Bling Libra's Tactical Evolution: The Threat Actor Group Behind ShinyHunters Ransomware

By Margaret Kelley, Chandni Vaya

Published: 2024-08-23 · Archived: 2026-04-05 20:12:39 UTC

Executive Summary

In an incident response engagement handled by Unit 42, the threat actor group Bling Libra (the group behind the ShinyHunters ransomware) showcased their new shift to extorting victims rather than their traditional tactic of selling/publishing stolen data. This engagement also displayed how the group acquires legitimate credentials, sourced from public repositories, to gain initial access to an organization's Amazon Web Services (AWS) environment.

While the permissions associated with the compromised credentials limited the impact of the breach, Bling Libra infiltrated the organization's AWS environment and conducted reconnaissance operations. The threat actor group used tools such as the Amazon Simple Storage Service (S3) Browser and WinSCP to gather information on S3 bucket configurations, access S3 objects and delete data.

Threat actors commonly use S3 Browser and WinSCP during their attacks. To expand incident responders' understanding of how these tools generate events in the logs, this research differentiates activity initiated by the threat actors versus activity automatically generated by each tool.

As businesses increasingly embrace cloud technologies, the threat posed by groups like Bling Libra underscores the importance of robust cybersecurity practices. By implementing proactive security measures and monitoring critical log sources, organizations can effectively safeguard their cloud assets and mitigate the impact of cyberthreats.

AWS log sources and services such as Amazon GuardDuty, AWS Config and AWS Security Hub play a crucial role in enhancing the security posture of organizations. When using AWS Organizations, using AWS Service Control Policies and permission boundaries add additional protection. These tools provide valuable insights and alerts to security analysts, enabling them to monitor and respond to security incidents effectively.

Palo Alto Networks customers are better protected from the threats discussed above through the following products:

- Cortex XDR for cloud can offer a comprehensive incident story by integrating activity from cloud hosts, cloud traffic and audit logs together with endpoint and network data.
- Palo Alto Networks customers can also take advantage of Prisma Cloud to monitor posture and maintain compliance across public clouds.
- Palo Alto Networks Cloud Security Agent (CSA) uses XSIAM to provide detection and monitoring capabilities to cloud infrastructure through both Prisma Cloud and Cortex cloud agents.

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team.

Bling Libra Background: The Threat Actors Behind ShinyHunters

Unless a threat actor leaves specific indicators behind, researchers have a difficult time performing attribution for cloud attacks. However, the threat actor group Bling Libra does not hold back in making sure their attacks explicitly link back to them.

Bling Libra first emerged in 2020 and has been linked to significant data breaches, such as the Microsoft GitHub and the Tokopedia attacks in 2020 as reported by Wired. While Bling Libra's targets span various industries and geographic regions, their modus operandi remains consistent.

This group typically acquires legitimate credentials before targeting database infrastructure to gather personally identifiable information (PII) for resale on underground marketplaces. In 2024, the group shifted from trying to sell data they've collected to extorting their victims, and targeting cloud environments.

Introduction to MITRE ATT&CK® Framework

This article uses the MITRE ATT&CK framework to categorize the different tactics present within the attack. The matrix comprises 14 unique tactics (Enterprise version 15) and categorizes various common characteristics seen by threat actors. For more information on the tactics present in this article, each header contains a link to each specific tactic and their corresponding techniques. Figure 1 represents the first step in the attack.

Figure 1. Initial access begins with stealing AWS access keys.

Initial Access (TA0001)

To gain initial access into the organization's AWS environment, the threat actors obtained AWS credentials from a sensitive file exposed on the internet. The file contained a variety of credentials, but the group specifically targeted the exposed AWS access key belonging to an identity and access management (IAM) user and a handful of other exposed credentials.

These cloud credentials allowed the threat actors to gain access to the AWS account where this IAM user resided and perform AWS application program interface (API) calls. The permissions associated with these credentials only allowed the attackers to successfully interact with S3 with the AmazonS3FullAccess policy. This AWS-managed policy grants unlimited permissions to S3 resources within an AWS account depending on what other organizational policies the AWS account might employ.

Unit 42 commonly investigates matters where overly permissive cloud credentials deployed by organizations lead to further exploitation of an environment. This Bling Libra attack exemplifies the results of not following the principle of least privilege.

