

Digital Forensics of Data Theft on the Google Cloud Platform



TJEERD SLOKKER | FRANK WIERSMA

SUPERVISOR: KORSTIAAN STAM



Monday February 3th

Introduction

Initial Access	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Collection	Exfiltration	Impact
Exploit Public-Facing Application	Account Manipulation	Valid Accounts	Redundant Access	Account Manipulation	Cloud Service Dashboard	Data from Cloud Storage Object	Transfer Data to Cloud Account	Resource Hijacking
Trusted Relationship	Create Account		Revert Cloud Instance	Cloud Instance Metadata API	Cloud Service Discovery	Data from Information Repositories		
Valid Accounts	Implant Container Image		Unused/Unsupported Cloud Regions	Credentials in Files	Network Service Scanning	Data from Local System		
	Redundant Access		Valid Accounts		Network Share Discovery	Data Staged		
	Valid Accounts				Remote System Discovery			
					System Information Discovery			
					System Network Connections Discovery			

MITRE ATT&CK Matrix

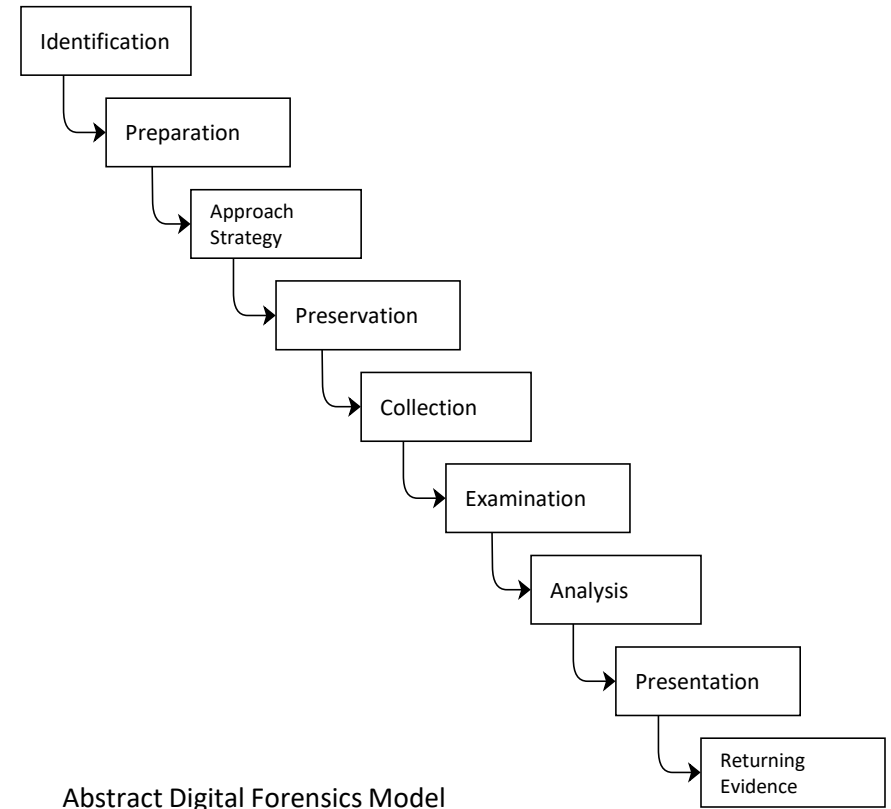
Research questions

What design, utilizing exclusively GCP native tooling, is required to establish digital forensic readiness on the Google Cloud Platform to investigate the Data from Cloud Storage Object and Data from Local System techniques from the MITRE ATT&CK Matrix?

1. What evidence needs to be acquired for investigation on the Data from Cloud Storage Object and Data from Local System techniques?
2. What are the sources for the evidence using exclusively GCP native tooling?
3. What evidence can be acquired with different GCP configurations?

