FIRSTCON25

Detection Engineering 101

Establishing a Structured Approach to Detection Engineering

2025.6.26 Tomohisa Ishikawa, Ph.D., CISSP, CSSLP, CCSP, CISA, CISM, PMP <u>scientia.admin@gmail.com</u> @scientia_sec



Practical Threat Detection Engineering

A hands-on guide to planning, developing, and validating detection capabilities

MEGAN RODDIE JASON DEYALSINGH | GARY J. KATZ



Tomo

Name :

Affiliation:

Certifications:

Expertise Area:

Tomo (Tomohisa Ishikawa)

Distinguished Cyber Security Architect @ Tokio Marine HD

Ph·D·, CISSP, CSSLP, CCSP, CISA, CISM, PMP etc.

Ex-Red Teamer and Current Blue Teamer

• Security Strategy, Security Architecture, Cyber Threat Intelligence, Security Operation...

- Member of National IT Exam Committee
- Cybersecurity Experts in Gov Agency (part-time)

External Activity:

- Speaker @ DEFCON24 SE Village, LASCON 2016, BSide Philadelphia 2016, FIRSTCON23, GCC 2024-2025...
- Author of book related to Cyber Thereat Intelligence
- Translate 7 cybersecurity books into Japanese

\$ whoam

Today's Goal

Today's Goal : Sharing structured approach of Detection Engineering & Understand how to operationalize Detection Engineering Process

Agenda :

- Part I: Defining Detection Engineering
- Part II: Detection Engineering Process Deep Dive
- Part III: "Successful" Detection Engineering Program

Part I : Defining Detection Engineering

What is Detection Engineering?

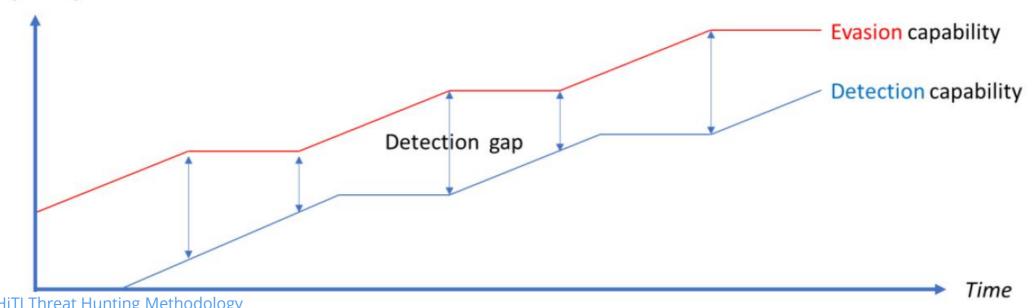
- Many definition is available, but definition from SentinelOne says:
 - "Detection engineering is a structured approach to developing, optimizing, and managing rules, alarms, and processes to detect threats or suspicious activity in real-time"

Propose Another Definition of "Detection Engineering"

- "Systematic approach to mind Detection Gap"

- **Detection Gap :** •
 - "Cat & Mouse Game" btw "Threat Actor" and "Blue Team"

< What is Detection Gap?>



Capability

- Three way to "Identify" Detection Gap
 - Cyber Threat Intel : Identify latest TTPs
 - Purple Teaming

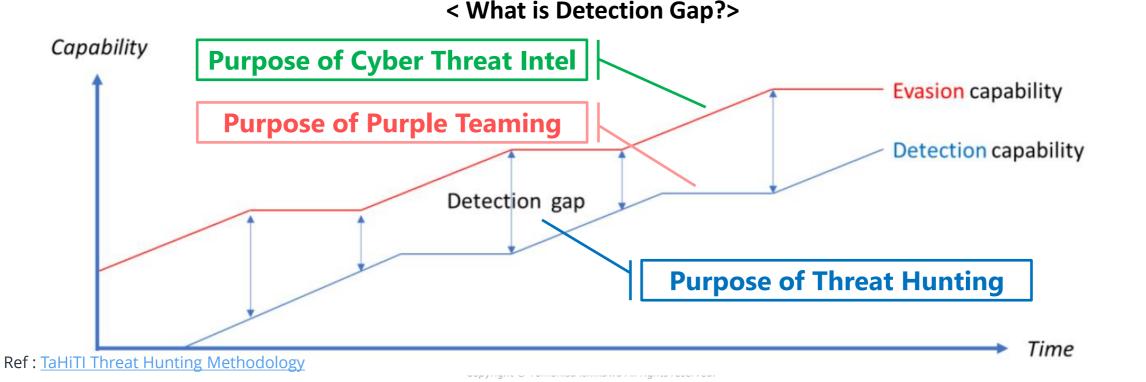
: Identify existing Prevention & Detection capability

