Digital Forensic Investigation of Data Theft on the Google Cloud Platform

A Design for Investigation of the Data from Cloud Storage Object and Data from Local System Techniques from the MITRE ATT&CK Matrix on the Google Cloud Platform.

Tjeerd Slokkker
tjeerd.slokker@os3.nl
Security and Network Engineering
University of Amsterdam

Frank Wiersma
frank.wiersma@os3.nl
Security and Network Engineering
University of Amsterdam

February 9, 2020

Abstract—**The emergence of cloud computing brings a new challenge to performing digital forensic investigations. When using public cloud services you are limited to the available resources provided by the cloud providers. This study aimed to create a design, with Google Cloud native tooling, for investigation of the Data from Cloud Storage Object and Data from Local System techniques from the MITRE ATT&CK Matrix on the Google Cloud Platform (GCP). First we defined what evidence had to be collected and then we determined the locations where this evidence needs to come from and where it should be stored. Then we performed experiments on these two techniques to determine how the GCP needs to be configured to be forensic ready. Based on the results of these experiments we can conclude that a design for forensic readiness on the GCP for the two MITRE attack techniques is not possible with only Google Cloud native tooling. However, to come as close as possible to forensic readiness we advise on using the Google Stackdriver logging agent for Data from Local System. For this technique we also suggest to enable periodic snapshots. For the Data from Cloud Storage Object technique we suggest to enable GCS data access audit logs. We cannot advise with certainty which location to choose for long term storage of evidence.**

I. Introduction

Digital forensics involves the investigation and recovery of data gathered from digital devices related to computer crime. The emerge of cloud computing brings a new challenge to performing digital forensic investigations. When using public cloud services you are limited to the available resources provided by the cloud providers. For example, you cannot get hold of a physical disk out of a data center of a cloud service provider to examine it. This research will focus on establishing forensic readiness on the Google Cloud Platform (GCP). It will focus on the offensive techniques 'Data from Cloud Storage Object' and 'Data from Local System' as specified in respectively `T1530` [**mitre˙gcp˙cso**] and `T1005` [**mitre˙gcp˙ls**] in the MITRE ATT&CK Matrix [**mitrematrix**]. The MITRE ATT&CK Matrix is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government and in the cybersecurity product and service community. The technique mentioned first, applies to Google Storage Buckets and the second technique applies to virtual machines on the Compute Engine module of GCP. This study aims to create a design on the GCP, with Google Cloud native tooling, for investigation of the Data from Cloud Storage Object and Data from Local System techniques.

II. Research Questions

The main research question is: *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?*

To formulate an answer to this question, the following sub-questions need to be answered:

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?*

A. Structure

The remainder of this paper has the following structure. In section III we look at highlights of work done by others that relates to ours. In section IV we define our approach to define a design for digital forensics on the GCP. In section V we present findings of our experiments. In section VI we discuss our findings. Using our findings, we will draw conclusions and present those in section VII. The last section, VIII, contains suggestions for future work.

III. Related work

Digital forensics is a field that consists of: computer, mobile device, database and network forensics [1]. Computer forensics includes disk forensics and live forensics where the latter is done by memory and live OS forensics. A suitable definition of computer forensics for the purpose of this research is ”the application of computer investigation and analysis techniques to determine potential evidence” [2]. With digital forensics comes the production of evidence. This production is a complex task, because digital evidence needs to be valid in court. This is only possible if the chain of custody can assure what happened with the evidence, why and how it was gathered, analysed and reported, and who had access to it in the process. [3]. For the chain of custody it is important that the integrity and reliability of the evidence can be verified. This needs to be taken into account while dealing with evidence.

A. Digital Forensics Process and Digital Evidence

The work of Reilly, Wren, and Berry [2] generally outlines the approach to performing digital forensics for cloud computing: ”Computer forensics investigations generally follows a linear process: identification, extraction, analysis and presentation of evidence.” They also state that the conventional legal requirements on evidence also apply to digital evidence. The digital evidence should be: authentic, reliable, complete, convincing to juries, and in conformity with common law and legislative rules (i.e. admissible). The work of Baryamureeba and Tushabe [4] gives a similar process for the approach of performing digital forensics. They define this as the Abstract Digital Forensics Model (ADFM) as shown in Figure 1.

