RSAConference2020
Asia Pacific & Japan

A Virtual Learning Experience | 15–17 July

HUMAN ELEMENT

SESSION ID: GPS-F08

Detection as Code – Detection Development using CI/CD

Patrick Bareiss

Senior Security Researcher
Splunk
@bareiss_patrick

Jose Hernandez

Principal Security Researcher
Splunk
@d1vious

Challenges

Threat landscape

Keep up with the ever changing threat landscape

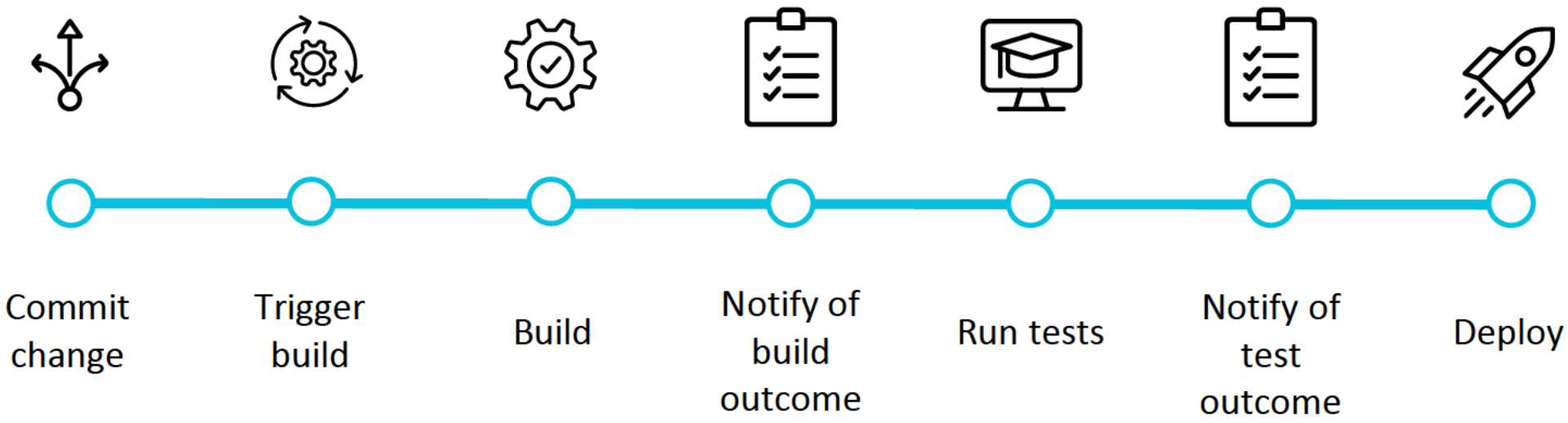
Test & Improve detections

Continuously testing and improvement of detections

Shorten development lifecycle

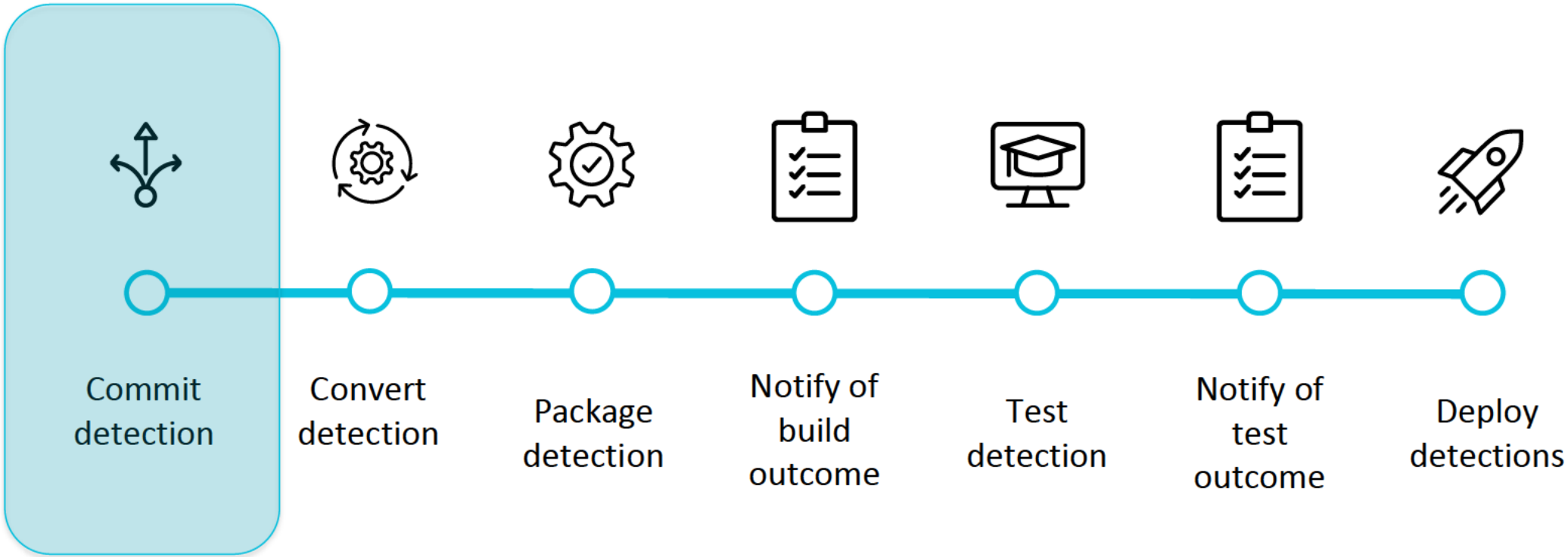
Reduce detection development lifecycle between new attack and new detection

CI/CD Workflow



CI/CD Workflow – Detection Engineering

CI/CD Workflow – Detection Engineering



Commit detection – detection schema

sysmon_mimikatz_detection_lsass.yml

```yaml
title: Mimikatz Detection LSASS Access
status: experimental
description: Detects process access to LSASS which is
references:
    - https://onedrive.live.com/view.aspx?resid=D026B4
    - https://cyberwardog.blogspot.com/2017/03/chronic
tags:
    - attack.t1003
    - attack.s0002
    - attack.credential_access
    - car.2019-04-004
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: 10
        TargetImage: 'C:\windows\system32\lsass.exe'
        GrantedAccess:
            - '0x1410'
            - '0x1010'
    condition: selection
falsepositives:
    - unknown
level: high
```

Sigma

creation_of_shadow_copy_with_wmic_and_po...

```yaml
name: Creation of Shadow Copy with wmic and powershell
id: 2ed8b538-d284-449a-be1d-82ad1dbd186b
version: 1
date: '2019-12-10'
description: This search detects the use of wmic and Powershell to create a shadow
  copy.
how_to_implement: You must enable Powershell scriptblock logging in order to detect
  this attack.This search uses an input macro named `sysmon`. We strongly recommend
  that you specify your environment-specific configurations (index, source, sourcety
  etc.) for Windows Sysmon logs. Replace the macro definition with configurations
  for your Splunk environment. The search also uses a post-filter macro designed to
  filter out known false positives.
type: ESCU
references:
- https://2017.zeronights.org/wp-content/uploads/materials/ZN17_Kheirkhabarov_Huntin
author: Patrick Bareiss, Splunk
search: '`sysmon` Message=*win32_shadowcopy* Message=*Create* | stats count min(_tim
  as firstTime max(_time) as lastTime by dvc User EventCode Message | rename User
  as user, dvc as dest | `security_content_ctime(firstTime)`| `security_content_ctim
  | `creation_of_shadow_copy_with_wmic_and_powershell_filter` '
known_false_positives: Legtimate administrator usage of wmic to create a shadow copy
tags:
  analytics_story:
  - Credential Dumping
  mitre_attack_id:
  - T1003
```

