de8ffeaebef22c’,
 'event_type_full': 'process_event',
 'opcode': 1,
 'user_domain': 'NT AUTHORITY'}
```

Data Normalization & Feature Engineering

How do we go from this... → to this?

```
{'timestamp_utc': '2018-08-15 01:14:44Z',
 'pid': 4856,
 'signature_status': 'trusted',
 'serial_event_id': 240227,
 'signature_signer': 'Microsoft Windows',
 'event_subtype_full': 'creation_event',
 'command_line': 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding',
 'ppid': 892,
 'sha256': 'c8533bb3b6088efb1d641b76fc7583c6bb7a...',
 'user_name': 'SYSTEM',
 'process_path': 'C:\\Windows\\System32\\wbem\\WmiPrvSE.exe',
 'user_sid': 'S-1-5-18',
 'timestamp': 131787692844718134,
 'process_name': 'WmiPrvSE.exe',
 'authentication_id': 999,
 'original_file_name': 'Wmiprvse.exe’,
 'md5': 'a782a4ed336750d10b3caf776afe8e70',
 'sha1': 'bdab221ccef7acd7a027447725de8ffeaebef22c’,
 'event_type_full': 'process_event',
 'opcode': 1,
 'user_domain': 'NT AUTHORITY'}
```

x_1 = [