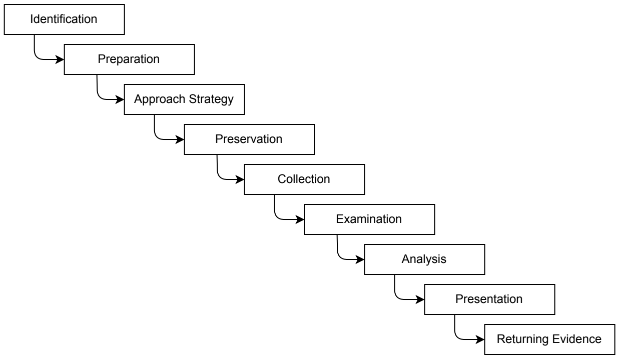

Fig. 1: Abstract Digital Forensics Model as defined by Baryamureeba and Tushabe [4]

For our research we use the model as follows. The identification and preparation phases are what we define as the forensic readiness part. The approach strategy is the procedure to follow when digital forensics is demanded. The preservation, collection, examination and analysis phases are used to determine the value of the evidence that results from tests of various configurations on the GCP. The last two phases are not included in our research scope, as they are not relevant to our research on forensic readiness.

B. Cloud Forensics Challenges

The National Institute of Standards and Technology (NIST) listed all the challenges that exist for digital forensics for cloud computing to date [5]. An important challenge is the time span of investigations since this can significantly increase, due to the distributed nature of cloud services. Data can potentially reside in multiple legal jurisdictions, leading to investigators relying on local laws and regulations regarding the collection of evidence. Besides that, the ephemeral existence of data forms another challenge. Potential evidence in the form of virtual instances, disks or databases can easily be deleted to make it near infeasible to find and retrieve them without aid of the Cloud Service Provider (CSP). Dependent on the cloud provider, sophisticated mechanisms could be available to facilitate access to this evidence [5].

C. Current Solutions

The researchers Zawoad and Hasan [6] proposed possible solutions to cope with and respond to the challenges that investigators might face when collecting evidence in a cloud environment. One of the proposed solutions is implementing a Log Management Solution in order to circumvent the challenge of decentralized logs in a myriad of formats from a lot of different cloud services. Another solution to cope with live forensics challenges is the Virtual Machine Introspection (VMI) technique. This is a technique for monitoring the runtime state of a system-level virtual machine, which is helpful for forensic analysis. Also, having access to the Cloud Management Plane, a collective term for the different control panels and cloud portals, is crucial for accessing the evidence. However this requires a high level of trust in the management plane. A user can alter or delete evidence that is under their control (intentionally or unintentionally). In a traditional evidence collection procedure, where we have physical access to the system, this level of trust is not required.

IV. Methodology

To answer the research questions we used the structure of the ADFM. The first phase in this model is identification. The identification phase is identifying the several vectors that will indicate the specific attacking technique. For this research these are the Data from Cloud Storage Object and Data from Local System techniques. The MITRE ATT&CK Matrix gives the following vectors for the technique ”Data from Cloud Storage Object”[**mitre˙gcp˙cso**]: Unusual queries to the cloud provider’s storage service, activity originating from unexpected sources, failed attempts by a user for a certain object, followed by escalation of privileges and eventually access to the object by the same user. For the Data from Local System technique, MITRE states that processes and command-line arguments are often used for the collection of files. Another attack vector can be the remote access tools with built-in features as these may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as Windows Management Instrumentation (WMI) and PowerShell. [**mitre˙gcp˙ls**]

The next phase in the model is preparation. This is in our research the preparation of the GCP. For this preparation it is important to determine the required evidence that is needed for investigating the Data from Cloud Storage Object and Data from Local System techniques and what the sources are of this evidence. These results are used for formulating experiments. With these experiments we want to determine how the GCP needs to be configured to facilitate forensic readiness. The experiments will be performed by changing the configuration parameters of the GCP Audit logs, Network flow logs, and Identity Access Management (IAM) logs settings. We will also test what evidence can be acquired with disks forensics and the difference of using a logging agent or not. As the integrity of evidence is an important part of the chain of custody, we will also do an experiment during the preservation and collection phases of the ADFM to see how integrity can be achieved. All experiments are done in a test environment.

