Google Workspace Malicious App Script analysis (english only)

By OWN

Published: 2024-04-25 · Archived: 2026-04-05 23:09:19 UTC

In this article, OWN-CERT presents a study at App Script, a development platform for creating add-ons to Google services used by the enterprise or automating certain actions. Through an example of persistence techniques using App Script, will illustrate the compromise chain by an attacker and the analyst's investigation possibilities. We will also explore methods for collecting and analyzing logs, along with ways to identify these malicious actions, either manually or through custom detection rules, to enhance detection and response capabilities.

Executive Summary

Google Workspace is a cloud platform designed to facilitate collaboration and communication among individuals, offering a variety of services including App Script, a development platform for creating add-ons to Google services used by the enterprise or automating certain actions.

• App Script becomes a critical service to investigate for identifying malicious activities on the platform following an account compromise.

• Real-time detection is feasible, and numerous events can serve as indicators of suspicious behaviors that analysts should verify.

• Understanding the threat, potential techniques, and investigation methods is crucial for quickly responding to an attacker: collecting logs, parsing them, and knowing what to look for.

In this article, an example of persistence techniques using App Script will illustrate the compromise chain by an attacker and the analyst's investigation possibilities. We will also explore methods for collecting and analyzing logs, along with ways to identify these malicious actions, either manually or through custom detection rules, to enhance detection and response capabilities

Introduction to Google Workspace

Google Workspace is a comprehensive suite of online tools developed by Google to promote collaboration and communication within businesses, universities, associations, and even for individuals. The suite includes various applications for communicating via email or instant messaging, managing calendars and organizing video conferences, storing and sharing files...

Google Workspace attracts businesses by offering various subscription plans with different levels of functionality depending on the chosen model, catering to startups, SMEs, or large corporations.

Many security measures are available within the environment and can be activated by administrators to secure accounts and services. These include enhancements in:

• User account security: 2FA (Two-Factor Authentication), password policy management, login session control, use of contextual access rules.

• Email security with options to activate restrictive measures on senders, creation of filtering rules, management of quarantine zones.

• File management including automatic content analysis and application of appropriate policies based on criticality level, protection against data leaks.

And what interests us most in the context of our blog post: the logs.

Many events performed on the platform generate logs which are stored online without the ability for a user, whether he is an administrator or not, to alter them by modifying or deleting them.

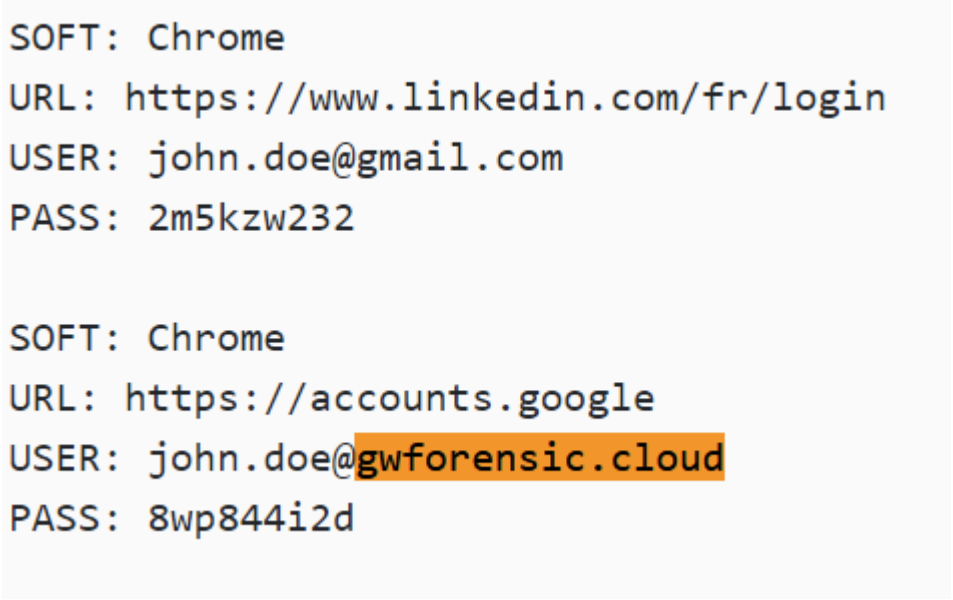
As a result, security teams can establish monitoring based on these sources to detect malicious behaviors within the cloud environment and respond to alerts in real-time, whether they are automatically generated by Google via the alert center (such as reclassification of emails as spam, device compromise detection...) or by security tools installed by the company like SIEM (Security Information and Event Management).

Compromise scenario

In the case of a compromise of a Cloud account, an attacker may recover the password using several methods, including the reuse of stolen credentials obtained through infostealers, a method that has been widely employed for some time.

In our scenario, we will play both the role of the attacker, by compromising an account and carrying out malicious actions, as well as the role of the security analyst, who will analyze the compromised account to identify malicious actions through the generated events.

We can hypothesize a compromise of a personal computer from which the employee logged into their professional account to check emails and documents. Once compromised by an infostealer, the credentials were exfiltrated via a log collection platform and then resold to other parties.



Preview of stolen credentials rendered by an infostealer in an exfiltrated log file.

Following the detection of a suspicious login from outside, log collection and analysis were performed on the suspicious account to understand what happened and implement remediation measures. We used the "GW Forensic", an internal tool published on our GitHub to collect and analyze Google Workspace logs. The tool is available at this link: https://github.com/ownsecurity/GWForensic/.

We save time through the automatic collection and indexing of logs within the OpenSearch tool. The analyst can focus on searching for malicious traces guided by the initial automated review of events by GW Forensic.