Related work

- Haag, Leuenberger and van Ginkel, described the basics of digital forensics
- Zawoad and Hasan, proposed a log management solution
- Baryamureeba and Tushabe, defined the Abstract Digital Forensics Model (ADFM)



Evidence

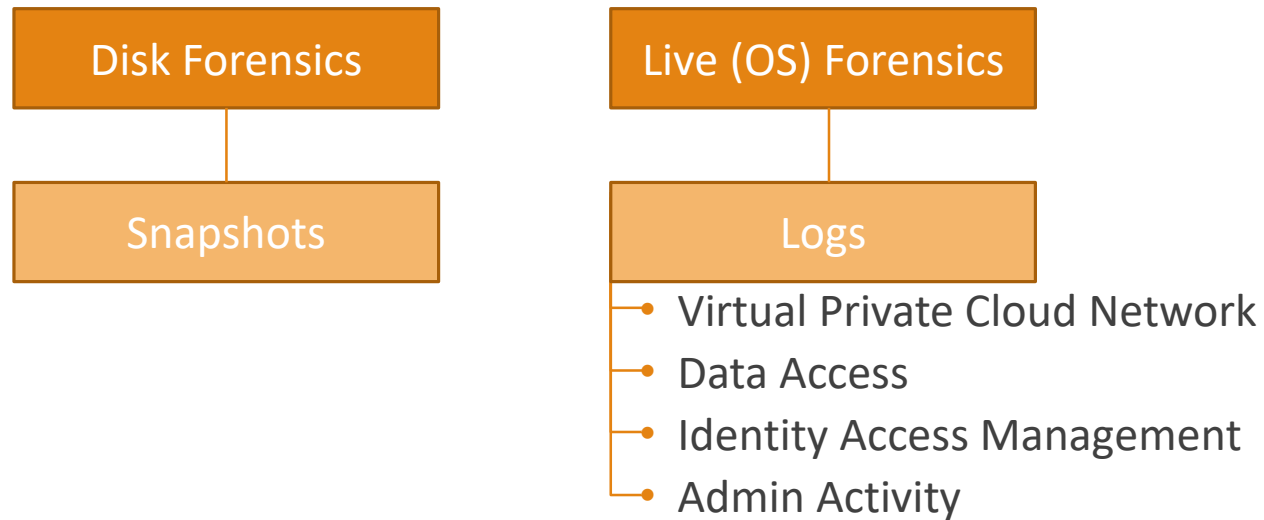
Data from Cloud Storage Object

- IP addresses
- Usernames
- Time of access
- What is accessed
- What operations
- Authentication attempts

Data from Local System

- + Network connections
- + Temp folders
- + Caches
- + Recycle bin
- + OS Event logs

