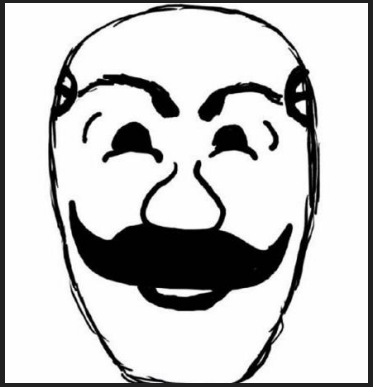
RATs & Socks abusing Google Services

Google Calendar RAT: Infrastructure-less Command&Control and GSSocks

Whoami

Valerio “MrSaighnal” Alessandroni

- Offensive Security Lead at EY Italy
- 10 years of experience as Pentester & Offensive Security Specialist
- Former Military
- Holder of multiple certifications: OSCP, OSEP, OSWE, OSWP, eWPTX, eCPTX, eCPPT, CEH, CRTO etc.
- Advanced Persistent Tortellini Crew Member
- Brazilian Jiu Jitsu Practitioner
- Passionate about space exploration



https://www.linkedin.com/in/valerio-alessandroni/

https://github.com/mrsaighnal

https://blog.keephack.ing

https://x.com/mrsaighnal

Index

1. Google Calendar RAT (GCR)
2. GCR - Technical Discussion
3. Security Considerations
4. DEF CON Bonus - Socks5 Over Google Services

1. Google Calendar RAT (GCR)

"Hacking is like art. It's about taking something that already exists and making it do something that it was never intended to do"

Dan Kaminsky

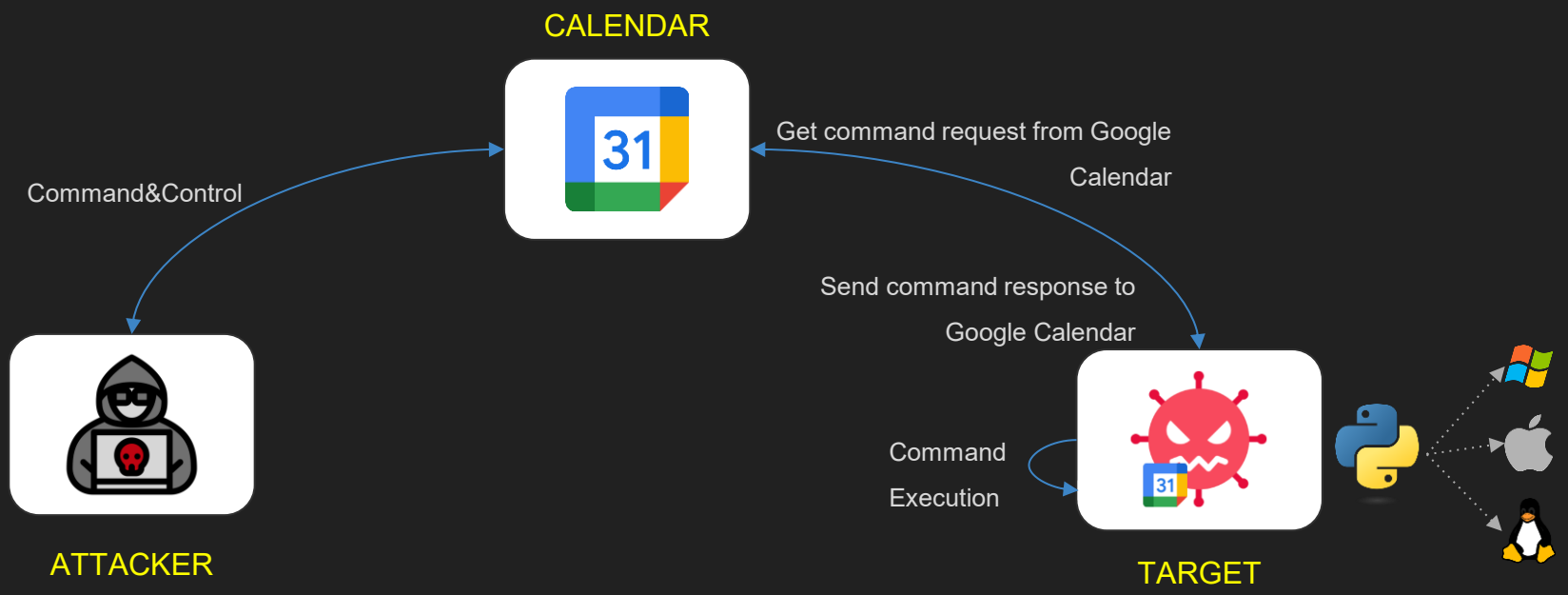
1. Google Calendar RAT

Initial Idea

- Perform C2 Without the Hassle of Building Infrastructure
- Save Time and Budget by Leveraging Existing Services
- Turn Trusted Services into C2 Channels (Living Off the Land)
- Developing a Tool which stays under the radar
- Researching New, Creative C2 Techniques
- Exploring Innovative Ideas While Keeping It Fun