Discovery (TA0007)

Once Bling Libra gained access to the organization's AWS environment, they performed a variety of API calls to determine the extent of the permissions the compromised credentials contained. To determine what API calls the threat actors made, Unit 42 used CloudTrail logs to track the group's activities in the environment. CloudTrail logs all the management events that occur within an AWS account. Figure 2 captures the various discovery attempts.

Figure 2. Discovery begins the attacker's process of learning more about the environment.

Against the IAM service, the threat actor performed ListUsers, which returns a list of the existing users within the AWS account. Due to the limited permissions associated with the access key, the API call failed.

To learn more about the S3 buckets that existed within the AWS account, the group performed the ListBuckets API call using the AWS Command Line Interface (CLI). The AWS CLI provides people the ability to interact with an AWS account through the command line, and it provides the building blocks for automation.

From there, the threat actors switched to using the S3 Browser tool to iterate through the S3 buckets in the account and that generated both GetBucketLocation and GetBucketObjectLockConfiguration events in the CloudTrail logs. The S3 Browser tools provide a graphical user interface (GUI) to interact with the S3 buckets within an AWS account.

The S3 Browser analysis section discusses in more depth which API calls the tool automatically generates to help incident responders differentiate between tool automation and threat actor activity.

Data Access and Impact (TA0010 and TA0040)

Following the discovery operations, the threat actor waited almost a month before returning and taking disruptive actions within the organization's AWS account. Due to both CloudTrail S3 data logging and S3 server access logging not being enabled within the organization's AWS environment, no logs existed that showed exfiltration activity from the S3 buckets. Figure 3 illustrates the next two stages of the attack.

Figure 3. Data access and impact of the attacker's actions.

After waiting that extended period of time, the threat actor group used WinSCP to graphically view all the S3 buckets in the account. After selecting the Amazon S3 file protocol and entering the access key ID and secret access key, the tool automatically generates the ListBuckets API call to populate the list of buckets in the GUI. Due to the lack of object level logging, no other API calls appeared in the CloudTrail logs until the threat actor deleted a handful of buckets, which resulted in the DeleteBucket API call.

As described below, WinSCP provides various methods to interact with the S3 storage objects in an AWS account**.** The ransom note later sent by the threat actor provided proof of data access, but it did not provide enough specifics to determine what S3 data left the environment.

Execution (TA0002)

Following the deletion of all the S3 buckets, the threat actor used an automated script with the AWS CLI and attempted to create new S3 buckets. These buckets had various name variations of contact-shinycorp-tutanota-com-# with the # replaced with ascending numbers. Figure 4 shows the final step – execution of the script.

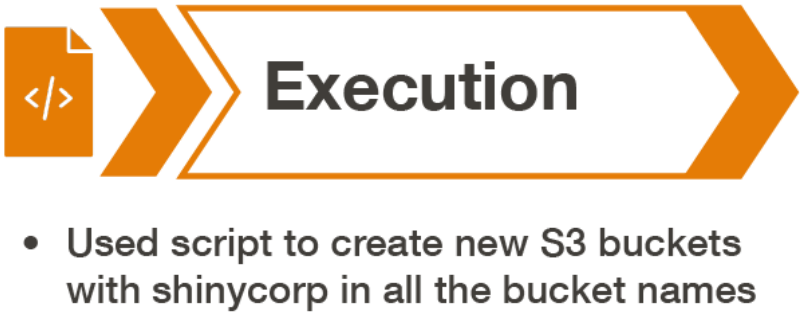


Figure 4. The attacker creates new S3 buckets.

All the buckets were created within ten minutes of starting the CreateBucket operations. We see no motive behind creating these S3 buckets other than to mock the organization about the attack. Figure 5 shows the full MITRE timeline based on key events detailed above.