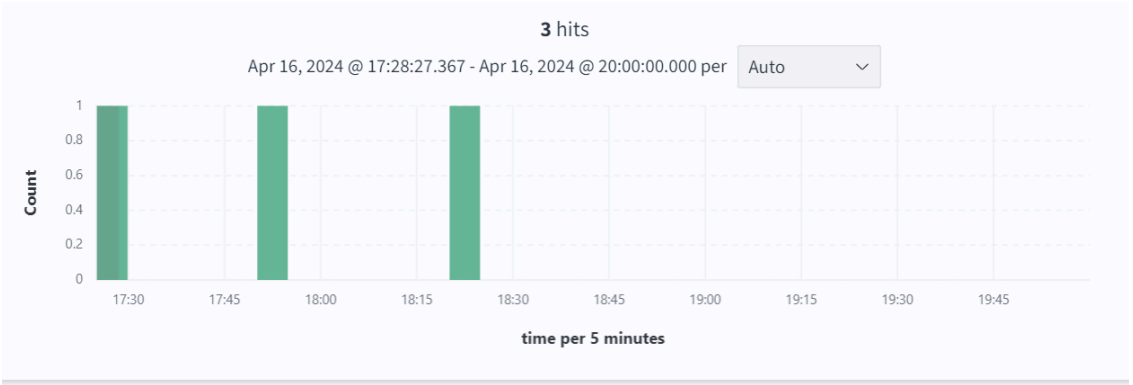
Configuration used in this scenario:

```
sources:
  - "all"
users:
  - john.doe@gwforensic.cloud
date:
  start: "2024-04-06T15:00:00.120Z"
  end: "2024-04-16T23:00:00.120Z"
export: "opensearch"
exportFolder: "./export/"
opensearch:
  url: "opensearch-node"
  port: 9200
  user: "admin"
  password: "admin"
  index_name: "gwforensic-invest-johndoe"
```

Launching GW Forensic with the detailed configuration above.

- Suspicious login

By filtering on the "login" service, we found 3 related events during the incident timeframe:



Columns Sort fields

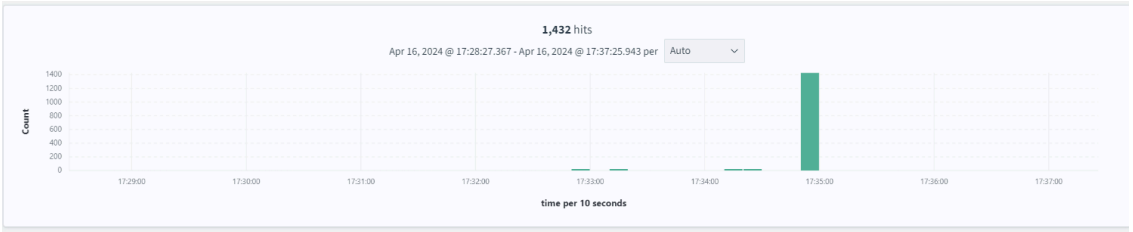
Time (time)	email	applicationName	eventsName	ipAddress
Apr 16, 2024 @ 18:23:11.103	john.doe@gwforensi…	login	logout	173.44.36.176
Apr 16, 2024 @ 17:51:37.059	john.doe@gwforensi…	login	login_success	173.44.36.176
Apr 16, 2024 @ 17:29:55.119	john.doe@gwforensi…	login	login_success	173.44.36.176

Events related to the "login" service on the account john.doe.

A connection was made to the account *john.doe@gwforensic.cloud* at 17:29:55+02:00 from 173.44.36[.]176, located in Miami (USA) and owned by a free VPN company, while the user is located and working in France. A reauthentication occurred 22 minutes later at 17:51:37+02:00, followed by a disconnection at 18:23:11+02:00.

- File download

Shortly after the malicious login, we observed several accesses to documents located on the Drive as well as the download of numerous files:



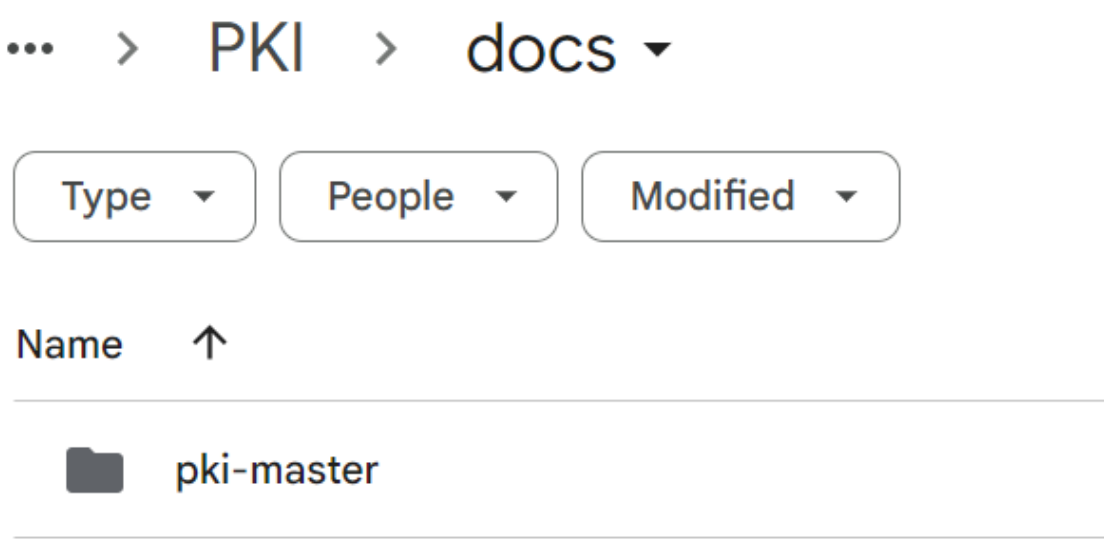
Columns Sort fields

Time (time)	email	applicationName	eventsName	ipAddress	doc_title	tactics	technique
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	caInternalAuthDRMstorag…	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	OpenSans-Italic-webfont.…	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	TryAgainButton.gif	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	CMCResponse.1.md	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	caSigningUserCert.cfg	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	.pydevproject	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	setup.py	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	conf.py	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	sslserver.conf	Exfiltration	Exfiltration Over Web Ser…
Apr 16, 2024 @ 17:34:54.753	john.doe@gwforensic.clo…	drive	download	173.44.36.176	404.html	Exfiltration	Exfiltration Over Web Ser…

Significant volume of documents downloaded from Drive.