```
0,1,1,0,0,0,0,321451,0,0,0,
1,0.33,0.25,.75,0,0,1,0,1,
0,0,0,0,1,1,0,0,0,0,1,0.33,
0.25,.75,0,0,1,0,1,1,0,0,0,
0,1,1,0,0,0
```

]

Data Normalization & Feature Engineering

Filepath C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe

Data Normalization & Feature Engineering

Filepath	C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe
Username	is_system = {1, 0}

Data Normalization & Feature Engineering

Filepath	C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe
Username	is_system = {1, 0}
Timestamp (Delta)	'2018-08-15 01:14:44Z' == 131787692844718134

Data Normalization & Feature Engineering

Filepath	C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe
Username	is_system = {1, 0}
Timestamp (Delta)	'2018-08-15 01:14:44Z' == 131787692844718134
Process & Parent Process	WmiPrvSE.exe svchost.exe

Data Normalization & Feature Engineering

Filepath	C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe
Username	is_system = {1, 0}
Timestamp (Delta)	'2018-08-15 01:14:44Z' == 131787692844718134
Process & Parent Process	WmiPrvSE.exe svchost.exe

Data Normalization & Feature Engineering

Filepath C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe

Username is_system = {1, 0}

Timestamp (Delta) '2018-08-15 01:14:44Z' == 131787692844718134

Process & Parent Process WmiPrvSE.exe
svchost.exe

Command Line Arguments 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding'

Perform TF-IDF

0	0	.87	0	.25	0	.33	0	0	0.12	1	0

02

Detecting Anomalous Events.

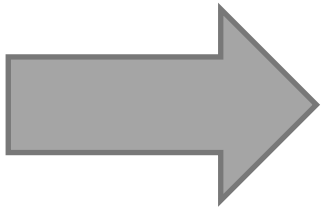