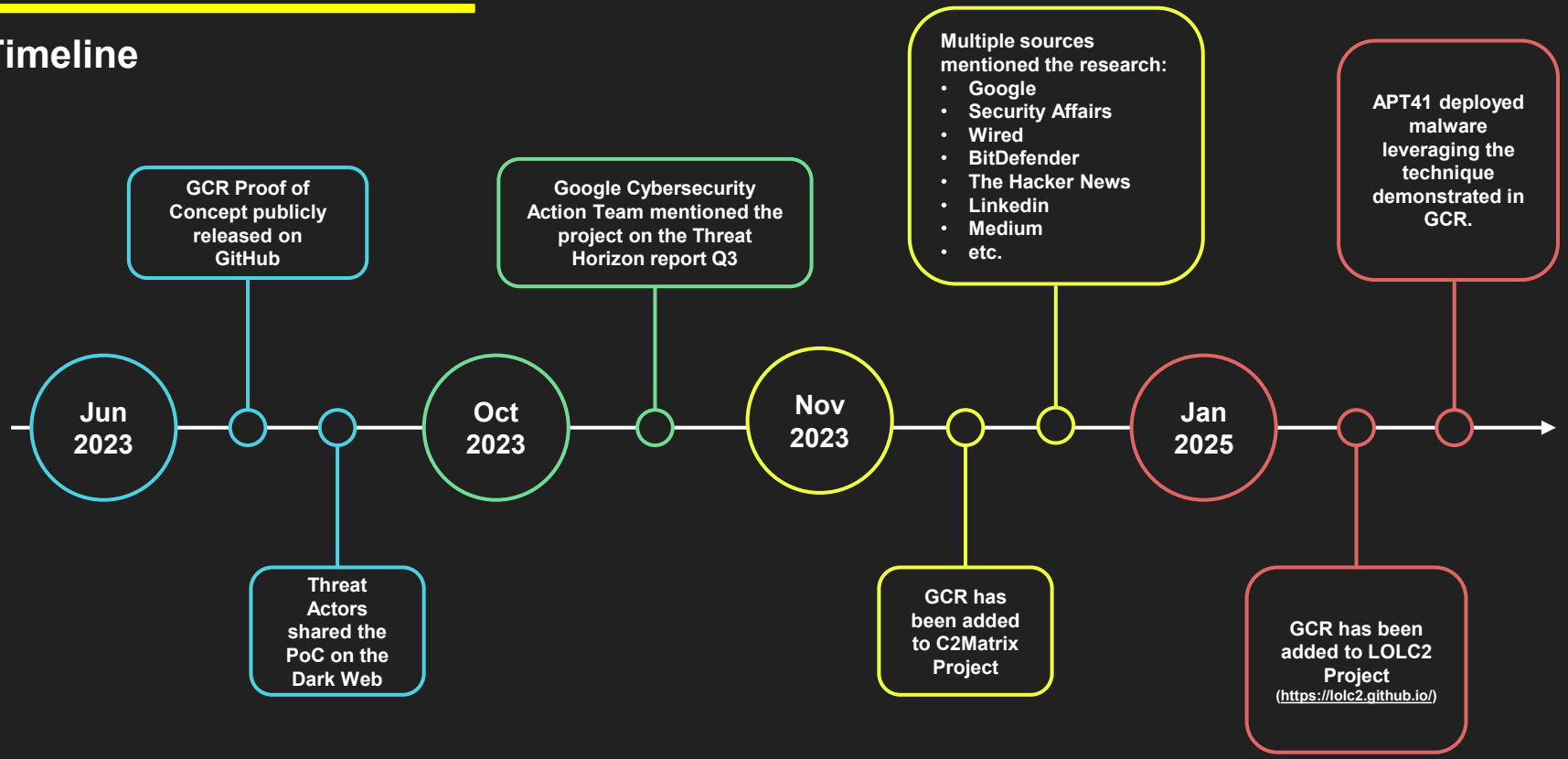
1. Google Calendar RAT

GCR Diagram Flow



1. Google Calendar RAT

Timeline



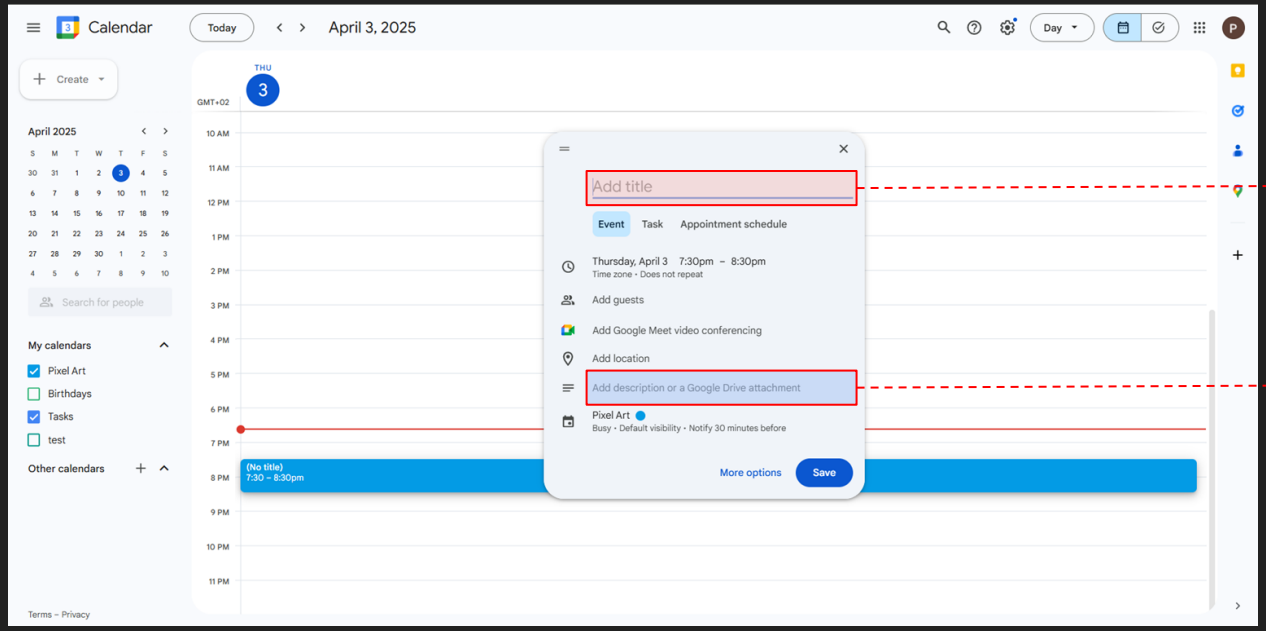
2. GCR - Technical Details

"Simplicity is the ultimate sophistication"