GW Forensic automatically tagged these events as a potential exfiltration technique "**Exfiltration Over Web Services**".

Upon analyzing the documents on the collaborator's Drive, we found that the majority came from a folder named "pki-master":



Folder "pki-master" on the Drive of user john.doe.

Downloading a folder on Google Drive (right-click > Download) creates a ZIP file containing all the files and folders for the user. However, in the Workspace logs, this action generates a download event for each file, hence the significant volume observed on the graph.

- Set up of a malicious script

We notice the creation, through the "change_user_access" modification event, of a file named "Projet sans Titre" referring to a Google App Script:

doc_id	1tu1oWj6WtOePUg7hyUpfX8AQszyxTnZ3JYmCjxVlLILUAm4bVacdyCm8
doc_title	Projet sans titre
doc_type	script
email	john.doe@gwforensic.cloud
eventsName	change_user_access
eventsType	acl_change

File details.

It is renamed within the following minute to "curriculum_vitae.txt" to blend in with other files.

Time (time)	email	applicationName	eventsName	ipAddress	doc_title
Apr 16, 2024 @ 18:13:49.588	john.doe@gwforensic.clo…	drive	edit	173.44.36.176	curriculum_vitae.txt
Apr 16, 2024 @ 18:01:12.610	john.doe@gwforensic.clo…	drive	edit	173.44.36.176	curriculum_vitae.txt
Apr 16, 2024 @ 18:00:56.985	john.doe@gwforensic.clo…	drive	edit	173.44.36.176	curriculum_vitae.txt
Apr 16, 2024 @ 17:50:53.337	john.doe@gwforensic.clo…	drive	edit	173.44.36.176	curriculum_vitae.txt
Apr 16, 2024 @ 17:44:56.262	john.doe@gwforensic.clo…	drive	edit	173.44.36.176	curriculum_vitae.txt
Apr 16, 2024 @ 17:42:45.556	john.doe@gwforensic.clo…	drive	rename	173.44.36.176	curriculum_vitae.txt
Apr 16, 2024 @ 17:41:15.552	john.doe@gwforensic.clo…	drive	change_user_access	/	Projet sans titre

Actions on the script.

Several edits are made to the file until 18:13:49+02:00. Alongside these actions, the logs from the "token" service in Google Workspace indicate an "**authorize**" event related to the script name "curriculum_vitae.txt":

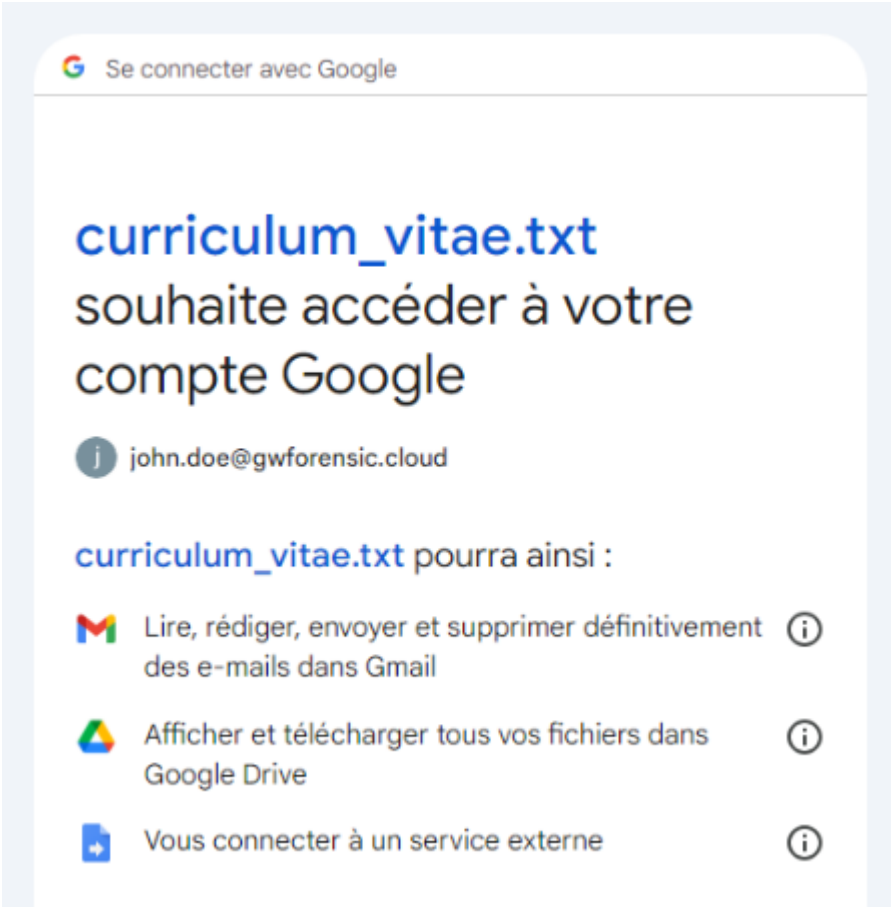
Time (time)	email	applicationName	eventsName	ipAddress	app_name	tactics	technique
Apr 16, 2024 @ 17:52:32.074	john.doe@gwforensic.cloud	token	authorize	173.44.36.176	curriculum_vitae.txt	Credential Access	Steal Application Access T…

"Authorize" event on the account of john.doe.

