Threat-Driven Development

with

Stratus Red Team

Ryan Marcotte Cobb

Principal Researcher, Secureworks

@detectdotdev

whoami

Link to Slides

https://bit.ly/3Vt100B

SPEAKER

Ryan Marcotte Cobb

Principal Information Security Researcher | Secureworks

@detectdotdev

Full-stack developer and researcher

Avid Pythonista with Jupyter notebooks

Based in Providence, Rhode Island

10 years of DFIR/hunting/purple teaming @ Secureworks

The Problem

Can we **detect** this threat?

Can we **still** detect this threat?

The Problem

Everything is in flux.
Detection engines are **complicated**.

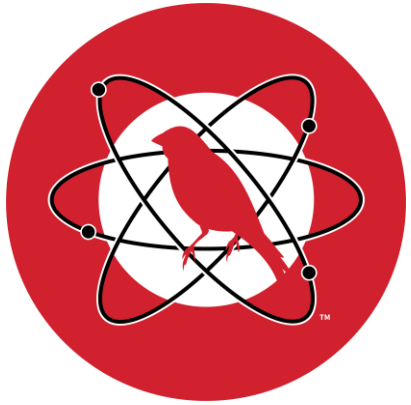
Testing with Automated Attacks

Emulated attacks performed in a **repeatable**, **consumable** and **actionable** way.

Compare predictable side effects with prior detonations of the same attack.

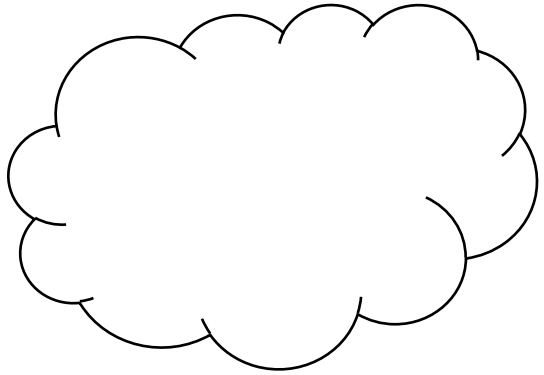
Used for calibration and **E2E testing** of detection engines.

Testing with Automated Attacks



Atomic Red Team

+



The Cloud(s)

=

?

	Atomic Red Team	?
Tests cases mapped to	ATT&CK techniques	Nascent ATT&CK cloud matrix
Test cases defined as	Configuration in YAML	?
Test runner uses	pwsh	?
Test infrastructure	BYOendpoint	Cloud-specific resources
Attack logic implementation	pwsh, lolbins, BYOmalware	?
Ships as	pwsh module, YAML repo	?

Stratus Red Team

	Atomic Red Team	Stratus Red Team
Tests cases mapped to	ATT&CK techniques	ATT&CK tactics
Test cases defined as	Configuration in YAML	Code in golang
Test runner uses	pwsh	golang
Test infrastructure	BYOendpoint	Cloud resources via Terraform
Attack logic implementation	pwsh, lolbins, BYOmalware	Golang using cloud provider SDKs
Ships as	pwsh module, YAML repo	Single-file native executable