A. Test Environment

The test environment as shown in Figure 2 is designed with the help of the ADFM. preserving, collecting, examination, and analysing potential evidence is done with the help of Plaso [7] and Splunk [8]. With Plaso it is possible to create a timeline of events of a snapshot from a Virtual Machine (VM). Splunk is an intelligent tool to search through all events in a timeline. This part of the design is disk forensics. Live OS forensics is done with the help of Stackdriver, Google Cloud Storage (GCS) bucket and BigQuery. Under Live OS forensics we consider two log categories: platform-generated and user-generated logs. An example of a platform-generated log is data access on GCP products. An example of a user-generated log is a log entry generated by an operating system [9]. For user-generated logs we deploy a Windows and Linux VM on the Infrastructure as a Service module of GCP. Both VMs will have the Stackdriver logging agent installed and will store some documents marked as valuable. With Stackdriver it is possible to collect and aggregate the different types of logs. A log router will be configured with a few log paths in order to to send the logs from Stackdriver to multiple applications for processing, filtering, and analysis [10]. For long term storage and integrity preservation purposes the logs will also be written to a GCS bucket and to the Google managed data warehouse called BigQuery. Thereby, the logs can be saved for a longer period of time, as data access logs have a retention period of 30 days in Stackdriver [11].

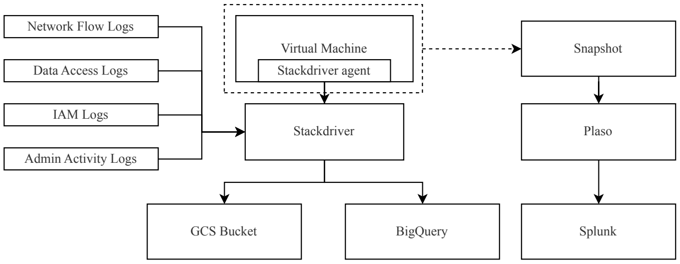

Fig. 2: Test Environment that contains a Live (OS) forensics path with Stackdriver as the central log management solution and a Disk Forensics path using snapshots.

B. Experiments

The following experiments will simulate actions that collect data either authorized or unauthorized. The experiments are based on real-life attack procedures listed in the MITRE ATT&CK Matrix [**mitrematrix**].

1) Local System (Virtual Machines): For testing the forensic readiness for Windows and Linux VMs on the established environment on GCP, we will simulate suspicious behaviour to see what information is generated, that eventually can be used as evidence in a court of law. According to MITRE [**mitre˙gcp˙cso**], actions that can be seen as suspicious behaviour are: sudden permission changes on a lot of files; frequency of opening specific files; sudden deletion of a large data set; copy operations on a large data set and compress archiving a large data set. To simulate this behaviour we will create a malicious script based on a data collection technique used by the threat group APT28 on the German Parliament in 2015 [12][13]. The script will be used in a simulation where a compromised Windows Server 2019 (with Stackdriver Logging Agent installed), located inside a victim's environment, now runs this malicious script which copies a lot of internal documents that might be valuable and confidential. The procedure of stealthily installing the script is out of the scope of this research. The functionality of the script will be fairly simple using just the combination of a tunnel to an FTP server outside the network and a command execution utility. It will utilize 'forfiles', a Windows native utility, that can be used to locate certain types of files/directories in a system. On Linux it will use the equivalent utility called 'find'. This utility stages documents in a temporary folder and eventually sends it to a compressed archive. The targeted file extensions of the documents are .pdf, .xls(x), .doc(x), .ppt(x), and .odt.