The event, annotated here as the "Credential Access" tactic, corresponds to granting permission to an application/script via an OAuth token to access user data: profile information, email action rights, calendar access rights, etc.

Here, the requested accesses correspond to those needed by the script within the code.

During the initial execution, the attacker manually authorized it to access certain services as shown in the pop-up:



Access request pop-up for the application "curriculum_vitae.txt".

The advantage for the attacker is that the access request only needs to be done once: the token is generated and can then be reused by the application if it is not revoked.

Nine minutes after the last execution of the “curriculum_vitae.txt “ script, a "**create_script_trigger**" event is generated and classified as a persistence method in the form of a scheduled task.

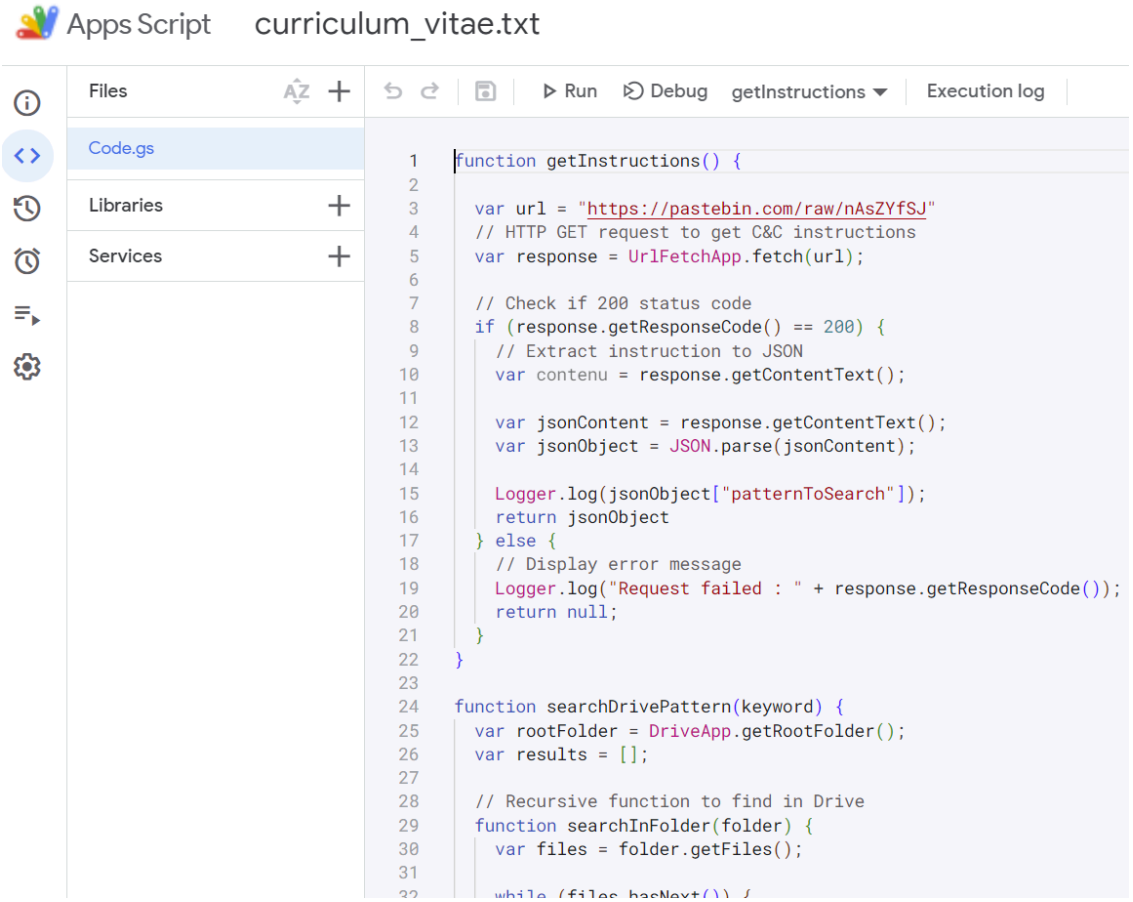
Time (time)	email	applicationName	eventsName	ipAddress	doc_title	tactics	technique
Apr 16, 2024 @ 18:22:35.859	john.doe@gwforensic.clo…	drive	create_script_trigger	173.44.36.176	curriculum_vitae.txt	Persistence	Scheduled Task/Job

Creation of a scheduled task by the attacker through the circumvented use of App Script.

Through this task, the script will be able to run without notifying the user at a frequency defined by the trigger configured in the script project: every minute, every hour, daily, weekly, etc.

If the token generated for the application is revoked, the script execution will fail.

After gaining access rights to the script (via an administrator account), it is possible to open and access the script content:



Preview of the Google Apps Script IDE page containing the script in the file "Code.gs".

We notice the presence of several functions within the script file:

• GetInstructions: function that retrieves instructions from a PasteBin file using the UrlFetchApp function of App Script (More information about UrlFetchApp on the Google official documentation)

• SearchDrivePattern: function that searches for a pattern inside the names of files stored on the Drive

• GetPath: function that returns the full path of a file

• SearchGmailPattern: function that searches for a pattern inside the subject or body of emails

• Main: function that calls the other mentioned functions.