Stratus Techniques and Providers

Stratus Red Team

Search

Stratus Red Team | User Guide | Attack Techniques Reference

Attack Techniques Reference

All Attack Techniques

Philosophy

Supported Platforms

Attack techniques

- AWS
- GCP
- Azure
- Kubernetes

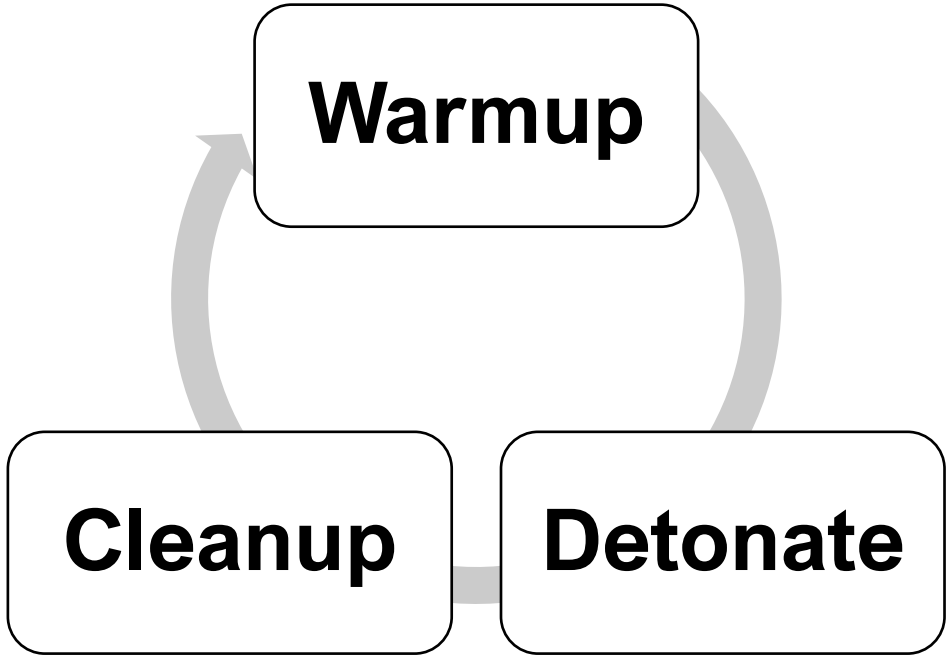
List of all Attack Techniques

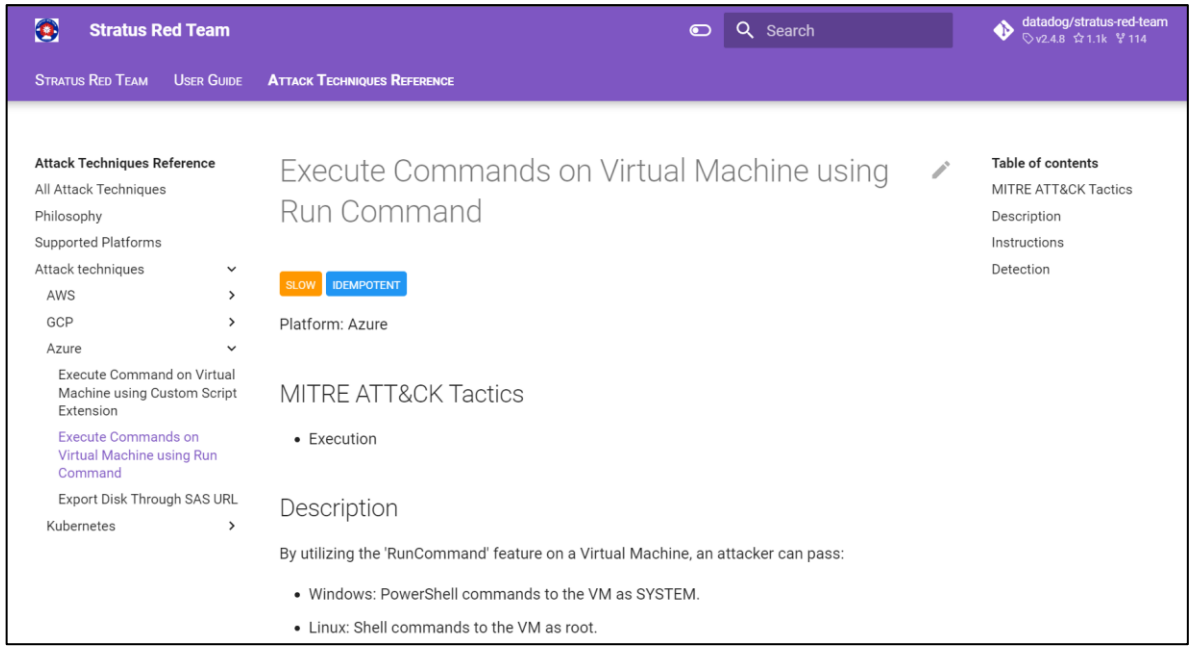
This page contains the list of all Stratus Attack Techniques.

Name	Platform	MITRE ATT&CK Tactics
Retrieve EC2 Password Data	AWS	Credential Access
Steal EC2 Instance Credentials	AWS	Credential Access
Retrieve a High Number of Secrets Manager secrets	AWS	Credential Access
Retrieve And Decrypt SSM Parameters	AWS	Credential Access
Delete CloudTrail Trail	AWS	Defense Evasion
Disable CloudTrail Logging Through Event Selectors	AWS	Defense Evasion

