



# OpenTIDE

*Threat-informed Detection Engineering  
management framework*

## When TI made actionable drives your Threat Detection



**Amine BESSON**  
*(Behemoth Cyberdefence, NL)*



**Remi SEGUY**  
*(EC DIGIT S2, LU)*



**Claus HOUMANN**  
*(EC DIGIT S1, LU)*



# Core Concepts

## Threat + Detection Modelling

- Structures how intelligence and other input should be processed
- Creates an ever-growing knowledge graph to support decision making
- Clarity to detection posture and prioritization
- As-code, YAML, Schema, LLM-ready

## Detection-as-Code

- Multi-system state of the art framework
- Multi-tenant ready
- Dynamic modifiers
- Staging and Production Workflows
- Powerful schemas and tooling
- Near or Full GUI Parity

Common DevOps Workflows, Threat Driven, Repeatable, Measurable

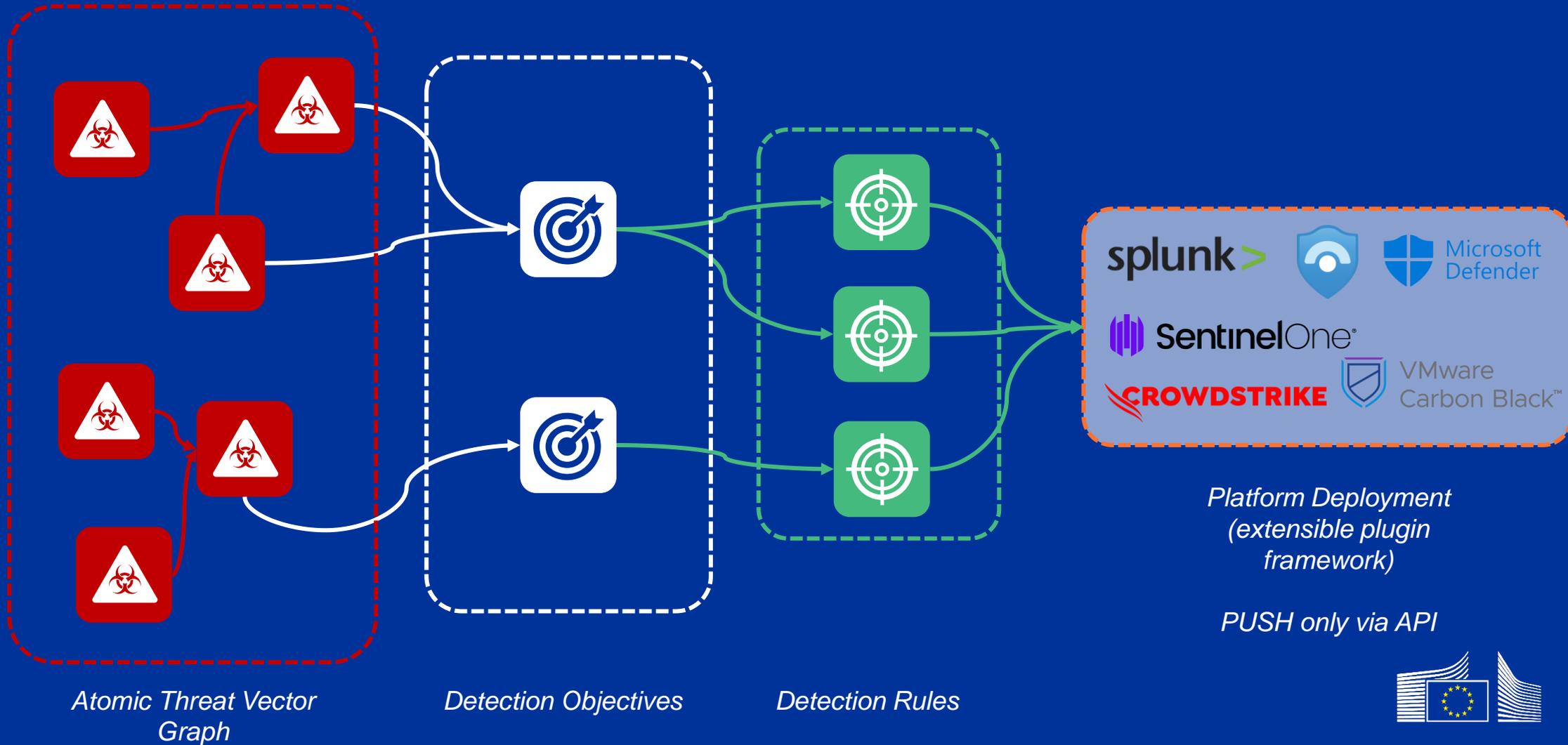


# OpenTide Objects and Graph

Standard Schemas and CI/CD Automation

Unstructured Threat Intelligence, Red Teaming Reports, Lessons Learned, Risk

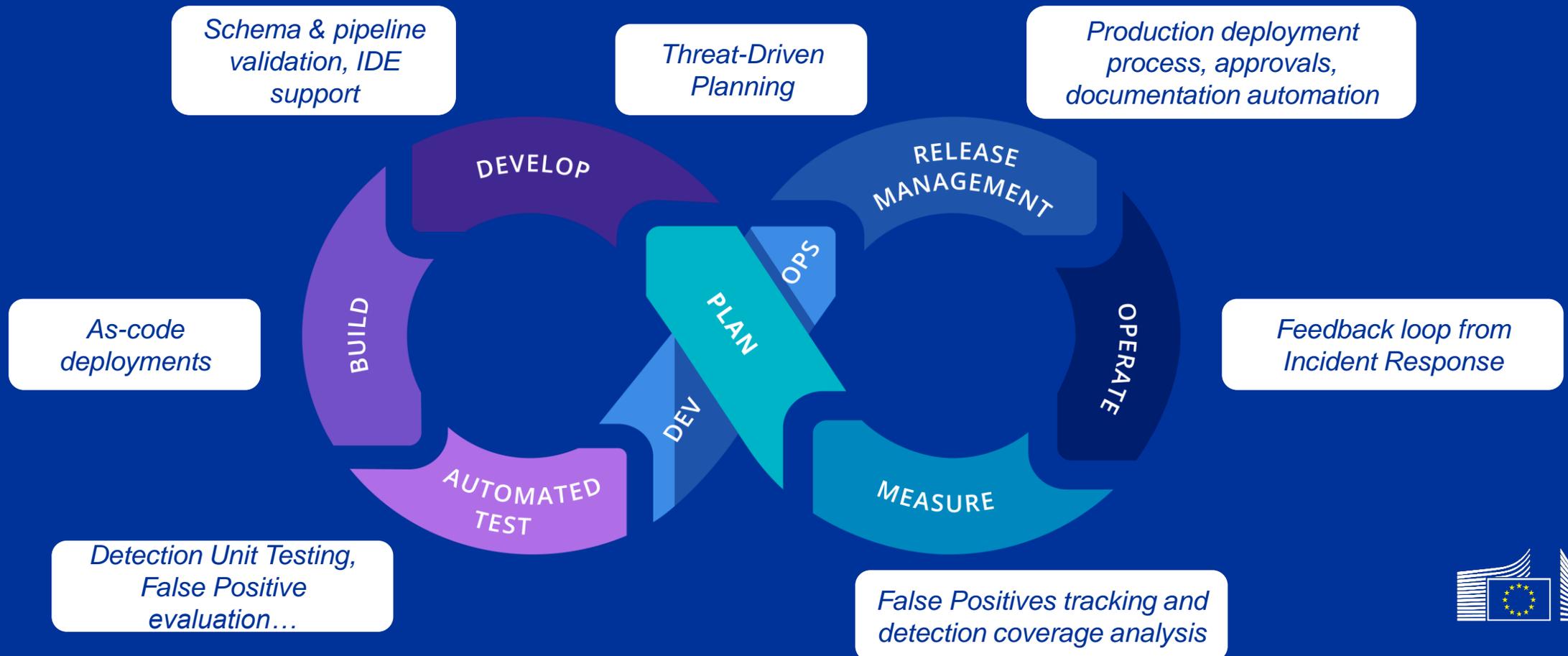
Analysis...