– Threat Hunting

: Identify unknown threat evading existing security mechanisms

7

• Detection Engineering is to "Mind" Detection Gap



- Three Characteristics of "Detection Engineering"
 - Proposed by Forrester Principal Analyst (Allie Mellen)

- #1 : Detection as Code

• Use and Manage detection rule as code (YARA, SIGMA etc.)

- #2 : Applying Software Engineering

- Structured Approach = Software Engineering
- "The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software" (definition of Software Engineering by IEEE)
- Use metrics for evaluating and managing detection engineering

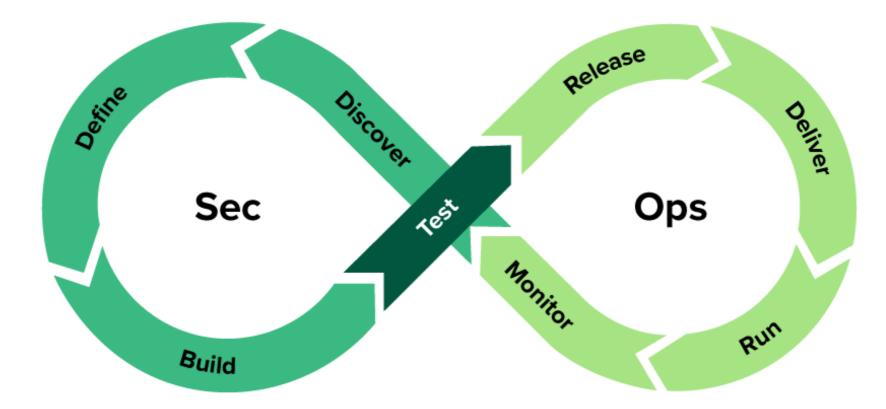
- #3 : "Agile" Approach

• Start Small, and Continuous Improvement

Ref: <u>https://www.forrester.com/blogs/announcing-the-detection-and-response-development-lifecycle-dr-dlc-for-detection-engineering/</u>

DR-DLC : Detection & Response Development Life Cycle

• Originally proposed by Forrester Principal Analyst (Allie Mellen)



https://www.forrester.com/blogs/announcing-the-detection-and-response-development-lifecycle-dr-dlc-for-detection-engineering/

Part II : Detection Engineering Process Deep Dive

DR-DLC : Detection & Response Development Life Cycle



< Overview of Each Phase>

Step	Description
Discover	Identify detection rules that need to be created or improved.
Define	Define the requirements for detection rules.
Build	Create detection rules that meet the defined requirements.
Test	Test whether the detection rules function as expected.
Release	Review and approve the created detection rules.
Deliver	Document, manage, and store the detection rules in the appropriate repository
Run	Deploy and execute the detection rules in the production environment.
Monitor	Manage the accuracy and quality of detections based on KPIs and functional metrics.



- Identify detection rules that need to be created or improved
- 4 Major Request Path

< 4 Major Request Path in Discover Phase >

Request from…	Request Overview
CTI Analyst	 By using CTI input for encountering new threat
SOC Analyst	 SOC analyst is working on the cyber attack frontier, and their insight will be helpful SOC analyst also request DE based on the results of purple teaming and threat hunting
Security Policy	 Business and security policy require additional control for securing the environments Ex) prohibiting specific services
"Monitoring phase"	 "Monitoring" phase is evaluation phase of detection rule quality



Phase 2 – Define

- Based on "Discover" phase, we will define/describe the requirement for detection rules
- Two activities is required.
 - Step 2-1 : Detection Requirement Definition
 - Step 2-2 : Triage (prioritization)