Sources for evidence



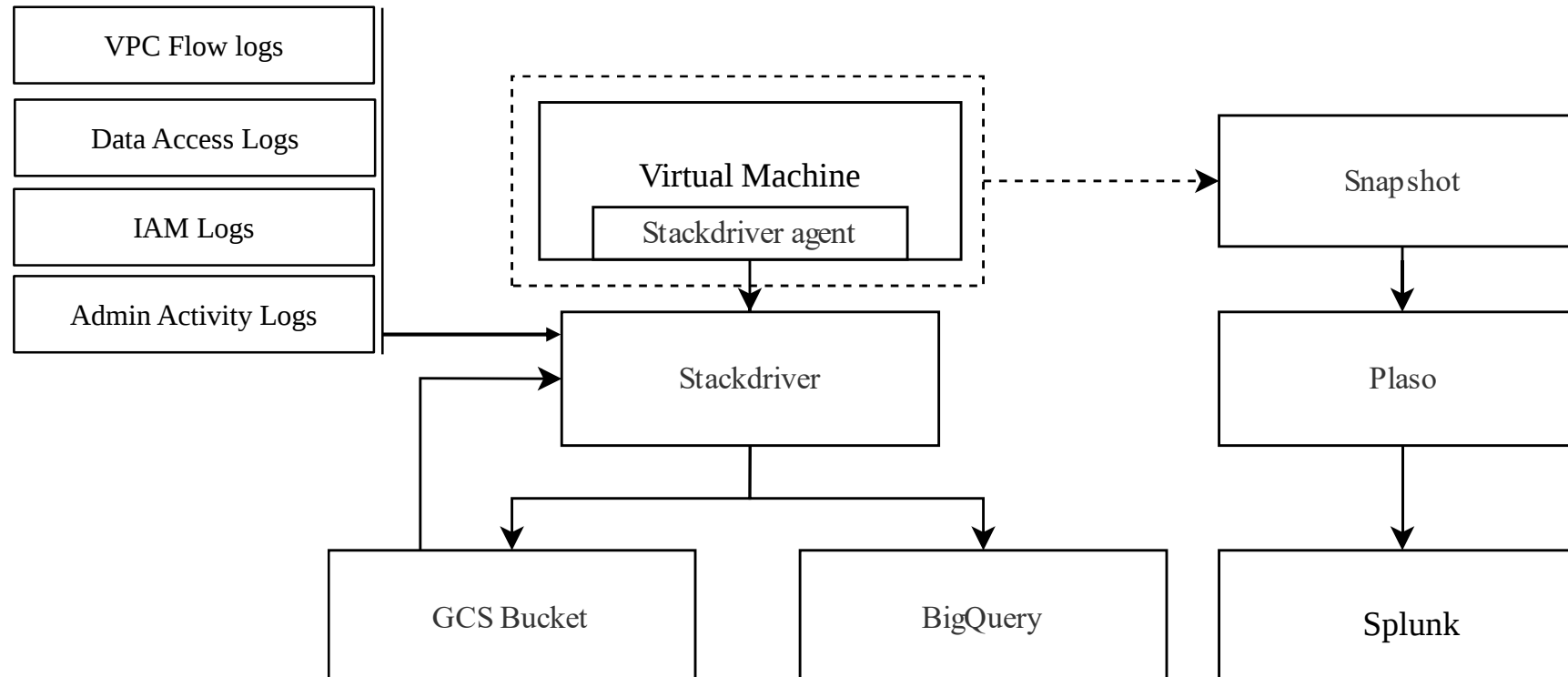
Storage locations

- BigQuery (data warehouse)
- Google Cloud Storage bucket

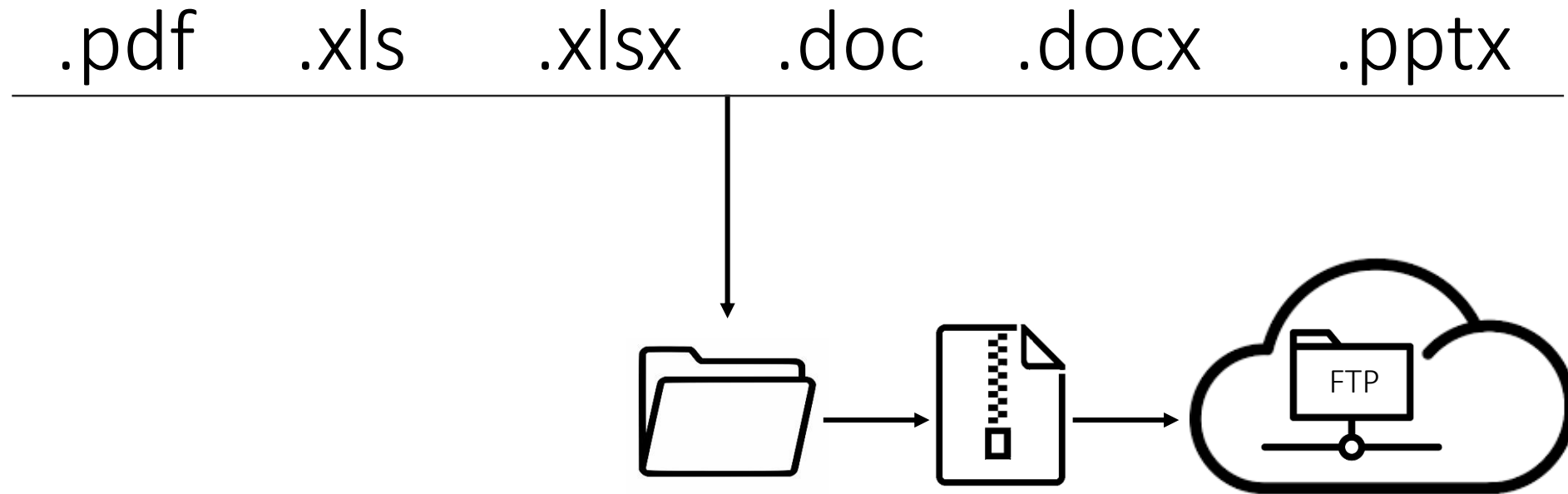
Methodology

- Forensic readiness
- Experiments
 - Data exfiltration from a virtual machine
 - Privilege escalation on a storage bucket
 - Integrity on storage location