Steve Jobs

2. GCR - Technical Details

Google Calendar

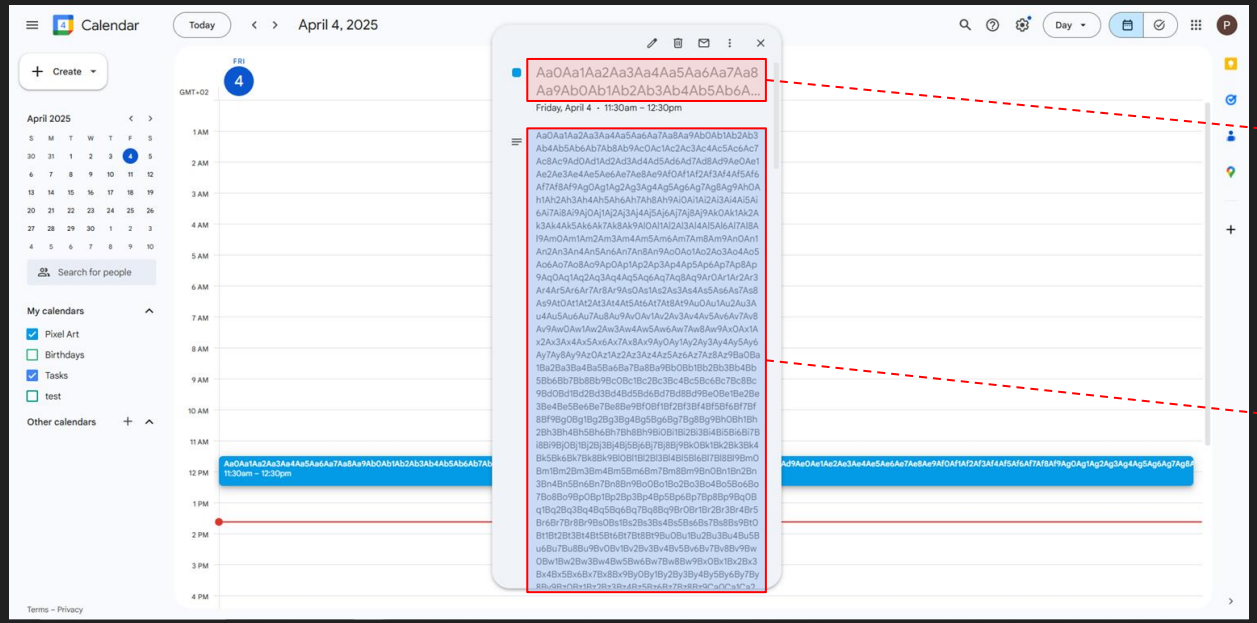


TITLE Field

DESCRIPTION Field

2. GCR - Technical Details

Google Calendar as a Shared Database



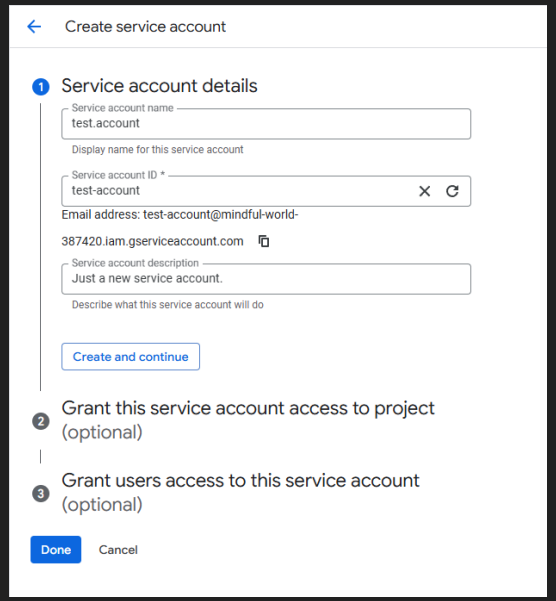
TITLE
Field
max length
1024 characters

DESCRIPTION
Field
max length
8191 characters

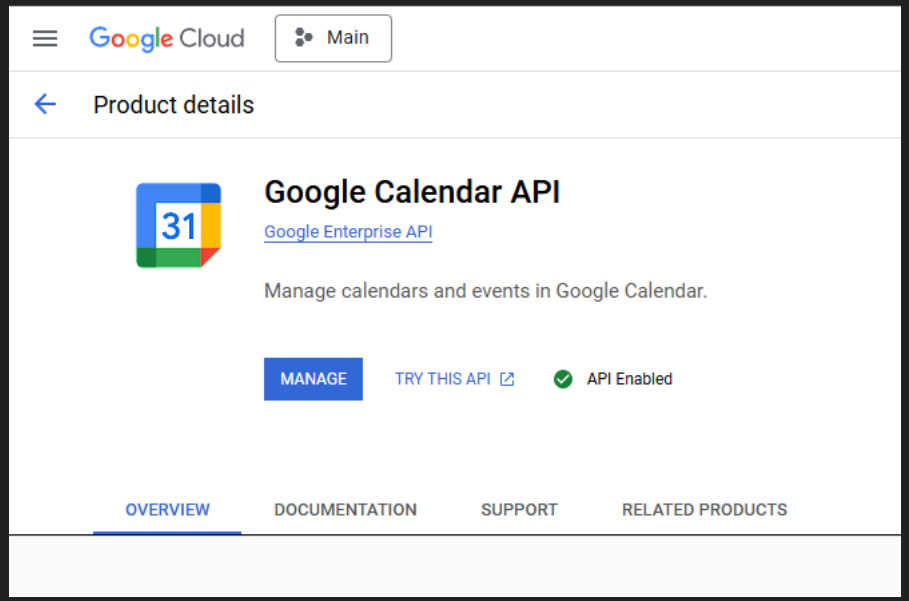
2. GCR - Technical Details

Google Calendar Setup

Service Account Creation



Enable Google API



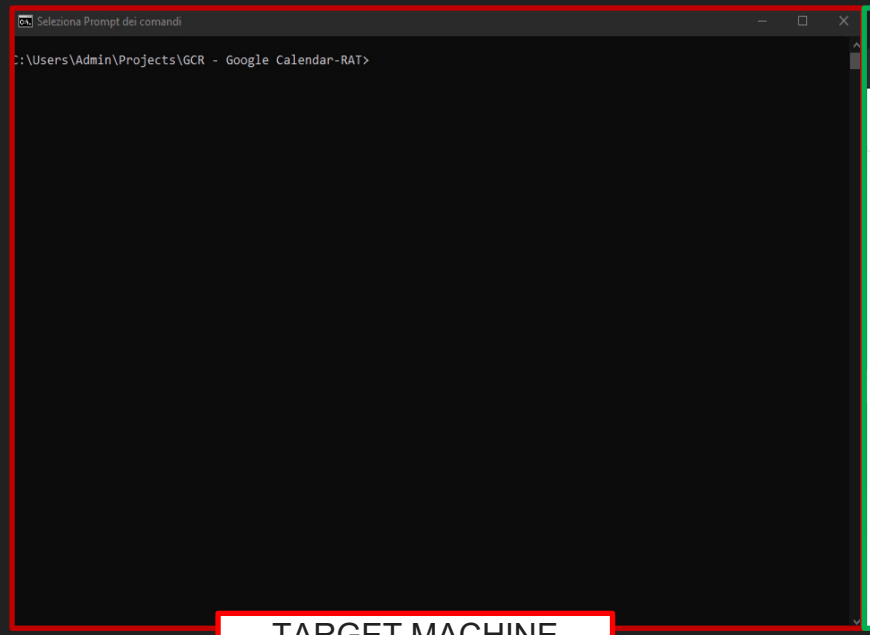
2. GCR - Technical Details

Google Calendar RAT Setup

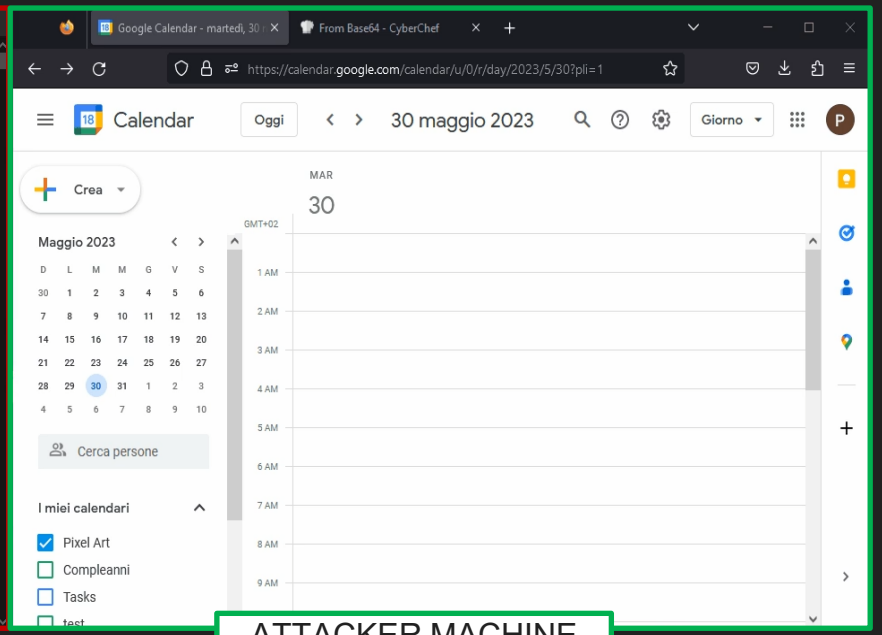
```
import subprocess
import hashlib
import socket
import uuid
import time

c2Calendar = "PUT_YOUR_CALENDAR_ADDRESS_HERE"#example "mycalendar@gmail.com"
pollingTime = 0

def print_banner():
    banner = """


                                        GOOGLE-CALENDAR-RAT - POC

                                        Command&Control via Google Calendar Events

                                        by: Valerio "MrSaighnal" Alessandroni
                                        https://github.com/MrSaighnal/GCR-Google-Calendar-RAT




    """
    print(banner)
    time.sleep(1.5)
```