Supervised Learning Model

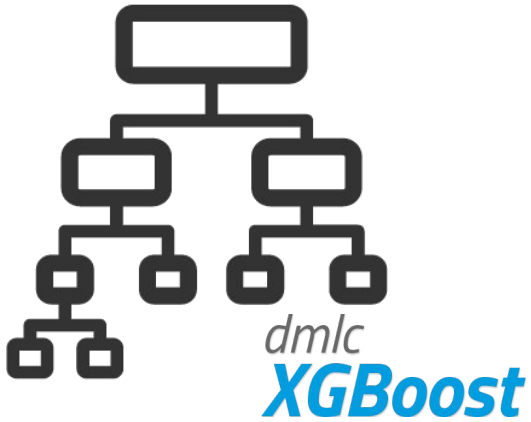
TRAINING DATA

X = [0,1,1,0,0,0,0,0.31,0,0,...,1
0,0,1,1,0,0,0,0.33,0.1,...,0
...]

Featurized Events

y = [0,1,1,0,0...,1] Labels





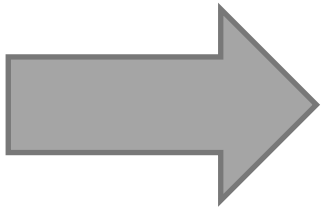
Supervised Learning Model

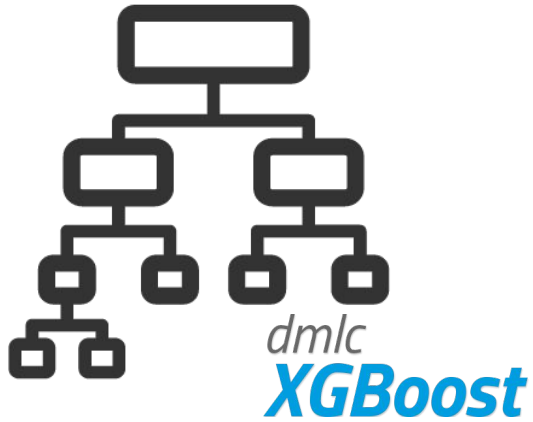
TRAINING DATA

X = [0,1,1,0,0,0,0,0.31,0,0,...,1
0,0,1,1,0,0,0,0.33,0.1,...,0
...]

Featurized Events

y = [0,1,1,0,0...,1] Labels





INPUT

'C:\Windows\system32\wbem\wmiprvse.exe -Embedding', 'SYSTEM', 1000700000, 'WmiPrvSE.exe', svchost.exe'

Feature Engineering

$\mathbf{X}_1$ = [0,1,1,0,0,0,0,0.31,0,0,...,1]

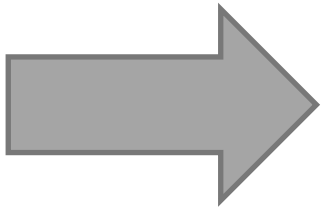
Supervised Learning Model

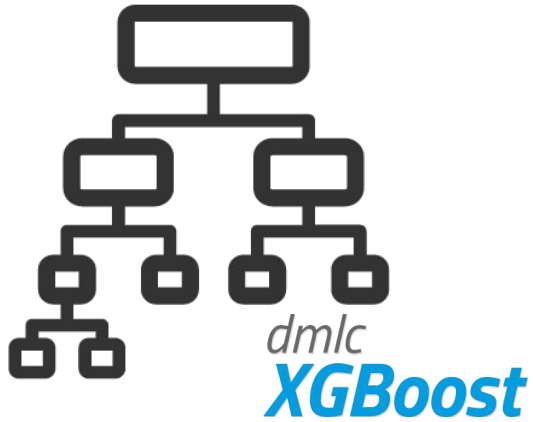
TRAINING DATA

X = [0,1,1,0,0,0,0,0.31,0,0,...,1
0,0,1,1,0,0,0,0.33,0.1,...,0
...]

Featurized Events

y = [0,1,1,0,0...,1] Labels

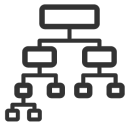




INPUT

'C:\Windows\system32\wbem\wmiprvse.exe -Embedding', 'SYSTEM', 1000700000, 'WmiPrvSE.exe', svchost.exe'

$\mathbf{X}_1$ = [0,1,1,0,0,0,0,0.31,0,0,...,1]



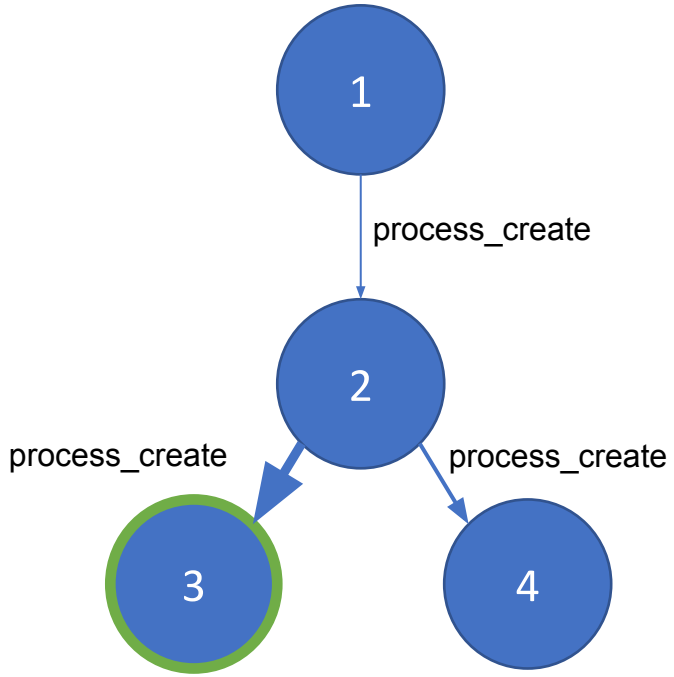
OUTPUT

$\mathbf{y}_1$ = 0.7561

0.7561 → how "malicious" the model thinks an event is
becomes an **edge weight** in our graph

Building a Graph

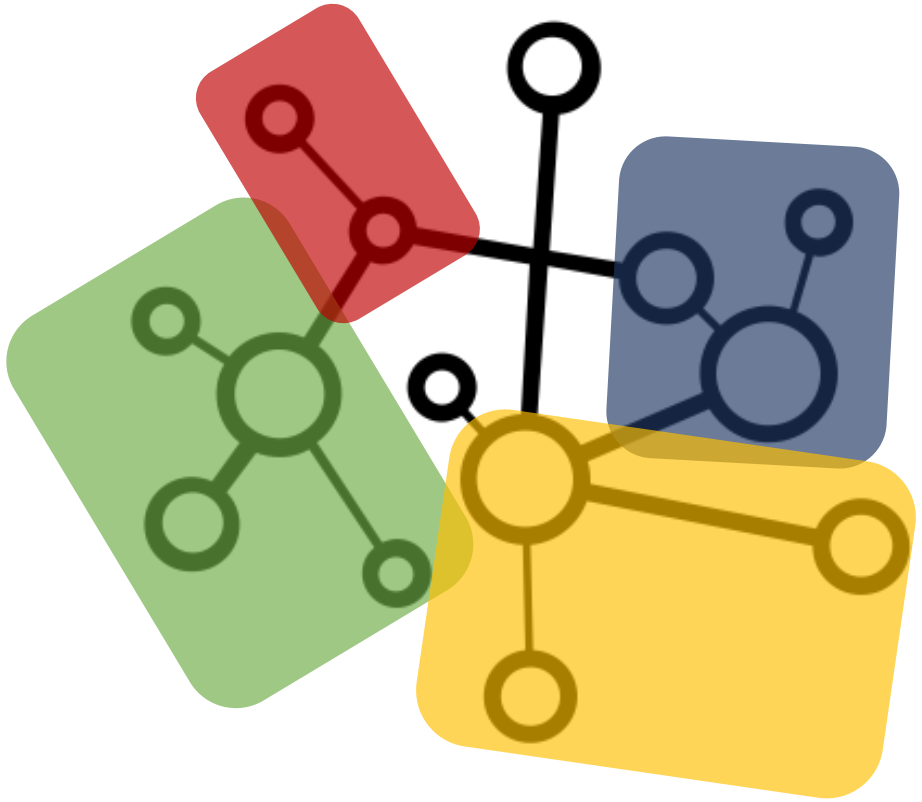
- Each event stored in a graphDB format
  - **Node** – process name or PID
  - **Edge** – *process_create* event
  - **Weight** – output from model
  - **Metadata** – cmd line args, timestamp