https://stratus-red-team.cloud/attack-techniques/list/

How Stratus Red Team Works

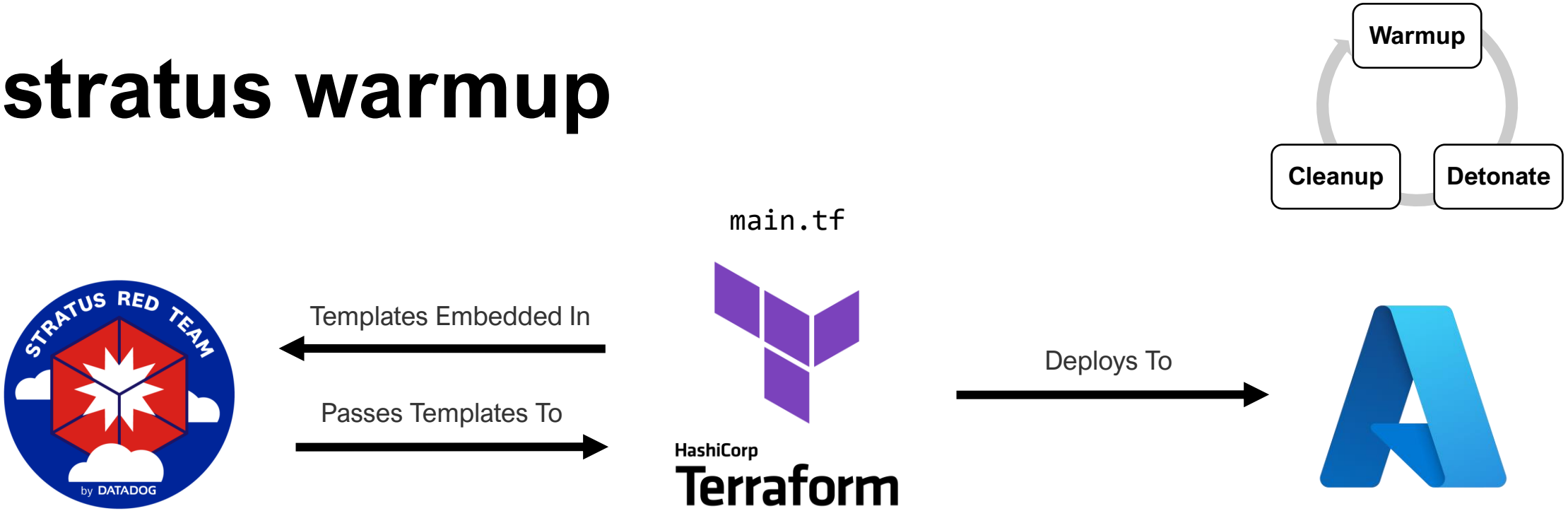




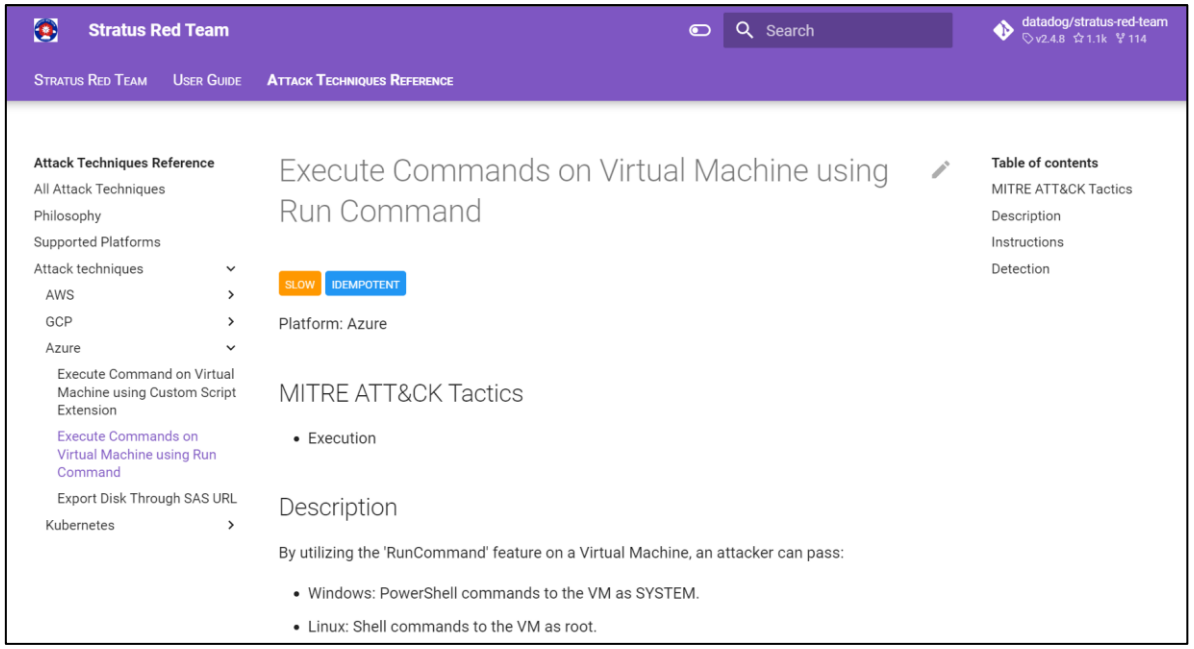
Example: `azure.execution.vm-run-command`

https://stratus-red-team.cloud/attack-techniques/azure/azure.execution.vm-run-command/

stratus warmup




```
C:\Users\rcobb\Desktop\repos\dash>stratus.exe warmup azure.execution.vm-run-command
2022/10/04 09:39:29 Checking your authentication against azure
2022/10/04 09:39:29 Installing Terraform in C:\Users\rcobb\.stratus-red-team\terraform
2022/10/04 09:39:33 Warming up azure.execution.vm-run-command
2022/10/04 09:39:33 Initializing Terraform to spin up technique prerequisites
2022/10/04 09:39:44 Applying Terraform to spin up technique prerequisites
2022/10/04 09:43:05 Virtual machine vm-i7c3plob ready in resource group rg-i7c3plob
```



Example: `azure.execution.vm-run-command`

main.tf

HashiCorp
Terraform

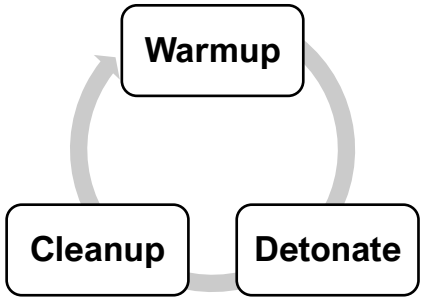
Deploys To

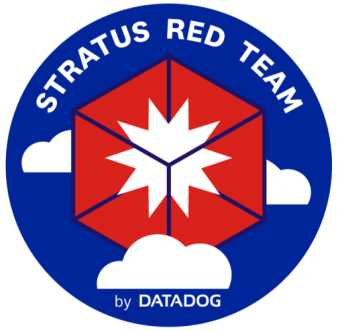
```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Resource Group
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

resource "azurerm_resource_group" "lab_environment" {
  name     = "rg-${random_string.lab_name.result}"
  location = "West US"
}

# # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Networking Resources
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

resource "azurerm_virtual_network" "lab_vnet" {
  name                = "vnet-${random_string.lab_name.result}"
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group
  resource_group_name = azurerm_resource_group
}

resource "azurerm_subnet" "lab_subnet" {
  name                 = "subnet-${random_stri
  resource_group_name  = azurerm_resource_grou
  virtual_network_name = azurerm_virtual_netwo
  address_prefixes     = ["10.0.2.0/24"]
}

resource "azurerm_network_interface" "lab_nic"
  name                = "nic-${random_string.l
  location            = azurerm_resource_group
  resource_group_name = azurerm_resource_group

  ip_configuration {
    name                          = "ip-${rand
    subnet_id                     = azurerm_su
    private_ip_address_allocation = "Dynamic"
  }
}
```