2. GCR - Technical Details

Google Calendar RAT Demo



TARGET MACHINE



ATTACKER MACHINE

VIDEO LINK: https://github.com/MrSaighnal/DEFCON33/blob/main/slide_13_GCR_demo.mp4

3. Security Considerations

"Simplicity is the ultimate sophistication"

Steve Jobs

3. Security Considerations

Analysis

Pros

- Infrastructure-less mechanism
  - No need to buy domain name
  - No need to buy Server and/or VPS
  - No need to make a history for domain or IP (trust making)
  - Exploit Google Trust
- Hard to be detected via traffic inspection
- High availability (Thanks to Google Infrastructure)
- Traffic is encrypted by default (HTTPS)
- High anonymity by using Google as “proxy”

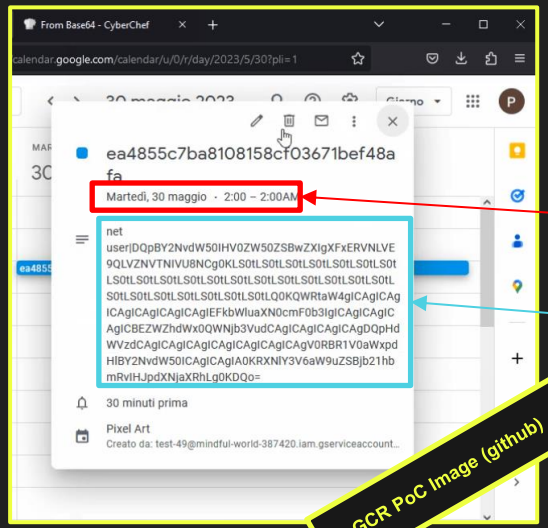
Cons

- Google domains are a single point of failure
  - Not all enterprises allow interaction with Google domains (due to policy or DLP restrictions)
- Limited to HTTPS Protocol
- Limited to 443 Port
- Google APIs quota limit
- Polling based communication

3. Security Considerations

From PoC to APT: GCR-like C2 Observed in the Wild

In 2025, **APT41** used **Google Calendar** as a **C2 channel**, with a method strikingly **similar** to what was demonstrated in the **GCR Poc** and later attributed to their malware **TOUGHPROGRESS** *.



GCR PoC Image (github)

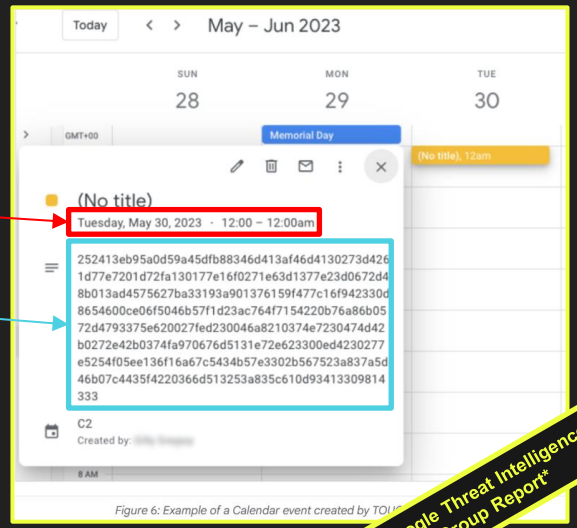


Figure 6: Example of a Calendar event created by TOU

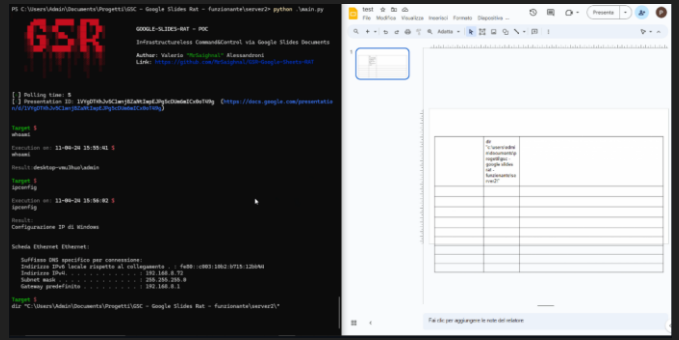
Google Threat Intelligence Group Report*

- Similar technique leveraging the event description field
- Same date (May 30, 2023)

*https://cloud.google.com/blog/topics/threat-intelligence/apt41-innovative-tactics

3. GCR - Technical Discussion

Other C2s Abusing Legitimate Services



Google Slides RAT

Known Abused Google Services

Gmail

Drive

Sheets

Slides

Calendar

Translate

Limitations

- Limited to an asynchronous request-response communication model for command execution
- Does not support TCP socket redirection
- Cannot tunnel other network protocols
- Unable to handle multiple simultaneous connections