- 2-1 : Detection Requirement Definition
 - Define the requirement for creating "detection rule"

< RESCUE framework for Detection Requirement Definitions >

Components	Details
Requester	Person to request to create detection rule (Need to satisfy requester's expectation)
Evidence	Provide CTI and log data that form the basis for creating detection rules
Scope	Define the scope for detection rules (target scope and time period to be applied)
Contents	Define the technical details of detection rule
Utility / Urgency	The reason that this specific detection rule needs to be created (used for deciding priority)
Exception	Describe exceptions of detection rule for false positives prevention



• 2-2 : Triage (Prioritization)

- Based on defined "requirement", we decide the priority
- Triage criteria in addition to "Utility / Urgency" of RESCUE framework is as follows.

< Triage Criteria >

Viewpoint	Details
Severity	Does this requirement bring a significant impact if detection rules are not created?
Consistency	Does the detection requirement align organization profile with the attack targets, intent, and capabilities of each threat actors?
Coverage	Does the requirement enhance the organization's detection coverage based on identified "detection gaps" from purple teaming or threat hunting?



- After "Define" phase, we will build "detection code" that meet the defined requirements.
- Two activities is required.
 - Step 3-1 : Design
 - Step 3-2 : Development



- : What kind of detection logic are required?
- Conditions

– Logic

- : What kind of conditions are required to avoid "false positive"?
- (B) Describe "Technical Specification"
 - Use H.O.P.E. framework for concise "Technical Specification"

– (C) Create Validation Criteria

• Prepare test data and validation criteria for "Test" phase



H.O.P.E. framework can help to build "Technical Specification"

< H.O.P.E framework for Technical Specification>

Components	Examples				
Hypothesis	"Threat actor create suspicious Domain Account"				
Object of Investigation	Windows Event Log in Domain Controller Server				
Procedure	Filter by Event ID (4720) and search log entries of account creation when helpdesk is closed				
Evaluation Criteria	If expected entry is found, code generate alerts				



- 3-2 : Development
 - Based on "design", we will develop actual "detection rule".
 - SIGMA
 - YARA



• Based on Detection Code created in "Build" phase, we test whether the detection rules function works as expected.

The purpose of "Test" phase

 To ensure that the detection code is implemented appropriately by aligning with the requirements definition and returns the expected behavior and results

Use Two Types of Test Data

- Known Good : Confirming no false positive
- Known Bad : Confirming no false negative



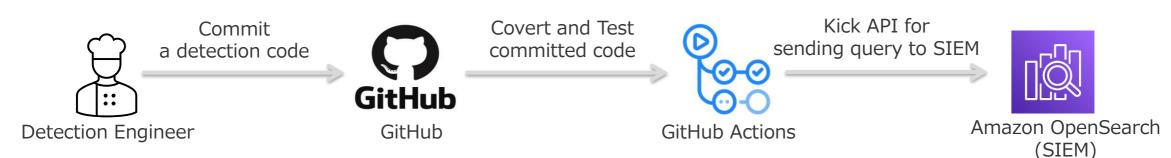
• Review and approve the created detection rules.

Phase 6 – Deliver

• Document, manage, and store the detection rules in the appropriate repository

For release and deploy management, Detection Engineering best practice typically recommends to use CI/CD pipeline.

< Sample workflow of CI/CD pipeline for Detection Engineering>





Phase 7 – Run

- Deploy and execute the detection rules in the production environment.
- Use "Deployment Tag" on sharing confidence level of detection rule to manage the commitment level of SOC analyst.

< Deployment Tag >

Viewpoint	Details
Experimental	Detection rules that have not undergone extensive testing in a live environment. It is desirable to limit the scope of application and restrict alert reviews to only the rule creator, excluding other SOC analysts.
Testing	Detection rules that have progressed from the "Experiment" phase and are now available for broader use. From the perspective of detection quality, all SOC analysts review alerts at a lower priority compared to "Stable" rules.
Stable	Rules that have been deemed stable and are fully deployed in the production environment. SOC analysts review these rules with high priority. If false positives or improvements are needed after deployment, consultation with a Detection Engineer is required.