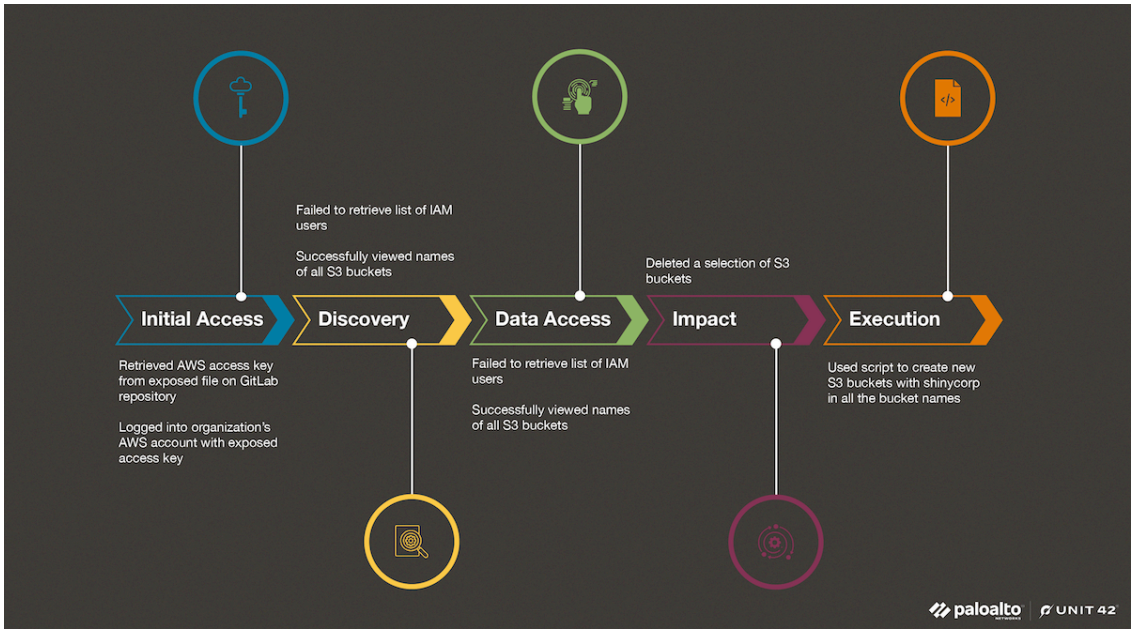
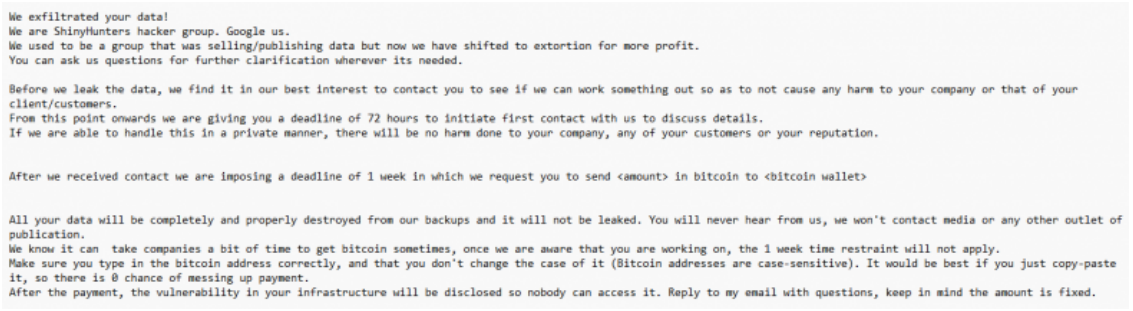


Figure 5. The MITRE timeline details the attack path taken by the threat actor.

Extortion

After Bling Libra performed all the aforementioned actions (i.e., data access via API key, discovery, deletion and creation of buckets), they completed their attack by sending an extortion email to the victim organization. The note stated the group had shifted to extortion to make more money and that the victim organization had one week to pay, as seen in Figure 6.



We exfiltrated your data!
We are ShinyHunters hacker group. Google us.
We used to be a group that was selling/publishing data but now we have shifted to extortion for more profit.
You can ask us questions for further clarification wherever its needed.

Before we leak the data, we find it in our best interest to contact you to see if we can work something out so as to not cause any harm to your company or that of your client/customers.
From this point onwards we are giving you a deadline of 72 hours to initiate first contact with us to discuss details.
If we are able to handle this in a private manner, there will be no harm done to your company, any of your customers or your reputation.