4. DEF CON Bonus - Socks5 Over Google Services

“Hackers produce new concepts, perceptions, and sensations out of the raw data of existence”

McKenzie War

A Hacker Manifesto (2004)

4. Socks5 Over Google Services

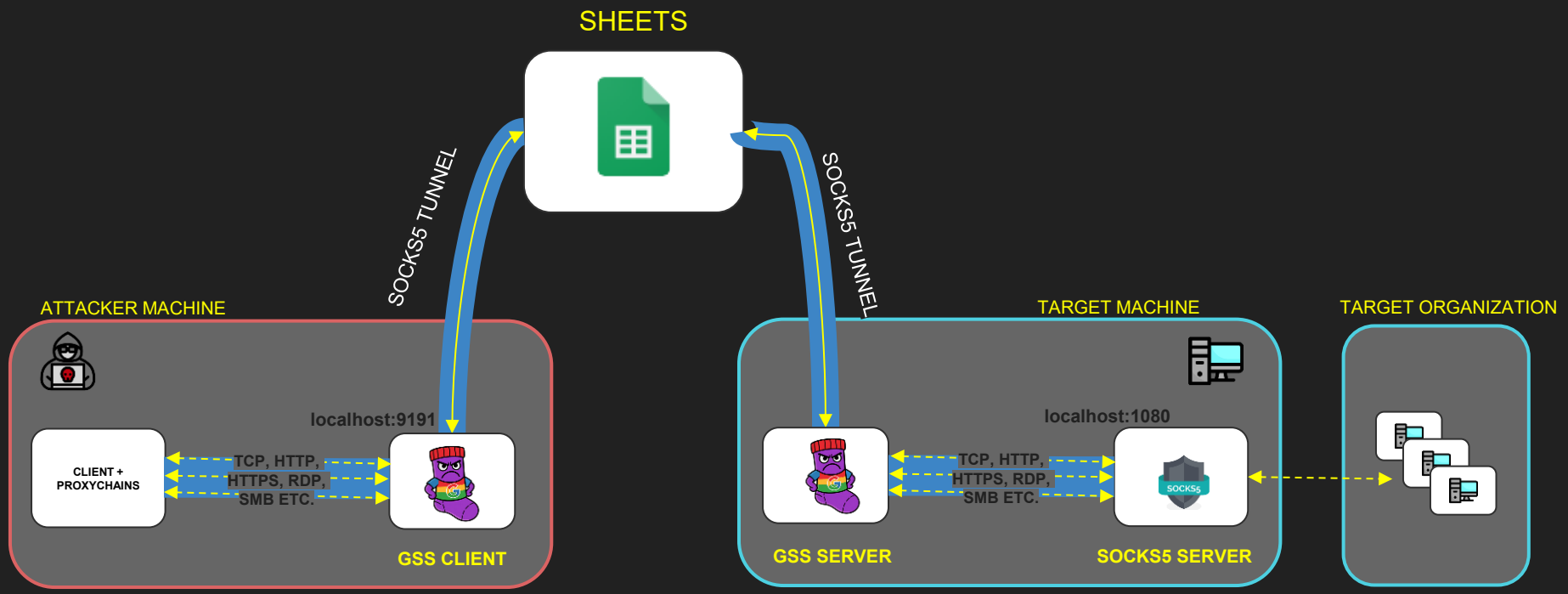
Presenting Google Sheets Socks (GSSocks)

- Post-Initial Access Tool which aims to stay under the radar
- Multiplatform Client and Server written in Go
- Provide SOCKS5 over Google Sheets.
  Usable via proxychains or other tools
- Multiplexing mechanism.
  Multiple bidirectional connections.