Test environment



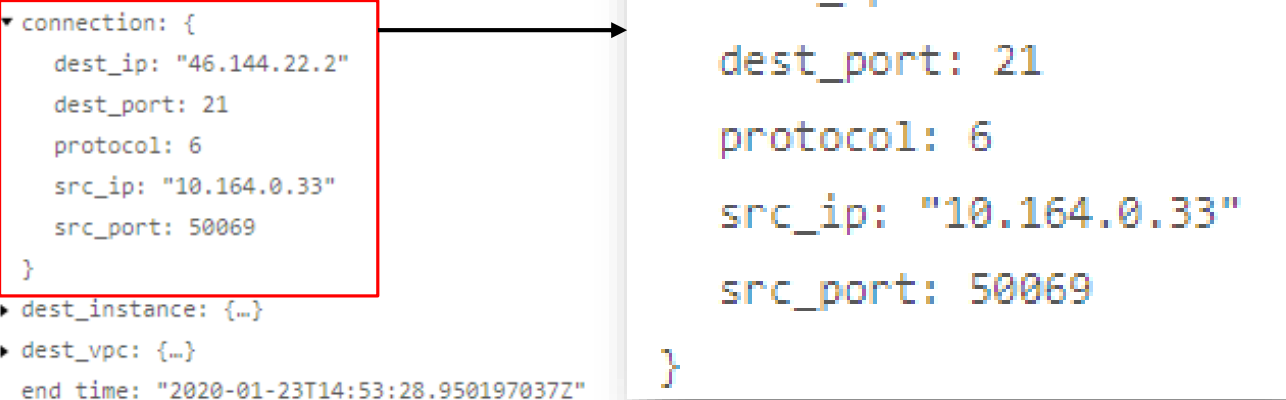
Experiment I – Data exfiltration from a VM



Experiment I – VM data exfiltration

Generated Logs

```
▼ * 2020-01-23 15:53:37.299 CET {"packets_sent": "4", "src_vpc": {"subnet_name": "default", "vpc_name": "default", "pr...  ⋮
  ▼ {
    insertId: "ivsmgbfyt38oc"
    ▼ jsonPayload: {
      bytes_sent: "24"
      ▼ connection: {
        dest_ip: "46.144.22.2"
        dest_port: 21
        protocol: 6
        src_ip: "10.164.0.33"
        src_port: 50069
      }
      ▶ dest_instance: {...}
      ▶ dest_vpc: {...}
      end_time: "2020-01-23T14:53:28.950197037Z"
      packets_sent: "4"
      reporter: "DEST"
      ▼ src_instance: {
        project_id: "public-cloud-forensics"
        region: "europe-west4"
        vm_name: "windows-test-machine"
        zone: "europe-west4-a"
      }
    }
  }
  Expand all | Collapse all
```



```
connection: {
  dest_ip: "46.144.22.2"
  dest_port: 21
  protocol: 6
  src_ip: "10.164.0.33"
  src_port: 50069
}
```

Experiment I – VM data exfiltration

Disk Forensic Investigation

i	Time	Event
>	1/23/20 2:39:33.324 PM	2020-01-23T13:39:33.324000+00:00, Content Deletion Time, RECBIN, Recycle Bin, C:\temp\files.zip (from drive: UNKNOWN), recycle_bin, TSK:/\$Recycle.Bin/S-1-5-21-3257857652-940297410-486091494-1000/\$I6BZNKU.zip,- host = NLFTSACQ007 source = windows-test-machine-timeline.csv sourcetype = csv

- Firewall change
- Creation of temporary folder
- File copy operations
- Tracks of a temporary ftp connection file
- Deletion of the zip afterwards