Splunk Security Content

Commit detection – detection schema

sysmon_mimikatz_detection_lsass.yml

```yaml
title: Mimikatz Detection LSASS Access
status: experimental
description: Detects process access to LSASS which is
references:
    - https://onedrive.live.com/view.aspx?resid=D026B4
    - https://cyberwardog.blogspot.com/2017/03/chronic
tags:
    - attack.t1003
    - attack.s0002
    - attack.credential_access
    - car.2019-04-004
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: 10
        TargetImage: 'C:\windows\system32\lsass.exe'
        GrantedAccess:
            - '0x1410'
            - '0x1010'
    condition: selection
falsepositives:
    - unknown
level: high
```

Sigma

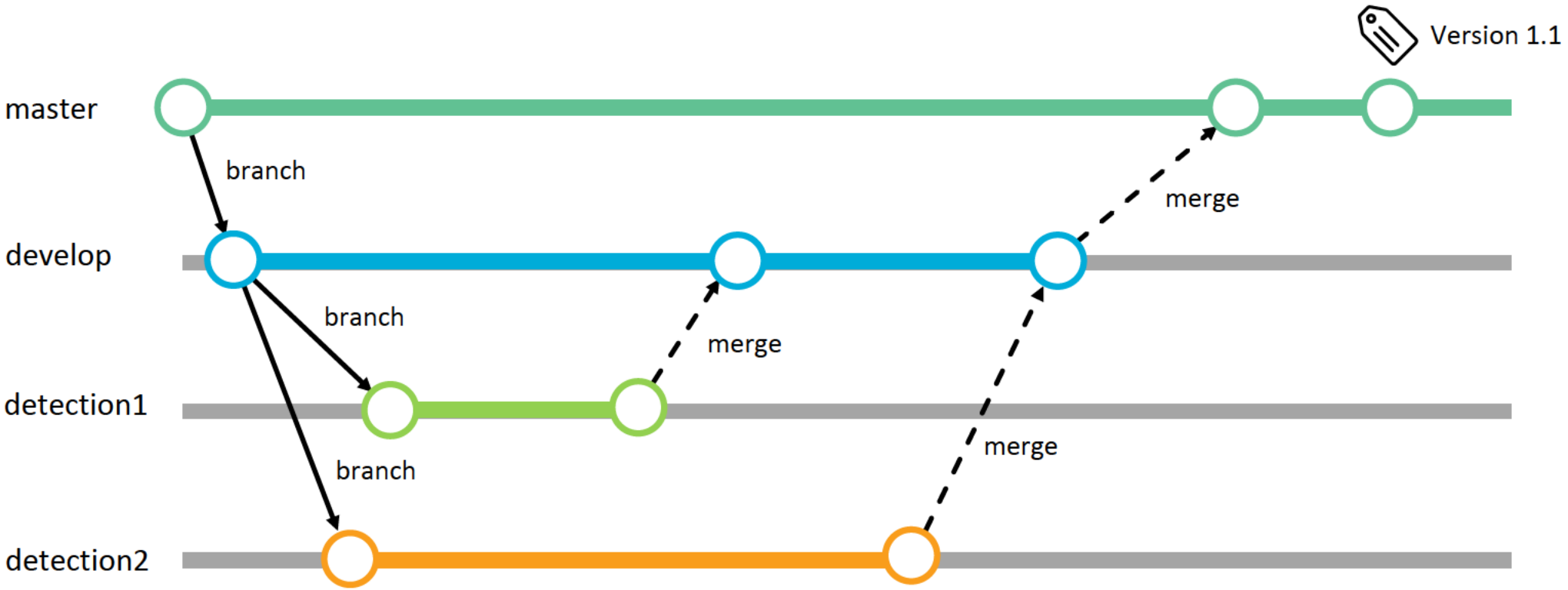
creation_of_shadow_copy_with_wmic_and_po...

```yaml
name: Creation of Shadow Copy with wmic and powershell
id: 2ed8b538-d284-449a-be1d-82ad1dbd186b
version: 1
date: '2019-12-10'
description: This search detects the use of wmic and Powershell to create a shadow
  copy.
how_to_implement: You must enable Powershell scriptblock logging in order to detect
  this attack.This search uses an input macro named `sysmon`. We strongly recommend
  that you specify your environment-specific configurations (index, source, sourcety
  etc.) for Windows Sysmon logs. Replace the macro definition with configurations
  for your Splunk environment. The search also uses a post-
  filter out known false positives.
type: ESCU
references:
- https://2017.zeronights.org/wp-content/uploads/materials
author: Patrick Bareiss, Splunk
search: '`sysmon` Message=*win32_shadowcopy* Message=*Crea
  as firstTime max(_time) as lastTime by dvc User EventCod
  as user, dvc as dest | `security_content_ctime(firstTime
  | `creation_of_shadow_copy_with_wmic_and_powershell_filt
known_false_positives: Legtimate administrator usage of wm
tags:
  analytics_story:
  - Credential Dumping
  mitre_attack_id:
  - T1003
```