The script contacts the URL configured within its “GetInstructions” function to retrieve the keywords for searching the drive and the exfiltration method. Below is the content of the retrieved Pastebin file which corresponds to the instructions that the script will follow:

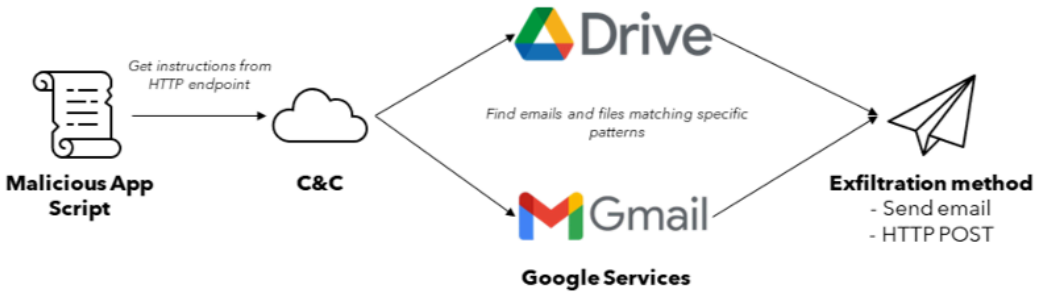
```
{
   "patternToSearch": [
      {
         "value": "pass: ",
         "service": "gmail"
      },
      {
         "value": "project",
         "service": "drive"
      }
   ],
   "export": {
      "method": "gmail",
      "value": "anonymous@yopmail.com"
   }
}
```

We observe the following instructions:

• Search for the string patterns "**project**" on the Drive service, and "**pass:**" for the Gmail service.

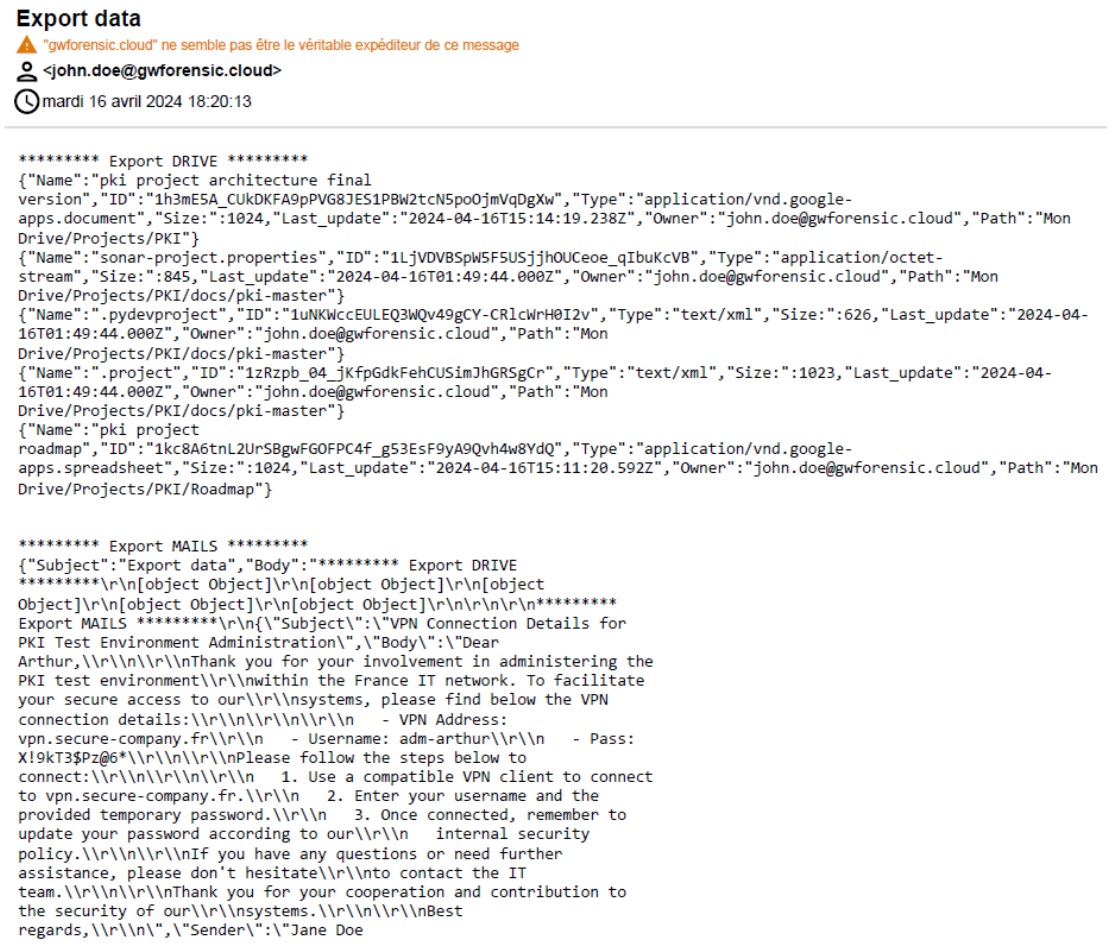
• Exfiltration of the stolen data via email to a temporary email address.

Upon analyzing the script content, the following process resembles the operation of a SaaS infostealer:



Infostealer actions.

The information regarding files and emails is then sent to the attacker's temporary address:



Export data

"gwforensic.cloud" ne semble pas être le véritable expéditeur de ce message

<john.doe@gwforensic.cloud>

mardi 16 avril 2024 18:20:13

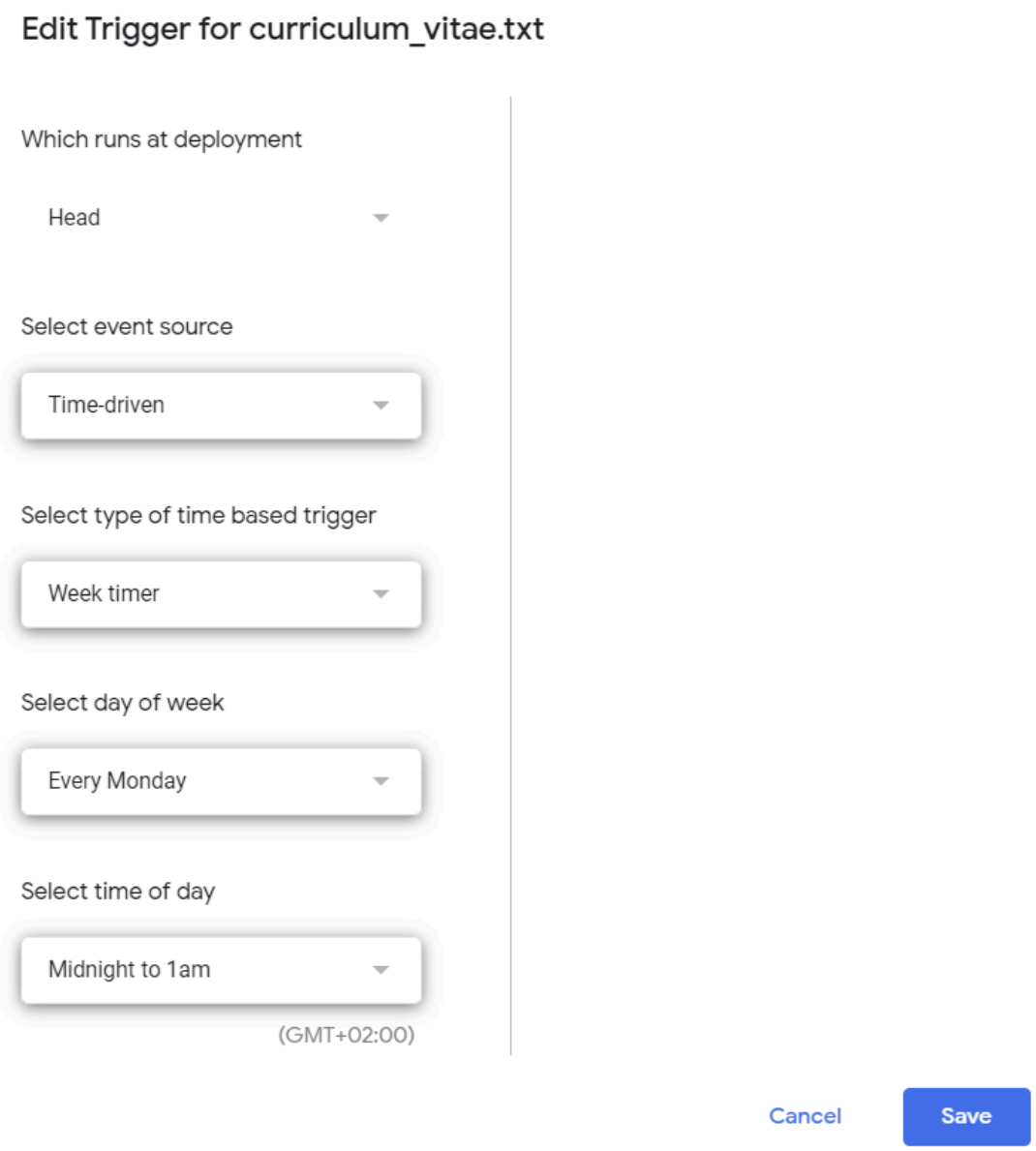

```
********* Export DRIVE *********
{"Name":"pki project architecture final
version","ID":"1h3mE5A_CUkDKFA9pPVG8JES1PBW2tcN5poOjmVqDgXw","Type":"application/vnd.google-
apps.document","Size:":1024,"Last_update":"2024-04-16T15:14:19.238Z","Owner":"john.doe@gwforensic.cloud","Path":"Mon
Drive/Projects/PKI"}
{"Name":"sonar-project.properties","ID":"1LjVDVBSpW5F5USjjhOUCeoe_qIbuKcVB","Type":"application/octet-
stream","Size:":845,"Last_update":"2024-04-16T01:49:44.000Z","Owner":"john.doe@gwforensic.cloud","Path":"Mon
Drive/Projects/PKI/docs/pki-master"}
{"Name":".pydevproject","ID":"1uNKWccEULEQ3WQv49gCY-CRlcWrH0I2v","Type":"text/xml","Size:":626,"Last_update":"2024-04-
16T01:49:44.000Z","Owner":"john.doe@gwforensic.cloud","Path":"Mon
Drive/Projects/PKI/docs/pki-master"}
{"Name":".project","ID":"1zRzpb_04_jKfpGdkFehCUSimJhGRSgCr","Type":"text/xml","Size:":1023,"Last_update":"2024-04-
16T01:49:44.000Z","Owner":"john.doe@gwforensic.cloud","Path":"Mon
Drive/Projects/PKI/docs/pki-master"}
{"Name":"pki project
roadmap","ID":"1kc8A6tnL2UrSBgwFGOFPC4f_g53EsF9yA9Qvh4w8YdQ","Type":"application/vnd.google-
apps.spreadsheet","Size:":1024,"Last_update":"2024-04-16T15:11:20.592Z","Owner":"john.doe@gwforensic.cloud","Path":"Mon
Drive/Projects/PKI/Roadmap"}


********* Export MAILS *********
{"Subject":"Export data","Body":"********* Export DRIVE
*********\r\n[object Object]\r\n[object Object]\r\n[object
Object]\r\n[object Object]\r\n[object Object]\r\n\r\n\r\n\r\n*********
Export MAILS *********\r\n{\"Subject\":\"VPN Connection Details for
PKI Test Environment Administration\",\"Body\":\"Dear
Arthur,\\r\\n\\r\\nThank you for your involvement in administering the
PKI test environment\\r\\nwithin the France IT network. To facilitate
your secure access to our\\r\\nsystems, please find below the VPN
connection details:\\r\\n\\r\\n\\r\\n   - VPN Address:
vpn.secure-company.fr\\r\\n   - Username: adm-arthur\\r\\n   - Pass:
X!9kT3$Pz@6*\\r\\n\\r\\nPlease follow the steps below to
connect:\\r\\n\\r\\n\\r\\n   1. Use a compatible VPN client to connect
to vpn.secure-company.fr.\\r\\n   2. Enter your username and the
provided temporary password.\\r\\n   3. Once connected, remember to
update your password according to our\\r\\n   internal security
policy.\\r\\n\\r\\nIf you have any questions or need further
assistance, please don't hesitate\\r\\nto contact the IT
team.\\r\\n\\r\\nThank you for your cooperation and contribution to
the security of our\\r\\nsystems.\\r\\n\\r\\nBest
regards,\\r\\n\",\"Sender\":\"Jane Doe
```