After we received contact we are imposing a deadline of 1 week in which we request you to send <amount> in bitcoin to <bitcoin wallet>

All your data will be completely and properly destroyed from our backups and it will not be leaked. You will never hear from us, we won't contact media or any other outlet of publication.
We know it can take companies a bit of time to get bitcoin sometimes, once we are aware that you are working on, the 1 week time restraint will not apply.
Make sure you type in the bitcoin address correctly, and that you don't change the case of it (Bitcoin addresses are case-sensitive). It would be best if you just copy-paste it, so there is 0 chance of messing up payment.
After the payment, the vulnerability in your infrastructure will be disclosed so nobody can access it. Reply to my email with questions, keep in mind the amount is fixed.

Figure 6. Extortion email.

S3 Browser and WinSCP Analysis

In many Unit 42 investigations involving cloud compromises, S3 Browser and WinSCP appear in the logs as threat actors use them for nefarious activity. Unit 42 performed tests to determine which activity in the AWS CloudTrail logs came from specific actions taken in the tools' GUIs versus automation API calls performed by the tools themselves.

The following tests used S3 Browser version 11.6.7 and WinSCP version 5.21.7.0.

S3 Browser

Unit 42 performed tests to understand which API calls automatically happen in the background by the tool, and which API calls get logged because of specific actions performed by a user in the GUI. The API calls can be tracked via the user agent field in the CloudTrail logs, which would contain a value of S3 Browser/<Version> (https://s3browser[.]com).

The test activities we discuss here took place against a bucket we created with data events logging enabled in CloudTrail, to compare the visibility in management versus data events. Data events provide visibility into the resource operations performed on or within a resource (e.g., creating, downloading or deleting an object within the S3 buckets). We denote object-level logging events with an asterisk (*) in the following sections.

- The first connection to the S3 storage service after entering the access key credentials resulted in the API calls listed below. As noted in Figures 7 and 8, S3 Browser automatically queries the object list, location and object lock configuration for the first S3 bucket alphabetically. If the S3 Browser application stays open when adding and removing the access key, only ListBuckets and ListDistributions appear in the CloudTrail logs (shown in Figures 9 and 10):
    - ListBuckets
    - ListObjects*
    - GetBucketLocation
    - GetBucketObjectLockConfiguration
    - ListDistributions