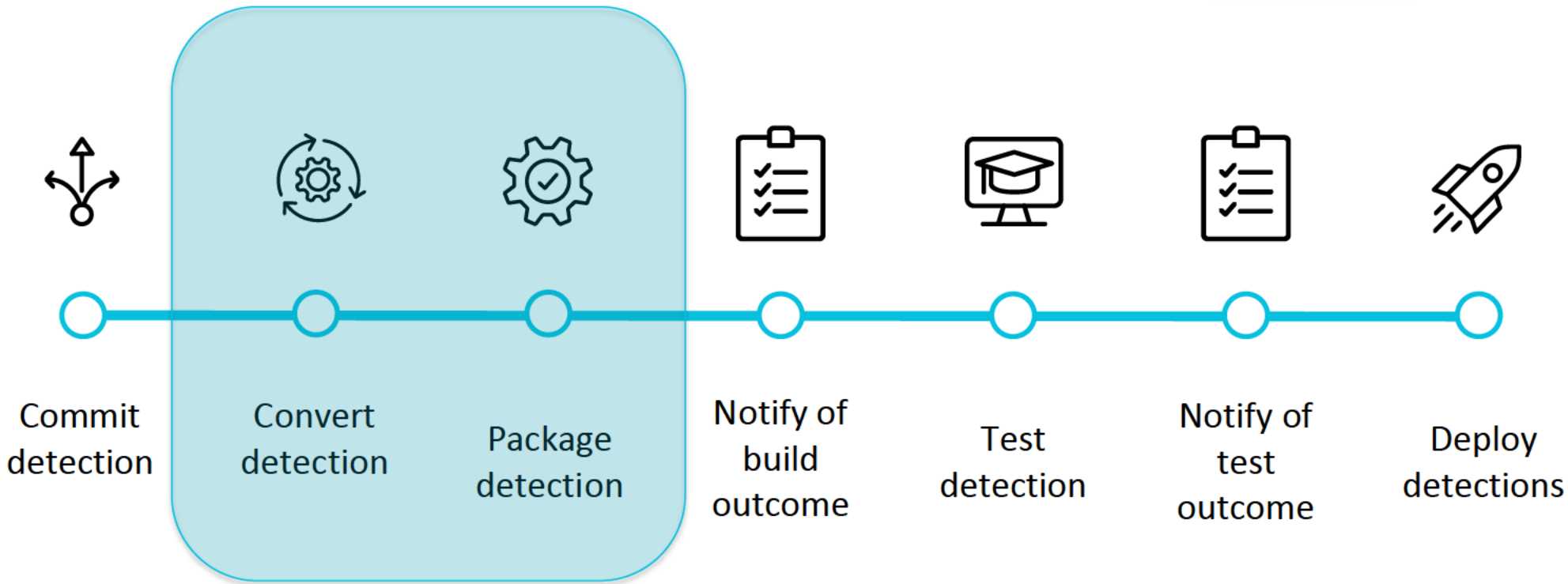
Splunk Security Conte

splunk> turn data into doing™

Commit detection – Branching workflow



CI/CD Workflow – Detection Engineering



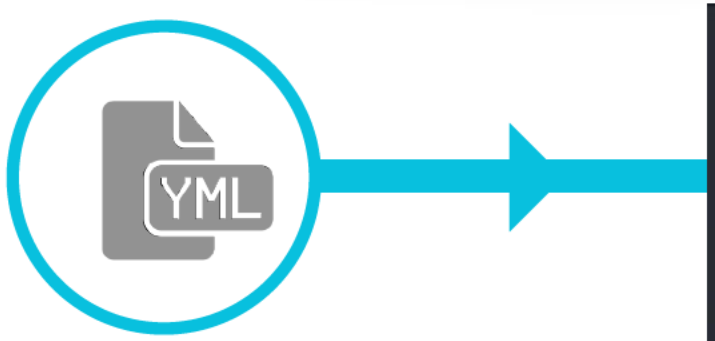
Convert Detection & Package Detection

detection schema format

SIEM specific search format

SIEM specific configuration format

Convert Detection & Package Detection




```yaml
name: Detect Credential Dumping through LSASS access
id: 2c365e57-4414-4540-8dc0-73ab10729996
version: 3
date: '2019-12-03'
description: This search looks for reading lsass memory consistent with credential
  dumping.
how_to_implement: This search needs Sysmon Logs and a sysmon configuration, which
  includes EventCode 10 with lsass.exe. This search uses an input macro named `sysmon`.
  We strongly recommend that you specify your environment-specific configurations
  (index, source, sourcetype, etc.) for Windows Sysmon logs. Replace the macro definition
  with configurations for your Splunk environment. The search also uses a post-filter
  macro designed to filter out known false positives.
type: ESCU
references: []
author: Patrick Bareiss, Splunk
search: '`sysmon` EventCode=10 TargetImage=*lsass.exe (GrantedAccess=0x1010 OR GrantedAcc
  | stats count min(_time) as firstTime max(_time) as lastTime by Computer, SourceImage,
  SourceProcessId, TargetImage, TargetProcessId, EventCode, GrantedAccess | rename
  Computer as dest | `security_content_ctime(firstTime)`| `security_content_ctime(lastTim
  | `detect_credential_dumping_through_lsass_access_filter` '
known_false_positives: The activity may be legitimate. Other tools can access lsass
  for legitimate reasons, and it's possible this event could be generated in those
  cases. In these cases, false positives should be fairly obvious and you may need
  to tweak the search to eliminate noise.
tags:
  analytics_story:
  - Credential Dumping
  mitre_attack_id:
  - T1003
```

splunk> turn data into doing™

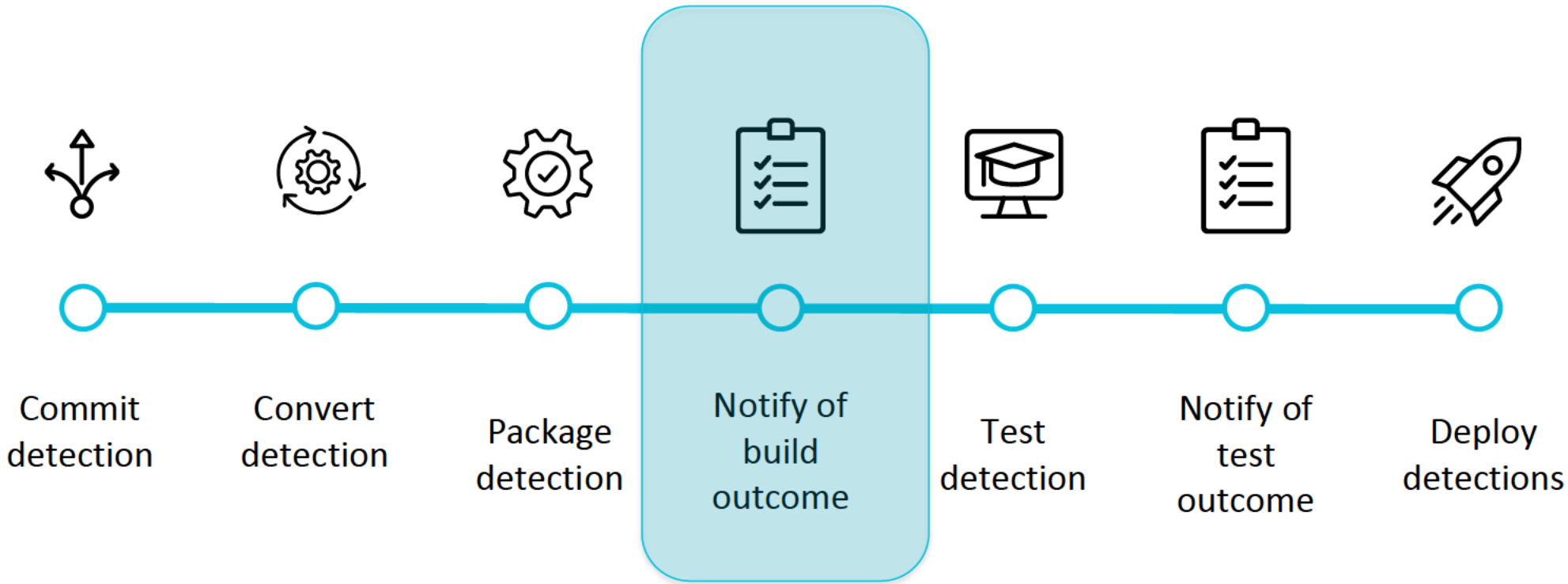
Convert Detection & Package Detection


```
`sysmon` EventCode=10 TargetImage=*lsass.exe (GrantedAccess=0x1010 OR GrantedAccess=0x1410)
  | stats count min(_time) as firstTime max(_time) as lastTime by Computer, SourceImage,
  SourceProcessId, TargetImage, TargetProcessId, EventCode, GrantedAccess | rename
  Computer as dest | `security_content_ctime(firstTime)`| `security_content_ctime(lastTime)`
  | `detect_credential_dumping_through_lsass_access_filter`
```