# DetectionOps

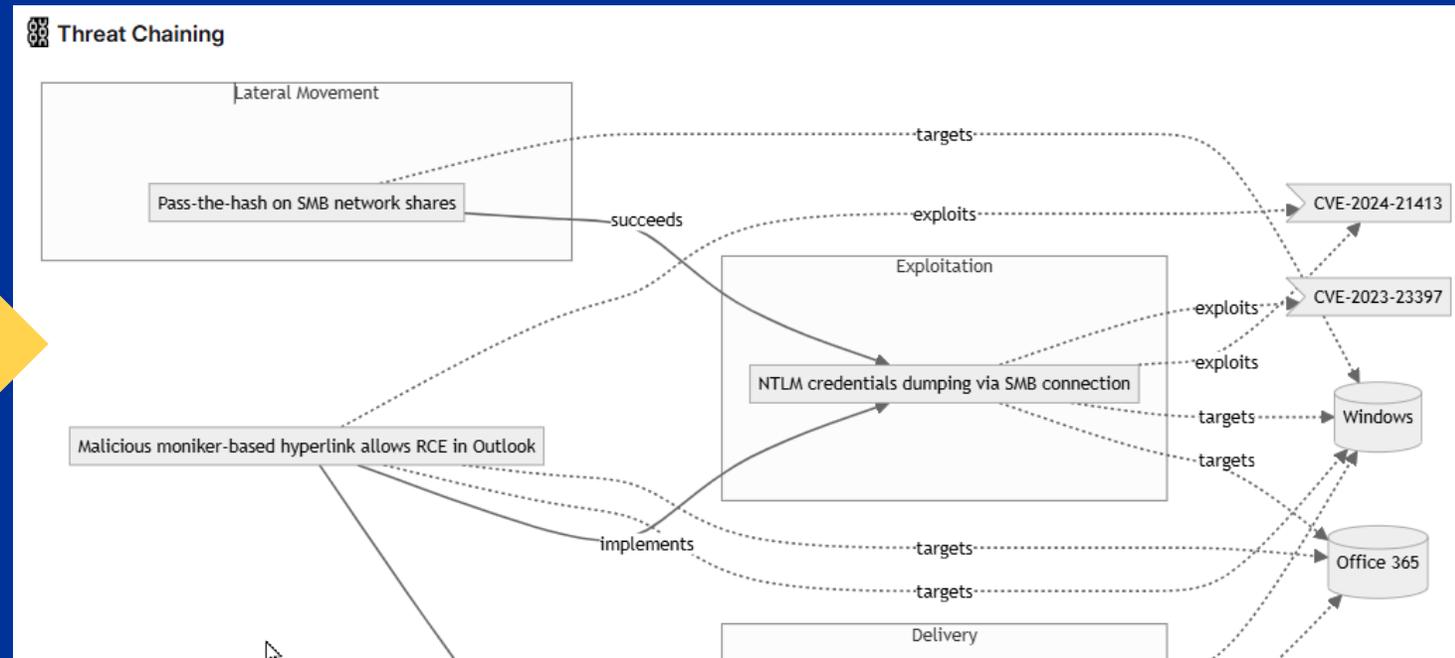
Thesis : we can map out all Detection Engineering tasks on a DevOps-like loop, and leverage existing tools, know-how and approaches across all stages – but we need a framework.

**OpenTide supports the end-to-end Detection lifecycle by structuring the DetectionOps workflow**



# Threat Vector Chaining = build dynamic attack path without knowing the full shape of the graph

```
att&ck:  
- T1566.002 #Phishing: Spearphishing Link  
- T1189 #Drive-by Compromise  
- T1557.001 #Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Re  
- T1204.002 #User Execution: Malicious File  
chaining:  
- relation: sequence::succeeds  
vector: 1a68b5eb-0112f424d-a21f-88dda0b6b8df # Spearphishing Link  
description: |  
Threat actor uses crafted e-mails with embedded hyperlinks  
in attempt to trick the user to click on them and grant  
access bypassing the protected view mechanism of  
Microsoft Outlook.  
- relation: atomicity::implements  
vector: 02311e3e-b7b8-4369-9e1e-74c8a844ae0f # NTLM credentials dumpi  
description: |  
In an initially prepared campaign, a threat actor can steal NTLM  
hashes, using Moniker-based hyperlink which points to SMB share.  
cve:  
- CVE-2024-21413 # Successful exploitation of this vulnerability would
```



# Detection Coverage Knowledge Graph

- Next-generation detection coverage evaluation
- Move away from static “taxonomies” and move at the speed of threats
- Full of metadata to slice and dice – e.g. what is my coverage for Evasion type of TTPs in Azure ?
- The Detection Engineering backlog is now a constant, continuously changing picture but accurate against threats we know of



Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection
45 techniques	17 techniques	33 techniques	9 techniques	17 techniques
Abuse Elevation Control Mechanism (3/6)	Adversary-in-the-Middle (2/4)	Account Discovery (4/4)	Exploitation of Remote Services	Adversary-in-the-Middle (2/4)
Access Token Manipulation (4/5)	Brute Force (3/4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (1/3)
BITS Jobs	Credentials from Password Stores (4/6)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture
Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (0/2)	Automated Collection
Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (5/8)	Browser Session Hijacking
Deobfuscate/Decode Files or Information	Forge Web Credentials (1/2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data
Deploy Container	Input Capture (2/4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage
Direct Volume Access	Modify Authentication Process (2/9)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (0/2)
Domain or Tenant Policy Modification (2/2)	Multi-Factor Authentication	Debugger Evasion	Use Alternate Authentication	Data from Information
Email Spoofing		Device Driver Discovery		
Execution Guardrails (0/2)		Domain Trust Discovery		
Exploitation for Defense Evasion				

# What about ATT&CK then ?

- ATT&CK is still a great way to identify if there's any gaps we don't know about
- All Threat Vector Objects map on ATT&CK
- By extension, every related artefact (down to the rule) has a connection to ATT&CK
- OpenTide leverages always the latest version of ATT&CK, and updates its dependencies frequently (no deprecated versions like in other tools or projects)