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Virtual Machine Resources
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

resource "azurerm_windows_virtual_machine" "lab_windows_vm" {
  name                = "vm-${random_string.lab_name.result}"
  resource_group_name = azurerm_resource_group.lab_environment.name
  location            = azurerm_resource_group.lab_environment.location
  size                = "Standard_F2"
  admin_username      = "local_admin_user"
  admin_password      = random_password.password.result
  user_data           = base64encode(random_string.lab_name.result)

  network_interface_ids = [
    azurerm_network_interface.lab_nic.id,
  ]

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  source_image_reference {
    publisher = "MicrosoftWindowsServer"
    offer     = "WindowsServer"
    sku       = "2016-Datacenter"
    version   = "latest"
  }
}
```

https://github.com/DataDog/stratus-red-team/blob/main/v2/internal/attacktechniques/azure/execution/vm-run-command/main.tf

stratus detonate





Dependency of

Attack Implemented In

main.go

azure-sdk-for-go

Performs Attack On

```
C:\Users\rcobb\Desktop\repos\dash>stratus.exe detonate azure.execution.vm-run-command
2022/10/04 10:05:14 Checking your authentication against azure
2022/10/04 10:05:14 Installing Terraform in C:\Users\rcobb\.stratus-red-team\terraform
2022/10/04 10:05:17 Not warming up - azure.execution.vm-run-command is already warm. Use --force to force
2022/10/04 10:05:17 Issuing Run Command for VM instance /subscriptions/3aa63cc3-c333-45e1-b06e-801c7bc0a3ac/
resourceGroups/rg-i7c3plob/providers/Microsoft.Compute/virtualMachines/vm-i7c3plob
2022/10/04 10:05:20 Waiting for command to be run on the VM
2022/10/04 10:06:02 Command successfully executed on the virtual machine
```

```go
func detonate(params map[string]string) error {
	vmObjectId := params["vm_instance_object_id"]
	vmName := params["vm_name"]
	resourceGroup := params["resource_group_name"]

	cred := providers.Azure().GetCredentials()
	subscriptionID := providers.Azure().SubscriptionID
	clientOptions := providers.Azure().ClientOptions

	log.Println("Issuing Run Command for VM instance " + vmObjectId)
	vmClient, err := armcompute.NewVirtualMachinesClient(subscriptionID, cred, clientOptions)
	runCommandInput := armcompute.RunCommandInput{
		CommandID: to.Ptr("RunPowerShellScript"),
		Script:    []*string{to.Ptr("Get-Service")},
	}
	if err != nil {
		return errors.New("unable to instantiate Azure virtual machine client: " + err.Error())
	}

	commandCreation, err := vmClient.BeginRunCommand(context.Background(), resourceGroup, vmName, runCommandInput, nil)
	if err != nil {
		return errors.New("unable to run a command on the virtual machine: " + err.Error())
	}

	log.Println("Waiting for command to be run on the VM")
	ctxWithTimeout, done := context.WithTimeout(context.Background(), 60*3*time.Second) // This can sometimes be quite slow
	defer done()
	commandResult, err := commandCreation.PollUntilDone(ctxWithTimeout, &runtime.PollUntilDoneOptions{Frequency: 2 * time.Second})
	if err != nil {
		return errors.New("unable to retrieve the output of the command ran on the virtual machine: " + err.Error())
	}