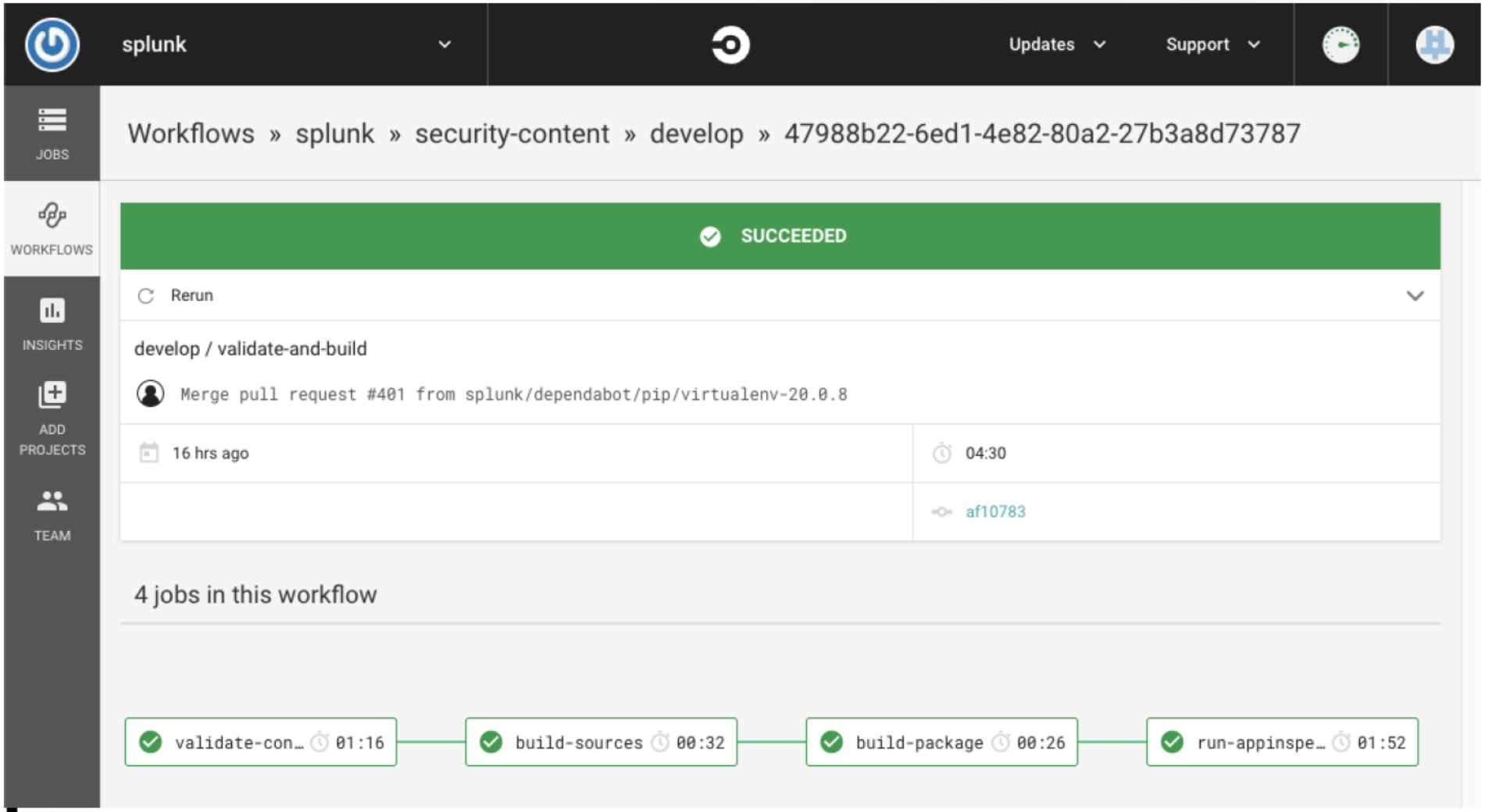
Convert Detection & Package Detection

```
[ESCU - Detect Credential Dumping through LSASS access - Rule]
action.escu = 0
action.escu.enabled = 1
description = This search looks for reading lsass memory consistent with credential dumping.
action.escu.mappings = {"cis20": ["CIS 3", "CIS 5", "CIS 16"], "kill_chain_phases": ["Actions on Objectives"], "mitre_a
action.escu.data_models = []
action.escu.eli5 = This search looks for reading lsass memory consistent with credential dumping.
action.escu.how_to_implement = This search needs Sysmon Logs and a sysmon configuration, which includes EventCode 10 wi
action.escu.known_false_positives = The activity may be legitimate. Other tools can access lsass for legitimate reasons
action.escu.creation_date = 2019-12-03
action.escu.modification_date = 2019-12-03
action.escu.confidence = high
action.escu.full_search_name = ESCU - Detect Credential Dumping through LSASS access - Rule
action.escu.search_type = detection
action.escu.providing_technologies = []
action.escu.analytic_story = ["Credential Dumping"]
cron_schedule = */30 * * * *
dispatch.earliest_time = -30m
dispatch.latest_time = now
action.correlationsearch.enabled = 1
action.correlationsearch.label = ESCU - Detect Credential Dumping through LSASS access - Rule
schedule_window = auto
action.notable = 1
action.notable.param.nes_fields = ['dest']
action.notable.param.rule_description = This search looks for reading lsass memory consistent with credential dumping.
action.notable.param.rule_title = Detect Credential Dumping through LSASS access
action.notable.param.security_domain = endpoint
action.notable.param.severity = high
alert.digest_mode = 1
action.escu.earliest_time_offset = 3600
action.escu.latest_time_offset = 86400
disabled = true
enableSched = 1
counttype = number of events
relation = greater than
quantity = 0
realtime_schedule = 0
is_visible = false
search = `sysmon` EventCode=10 TargetImage=*lsass.exe (GrantedAccess=0x1010 OR GrantedAccess=0x1410) | stats count min(
```