# Best in class tooling to draft YAML objects

OpenTide goes very far with pushing JSON Schemas to create an IDE-style experience

- Auto-completion
- Continuous validation
- Self-indexation to reference other objects as they get merged
- Templates and snippets that evolve with schema evolution
- For users : once familiar, from branch to PR in less than a minute

```
detection_model: entrata
response:
  alert_severity:
  #playbook: https://
  #responders:
  #procedure:
    #analysis: |
    #...
    #searches:
      #- purpose: |
        #...
        #system:
        #query: |
        #...
    #containment: |
    #...

configurations:
  #carbon_black_cloud:
  #sentinel:
  splunk:
    schema: splunk:2.1
    status: DEVELOPMENT
  #contributors:
  #-
  #threshold: 0
  #threatling:
```

**Entrata ID - Device Code Authentication**

Identifier: ab76305a-0c11-465e-b435-dd178e78d734

Criticality: ▲ High | ● TLP: CLEAR

**Device Code flow**

OAuth 2.0 Device Authorization Grant from IETF:

The OAuth 2.0 device authorization grant is designed for Internet-connected devices that either lack a browser to perform a user-agent-based authorization or are input constrained to the extent that requiring the user to input text in order to authenticate during the authorization flow is impractical. It enables OAuth clients on such devices (like smart TVs, media consoles, digital picture frames, and printers) to obtain user authorization to access protected resources by using a user agent on a separate device.

**Detection**

There is a way to reliably detect this attack

006 - MDR demo for CERT-EU workshop.yaml Models Library\Managed Detection Rules 12

- Incorrect type. Expected "string". yaml-schema: Display name [Ln 1, Col 7]
- Incorrect type. Expected "string". yaml-schema: UUID [Ln 9, Col 9]
- Incorrect type. Expected "integer". yaml-schema: Version [Ln 11, Col 12]



# Drafting Managed Detection rules example Sentinel & Splunk sub-sections

```
! demo.yaml 9+, U • {} settings.json
tests > ! demo.yaml > {} configurations > {} sentinel > [ ] entities > {} 2 > entity
60 configurations:
62 sentinel:
73 alert:
75 description: |
76 | {{description}}
77 #custom_details:
78 #- key:
79 | #value:
80
81 grouping:
82 event: AlertPerResult
83 #alert:
84 #enabled: true
85 #grouping_lookback: 5h
86 #matching:
87
88 entities:
89 - entity: Process
90 mappings:
91 - identifier: CommandLine
92 | column: command_line
93
94 - entity: Host
95 mappings:
96 - identifier: HostName
97 | column: src_host
98 - identifier: DnsDomain
99 | column: src_nt_domain
100
101 - entity: Account
102 mappings:
103 - identifier: Name
104 | column: src_user
105
106 query: |
107 let Base64EncodedPS = (
108 | _GetWatchlist('CSIRC_WL_005')
109 | project command_line);
110 SecurityEvent
111 //we look for process creation
112 | where EventID==4688
113 | where isnotempty(CommandLine)
114 //field renaming
115 | extend command_line = CommandLine
116 | extend src_host = tostring(split(Computer, '.', 0)[0])
117 | extend src_user = SubjectUserName
118 | extend src_nt_domain = tostring(strcat_array(array_slice(split(Computer, '.'), 1, -1), '.'))
119 | extend user_type = AccountType
120 | extend event_source = EventSourceName
121 | extend signature = Activity
122 | extend process = NewProcessName
123 | extend parent_process = ParentProcessName
124 | extend subscription_templ = split(ResourceId, "/subscriptions/")[1]
```