2) Cloud Storage Object (Google Cloud Storage bucket): A second experiment was required in order to see what configuration is needed to generate evidence on attacks on GCP buckets. Rhino Security Labs, experienced with penetration testing of the GCP [14], state that they often detect misconfigurations for the permissions of GCS buckets, despite the fact that they are created as private by default. Some misconfigurations make these buckets vulnerable to privilege escalation. To perform privilege escalation, somehow the permission to change the permissions for GCS needs to be granted to either everyone or all users authenticated with a Google account, to allow writing to the bucket's policy. We will grant the `storage.buckets.setIamPolicy` permission to simulate a severe misconfiguration and use this vulnerability to escalate privileges to grand storage admin permissions. We will use the TestIamPermissions API to verify if the specified permission has been granted. If so, this would mean that the bucket is then vulnerable. Then we will use the command-line interface for GCS, called 'gsutil' to append the Storage Admin role and try to download a file unauthenticated from the bucket. Ultimately, we will check if sufficient evidence is generated by the data access logs and identity access logs, to point out what happened and if it reveals some intelligence about a potential suspect.

3) Integrity potential Evidence: To evaluate integrity of potential evidence we are going to compare GCS and BigQuery as long time storage locations for the logs received by Stackdriver. For this experiment we are going to look at the possibilities of securing the storage locations, how mutation of evidence can be prevented, and how the evidence can be retrieved. This way we can determine what the optimal solution is for the design to ensure integrity.

V. Results

First of all, an analysis on what evidence is required for the investigation on the Data from Cloud Storage Object and Data from Local System techniques was made. With the help of the vectors as described by MITRE [15][16] and the work of Haag, Leuenberger, and Ginkel [1] we were able to determine what evidence needs to be acquired. For the investigation on the Data from Cloud Storage Object technique the following evidence needs to be collected: The IP addresses of all users that have accessed the Storage Object, the usernames of the people that used to access the data, what time the users had access, which files they accessed, what operations were performed, e.g., download/delete, API requests, and the failed and successful authentication attempts. All this evidence also needs to be acquired for the Data from Local System technique. Additionally, for this technique the network connections, temporary folders, caches, recycle bin, and OS event logs need to be acquired.

The IP addresses are important for pinpointing a location of the attack. It can for example help to determine if the attack could come from inside or outside the company. In the case that the attack came from inside the company it is also important to have the usernames of the people that had access to the data. That way it is possible to see which account was used for the attack. With the information of the time that users had access to the data it is easier to search through all possible evidence. The information on the files that have been accessed and what happened with them is crucial to determine whether or not the user could be a suspect. Data on API requests can reveal unusual behaviour. It is also important to monitor the failed and successful authentication attempts. An unusual peak in one or both of them could indicate malicious activity. Information on network connections needs to be acquired to see what kind of connections have been made to and from a system. Temporary folders are important to see file staging and copying. Thumbnail caches can help with determining which files and folders were accessed. The recycle bin can contain valuable deleted data. At last the OS event logs are important to find traces for firewall changes, modified services and authentication attempts.

Secondly, this study determined what sources of evidence are present using exclusively GCP native tooling. The first source of the evidence is the generation of the actual potential evidence. This evidence is collected from disk snapshots or logs. The second source for the evidence is the storage location of the potential evidence. From this location the evidence can be collected and further processed. For the collection of logs there are two source categories: platform-generated and user-generated logs. Platform-generated logs are data access audit logs (applies to GCS), VPC network flow logs, and admin activity audit logs. User-generated logs are application direct logs and logs from the OS level. User-generated logs are collected by the Google Stackdriver logging agent. Stackdriver can route these logs to different locations. By default all logs get routed to the Stackdriver log storage. For further processing and long term storage, the Stackdriver logs router can also send the logs to other locations. GCP native locations are Google Cloud Storage and BigQuery. To send the logs to these locations it is important to define what needs to be stored, because routing to these locations uses an inclusion filter. Another source for evidence is disk snapshots. When processed this is a source for evidence for file and registry operations on a file system level, and operating system event logs.