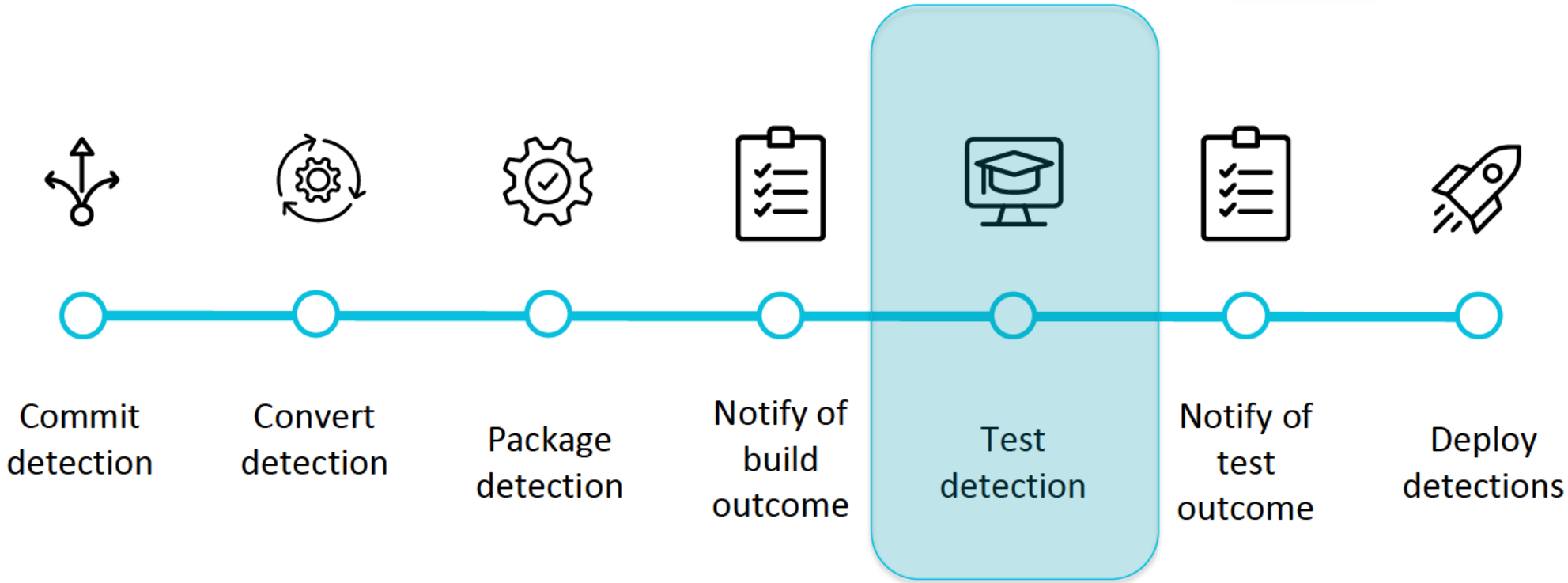
CI/CD Workflow – Detection Engineering



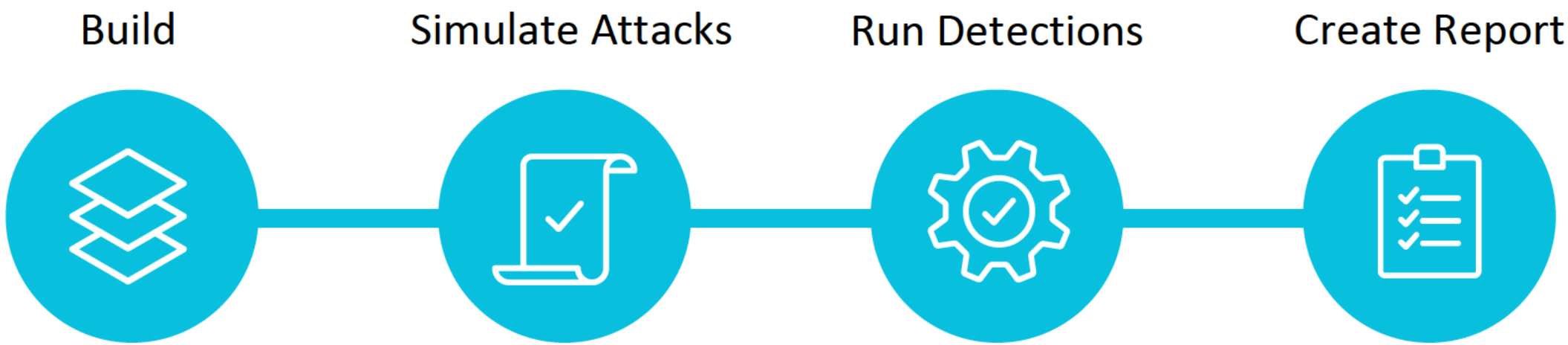
Notify of build outcome - CircleCI



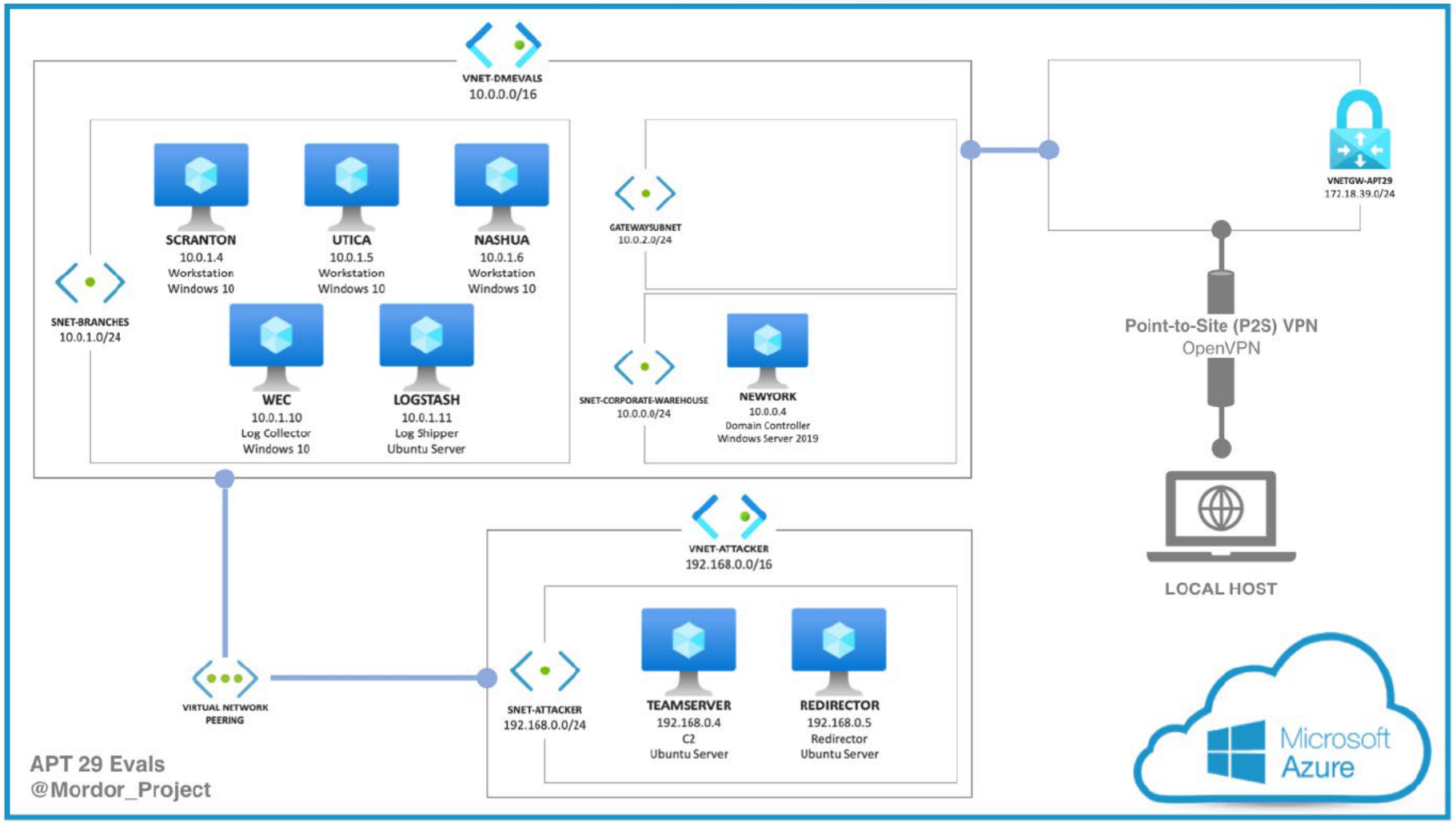
CI/CD Workflow – Detection Engineering



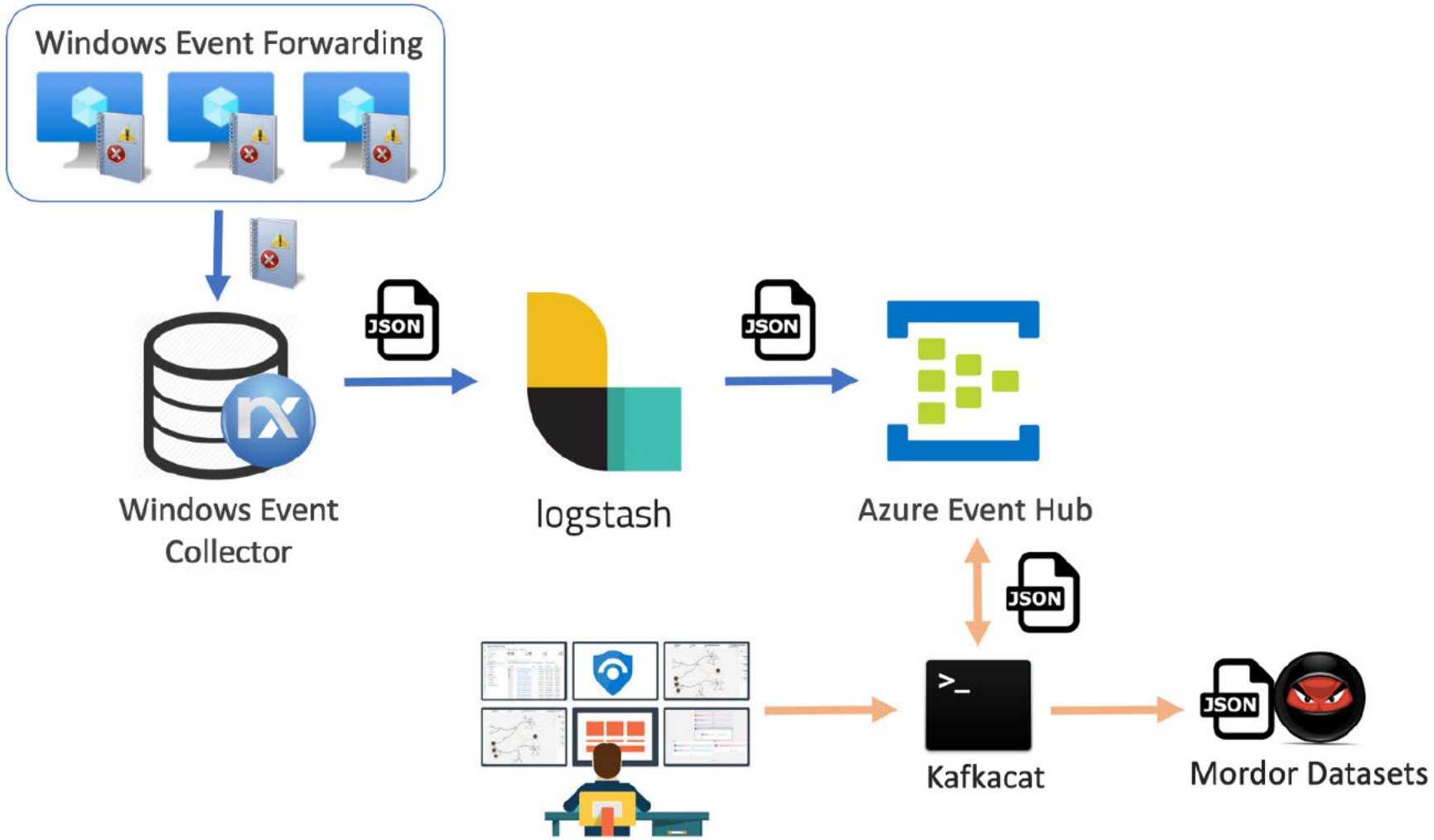
Test Detections



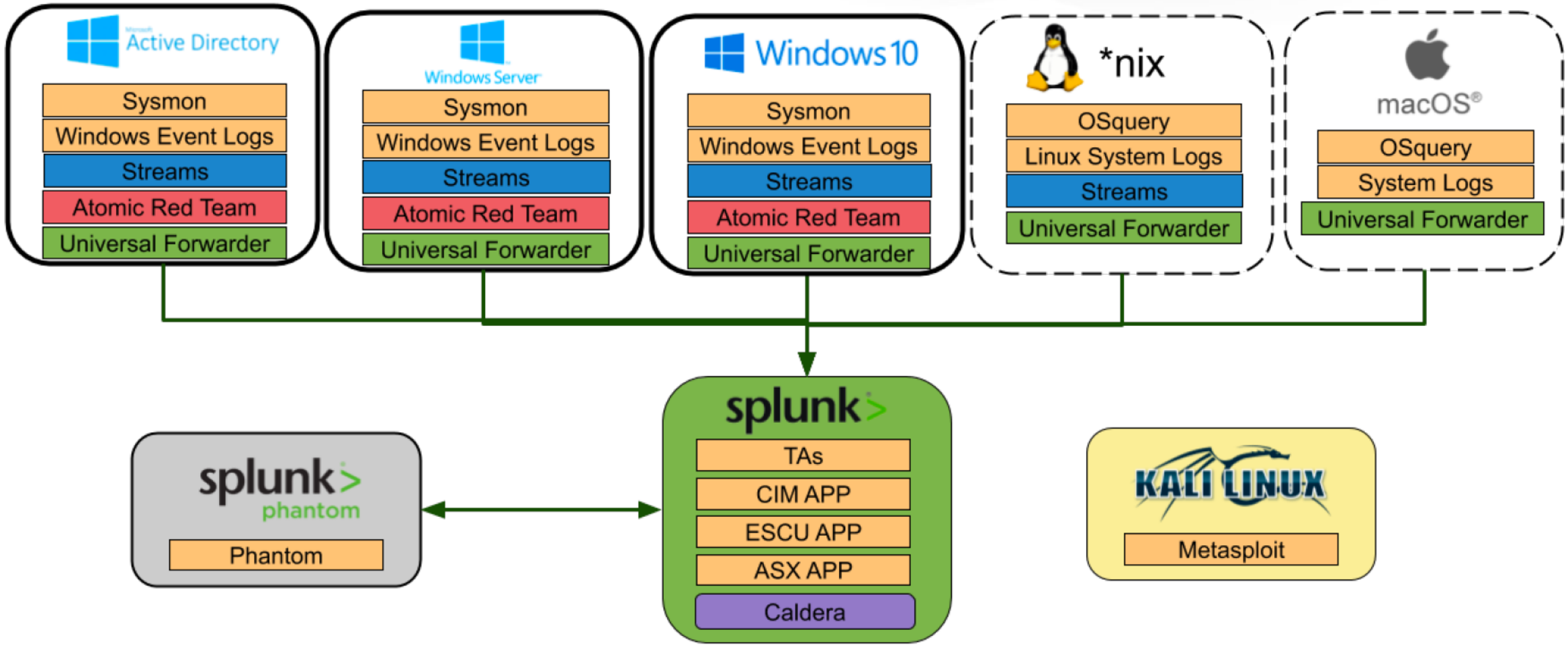
Mordor Labs


VNET-DMEVALS
10.0.0.0/16
SCRANTON
10.0.1.4
Workstation
Windows 10
UTICA
10.0.1.5
Workstation
Windows 10
NASHUA
10.0.1.6
Workstation
Windows 10
SNET-BRANCHES
10.0.1.0/24
WEC
10.0.1.10
Log Collector
Windows 10
LOGSTASH
10.0.1.11
Log Shipper
Ubuntu Server
GATEWAYSUBNET
10.0.2.0/24
SNET-CORPORATE-WAREHOUSE
10.0.0.0/24
NEWYORK
10.0.0.4
Domain Controller
Windows Server 2019
VNETGW-APT29
172.18.39.0/24
Point-to-Site (P2S) VPN
OpenVPN
LOCAL HOST
VNET-ATTACKER
192.168.0.0/16
VIRTUAL NETWORK
PEERING
SNET-ATTACKER
192.168.0.0/24
TEAMSERVER
192.168.0.4
C2
Ubuntu Server
REDIRECTOR
192.168.0.5
Redirector
Ubuntu Server
Microsoft
Azure
APT 29 Evals
@Mordor_Project