- Manage the accuracy and quality of detections based on KPIs and functional metrics.
- Use 3M+C framework originally proposed by "Practical Threat Detection Engineering"

Metrics	 Evaluate "detection rules" performance from statistical KPI
Monitoring	 Continuously analyze actual "alerts" from detection rule to avoid false positive
Maintenance	 Update "detection rules" based on updated threat intelligence and additional insight from security operation including threat hunting
Continuous Validation	 Conduct purple teaming continuously to identify "detection gap"

< 3M +C Framework for "Monitor" phase>



- Deep Dive : "Metrics"
 - Evaluate "detection rules" performance from statistical KPI (Top-Down approach)

< Example of "Metrics" alerts >

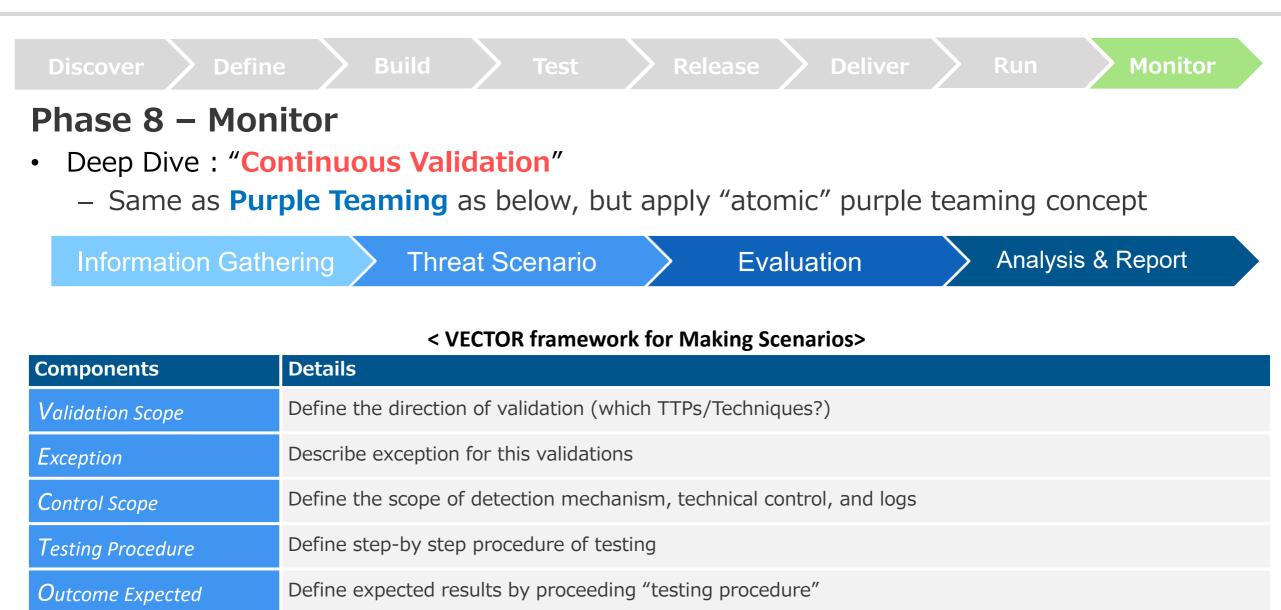
Metrics	Meaning
Number of Detection Alerts	The number of detection alerts that need to be reviewed by analysts.
Analysis Results	The number of cases where analysts reviewed alerts and determined whether they were false positives.
Average Time to Close	The average time required to analyze a alert and close it.
Standard Deviation of Time to Close	Time deviation to analyze and close alerts.
Change in Time to Close	The difference in the average time to close before and after updating detection rules.



- Continuously analyze actual "alerts" from detection rule (Bottom-Up approach)
- Even we have various validation in test phase, we might have "false positive"
- Ex) "nc" command detection => "rsync" or "sync"
- Ex) "Emotet" detection of Excel Macro execute "powershell.exe"



- "Agile" Approach "Start Small, and Continuous Improvement"
- Various Approach are available:
 - Use External Threat Intelligence
 - Analyze Similar Malware (i.e. VirusTotal Commercial License)
 - Utilize Deception Techniques for Detections



Review Criteria Define the review criteria whether or not existing "detection code" correctly works.