Moreover, this study performed experiments to determine what evidence can be acquired with different GCP configurations. To do this we looked at VMs, GCS buckets, and the integrity of the potential evidence on BigQuery and GCS buckets.

1) Virtual Machines: Without the logging agent installed, Stackdriver receives some logs from the virtual machines. These logs however show only metadata about the VM, i.e. start and stop logs, local user database mutations creation and OS config status logs. In our need for evidence this did not meet the requirements for clues on a potential attack. The user-generated logs acquired by the Stackdriver agent contained failed and successful user logons, including usernames and source IP addresses. The agent also retrieved logs from state changes of services running on the OS. But after execution of the data collection attack, the generated logs in the Stackdriver logs viewer did not show any evidence for it. It did not reveal the execution of the malicious data collection script nor any file operations. However with the enabled VPC network flow, logs were observed that showed proof of an established FTP connection to a specific IP address outside the VPC network around the time that the script had run. This is merely partial evidence therefore the disk forensics path was taken for more complete evidence. A disk image was collected from a snapshot. By analysing, the with Plaso timelined image, more proof for a data collection attack was found. We encountered event logs on changed FTP rules in the firewall and creation of a temporary folder that later contained a copy of the files that were marked as important. Subsequently, the timeline entries showed the creation of a compressed archive around the same time. We also observed tracks left by a temporary `ftp.txt` file that was used as a FTP connection file. The information also showed the logs of failed logon attempts, but these events were not created due to the actions of the malicious script. Table I shows the evidence retrieved by the different logging sources. The combination of the Stackdriver agent and Network flow logs enabled, plus the disk forensics gives the most complete evidence possible.

2) Google Cloud Storage Buckets: The results of the second experiment showed that most of the important evidence requirements were fulfilled, because with the data access logs enabled all executed file operations were logged. These logs showed whether the Google Storage API requests were denied or successful, it revealed the IP of the requester, and the different file or folder operations that were triggered (create, get, list, delete). The failed or successful attempts for listing the permissions on the test bucket were also logged, just like the event of the successful download of a file from the private bucket by an unauthenticated user. However in contrary to our expectations, the event of granting the storage admin rule to everyone was not logged, neither in the data access logs nor in the IAM logs. Additionally, the fact that the storage logs are generated once a day and contain the storage usage for the previous day, also makes the live forensics process for GCS buckets more difficult. Table II displays the types of evidence that was retrieved from the data access audit logs. The identity and access management audit logs did not log anything about the performed privilege escalation attack. Only when data access audit logs enabled, evidence was generated about the data collection. However the evidence provision for the Unusual API requests is set to 'Partially', as it did not provide the evidence for the privilege escalation. It logged the list permissions request, but there was no trace of a set permissions request.

3) Integrity potential Evidence: For BigQuery and GCS as storage locations, the Stackdriver router needs to be configured to specifically send logs. This means that only the logs that can contain evidence need to be stored on these locations. Securing the log data can be done on two different levels. The first level is encrypting the storage locations. For GCS it is possible to create a bucket with a customer-managed key. This way Google, as far as we know, is unable to access the data in the bucket. Doing something similar for BigQuery is not possible. As far as we could discover there is no option to encrypt the data. The second level of securing the data, is securing it from users. The evidence should not be accessible for users who have no use for it. This is possible with the GCP built-in permission profiles. BigQuery can with its own permission profile mutate the data in its storage. This is not possible for other profiles. The Stackdriver router sends logs to the GCS bucket in batches of one hour. Just as BigQuery, no one else can edit this data. The only way to edit the data is to download it and change it locally. This means that retrieving the logs, which can be potential evidence, is possible by downloading files. This retrieval is for logs on GCS done by downloading the files with all logs from a specific time slot. On BigQuery it is possible to do a query on only the data needed to download. Other than configuring where the data needs to go, it is not possible to control or have insight in the data flow. From the moment the data is retrieved it is possible to hash the data. With this hash it is possible to check if the data stays unchanged during the investigation.