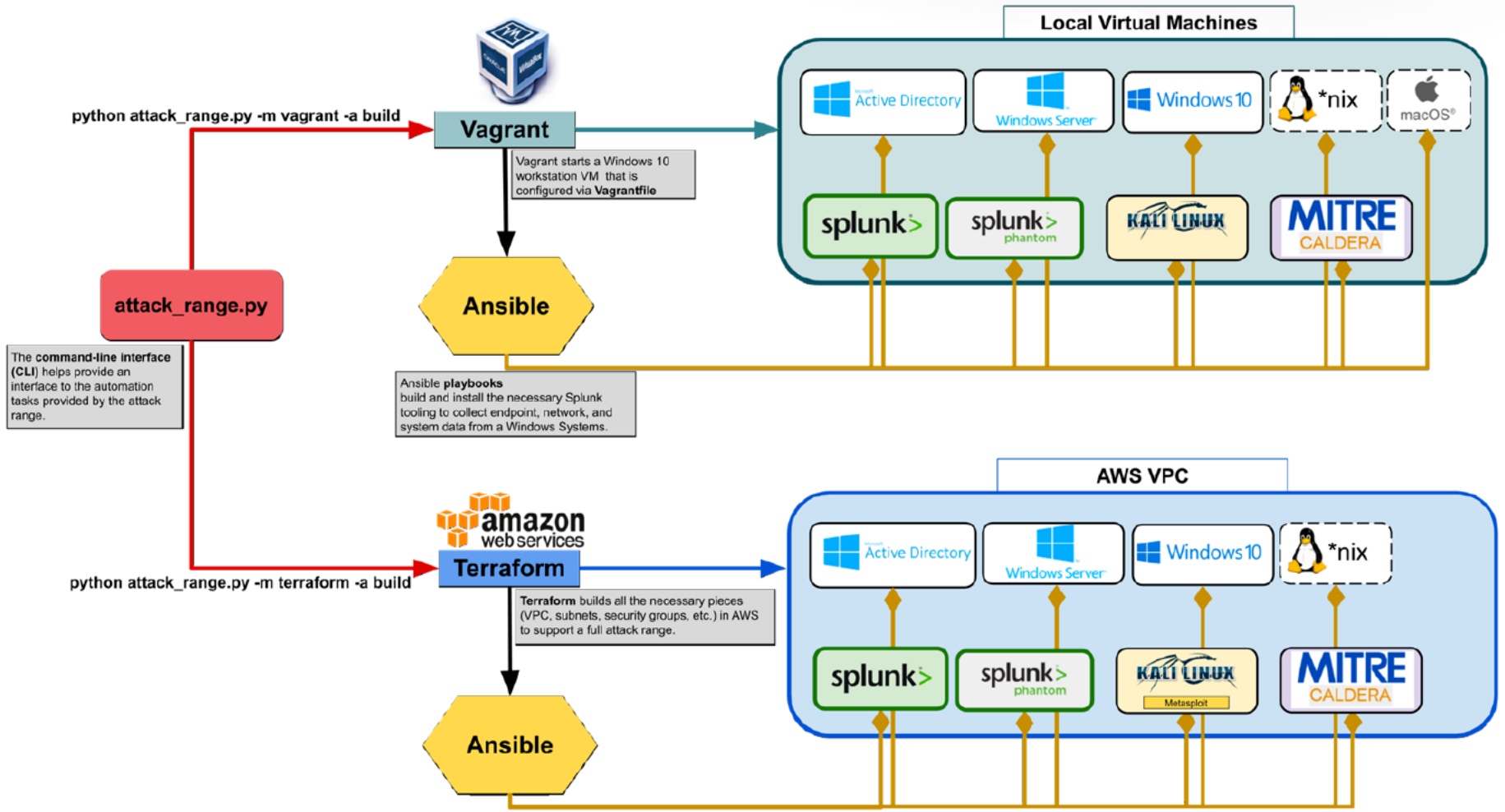
Mordor Labs


Windows Event Forwarding
JSON
JSON
JSON
Windows Event Collector
logstash
Azure Event Hub
Kafkacat
Mordor Datasets

Attack Range

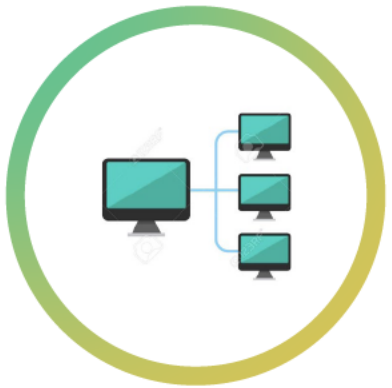


Attack Range Architecture



Attack Range Commands

Build



Automated building process with commands: **Build, destroy, stop, resume**

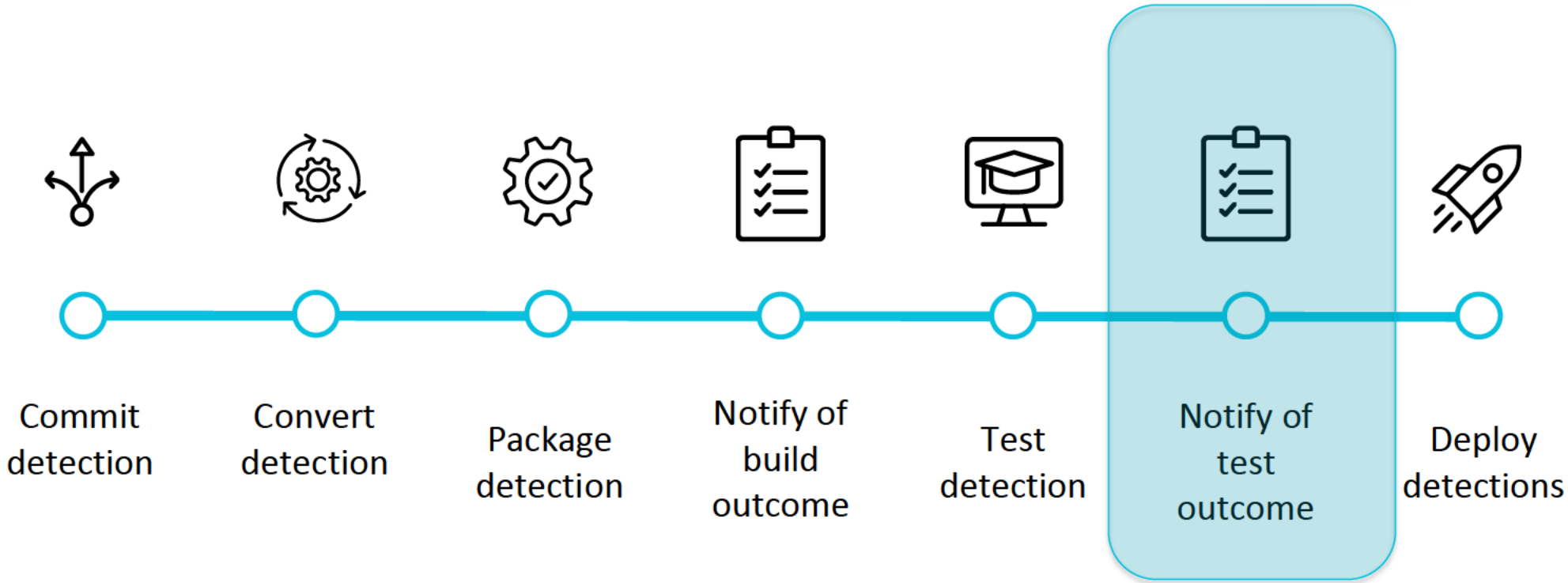
Simulate

Simulate attacks with Atomic Red Team with command: **simulate**

Test Detections

Run Splunk queries with the command: **search**

CI/CD Workflow – Detection Engineering



Notify of test outcome

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
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spearphishing via Service	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compile After Delivery	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Data from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Supply Chain	Exploitation for Client	Bootkit	Exploitation for Privilege	Compiled HTML File	Hooking	Password Policy	Remote File Copy	Email Collection	Domain Generation	Scheduled Transfer	Network Denial of
					Input Capt	ware					
					Input Prom	Model					
	LSASS Driver	Component Firmware	Hooking	Control Panel Items	Kerberoast						
	Launchctl	Component Object Model Hijacking	Image File Execution Options Injection	DCShadow	Keychain	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Stored Data Manipulation
	Local Job Scheduling	Create Account	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning and Relay	Remote System Discovery	Taint Shared Content		Multilayer Encryption		Transmitted Data Manipulation
	Mshta	DLL Search Order Hijacking	New Service	DLL Side-Loading	Network Sniffing	Security Software Discovery	Third-party Software		Port Knocking		
	PowerShell	Dylib Hijacking	Path Interception	Deobfuscate/Decode Files or Information	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote Access Tools		
	Regsvcs/Regasm	External Remote Services	Plist Modification	Disabling Security Tools	Private Keys	System Network Configuration Discovery	Windows Remote Management		Remote File Copy		
						System Network			Standard Application Layer		

```
python attack_range.py -m terraform -a simulate -st T1003
```

```
python attack_range.py -m terraform -a search -sn "ESCU - Attempted Credential Dump From Registry via Reg.exe - Rule"
```