Part III: "Successful" Detection Engineering Program

What is the "Successful" Detection Engineering Program

– 3 critical KPI will define what is "successful" in Detection Engineering Program

Metrics Viewpoint		Overview			
MTTD + MTTR	Time	Explain the resilience capability from time-basis			
Precision & Recall	Efficiency	Explain the "false positive" ratio and "false negative" ratio as the efficiency of detection			
Detection Coverage	Coverage	Covering entire MITRE ATT&CK			

< 3 Critical KPI for "successful" detection engineering program>

KPI #1 : MTTD + MTTR

- Detection Engineering shorten MTTD + MTTR
 - MTTD (Mean Time To Detect)
 - MTTR (Mean Time To Response)
- : Average time between attack start and have detected
- : Average time between detection and response

• Background Theory : Time-Based Security (by Winn Schwartau)

- System are secured if it satisfy following equation
- MTTA > MTTD + MTTR
 - MTTA (Mean Time To Attack)
- : Average time between attack start and end

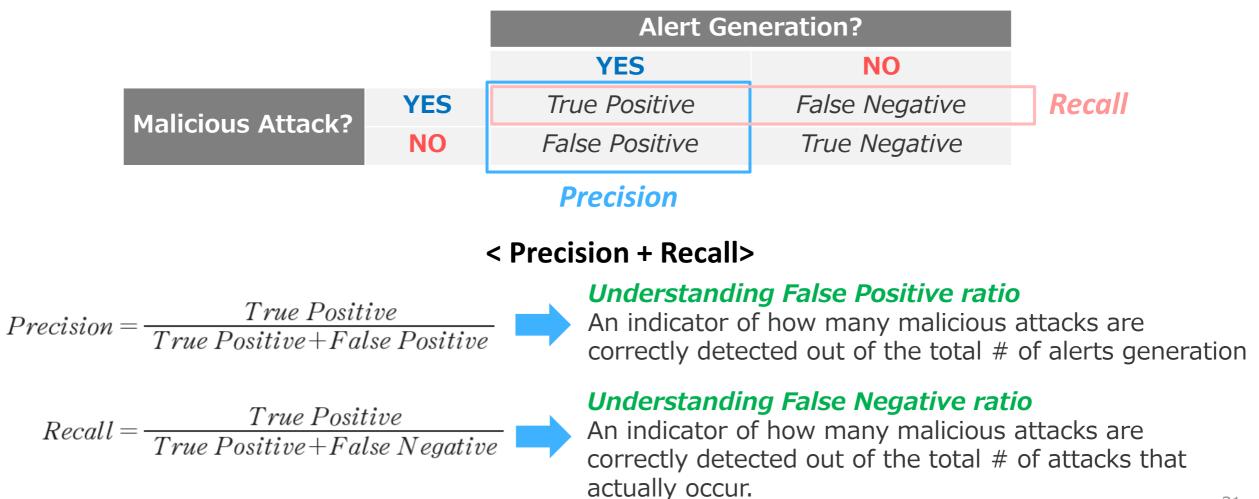


"Successful" Detection Engineering Program

KPI #2 : Precision + Recall

• Precision + Recall is defined as follows.

< Confusion Matrix >



KPI #3 : Detection Coverage

• Detection Coverage is as follows.