Preview of the email received by the attacker with the exfiltrated data.

Yes, the output isn't very pretty, but it's just a proof of concept! 😀

In the triggers page of the IDE, we find the scheduled task that generated the "**create_script_trigger**" event:



Preview of the trigger configuration.

Recommendations

To protect against the abuse of the App Script functionality, there are various methods detailed below:

Enable "Advanced Program Protection"

Google Workspace offers the "Advanced Program Protection" feature (Documentation), which can be activated on domain accounts to enhance security and prevent access to sensitive data. Enabling this feature has several consequences, such as:

• Mandatory use of 2FA (Two-Factor Authentication)

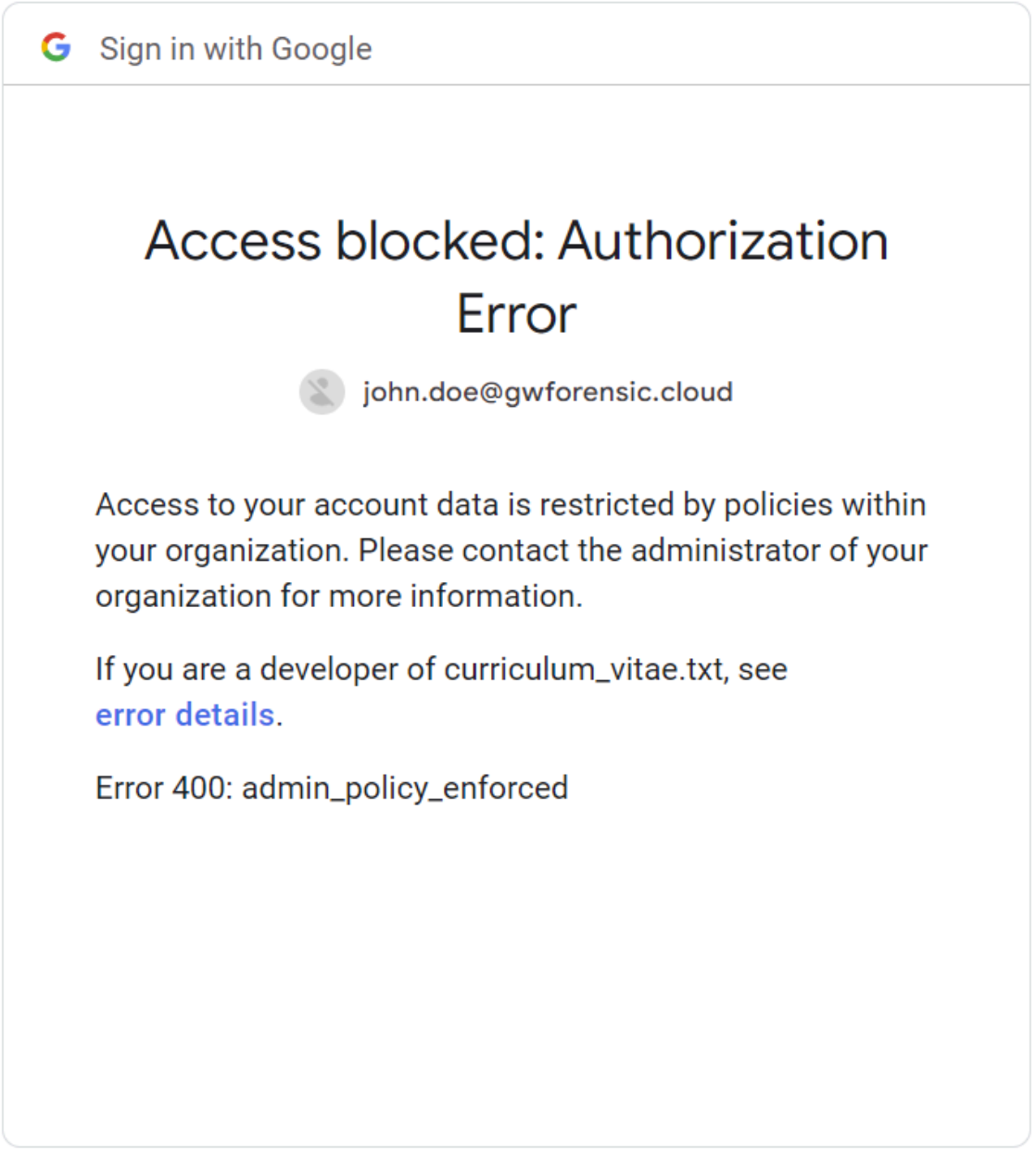
• Additional security for browsing in the Google Chrome browser, especially regarding downloads.

• Restricted access to data and services by third-party applications (via OAuth tokens)

Enable third-party app filtering

By default, each user can use their Google account on third-party applications that work via Google authentication. This allows them to grant access to their profile (name, email address) to an application for login purposes, avoiding traditional manual registration. Some applications may also access documents in Drive, such as file conversion SaaS applications like DOCX to PDF, JPEG to PDF, etc.