```
metadata = {
 'pid': 4856,
 'event_subtype_full': 'creation_event',
 'command_line': 'C:\\Windows\\system32\\wbem\\wmiprvse.exe
-Embedding',
 'ppid': 892,
 'sha256': 'c8533bb3b6088efb1d641b76fc7583c6bb7a...',
 'user_name': 'SYSTEM’,
 'timestamp': 131787692844718134,
 'process_name': 'WmiPrvSE.exe',
 'parent_process_name': 'svchost.exe',
 'weight': 0.7561
}
```

Community Detection

- We now have a weighted graph!
- Run Louvain community detection
- Segments a large graph into smaller “communities”
- Helps identify common attack patterns across datasets



03 Prevalence Service.

Process Prevalence Service

$$P(process)$$

normalize between **0-99**

"I've seen this process more than x% of other processes"

Parent-Child Prevalence Service

$$P(\text{child} \mid \text{parent})$$

normalize between **0-99**

"From this parent, I've seen this child more than x% of other child processes"

In Development: **Command Line Prevalence**

P(cmdline | process)

normalize between **0-99**

"From this process, I've seen this command line more than x% of other command lines associated w/ the process."

Prevalence Service – Use Cases

Answer Questions

- *"How rare is this parent-child relationship?"*
- *"Is it common to see this process with these siblings in a process tree?"*

Reduce/Rank Data

- Rank Communities based on Prevalence
- Suppress FPs by focusing on rarer patterns of behavior

04 Output.

1. Finding Bad Communities Pseudocode
2. Experiment Setup
3. Determining Ideal Threshold
4. APT3

Finding Bad Communities ~~Pseudocode~~ Python

Input: weighted graph *G, threshold*

Output: rank ordered list of bad communities