DataCoverage ×	†						ion controls	ayer controls B, ± 🏾 O =,		technique controls
nitial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	DataCoverage		Command And Control
10 items	33 items	58 items	28 items	63 items	19 items	20 items	17 items	description		21 items
vrive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	2019-03-23	Exfiltration	Commonly Used Port
ploit Public-Facing	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window	Application Deployment	Metadata	inessed	Communication Through
pplication	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Discovery	Software		pted	Removable Media
ardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Browser Bookmark Discovery	Distributed Componen Object Model	add more metadata	ier Size Limits	Connection Proxy
eplication Through emovable Media	Control Panel Items	AppInit DLLs	Application Shimming	Clear Command History	Credentials in Files	File and Directory Discovery	Exploitation of Remote		Cover Alternative	Custom Command and Contr Protocol
earphishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account	CMSTP	Credentials in Registry	Network Service Scanning	Services		Protocol	Custom Cryptographic
pearphishing Link	Execution through API	Authentication Package	Control	Code Signing	Exploitation for Credential	T1214irk Share Discovery Score: 2500	Logon Scripts	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Protocol
pearphishing via Service	Execution through Module	BITS Jobs	DLL Search Order Hijacking	Compiled HTML File	Access	Metadata: ""fing	Pass the Hash	Data from Removable	Exfiltration Over Other	Data Encoding
upply Chain Compromise	Load	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Windows Registry:Windows:4657: Score:	Pass the Ticket	Media	Network Medium	Data Obfuscation
rusted Relationship	Exploitation for Client Execution	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Pro pheral Device Discovery Windows Registry:Sysmon:12:	Remote Desktop Protoc	ol Data Staged	Exfiltration Over Physical	Domain Fronting
alid Accounts	Graphical User Interface	Change Default File	Extra Window Memory	Control Panel Items	Input Capture	Score: 285	Remote File Copy	Email Collection	Medium	Fallback Channels
and Accounts	InstallUtil	Association	Injection	DCShadow	Input Prompt	Windows Registry:Sysmon:13: Score: 285	Remote Services	Input Capture	Scheduled Transfer	Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions	Deobfuscate/Decode Files or	Kerberoasting	Windows Registry:Sysmon:14: Score: 285	Replication Through	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model	Weakness	Information	Keychain	Process command-line	Removable Media	Screen Capture		Multiband Communication
	LSASS Driver	Hijacking	Hooking	Disabling Security Tools	LLMNR/NBT-NS Poisoning	parameters:Windows:4688: Score: 58	Shared Webroot	Video Capture		Multilayer Encryption
	Mshta	Create Account	Image File Execution	DLL Search Order Hijacking	Network Sniffing	Process command-line	SSH Hijacking	÷2		Port Knocking
	PowerShell	DLL Search Order Hijacking	Options Injection	DLL Side-Loading	Password Filter DLL	 parameters:Sysmon:1: Score: 290 	Taint Shared Content			Remote Access Tools
	Regsvcs/Regasm	Dylib Hijacking	Launch Daemon	Exploitation for Defense Evasion	Private Keys	Process command-line parameters:Windows:4688:	Third-party Software			Remote File Copy
	Regsvr32	External Remote Services	New Service	Extra Window Memory Injection	Securityd Memory	Score: 58	Windows Admin Shares			Standard Application Layer
	Rundll32	File System Permissions Weakness	Path Interception	File Deletion	Two-Factor Authentication	Process monitoring:Windows:4688;	Windows Remote			Protocol
	Scheduled Task	Hidden Files and Directories	Plist Modification	File Permissions Modification	Interception	S Score: 59/mer/User	Management			Standard Cryptographic
	Scripting	Hooking	Port Monitors	File System Logical Offsets		Process monitoring:Windows:4689:			~	legend
	Service Execution	Hypervisor	Process Injection	Gatekeeper Bypass		Score: 295 Process monitoring:Sysmon:1:			a correct	Low Coverage
	Signed Binary Proxy Execution	Image File Execution Options	Scheduled Task	Hidden Files and Directories		Score: 295			#11111	Low Coverage
	Signed Script Proxy Execution	Injection	Service Registry Permissions Weakness	Hidden Users		Process monitoring:Sysmon:5: Score: 295			#4dd2fb	Medium Coverage
	Source	Kernel Modules and Extension	Setuid and Setgid	Hidden Window		Process monitoring:Sysmon:8: Score: 295				
	Space after Filename	Launch Agent	SID-History Injection	HISTCONTROL		Process monitoring:Windows			#0c1b33	Pretty Good Coverage
	Third-party Software	Launch Daemon	Startup Items	Image File Execution Options		Scheduled Tasks:100-200: Score: 0				dd Item Clear
	Trap	Launchctl	Sudo	Injection		Process monitoring:Windows Whitelist:8000-8027: Score: 0			1	Add Item Clear
	Trusted Developer Utilities	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking		Wintelist.0000-0027. Scole. 0				
	User Execution	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Windows Management	Login Item	Web Shell	Indicator Removal on Host						
	Instrumentation	Logon Scripts	web shell	Indirect Command Execution						

Source : https://twitter.com/olafhartong/status/1109569799863091201

"Successful" Detection Engineering Program

- "3M + C framework" can improve "3 Critical KPIs".
- "3M + C framework" makes Detection Engineering Program "successful".