VI. Discussion

The results of the first two sub-questions helped us to answer the last sub-question. The results of the experiment on the Data from Local System technique showed us that the GCP native tooling does provide some evidence of an attack with live OS forensics, but not all sufficient evidence that is needed for an investigation. However, disk forensics did provide all the evidence. The results of the experiment on the Data from Cloud Storage Object technique showed us that the IAM audit logs do not delivered any evidence needed for an investigation. The GCS data access audit logs deliver almost all evidence. However, the evidence for unusual API requests demonstrates that the logging on the Google storage buckets is just not yet complete, because the result missed crucial evidence on the privilege escalation event. The results of the experiment on integrity showed us that it is hard check if integrity of the long term storage of evidence can be guaranteed. For our tests on the integrity part of the chain of custody we only looked at the preservation and collection steps of the ADFM. However, there may be other ways to guarantee integrity of stored evidence. So we cannot with complete certainty conclude that one storage location is more suitable than the other, nor that the integrity can be guaranteed. So with these findings it becomes clear that a design for forensic readiness on the GCP for the two MITRE attack techniques is not possible with only Google Cloud native tooling.

The conducted experiments were quite specic which could mean that the results do not give a representative view of the attack techniques. Another limitation of the results of this study is the single conduction of the experiments contrary to multiple conductions to verify if there is a change that in generated evidence. However, if this is the case it can still be concluded that the tools are not sufcient. In the case of digital forensic investigation, you do not want uncertainty about whether or not all evidence exists.

One of the reasons that there is a shift towards public cloud is because of the sometimes possible reduction of costs. To improve the forensic readiness a lot of additional functionality still needs to be enabled. All these additional functionalities do not cost that much in our small test environment. However at large scale an enormous amount of logs will be generated and processed which might become costly and make the design less viable.

Potential Evidence	Stackdriver agent OFF	Stackdriver agent ON	Network flow logs OFF	Network flow logs ON	Disk forensics
IP addresses	No	Yes	No	Yes	No
Usernames	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

TABLE I: Evidence provision by the different evidence sources for the Data Collection from a Local
Yes = function provides evidence System technique
No = function does not provide evidence

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
Usernames	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

TABLE II: Evidence provision by the different evidence sources for the Data Collection from a Cloud Storage Object technique.
Yes = function provides evidence System technique
No = function does not provide evidence

VII. Conclusion

Through this project we have answered the three research questions set out in section II. Answering these questions was done by analysing the forensic readiness, for evidence generation for data collection, through experiments. Our experiment for data collection from a local system demonstrated that with the use of just live (OS) forensics i.e. the Stackdriver logging solution and multiple enabled logging sources, did not provide sufficient evidence of the performed attack. Additional disk forensics was needed to reveal the attack pattern and intelligence on the suspect. Regarding the cloud storage object data collection attack, the result also missed crucial evidence on the privilege escalation event. With these results we can conclude that a design for forensic readiness on the GCP for the two MITRE attack techniques is not possible with only Google Cloud native tooling. However, to come as close as possible to forensic readiness we advise on using the Google Stackdriver logging agent for both attack techniques. However, to come as close as possible to forensic readiness we advise on using the Google Stackdriver logging agent for Data from Local System. For this technique we also suggest to enable periodic snapshots. For the Data from Cloud Storage Object technique we suggest to enable GCS data access audit logs. We cannot advise with certainty which location to choose for long term storage of evidence, so we recommend further research on this topic.

VIII. Future work

A. Additional attack techniques

In this research we focused on the Data from Local System and Data from Cloud Storage Object. The MITRE ATT&CK framework contains multiple other techniques that can be a threat for the environments that are being built on the GCP. For example the 'Resource hijacking', which is a technique where cloud resources will be compromised for cryptocurrency mining purposes **mitre˙gcp˙resource˙hjck**. Researching other techniques could be interesting, as it could be possible that for these techniques the GCP native tooling could suffice for delivering required evidence.