GSSocks

4. Socks5 Over Google Services

Data Flow Analysis

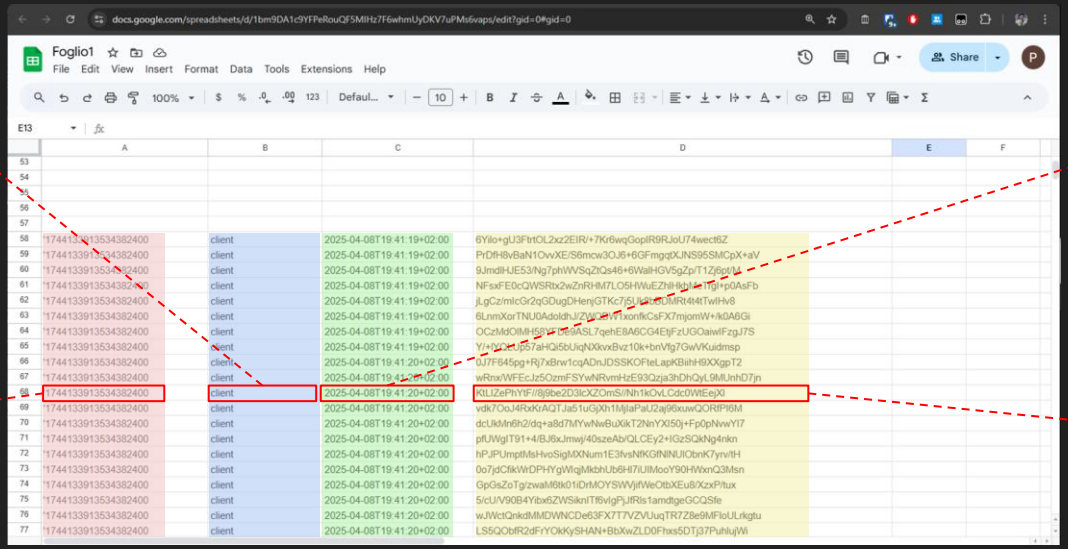


4. Socks5 Over Google Services

Google Sheets as a Shared Database

Sender label: client/server

Socket ID

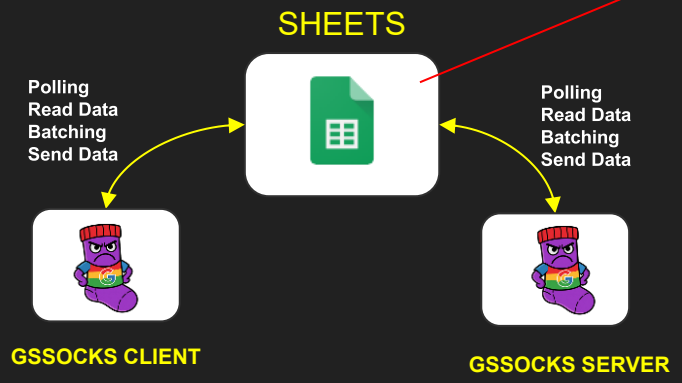


Chunk Timestamp

Base64 encoded data chunk

4. Socks5 Over Google Services

Limitations & Solutions/Optimizations



LIMITATIONS

- ~33% overhead introduced by base64 encoding.
- Maximum of 5000 characters per cell.
- Google quota limit per account (60 requests per minute) which resulted in very slow data transmission.

SOLUTIONS & OPTIMIZATION

- Splitting data to fill up to 5000 characters per cell.
- Batching read, write, and delete operations to reduce the number of requests.
- Implemented a rotational account system to increase the bitrate.

BEFORE

By using **1** account for the Client and **1** account for the Server:

- About 20 minutes to execute PSexec via Proxychains
- About 15 minutes to execute SecretsDump via proxychains

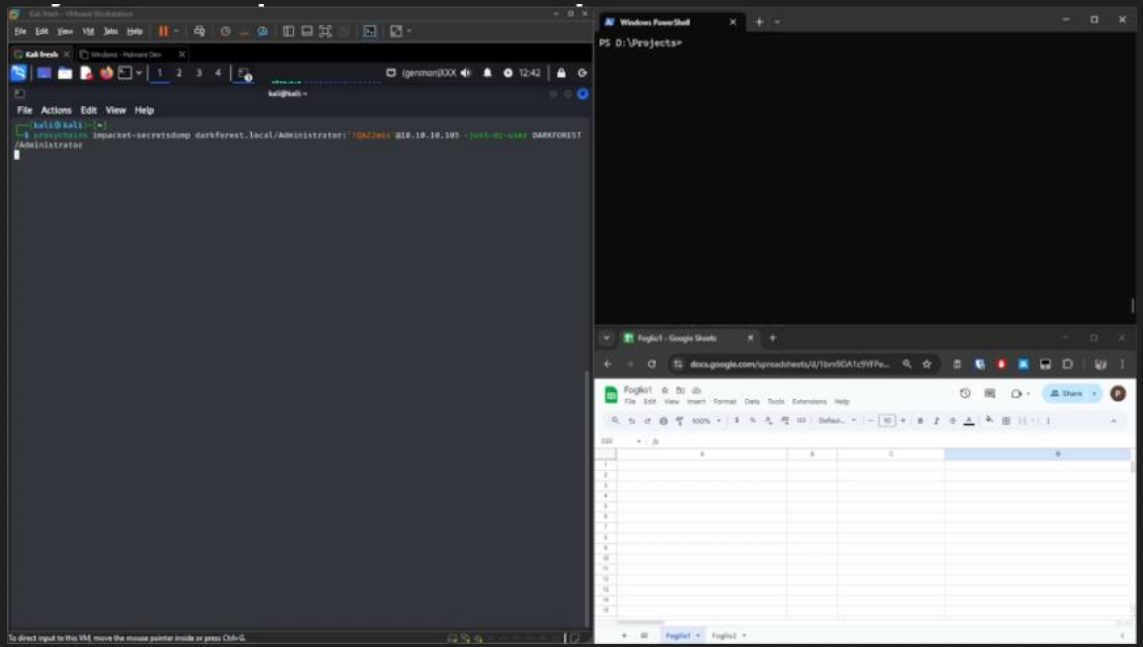
AFTER OPTIMIZATION

By using **4** account for the Client and **3** account for the Server:

- About 2 minutes to execute PSexec via Proxychains
- About 1 minute to execute SecretsDump via proxychains