< 3M + C framework & 3 Critical KPIs>

	Metrics	Monitoring	Maintenance	Continuous Validation
Evaluation Viewpoint	Top Down	Bottom-Up	Bottom-Up	Top Down
Approach	Reactive	Reactive	Proactive	Proactive
Evidence	Quantitative	Qualitative	Qualitative	Quantitative
↑ MTTD + MTTR		-	-	
↑ Precision + Recall	Precision	Precision	Recall	Recall
↑ Detection Coverage	-	-		

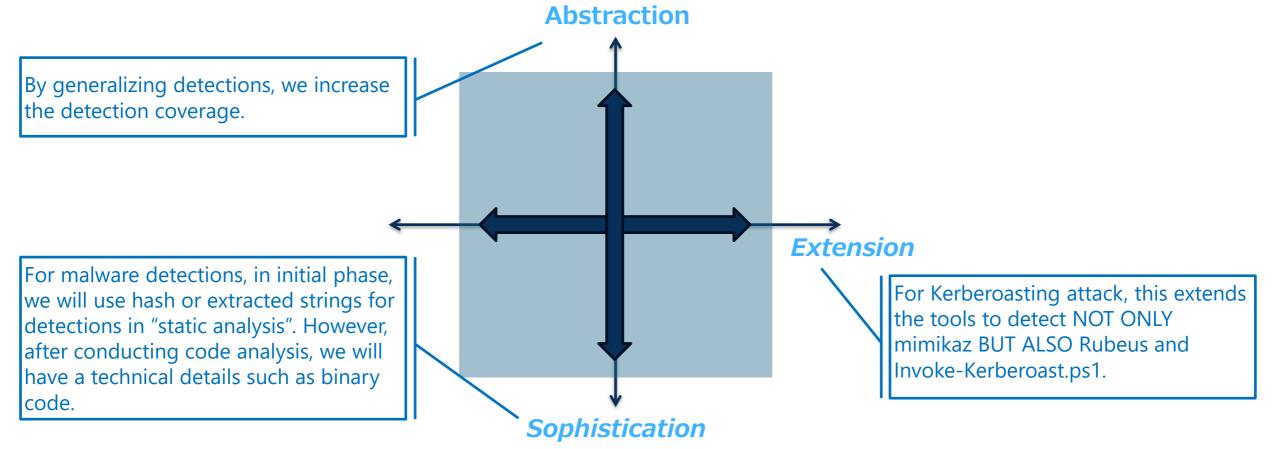
How to Expand the Detection Capability

Extension

: Extend the detection capability to similar attack methods or tools

- Sophistication
- Abstractions

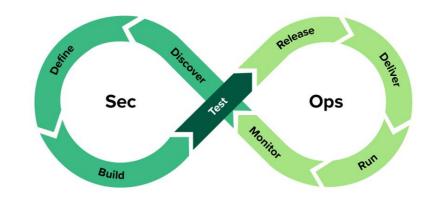
- : Deep-dive into technical details for further detections
- : Generalize detections (Follow the Idea of MITRE "Summiting The Pyramid")



Wrap-Up

Wrap-Up

- Part I: Defining Detection Engineering
 - "Systematic approach to mind Detection Gap"
 - Three Characteristics of Detection Engineering
 - Detection Engineering Process (DR-DLC Model)



• Part II: Detection Engineering Process Deep Dive

- Detailed Explanations of DR-DLC Process
- Sharin several framework/idea for structured approach such as RESCUE, HOPE, 3M+C, VECTOR

• Part III: "Successful" Detection Engineering Program

- High-Level KPIs for "Successful" Detection Engineering Program
- Discuss the relationship between 3 Metrics and 3M+C "Monitor" framework
- 3 Strategies for further improvement

Thank You!

@scientia_sec



 \square

https://www.linkedin.com/in/tomohisaishikawa/

scientia.admin@gmail.com