It is possible to enable a filtering feature or a list of authorized applications to manually approve which applications are allowed to access domain user data. This helps reduce the use of accounts on external sites and thus the risk of illegitimate access via third-party applications. App Script scripts will be blocked by default.



Preview of blocking third-party applications including App Script.

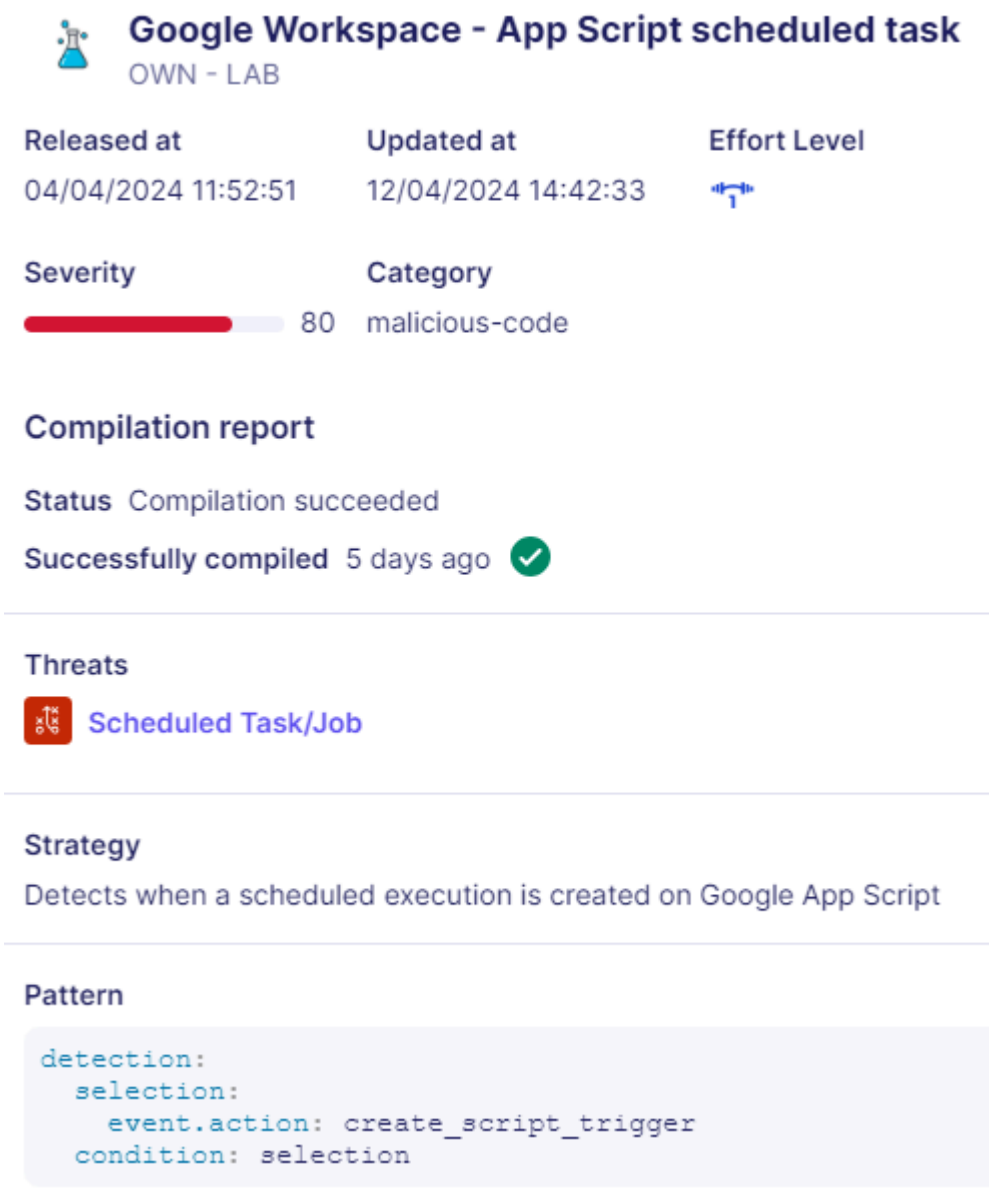
Monitor Google Workspace events in real time

It is possible to collect Google Workspace logs and analyze them in real time to identify suspicious events for investigation as early as possible. There are several Workspace sources: login, token, drive, calendar, users... where a significant number of events could correspond to compromise risks or malicious actions.

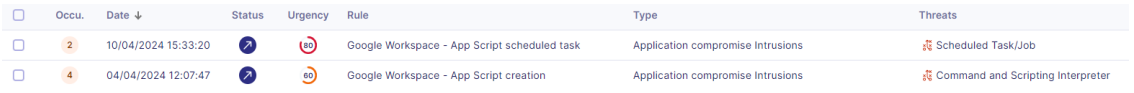
The GW Forensic project contains documentation with use cases to monitor and suggestions for detection based on existing events.

More information: https://github.com/ownsecurity/GWForensic/

It is possible to define SIGMA detection rules to integrate into log analysis tools like the Sekoia XDR solution. Once the Google Workspace connectors are integrated into the platform, it is possible to create SIGMA rules dedicated to Google Workspace events. In our lab, several rules have been created, including a detection rule related to the use of App Script:



Custom rule "App Script scheduled task" on Sekoia XDR.



Overview of alerts generated related to abusive use of App Script: script creation and execution scheduling.

Approximately 20 SIGMA rules are currently available and free to use in the GW Forensic project documentation to assist analysts in detecting malicious behaviors in real time on Google Workspace platforms.

More information: https://github.com/ownsecurity/GWForensic/