```
func find_bad_communities(G, threshold):

bad_communities = [ ]
communities  = community_detection(G, weight=weight)

for community in communities:
    for node in community:
        prevalence_score = prevalence_lookup[node.process_name]
        node['anomalous_score'] = node['weight'] * (1- prevalence_score)

    if max([node['anomalous_score'] for node in community]) >= threshold:
        bad_communities.append(community)

return sorted(bad_communities, reverse=True) # return most malicious communities first
```

Experiment(s) Setup

- **Overview**: Post-mortem red vs. blue team exercise.
- **Dataset**: ~500K process events from open source repo and internal testing
- **Results**
  - Yielded ideal threshold for “bad” communities
  - Evaluated performance against 2018 MITRE Evaluation Test (APT3)

ProblemChild Training Data

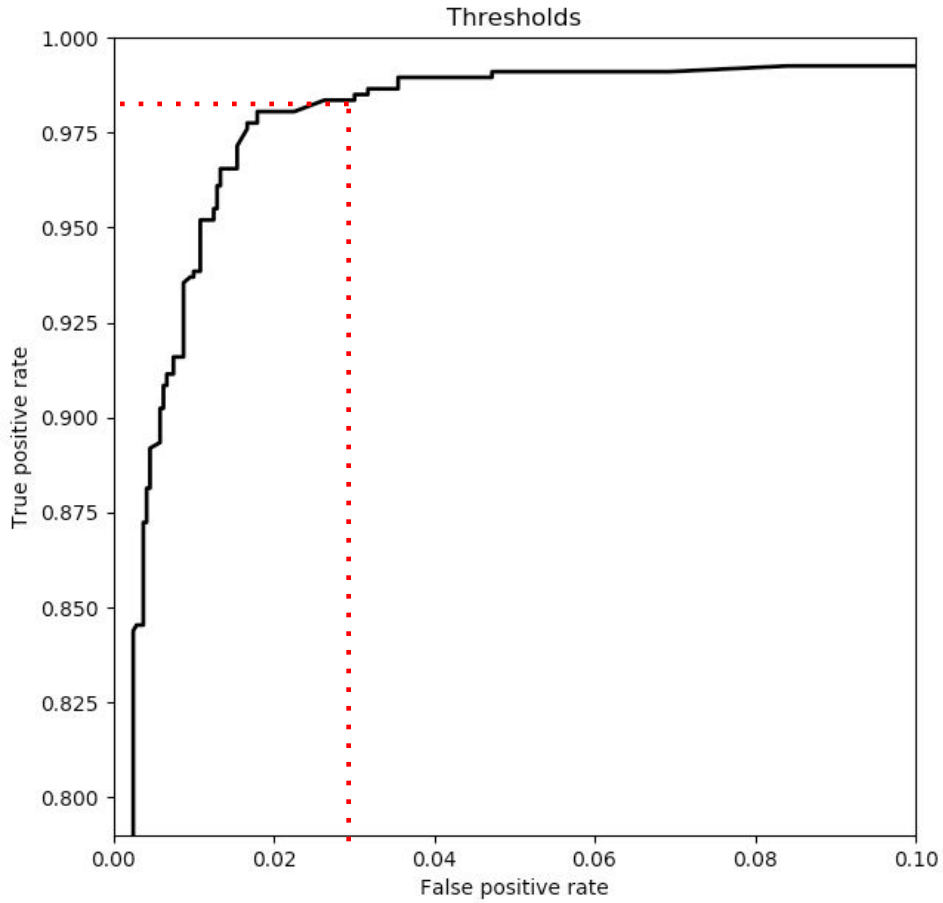
Open Source	*Closed Source*
Atomic Red Team	5 days clean Win Event Logs x 9
RTA	Executing 500 malware samples in a live environment*
Eqllib data	

**Various APT actors, Emotet, FIN7*

Leave-One-Out Retraining

- LOO was used to refine our “ideal” threshold for detecting malicious communities.
  - *Targeted a 3% FPR*

- FPs were anomalous PowerShell processes (e.g. *pushing updates, admin actions*), Rundll32 calling Rundll32



Threshold = 0.27 @ 3% FP Rate.

APT3 Emulation

Chinese-based threat actor associated with China’s Ministry of State Security.

2018 ATT&CK Evaluation emulated APT3 using FOSS & COTS tools

Unlike real world there is no user noise

Adversary activity is unrealistically loud

Data Reduction

- **~10K** process-related events per endpoint (***5*** *endpoints total*)
- ProblemChild identified **~6** malicious communities per endpoint
- Malicious communities consisted of **~4-6** process creation events

~10,000 process events reduced to **~36** events

APT3 Results: Initial Discovery

5724: {'score': **0.5600073383155507**,
 'chain': [('powershell.exe',
'"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -noP -sta -w 1 -enc <b64encoded_str>'),
('route.exe', '"C:\\Windows\\system32\\ROUTE.EXE" print’),
('ipconfig.exe', '"C:\\Windows\\system32\\ipconfig.exe" /all’),
('whoami.exe', '"C:\\Windows\\system32\\whoami.exe" /all /fo list’),
('qprocess.exe', '"C:\\Windows\\system32\\qprocess.exe" *’),
('netstat.exe', '"C:\\Windows\\system32\\NETSTAT.EXE" -ano’),
('net.exe', '"C:\\Windows\\system32\\net.exe" use’),
('reg.exe’,"C:\\Windows\\system32\\reg.exe" query HKLM\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System')]}

8611: {'score': **0.34580638056549406**,
 'chain': [
('cmd.exe','C:\\Windows\\system32\\cmd.exe /C
ipconfig /all & arp -a & echo
USERDOMAIN%\\%USERNAME% & tasklist /v & sc query &
net start & systeminfo &
net config workstation'),
('ipconfig.exe', 'ipconfig /all '),
('arp.exe', 'arp -a '),
('tasklist.exe', 'tasklist /v '),
('sc.exe', 'sc query '),
('systeminfo.exe', 'systeminfo ')]}

Techniques Used:

- PowerShell (T1086)
- Process Discovery (T1057)
- System Service Discovery (T1007)
- System Owner/User Discovery (T1033)
- System Network Connections Discovery (T1049)
- Query Registry (T1012)
- System Network Configuration Discovery (T1016)
- System Information Discovery (T1082)

APT3 Results: Lateral Movement

5864: {'score': **0.48416015079354735**,
'chain': [