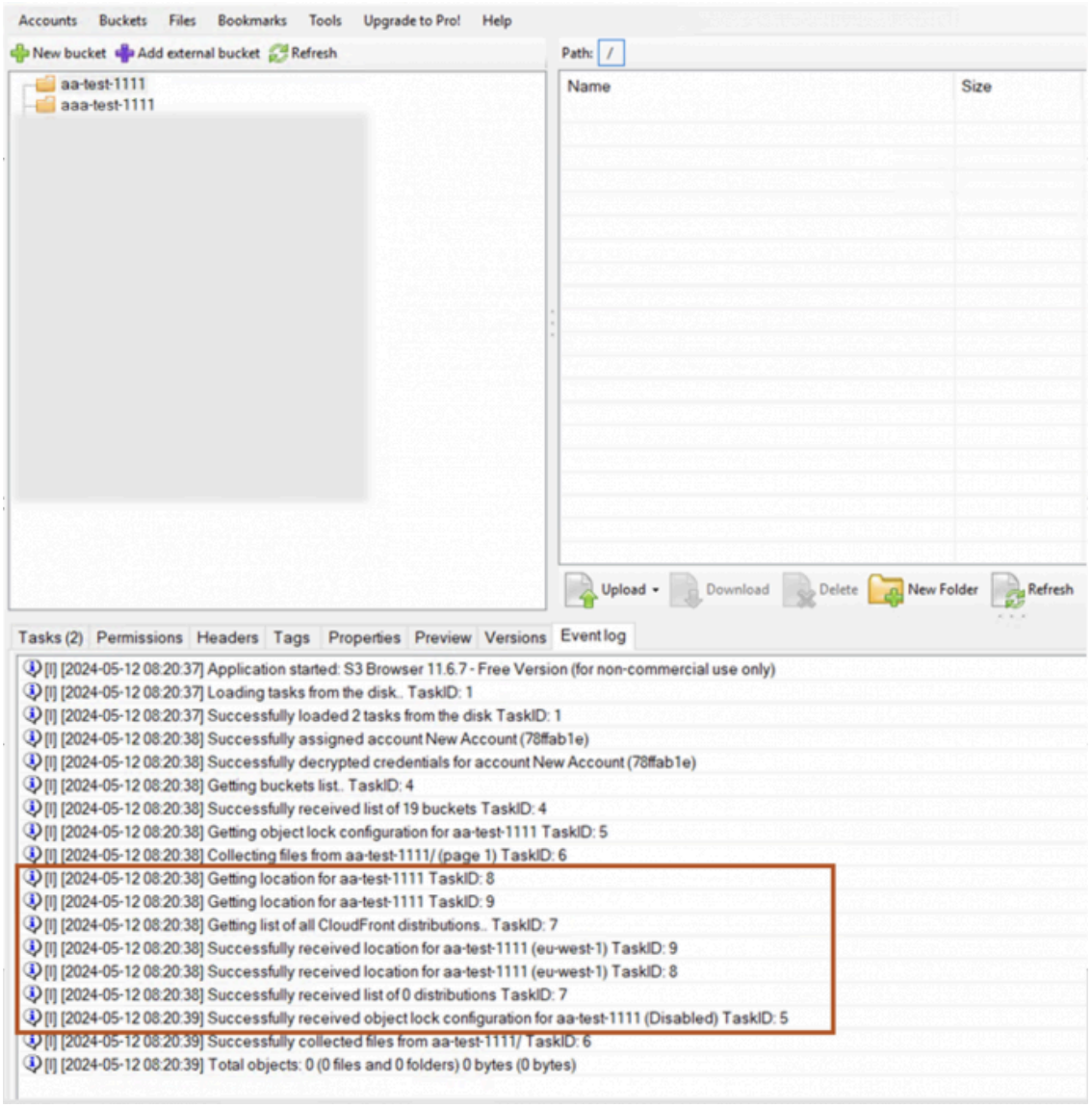


Figure 7. First connection to S3 service using S3 Browser.

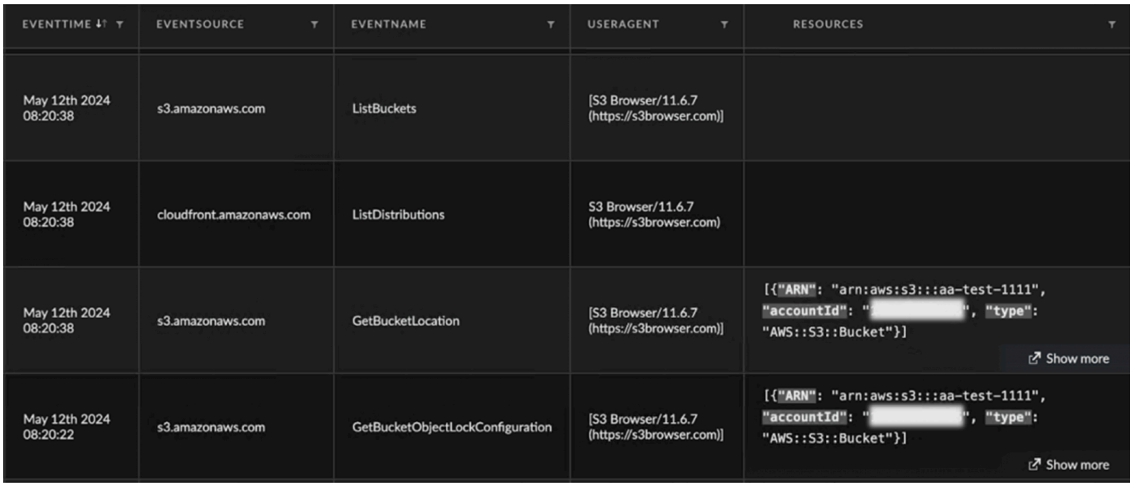


Figure 8. API calls for first connection to S3 service using S3 Browser.