	_ = *commandResult.RunCommandResult.Value[0].Message // contains the output of the command executed
	log.Println("Command successfully executed on the virtual machine")
	return nil
}
```

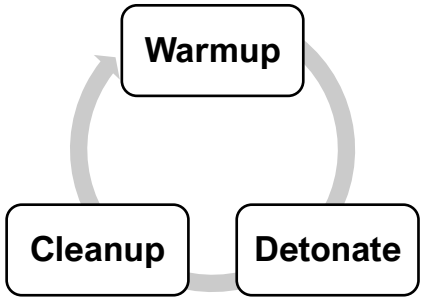
main.go

azure-sdk-for-go

Performs Attack On

https://github.com/DataDog/stratus-red-team/blob/main/v2/internal/attacktechniques/azure/execution/vm-run-command/main.go

stratus cleanup

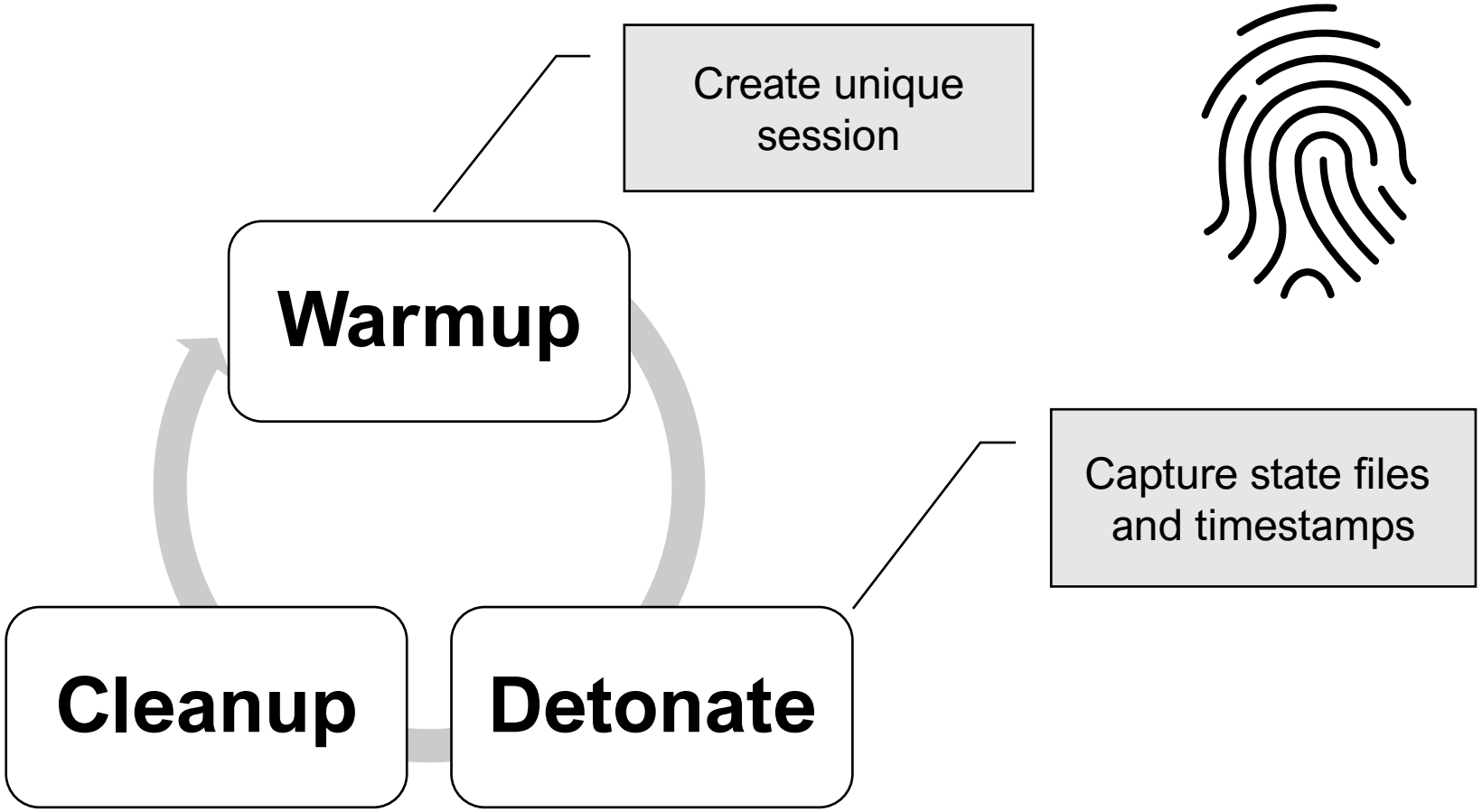


No need to maintain persistent testing infrastructure.

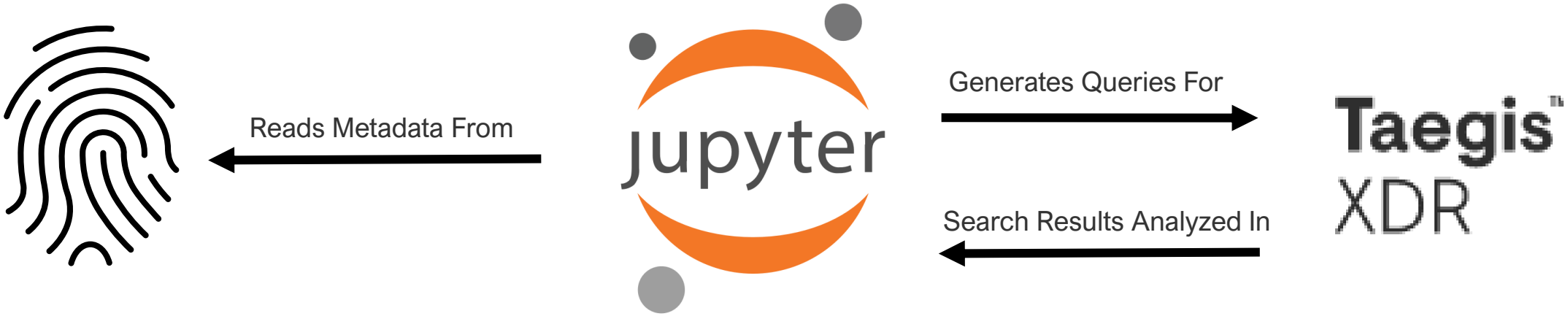
Testing with Stratus Red Team

1. Fingerprint
2. Detonate
3. Query XDR
4. Validate and Compare
5. Create XDR Investigation
6. Review and Tune
7. Rinse and Repeat

Detonation Fingerprinting



Query XDR for Detonation Side Effects