**Account**

User account entity type

Required identifiers: Sid, Name, AadUserId, PUID, ObjectGuid

**Strong identifiers of an account entity:**

- Name + UPNSuffix
- AadUserId
- Sid + Host (required for SIDs of builtin

```
! demo.yaml 9+, U • {} settings.json
tests > ! demo.yaml > {} configurations > {} splunk > query
60 configurations:
144 splunk:
156 scheduling:
158 frequency: 15m
159 lookback: 15m
160
161 #notable:
162 #event:
163 | #title: Suspicious PowerShell Encoded Payload launched by $src_user$ on $orig_host$
164 | #description:
165
166 risk:
167 message: Suspicious PowerShell Encoded Payload ($matching_pattern$) launched by $src_user$ on $
168 risk_objects:
169 - field: src_user
170 | type: user
171 | score: 10
172 - field: host
173 | type: system
174 | score: 10
175 threat_objects:
176 - field: CommandLine
177 | type: command_line
178
179 #drilldown:
180 #name:
181 #search: |
182 | #Type Here
183
184 #security_domain:
185
186 query: |
187 'win_security_logs'
188 AND TERM(EventID=4688)
189 AND NewProcessName IN ("C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe", "C:\\
190 AND ( TERM(C:\\WINDOWS\\system32\\WindowsPowerShell\\v1.0\\PowerShell.exe) OR TERM(C:\\WINDO
191 AND (TERM(-e) OR TERM(-enc) OR TERM(-EncodedCommand))
192 NOT TERM(C:\\Windows\\CCM\\CcmExec.exe)
193 | rex field=CommandLine "(?i)\\s+(-e|-enc|-EncodedCommand)\\s+(?<payload_b64>(?:[A-Za-z0-9+]{4})*
194 | where isnotnull(payload_b64)
195 | search `exclude isnotnull
196 | `soc_macro_decode_b64(payload_b64)`
197 | fields host, ParentProcessName, src_user, pa
198 | stats count min(_time) as et, max(_time) as
199 | fields host, count, et, lt, ParentProcessNa
200 | rename payload_* as payloadlevel1_*
201 | rex field=payloadlevel1_b64_ascii "(?i)\\s?(
202 | search `exclude(SOC_LT_289_WIN_reviewed_base
203 | `soc_macro_decode_b64(payload_b64)`
204 | rename payload_* as payload_level2_*, paylo
205 | rex field=payload_level2_b64_ascii "(?i)\\s?(
```

**Signature:**

isnotnull(<value>)

**Returns:** boolean

**Examples:**



# Nobody wants to write documentation

CATCH / EC Tide / Wiki / Models / Threat vectors / Use sophisticated Microsoft device code phishing

## Use sophisticated Microsoft device code phishing

Last edited by CATCH Bot 2 days ago

**Criticality: High** : A High priority incident is likely to result in a demonstrable impact to public health or safety, national security, economic security, foreign relations, civil liberties, or public confidence.

**TLP: AMBER+STRICT** : Similar to TLP: AMBER, but restricts sharing to the organization only.

**ATT&CK Techniques** T1566.001 : Phishing: Spearphishing Attachment, T1071.001 : Application Layer Protocol: Web Protocols, T1071.002 : Application Layer Protocol: File Transfer Protocols

UUID : 00ee99b5-27be-46fe-9674-50f71263069f | Version : 1 | Creation Date : 2025-02-18 | Last Modification : 2025-02-21 | Model author | Schema Identifier : tvm::2.1

### Description

The Device Code Phishing Technique is a sophisticated process that exploits the OAuth 2.0 Device Authorization Grant flow. Here's a more extensive explanation of how this attack unfolds:

### Mechanism of Attack

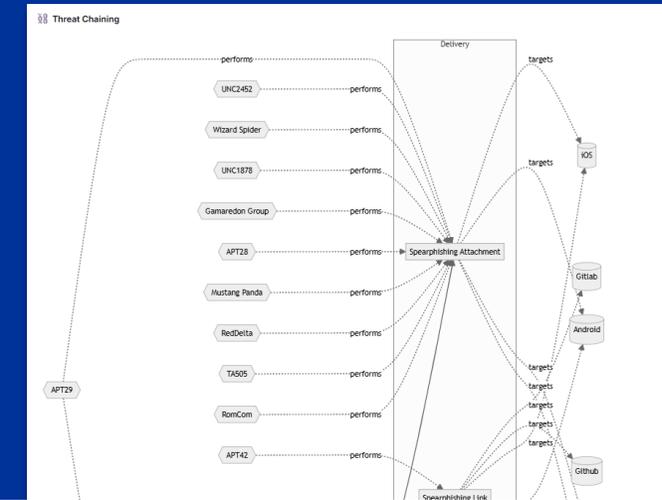
1. Device Code Generation
2. Crafting the Phishing Lure
3. Initial Contact
4. Victim Interaction
5. Authentication Process
6. Token Interception
7. Token Acquisition
8. Account Exploitation