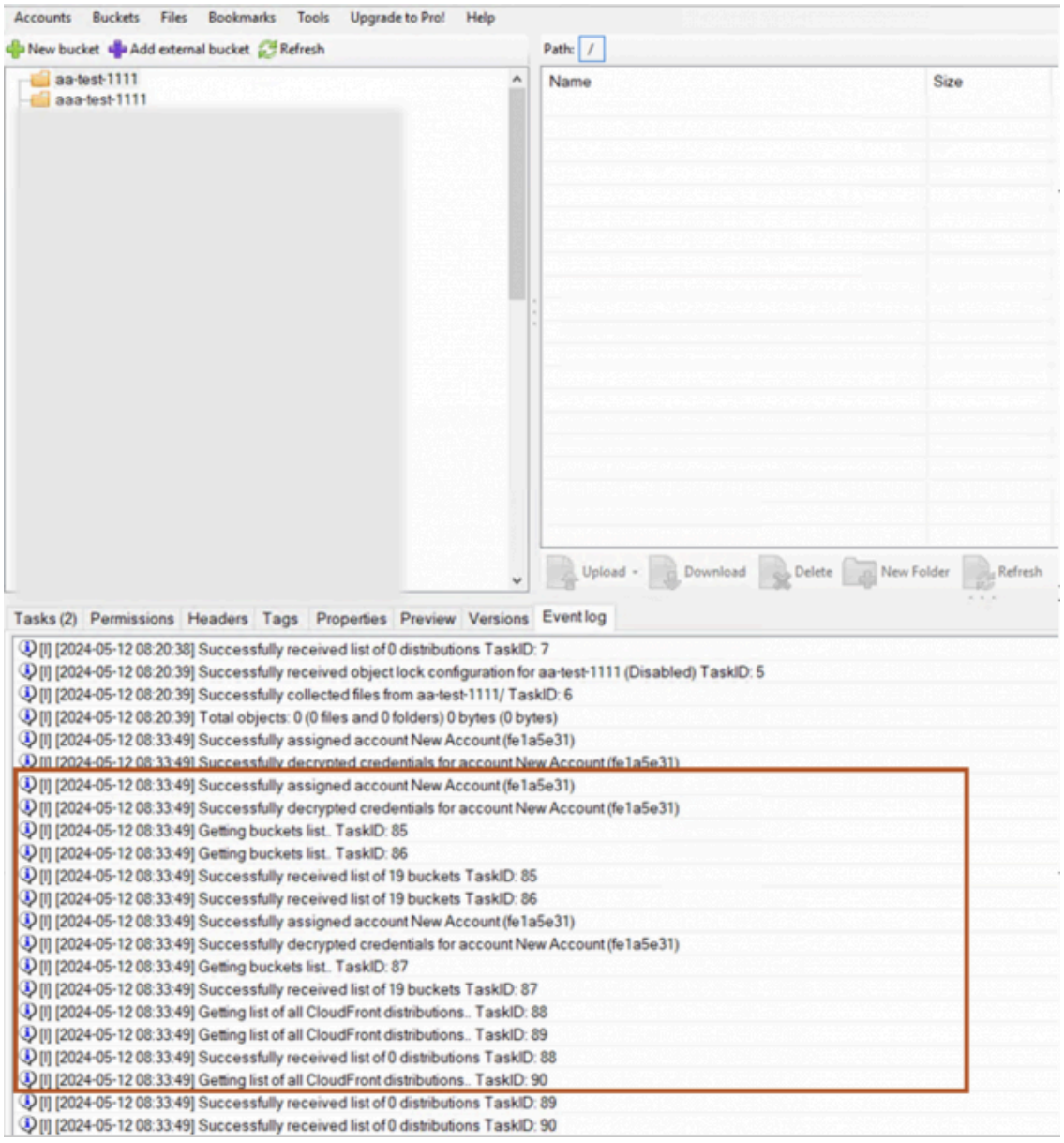
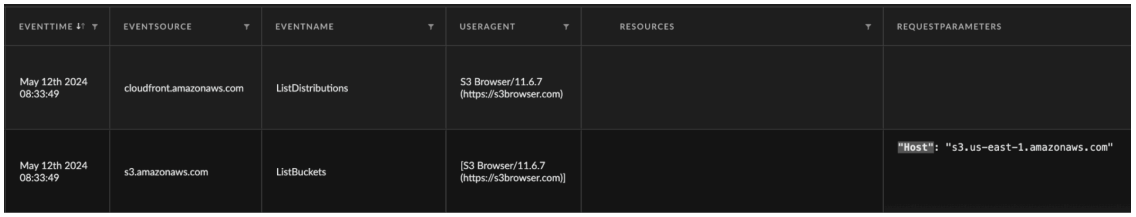


Figure 9. Connection using the same API keys without closing the S3 Browser.