Experiment I – VM data exfiltration

Evidence collection

Potential evidence	Stackdriver logging-agent OFF	Stackdriver logging-agent ON	Network flow logs OFF	Network flow logs ON	Disk forensics
IP addresses	No	Yes	No	Yes	No
Username	No	Yes	No	No	Yes
Time of access	No	Yes	No	Yes	Yes
What is accessed	No	No	No	Yes	Yes
What file operations	No	No	No	No	Yes
Authentication attempts	No	Yes	No	No	Yes
Network connections	No	No	No	Yes	Yes
Temporary folders	No	No	No	No	Yes
Caches	No	No	No	No	Yes
Recycle bin	No	No	No	No	Yes
OS event logs	No	Yes	No	No	Yes

Yes = did provide evidence

No = did not provide evidence

Experiment II – Storage Bucket Privilege escalation

```
PS C:\Users\frank\Documents\GCPBucketBrute> python3 .\gcpbucketbrute.py -k data-collection-simulation -u
Generated 1216 bucket permutations.

UNAUTHENTICATED ACCESS ALLOWED: data-collection-simulation
- VULNERABLE TO PRIVILEGE ESCALATION (storage.buckets.setIamPolicy)
- ALL PERMISSIONS:
  [
    "storage.buckets.getIamPolicy",
    "storage.buckets.setIamPolicy"
  ]
```

Experiment II – Storage Bucket Privilege escalation

☐ Type	Members ^	Role(s)	
☐ 	443704278198@cloudbuild.gserviceaccount.com	Cloud Build Service Account inherited	 
☐ 	allAuthenticatedUsers	Storage Admin	 
☐ 	allUsers	Storage Admin	 
☐ 	frankwiersma1997@gmail.com	Security Reviewer inherited	 
☐ 	private-storage-access@public-cloud-forensics.iam.gserviceaccount.com	Storage Admin inherited	 

Experiment II – Storage Bucket Privilege escalation Success!

2020-01-29 18:02:48.233 CET **Cloud Storage** **get** **data-collection-simulation:short-csv.csv** 

[Expand all](#) | [Collapse all](#)

```
▼ {
  insertId: "-1ctpxdd8cwe"
  logName: "projects/public-cloud-forensics/logs/cloudaudit.googleapis.com%2Fdata_access"
  ▼ protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    ▼ authenticationInfo: {
      }
    ▼ authorizationInfo: [
      ▼ 0: {
        granted: true
        permission: "storage.objects.get"
        resource: "projects/_/buckets/data-collection-simulation/objects/short-csv.csv"
        ▶ resourceAttributes: {...}
      }
    ]
    methodName: "storage.objects.get"
    ▼ requestMetadata: {
      callerIp: "46.144.22.2"
    }
  }
}
```

Experiment II – Storage Bucket Privilege escalation

Evidence collection

Potential evidence	GCS data access audit logs OFF	GCS data access audit logs ON	IAM audit logs OFF	IAM audit logs ON
IP addresses	No	Yes	No	No
Username	No	Yes, if authenticated	No	No
Time of access	No	Yes	No	No
What is accessed	No	Yes	No	No
What file operations	No	Yes	No	No
Authentication attempts	No	Yes	No	No
Unusual API requests	No	Partially	No	No

Yes = did provide evidence

No = did not provide evidence

Experiment III – Integrity

Storage Location	Mutation prevention	Security options evidence	Retrievability evidence
BigQuery	Permissions	-	Querying Downloading
Google Cloud Storage bucket	Permissions	Customer-managed key	Downloading

What design, utilizing exclusively GCP native tooling, is required to establish digital forensic readiness on the Google Cloud Platform, to investigate the Data from Cloud Storage Object and Data from Local System techniques from the MITRE ATT&CK Matrix?

Conclusion

- GCP native tooling not sufficient for live forensics
- Combine logs & disk forensics

Key findings:

- Stackdriver agent collects minimal OS event logs
- No traces of the intentional privilege escalation
- Hard to check integrity during the preservation and collection phase
- Disk forensics provided the most evidence

Future work

- More tests within MITRE matrix
- Try to get Google's help with evidence collection
- Research on Chain of Custody
- Third party agents