4. Socks Over Google Services

Demo Proxychains + Impacket-SecretsDump

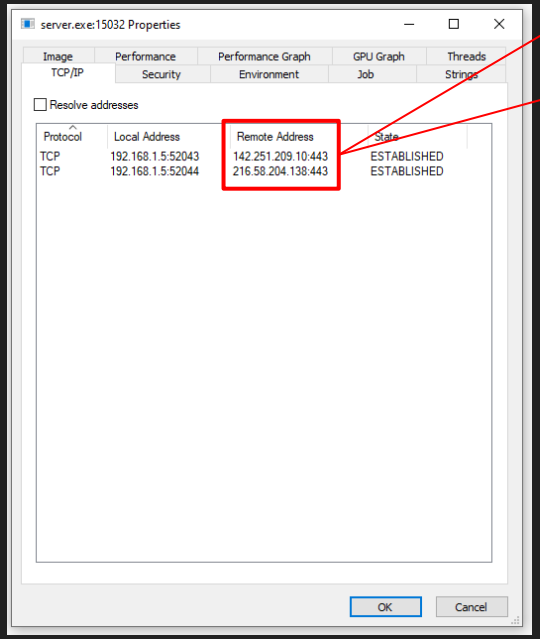


VIDEO LINK: https://github.com/MrSaighnal/google-sheets-socks/blob/main/video/slide_23_SecretsDump.mp4

4. Socks Over Google Services

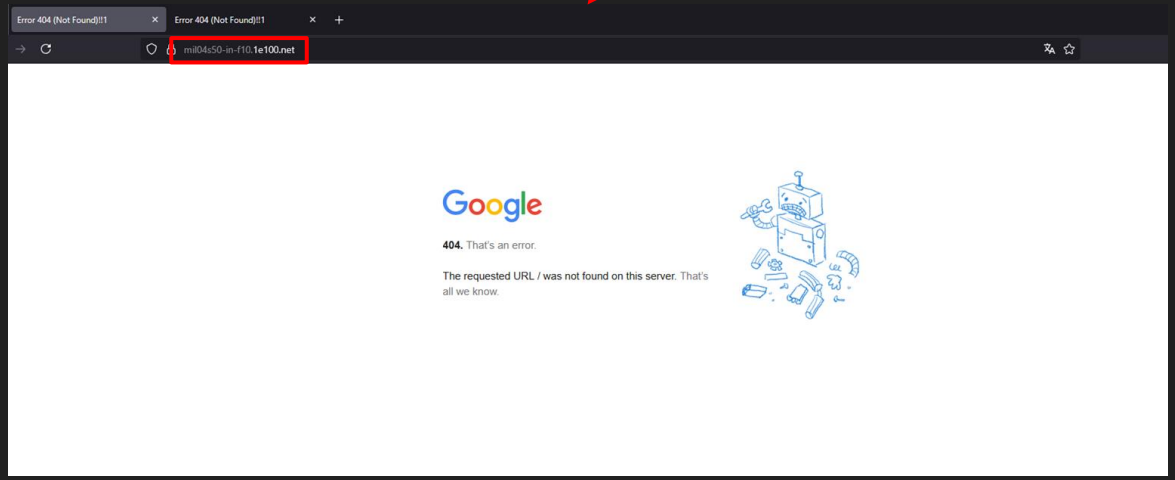
Detection

Process Explorer



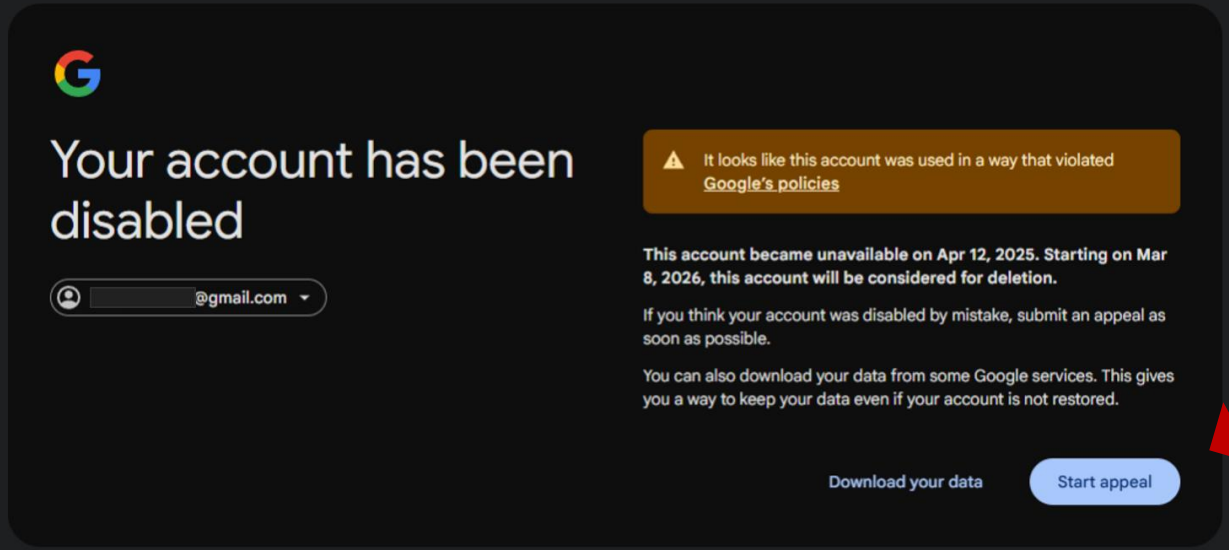
142.251.209.10 → mil04s50-in-f10.1e100.net

216.58.204.138 → par21s05-in-f10.1e100.net



4. Socks Over Google Services

Google Mitigations



GSSOCKS Github Project

Thank You!