```
FROM cloudaudit
WHERE sensor_type contains 'AWS CloudTrail' AND (
    access_key contains 'ASIAVRCHE4GS2B5NXA6F' OR
    user_name contains 'credential-access.ec2-steal-instance-credentials-1646237061' OR
    original_data contains 'i-019fb6bd6ab61639a' OR
    original_data contains 'stratus-ec2-credentials-instance-role'
)
EARLIEST='2022-03-02 16:04:21'
LATEST='2022-03-02 17:04:53'
```

Validate and Compare

Alerts

Detectors are mapped to each Stratus technique name

Assert that the query results contain the relevant alerts

Events

Summarize results for each event type

Identify and diff against related investigations

Validate and Compare

Did we receive the **data in a timely fashion**?

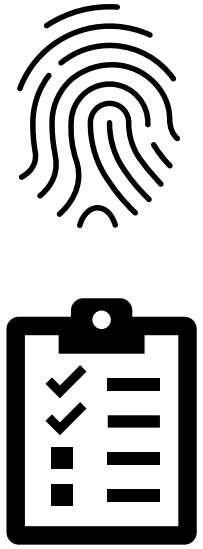
Do we see the **expected alerts** for this attack?

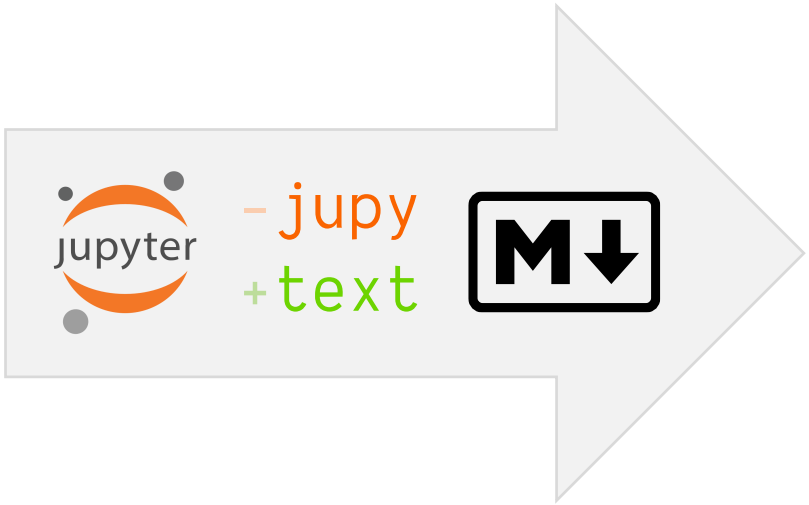
How do these side effects **compare to previous detonations** of the same attack?

pandas

```
#### Alerts Summary

{{ (
    detonation
    .related_alerts
    .query_results
    .pipe(display_summary, ["metadata.creator.detector.detector_id", "metadata.title"
) }}

#### Time to Detect

**Event Filters: Event Time to Ingest Time**

{{ (
    detonation
    .related_alerts
    .query_results
    .pipe(calculate_alert_timedeltas)
    .groupby("schema")["event_to_ingest_timedelta"]
    .describe()
    .to_html()
) }}

**Event Filters: Ingest Time to Alert Creation Time**

{{ (
    detonation
    .related_alerts
    .query_results
    .pipe(calculate_alert_timedeltas)
    .groupby("schema")["ingest_to_alert_creation_timedelta"]
    .describe()
    .to_html()
) }}

**Other Detectors: Ingest Time to Alert Creation Time**

{{ (
    detonation
    .related_alerts
    .query_results
    .pipe(calculate_alert_timedeltas)
    .groupby("metadata.creator.detector.detector_id")["ingest_to_alert_creation_timed
    .describe()
    .to_html()
) }}
```

Create XDR Investigation





Taegis™
XDR
Investigation

Query results **linked** to investigation.
Comparative summaries provided as markdown tables.
Investigation **assigned** to relevant security/product team.

Taegis XDR | CTU Dark Cloud - Testing | Quick Search

- Dashboards
- Alerts
- Investigations
- Advanced Search
- Endpoint Agents
- Integrations
- Automations
- Tools
- Downloads
- Reports
- Tenant Settings

Investigations

219	213	0	6	0	63
Total	New	Ongoing	Closed	Suspended	Archived

Showing 101 - 200 of 219 investigations

Export All | + Add New

NAME	ALERTS	EVENTS	STATUS	PRIORITY
2022-03-03 - [Stratus Automated Attack Scenario] - exfiltration.ec2-share-ebs-snapshot-1646238246	2	7	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - exfiltration.ec2-share-ebs-snapshot-1644938026	2	7	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - exfiltration.ec2-share-ami-1646238180	0	8	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - exfiltration.ec2-share-ami-1644938864	0	8	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - exfiltration.ec2-security-group-open-port-22-ingres-1646238112	3	14	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - exfiltration.ec2-security-group-open-port-22-ingres-1644611790	3	29	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - execution.ec2-user-data-1646237749	3	79	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - discovery.ec2-enumerate-from-instance-1646237621	2	42	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - discovery.ec2-enumerate-from-instance-1644940253	1	32	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - discovery.ec2-download-user-data-1646237738	0	27	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.vpc-remove-flow-logs-1646237476	1	17	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.organizations-leave-1646237450	0	13	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-lifecycle-rule-1646237378	7	30	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297	5	9	Open	Low
2022-03-03 - [Stratus Automated Attack Scenario] - credential-access.ssm-retrieve-securestring-paramet-1646237241	0	53	Open	Low

Columns | Filters

Items per page: 100 | 101 – 200 of 219

Taegis XDR | CTU Dark Cloud - Testing

Investigations / 2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297

2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297

SUMMARY | EVIDENCE | ENTITIES

Status: Open
Assignee: Ryan Cobb
Priority: Low
Type: ManagedXDR Threat Hunt
Author: Ryan Cobb
Created: 2022/03/03 17:51:26 UTC
Updated: 2022/10/04 15:10:33 UTC
Archived: N/A
Ticket: No associated ticket

KEY FINDINGS

Delete CloudTrail Trail

Platform: AWS

MITRE ATT&CK Tactics

- Defense Evasion

Description

Delete a CloudTrail trail. Simulates an attacker disrupting CloudTrail logging.

Warm-up:

- Create a CloudTrail trail.

Detonation:

- Delete the CloudTrail trail.

Instructions

```
stratus detonate aws.defense-evasion.cloudtrail-delete
```

Detection

Identify when a CloudTrail trail is deleted, through CloudTrail's `DeleteTrail` event.

GuardDuty also provides a dedicated finding type, Stealth:IAMUser/CloudTrailLoggingDisabled

Execution Details

- Start Time: 2022-03-02 16:08:17
- End Time: 2022-03-02 17:08:20
- Unique AWS Access Key ID: `ASIAVRCHE4GSSELAFXWR`
- Session Name: `defense-evasion.cloudtrail-delete-1646237297`

Execution Log