**OpenTide Objects**

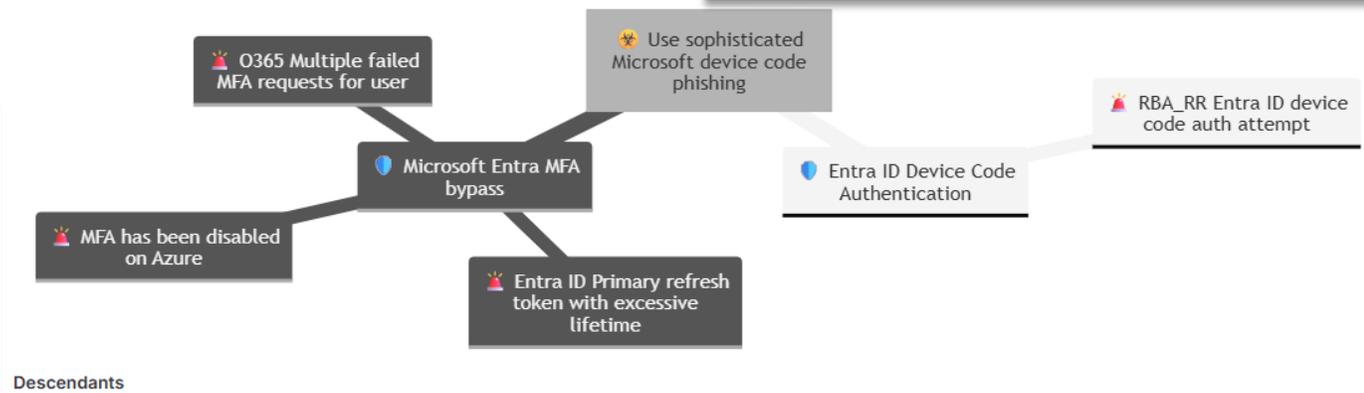
`https://login.microsoftonline.com/common/oauth2/deviceauth`

This request does not require authentication, allowing the attacker to generate codes at will.

- On this page
- Description
  - Mechanism of Attack
  - 1. Device Code Generation
  - 2. Crafting the Phishing Lure
  - 3. Initial Contact
  - 4. Victim Interaction
  - 5. Authentication Process
  - 6. Token Interception
  - 7. Token Acquisition
  - 8. Account Exploitation
  - Terrain
  - Relations
  - Actors sightings
  - OpenTide Objects
  - Threat Chaining
  - Actors sightings
  - Model Data
  - Cyber Kill Chain
  - Domains
  - Targets



Actor	Description	Aliases	Source	Sighting
[Enterprise] APT29	APT29 is threat group that has been attributed to Russia's Foreign Intelligence Service (SVR). (Citation: White House Imposing Costs RU Gov April 2021)(Citation: UK Gov Malign IRIS Activity April 2021) They have operated since at least 2008, often targeting government networks in Europe and NATO member countries, research institutes, and think tanks. APT29 reportedly compromised the Democratic National Committee starting in the summer of 2016. (Citation: P-Secure The Dukes)(Citation: GRIZZLY STEPPED AHEAD)(Citation: Crowdstrike DNC June 2016)(Citation: UK Gov UK Exposes Russia SolarWinds April 2021) In April 2021, the US and UK governments attributed the SolarWinds compromise to the SVR; public statements included citations to APT29, Cozy Bear, and The Dukes. (Citation: NSA Joint Advisory SVR SolarWinds April 2021)(Citation: UK NCSC Russia SolarWinds April 2021) Industry reporting also referred to the actors involved in this campaign as UNC2452, NOBELIUM, StellarParticle, Dark Halo, and SolarStorm. (Citation: FireEye SUNBURST Backdoor December 2020) (Citation: MSTIC NOBELIUM Mar 2021)(Citation: CrowStrike SUNSPOT Implant January 2021)(Citation: Volexity SolarWinds)(Citation: Cybersecurity Advisory SVR TTP May 2021)(Citation: Unit 42 SolarStorm December 2020)	Blue Kitsune, Cozy Bear, CozyDuke, Dark Halo, IRON HELMLOCK, IRON RITUAL, Midnight Blizzard, NOBELIUM, NobleBaron, SolarStorm, The Dukes, UNC2452, UNC3524, YTRILUM	MITRE ATTACK Groups	Threat actor that has been conducting sophisticated spearphishing attacks targeting Microsoft 365 accounts. They impersonated entities such as the US Department of State, and lured victims to access applications within the M365 tenant or join a Microsoft Teams chat named "Measuring Influence Operations".