B. Google's involvement

For further research it would be interesting to try to get Google involved in the digital forensic investigation. This research focused on the ability to create forensic readiness from the perspective of a GCP user. However we do not know to what extend Google keeps track of actions on their platform.

C. Chain of Custody

As mentioned before the chain of custody is an important part of (digital) forensics. We only looked at integrity in two steps of the ADFM. Future research could focus on integrity of different steps of the ADFM. It could also focus on the whole chain of custody and how to administer this in the whole process of digital forensic investigation on the GCP as well as on other cloud platforms.

D. Third party logging agents

We found that for the two attack techniques the GCP native tooling did not suffice in delivering all required evidence with just live (OS) forensics. In future research, different third party logging agents could be compared to the Stackdriver logging agent. It is possible that a third party logging agent could deliver all required evidence.

E. Third party storage

Just as using third party logging agents, there could also be looked at the possibilities of storing the evidence on premise. For that research the GCP storage locations could be compared to storage locations outside of the GCP.

References

[1] P. Haag, A. Leuenberger, and J. van Ginkel, "Transits-ii — digital forensics module part 2," OS3, 2019.

[2] D. Reilly, C. Wren, and T. Berry, "Cloud computing: Forensic challenges for law enforcement," in *2010 International Conference for Internet Technology and Secured Transactions*, Available at https://ieeexplore.ieee.org/abstract/document/5678033, IEEE, 2010, pp. 1–7.

[3] G. Giova, "Improving chain of custody in forensic investigation of electronic digital systems," *International Journal of Computer Science and Network Security*, vol. 11, no. 1, pp. 1–9, 2011.

[4] V. Baryamureeba and F. Tushabe, "The enhanced digital investigation process model," in *Proceedings of the Fourth Digital Forensic Research Workshop*, Available at https://dfrws.org/sites/default/files/session-files/paper-the_enhanced_digital_investigation_process_model.pdf, 2004, pp. 1–9.

[5] NIST *et al.*, "Nist cloud computing forensic science challenges," National Institute of Standards and Technology, Tech. Rep., 2014, Available at https://csrc.nist.gov/csrc/media/publications/nistir/8006/draft/documents/draft_nistir_8006.pdf.

[6] S. Zawoad and R. Hasan, "Cloud forensics: A metastudy of challenges, approaches, and open problems," *arXiv preprint arXiv:1302.6312*, 2013, Available at https://arxiv.org/pdf/1302.6312.pdf.

[7] Log2Timeline, *Plaso*, Available at https://github.com/log2timeline/plaso.

[8] Splunk, *Splunk*, Available at https://www.splunk.com/.

[9] *Google cloud next '18 - forensics*. [Online]. Available: https://cloud.withgoogle.com/next18/sf/sessions/session/156791.

[10] Google, *Logs router overview — stackdriver logging — google cloud*, Available at https://cloud.google.com/logging/docs/routing/overview.

[11] GoogIe, *Quotas and limits — Stackdriver Logging — Google Cloud*, Available at https://cloud.google.com/logging/quotas.

[12] Gastbeitrag, *Digital attack on german parliament: Investigative report on the hack of the left party infrastructure in bundestag*, 2017. [Online]. Available: https://netzpolitik.org/2015/digital-attack-on-german-parliament-investigative-report-on-the-hack-of-the-left-party-infrastructure-in-bundestag/.

[13] D. Church, Splunk, E. Ratliff, IBM, R. Gold, and D. Shadows, *Apt28*. [Online]. Available: https://attack.MITRE.org/groups/G0007/.

[14] S. Gietzen, *Google cloud platform (gcp) bucket enumeration privilege escalation*, Mar. 2019. [Online]. Available: https://rhinosecuritylabs.com/gcp/google-cloud-platform-gcp-bucket-enumeration/.

[15] Netskope and Preatorian, *Data from cloud storage object*, Available at https://attack.MITRE.org/techniques/T1530/.

[16] Preatorian, *Data from local system*, Available at https://attack.MITRE.org/techniques/T1005/.