```
2022/03/02 16:08:19 Checking your authentica
2022/03/02 16:08:19 Not warming up - aws.def
2022/03/02 16:08:19 Deleting CloudTrail trai
```

Taegis XDR | CTU Dark Cloud - Testing

Investigations / 2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297

2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297

SUMMARY | EVIDENCE | ENTITIES

Status: Open
Assignee: Ryan Cobb
Priority: Low
Type: ManagedXDR Threat Hunt
Author: Ryan Cobb
Created: 2022/03/03 17:51:26 UTC
Updated: 2022/10/04 15:12:22 UTC
Archived: N/A
Ticket: No associated ticket

KEY FINDINGS

Review

Alerts Generated

creator	severity_category	message	count
app:event-filter	High	AWS CloudTrail trail deletion event was performed.	4
	Research	RESEARCH: AWS CloudTrail stop trail event StopLogging was performed.	1

Events Generated

sensor_id	event_source	event_name	count
arn:aws:cloudtrail:us-west-2:380253823397	cloudtrail.amazonaws.com	DeleteTrail	1
		DescribeTrails	3
	ec2.amazonaws.com	DescribeAccountAttributes	1

Event Timeline

event_time_usec	sensor_event_id	source_address	event_source	event_name	status
2022-03-02 16:08:19	bd87578b-d7e7-406f-b1e3-d4639dd8ed3f	162.84.137.138	ec2.amazonaws.com	DescribeAccountAttributes	Succeeded
					Succeeded
					Succeeded
					Succeeded
					Succeeded

Taegis XDR | CTU Dark Cloud - Testing

Investigations / 2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297

2022-03-03 - [Stratus Automated Attack Scenario] - defense-evasion.cloudtrail-delete-1646237297

SUMMARY | EVIDENCE | ENTITIES

ALERTS (5) | EVENTS (9) | AGENTS (0) | SEARCHES (2) | ATTACHMENTS (0) | HISTORY

5 alerts

CREATED AT	TITLE	DETECTOR NAME	SENSOR TYPE	MITRE ATT&CK
2022/03/02 16:11:12 UTC	AWS CloudTrail trail deletion event was performed.	TDR Watchlist	AWS CloudTrail	
2022/03/02 16:11:12 UTC	RESEARCH: AWS CloudTrail stop trail event StopLogging was performed.	TDR Watchlist	AWS CloudTrail	
2022/03/02 16:11:12 UTC	AWS CloudTrail trail deletion event was performed.	TDR Watchlist	AWS CloudTrail	
2022/03/02 16:11:12 UTC	AWS CloudTrail trail deletion event was performed.	TDR Watchlist	AWS CloudTrail	
2022/03/02 16:11:12 UTC	AWS CloudTrail trail deletion event was performed.	TDR Watchlist	AWS CloudTrail	

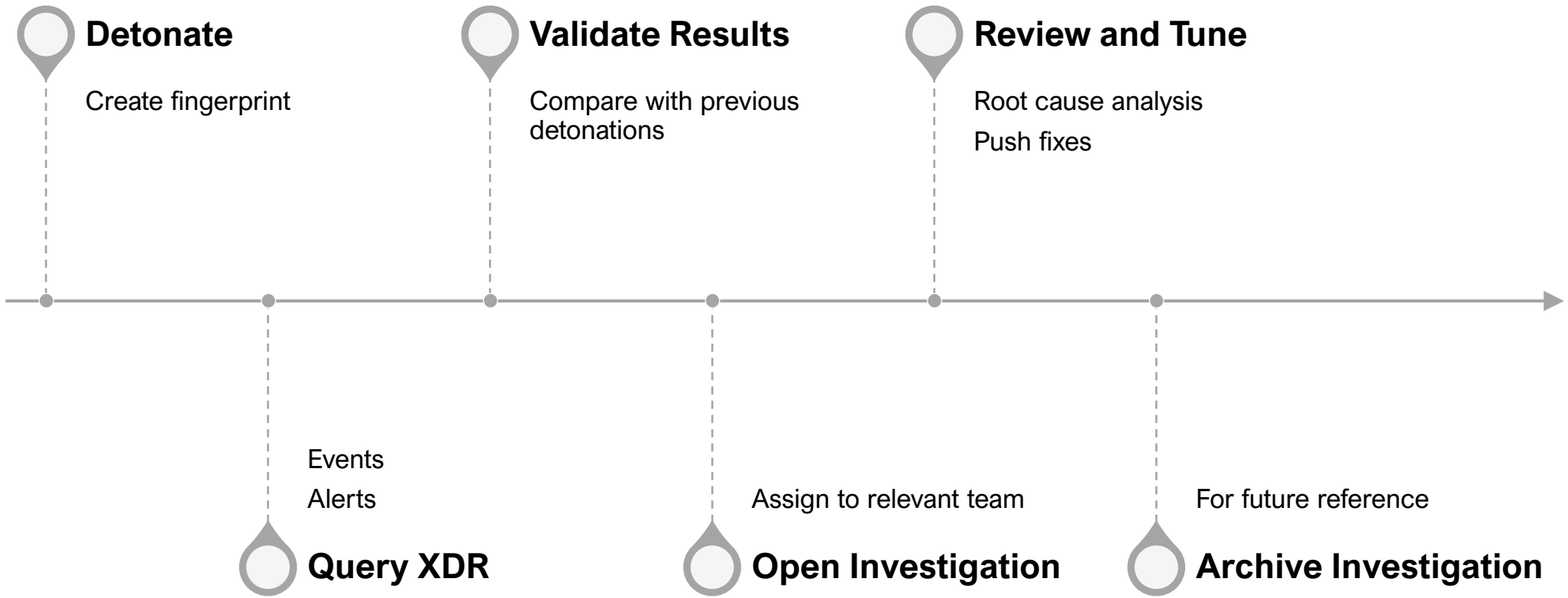
jupyter

Review and Tune

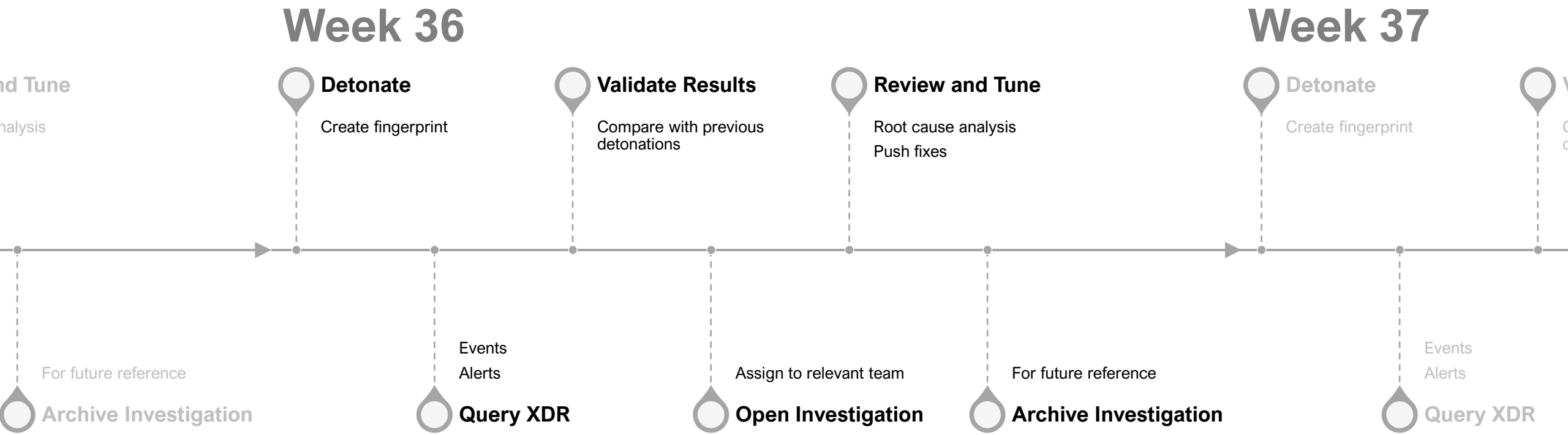
Relevant team performs **root cause analysis**.
Make changes to monitoring, logic, pipeline, or **expectations**.
Ad hoc re-detonation to validate changes.
Leverages existing investigation-based workflows.

========== Tests Passed ==========

Rinse and Repeat



Rinse and Repeat



Lessons Learned

Good	Bad
Stratus Red Team works great	Manually running detonator scripts and Jupyter notebooks
Proactively identified and remediated gaps	e2e tests are inherently more flakey, time-consuming to investigate
Automated attacks as a new team deliverable	Scaling up and across multiple deployments
Leveraging XDR investigation workflow made it easy to share and save results	Rudimentary way to make assertions

Into the Future