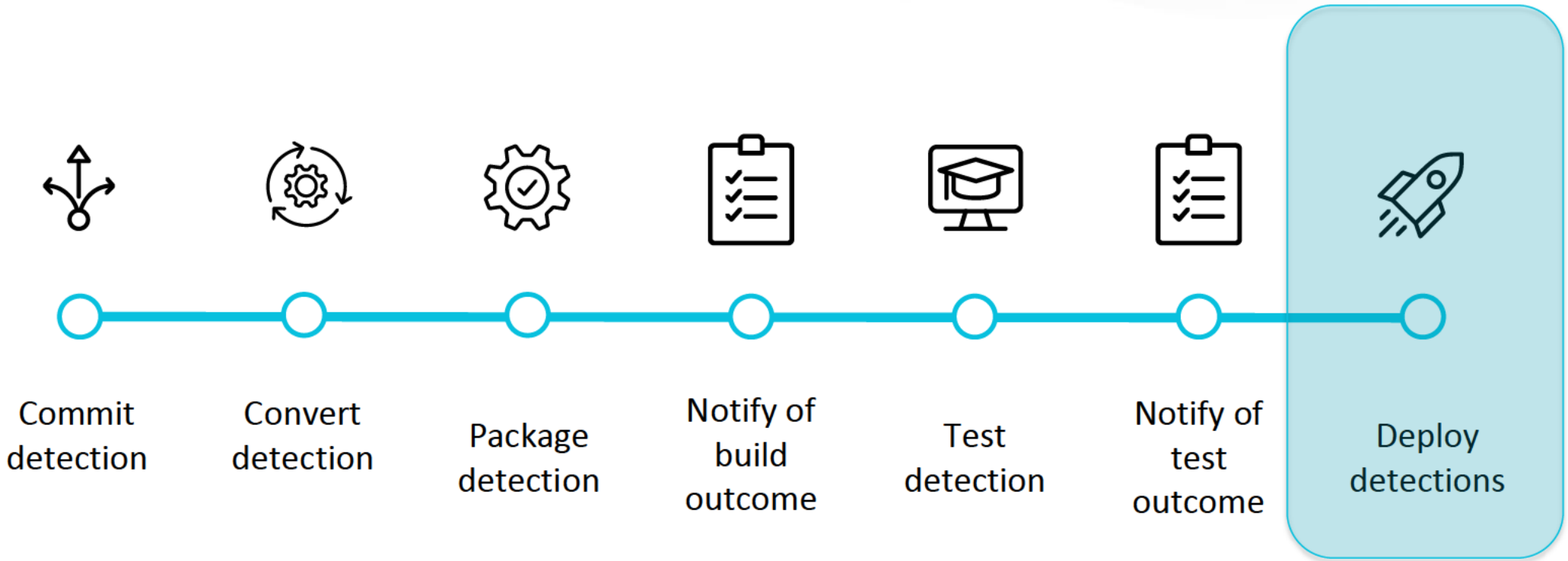
Notify of test outcome

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evas	edential Access	Disc	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulatio	Account Manipulation	Acco covery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS	Bash History	A tion 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
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spearphishing via Service	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compile After Delivery	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Data from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Supply Chain	Exploitation for Client	Bootkit	Exploitation for Privilege	Compiled HTML File	Hooking	Password Policy	Remote File Copy	Email Collection	Domain Generation	Scheduled Transfer	Network Denial of
				ware	Input Capt						
				Model	Input Prom						
	LSASS Driver	Component Firmware	Hooking	Control Panel Items	Kerberoast						
	Launchctl	Component Object Model Hijacking	Image File Execution Options Injection	DCShadow	Keychain	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Stored Data Manipulation
	Local Job Scheduling	Create Account	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning and Relay	Remote System Discovery	Taint Shared Content		Multilayer Encryption		Transmitted Data Manipulation
	Mshta	DLL Search Order Hijacking	New Service	DLL Side-Loading	Network Sniffing	Security Software Discovery	Third-party Software		Port Knocking		
	PowerShell	Dylib Hijacking	Path Interception	Deobfuscate/Decode Files or Information	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote Access Tools		
	Regsvcs/Regasm	External Remote Services	Plist Modification	Disabling Security Tools	Private Keys	System Network Configuration Discovery	Windows Remote Management		Remote File Copy		
						System Network			Standard Application Layer		

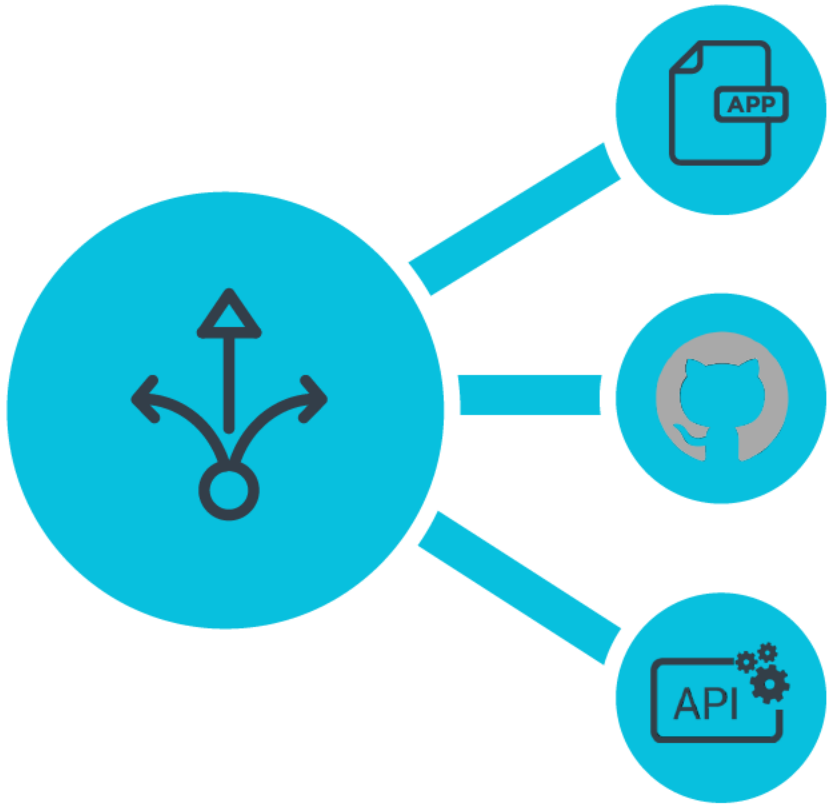
```
python attack_range.py -m terraform -a simulate -st T1003
```

```
python attack_range.py -m terraform -a search -sn "ESCU - Attempted Credential Dump From Registry via Reg.exe - Rule"
```

CI/CD Workflow – Detection Engineering



Deploy detections



Package detections into an app

Forking from a Git repository

Provide detections over REST API

Apply What You Have Learned Today

- Next week you should:
  - Download and install Attack Range or Mordor Labs
- In the first three months following this presentation you should:
  - Establish a CI/CD workflow for your SIEM detections
  - Continuously test your detections
- Within six months you should:
  - Share your detections with the InfoSec community
  - Establish automated testing of detections