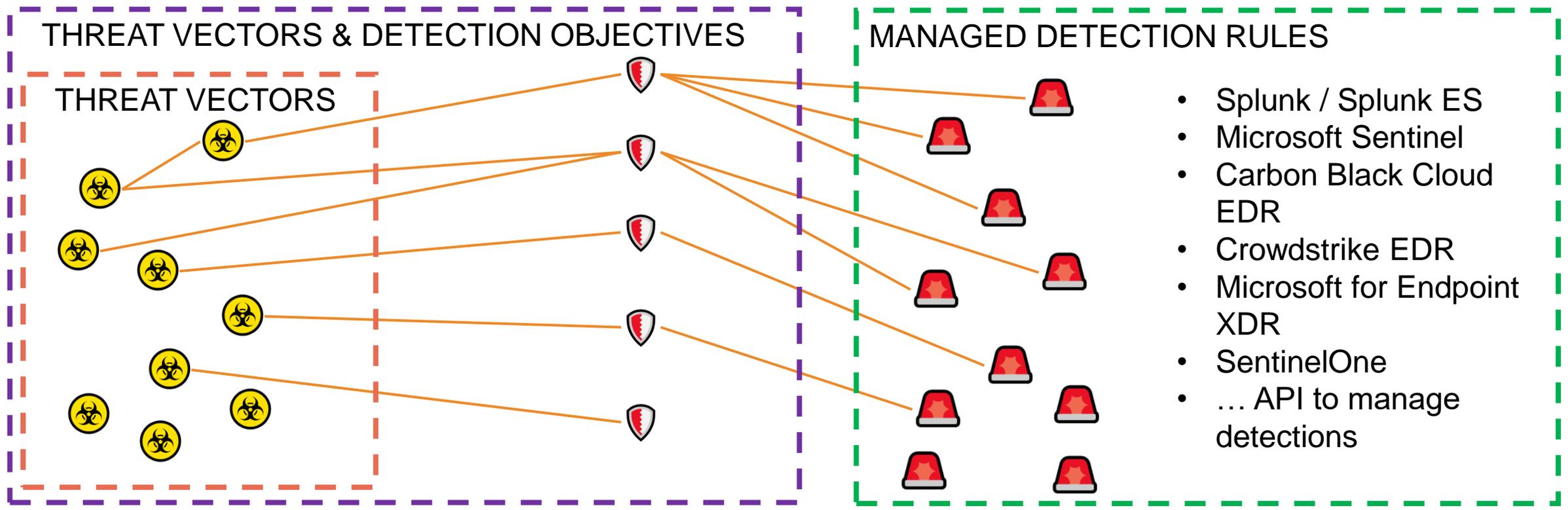


Descendants



# Use cases for OpenTIDE

## OpenTIDE



- Splunk / Splunk ES
- Microsoft Sentinel
- Carbon Black Cloud EDR
- CrowdStrike EDR
- Microsoft for Endpoint XDR
- SentinelOne
- ... API to manage detections

 435 Threat Vectors

 215 Detection Models

 475 Active Detection Rules



# OpenTide key points

- Managed Knowledge **Graph**
- Rich documentation including metadata
  - Linking threats to detection scenarios to deployed rules
  - Consistency and quality of documentation, enforced via validation
- Detection “as-code” – Automation
  - YAML objects structured and validated against a schema
  - Prioritisation of the backlog
  - Versioning, history, BCP (rules can be rolled out in short time)



# Threat Vector object key points

- Turn TI inputs into actionable objects (purple teaming)
- Allow to document the attack vectors and attack paths
  - Threat vector relevant for the organisation
  - Granularity can go as deep as needed below the ATT&CK sub-techniques
- Prioritise the backlog of threat to be covered

 415 Threat Vectors

Type to search

 UUID	 Model Name	 Vector criticality	 Traffic Light Protocol 2.0	 Description of the threat	 Last Modification	 Implementations	 Targets	 Platforms concerned	 ATT&CK Techniques
--	--	--	--	---	---	---	---	---	---



European  
Commission

# Detection Objective object key points

- Define clearly WHAT to detect
  - Linked to threat vectors
  - Document the detection strategies and which data sources are needed
  - Documentation of detection use cases that didn't work out
- Facilitate communication with system owners and risk managers
  - Detection objectives are descriptive vs rules that are platform specific code
- Drive the log collection to support the objective (visibility)
  - List the data sources required to implement the detection strategies

 212 Detection Models

Type to search

 UUID

 Model Name

 Vector  
criticality

 Traffic  
Light  
Protocol  
2.0

 ATT&CK  
Technique

 Guidelines

 Last  
Modification

Threat Vector Model

 Implementations

 Technical  
Detection  
Methods

# Managed Detection Rule object key points

- DE focuses on the quality of the query
- All the rest is automated within EC-Tide  
(no need to configure via GUI - 100+ clicks/copy/paste replaced by a YAML file)
- Peer review at the lines of code level
- **Joint workflow btw DE and SOC to review the rule & draft alert handling guidelines**
  - MDR contains the response section providing guidelines on handling the alert.

  470 Active Detection Rules



Type to search

 UUID

 Display name

 Detection description

 Last  
Modification

 Status

 MDR Technique

CoreTIDE  
Cyber  
Detecti  
Model



European  
Commission

# OpenTIDEhq Project Structure

Decoupled code backend, made to run without maintenance on every pipeline run



**CoreTide**



**InitTide**

Starting point template to provide the folder structure, minimal set of files, and CI to connect to CoreTide