```
Feature: Demo for bdd-detect
    An example attack using `stratus` and `pytest-bdd` to validate
    detection pipeline works as expected.

    Scenario: Detonate Azure VM Run Command
        Given we detonate stratus attack technique "azure.execution.vm-run-command"
        And we wait "15" "minutes"
        When we query azure activity logs
        Then we should see azure activity events containing
            """
            ['Microsoft.Compute/virtualMachines/runCommand/action'] in `operation_name.value`
            """
```

Inspirations:
https://github.com/pytest-dev/pytest-bdd
https://cucumber.io/
http://gauntlt.org/

```
vscode ➜ /workspaces/detect-dot-dev/bdd-detect (main ✗) $ pytest --gherkin-terminal-reporter src/bdd_detect/scenarios.py
============================================================================ test session starts ================================
==
platform linux -- Python 3.9.2, pytest-7.1.2, pluggy-1.0.0
rootdir: /workspaces/detect-dot-dev/bdd-detect, configfile: pytest.ini
plugins: anyio-3.6.1, bdd-5.0.0, respx-0.19.2
collected 2 items


src/bdd_detect/scenarios.py::test_detonate_azure_vm_run_command
------------------------------------------------------------------------------ live log call ------------------------------------
--
2022-10-04 15:31:41 INFO Executing stratus technique: azure.execution.vm-run-command
2022-10-04 15:31:41 INFO Sleeping 15 minutes until 2022-10-04 15:46:41.768575
2022-10-04 15:31:41 INFO Sleeping...
2022-10-04 15:31:44 INFO Sleeping...
2022-10-04 15:31:47 INFO Sleeping...
```

pytest-bdd is a very promising solution to some of these challenges.

Recap

e2e testing for detection systems using automated attacks

Stratus Red Team and how it works

Automating XDR investigations based on Stratus detonations

Lessons learned and next steps

Thank You!

Link to Slides

https://bit.ly/3Vt100B

SPEAKER

Ryan Marcotte Cobb

Principal Information Security Researcher | Secureworks

@detectdotdev