```
('powershell.exe','powershell.exe -w 1 -enc <b64encoded_str>'),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\morris\\ADMIN$ /user:morris\\kmitnick <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:nimda\\kmitnick <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:shockwave\\bob <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:shockwave\\frieda <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\conficker\\ADMIN$ /user:conficker\\kmitnick <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\conficker\\ADMIN$ /delete'),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\creeper\\C$ <PASSWORD> /user:shockwave\\kmitnick')
```

]}

Techniques Used:

- PowerShell (T1086)
- Brute Force (T1110)
- Windows Admin Shares (T1077)
- Valid Accounts (T1078)
- Network Share Connection Removal (T1126)

APT3 Results: Persistence, Execution, Exfiltration

6050: {'score': **0.34641108500338824**,
'chain': [

```
('powershell.exe','"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -noP -sta -w 1 -enc <b64encoded_str>'),
('sc.exe', '"C:\\Windows\\system32\\sc.exe" \\\\creeper query'),
('sc.exe','"C:\\Windows\\system32\\sc.exe" \\\\creeper create AdobeUpdater binPath= "cmd.exe /c
\\"C:\\Users\\kmitnick\\AppData\\Roaming\\Adobe\\Flash Player\\update.vbs\\"" displayName= "Adobe Flash Updater" start= auto'),
('sc.exe','"C:\\Windows\\system32\\sc.exe" \\\\creeper description AdobeUpdater "SynchronizewithAdobeforsecurityupdates"'),
('sc.exe','"C:\\Windows\\system32\\sc.exe" \\\\creeper qc AdobeUpdater'),
('sc.exe','"C:\\Windows\\system32\\sc.exe" \\\\creeper start AdobeUpdater'),
('recycler.exe','"C:\\Windows\\recycler.exe" a -hpfGzq5yKw C:\\$Recycle.Bin\\old.rar
C:\\$Recycle.Bin\\Shockwave_network.vsdx'),
('ftp.exe', '"C:\\Windows\\system32\\ftp.exe" -v -s:ftp.txt')]}
```

Techniques Used:

- PowerShell (T1086)
- New Service (T1050)
- Masquerading (T1036)
- System Service Discovery (T1007)
- Service Execution (T1035)
- Exfiltration Over Alternative Protocol (T1048)
- Data Encrypted (T1022), Data Compressed (T1002)

The Importance of the Prevalence Service

Pre-Prevalence

```
4287: {'score': 0.7948077773153566,
'chain': [('svchost.exe','C:\\Windows\\system32\\svchost.exe -k wsappx -p -s AppXSvc'),
  ('rundll32.exe','rundll32.exe AppXDeploymentExtensions.OneCore.dll,ShellRefresh'),
  ('rundll32.exe','rundll32.exe AppXDeploymentExtensions.OneCore.dll,ShellRefresh'),
  ('rundll32.exe','rundll32.exe AppXDeploymentExtensions.OneCore.dll,ShellRefresh')]},
```

Post-Prevalence

```
4287: {'score': 0.0440438385836174,
'chain': [('svchost.exe','C:\\Windows\\system32\\svchost.exe -k wsappx -p -s AppXSvc'),
  ('rundll32.exe','rundll32.exe AppXDeploymentExtensions.OneCore.dll,ShellRefresh'),
  ('rundll32.exe','rundll32.exe AppXDeploymentExtensions.OneCore.dll,ShellRefresh'),
  ('rundll32.exe','rundll32.exe AppXDeploymentExtensions.OneCore.dll,ShellRefresh')]},
```

What's Next?

Can we turn this...

5864: {'score': **0.48416015079354735**,
'chain': [
('powershell.exe','powershell.exe -w 1 -enc <b64encoded_str>'),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\morris\\ADMIN$ /user:morris\\kmitnick <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:nimda\\kmitnick <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:shockwave\\bob <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:shockwave\\frieda <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\conficker\\ADMIN$ /user:conficker\\kmitnick <PASSWORD>),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\conficker\\ADMIN$ /delete'),
('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\creeper\\C$ <PASSWORD> /user:shockwave\\kmitnick')
]}

What's Next? Auto-Rule Generation

Can we turn this...

```
5864: {'score': 0.48416015079354735,
  'chain': [
     ('powershell.exe','powershell.exe -w 1 -enc <b64encoded_str>'),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\morris\\ADMIN$ /user:morris\\kmitnick <PASSWORD>),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:nimda\\kmitnick <PASSWORD>),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:shockwave\\bob <PASSWORD>),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\nimda\\ADMIN$ /user:shockwave\\frieda <PASSWORD>),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\conficker\\ADMIN$ /user:conficker\\kmitnick <PASSWORD>),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\conficker\\ADMIN$ /delete'),
     ('net.exe','"C:\\Windows\\system32\\net.exe" use \\\\creeper\\C$ <PASSWORD> /user:shockwave\\kmitnick')
]}
```

Into this?

```
process where subtype.create and
  (process_name == "net.exe" or (process_name == "net1.exe" and parent_process_name != "net.exe"))
   and (command_line == "* use" or command_line == "* use *") and command_line == "* \\\\*"
  | unique parent_process_path, command_line, user_name
```

EQL

Challenges

- **Data**
  - Getting labeled data is **hard**
  - Getting realistic data is **harder**
  - Getting diverse data is **hardest**
- **Labels**
  - Good data **!=** good labels
- **Event types**
  - Additional events require additional feature engineering

ENDGAME.

PROBLEMCHILD

Questions?

Bobby Filar
@filar

References

- RTA: https://github.com/endgameinc/RTA/tree/master/red_ttp
- EQL: https://github.com/endgameinc/eqllib
- Atomic Red Team: https://github.com/redcanaryco/atomic-red-team
- Mitre ATT&CK: https://attack.mitre.org
- Metta: https://github.com/uber-common/metta