Copy

**Private clone of CoreTide**

Point to your local clone of CoreTide

**Your private Git content repo**

Pull from public

TLP:CLEAR content repository  
YAML opentide objects

200+ Threat Vectors shared  
More to come



**ShareTide**

Review and copy

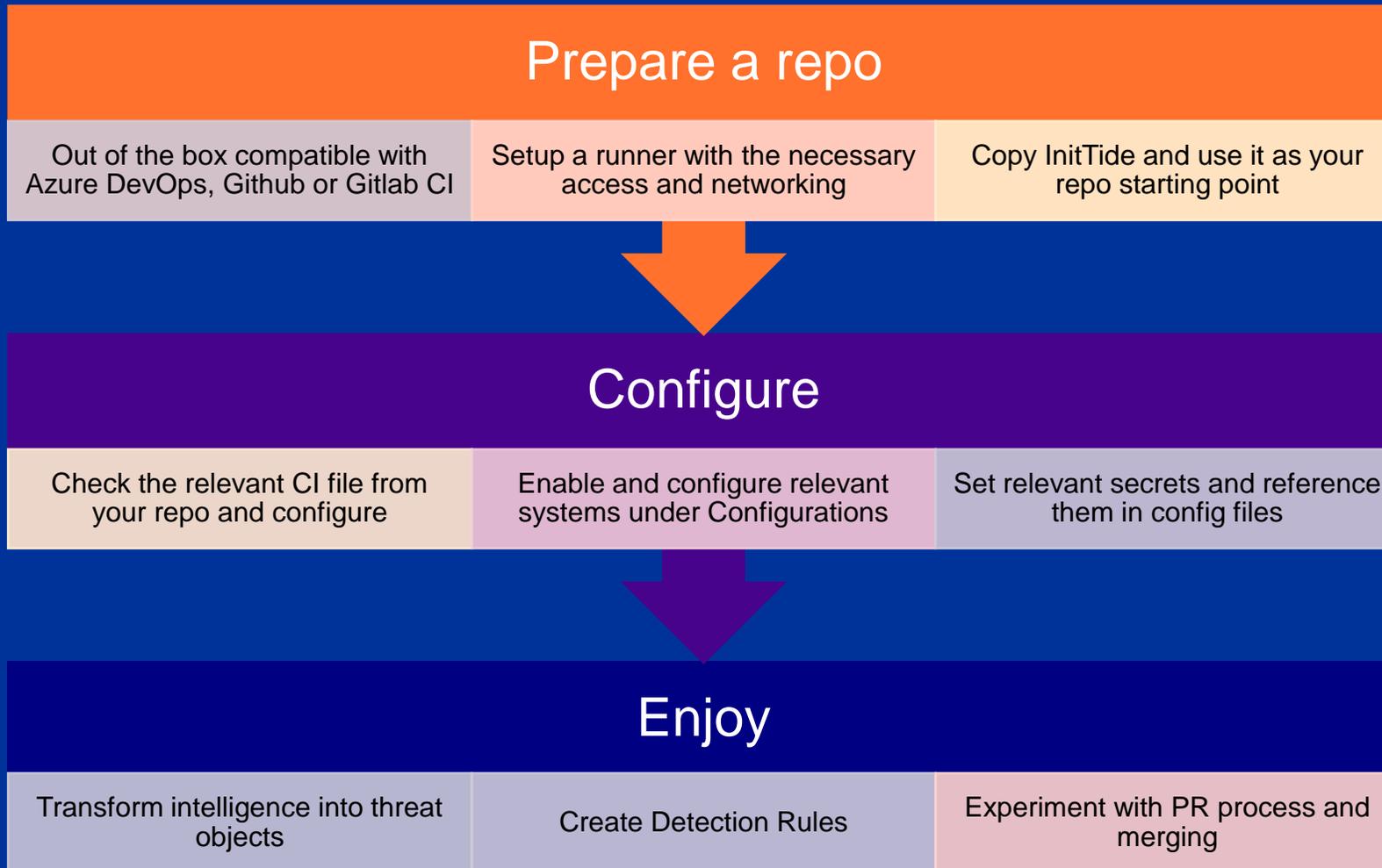


**WikiTide**

TLP:CLEAR Wiki content  
Markdown documentation of  
Objects published under  
ShareTIDE



# Give it a shot



# With OpenTIDE do

- Detection Engineering
  - Include TI context into your detection rules
  - Prioritise and measure your threat detection coverage
- DetectionOps
  - Backlog, iterations, issues, merge requests, review,
- Detection-as-Code
  - YAML, versioning, CI/CD pipelines, deployers, validation of queries





- OpenTide project

- <https://github.com/OpenTideHQ>
- <https://code.europa.eu/ec-digit-s2/opentide>

Please open issues or pull requests here !

- Mailbox: [DIGIT-EC-OPENTIDE@ec.europa.eu](mailto:DIGIT-EC-OPENTIDE@ec.europa.eu)

- [FIRST SIG DE&TH Detection Engineering & Threat Hunting](#)

# Thank you!



European  
Commission