EVENTTIME	EVENTSOURCE	EVENTNAME	USERAGENT	RESOURCES	REQUESTPARAMETERS
May 12th 2024 08:33:49	cloudfront.amazonaws.com	ListDistributions	S3 Browser/11.6.7 (https://s3browser.com)		
May 12th 2024 08:33:49	s3.amazonaws.com	ListBuckets	[S3 Browser/11.6.7 (https://s3browser.com)]		"Host": "s3.us-east-1.amazonaws.com"

Figure 10. API calls for the connection using the same API keys without closing the S3 Browser.

- When selecting and viewing the directory structure of another bucket in S3 Browser, it results in the following API calls.
    - ListObjects*
    - GetBucketLocation
    - GetBucketObjectLockConfiguration
- When previewing an object in S3 Browser (shown in Figure 11), the previewed file gets downloaded as a temporary file locally onto the host running the S3 Browser application. The file then immediately gets

deleted from disk (even if the preview window is still open with the file in the GUI). The S3 Browser stores the file in a temporary directory: C:\Users\<username>\AppData\Local\Temp\S3 Browser\<TempFileName>. The CloudTrail logs generate two data events for this action, one of them being GetObject, which appears in the logs since the object retrieval took place to populate the preview window. The HeadObject API call retrieves metadata about the object selected (e.g., its size, last modified date and metadata displayed by the application in the Properties tab).

- HeadObject*
- GetObject*
- C:\Users\<username>\AppData\Local\Temp\S3 Browser\<TempFileName>

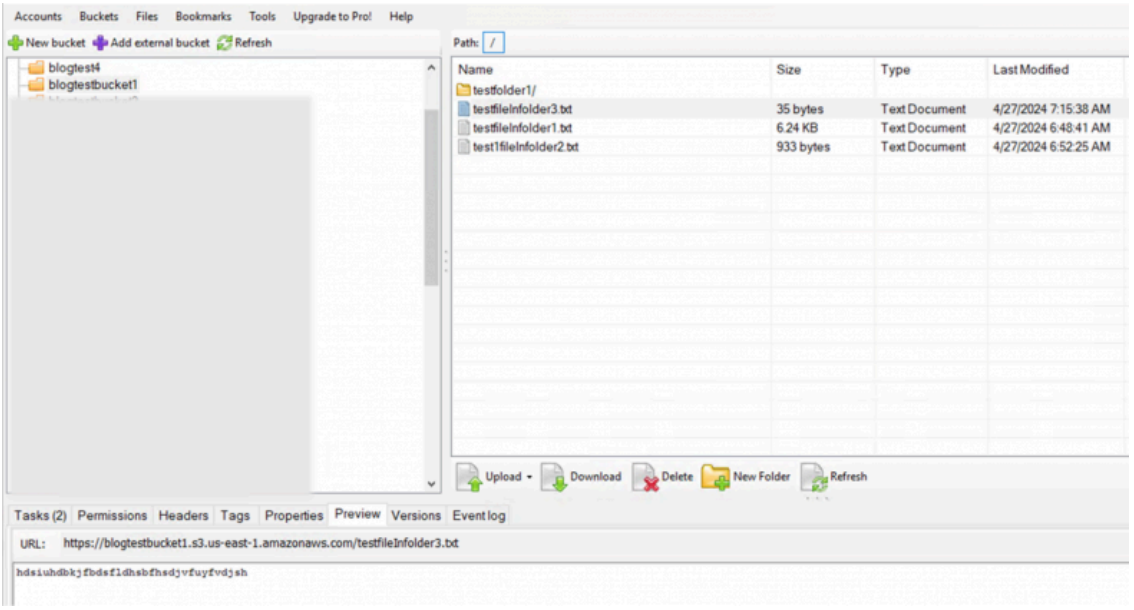


Figure 11. Previewing an object in S3 Browser.

- Creating and deleting buckets and objects resulted in Put, Create and Delete API calls as listed below:
  - DeleteObject*
  - PutObject*
  - CreateBucket
  - DeleteBucket
- Viewing the permissions in S3 Browser (shown in Figure 12) results in the API calls below for the access control list (ACL), OwnershipControls and PublicAccessBlock of the currently selected bucket.
  - GetBucketAcl
  - GetBucketOwnershipControls
  - GetBucketPublicAccessBlock

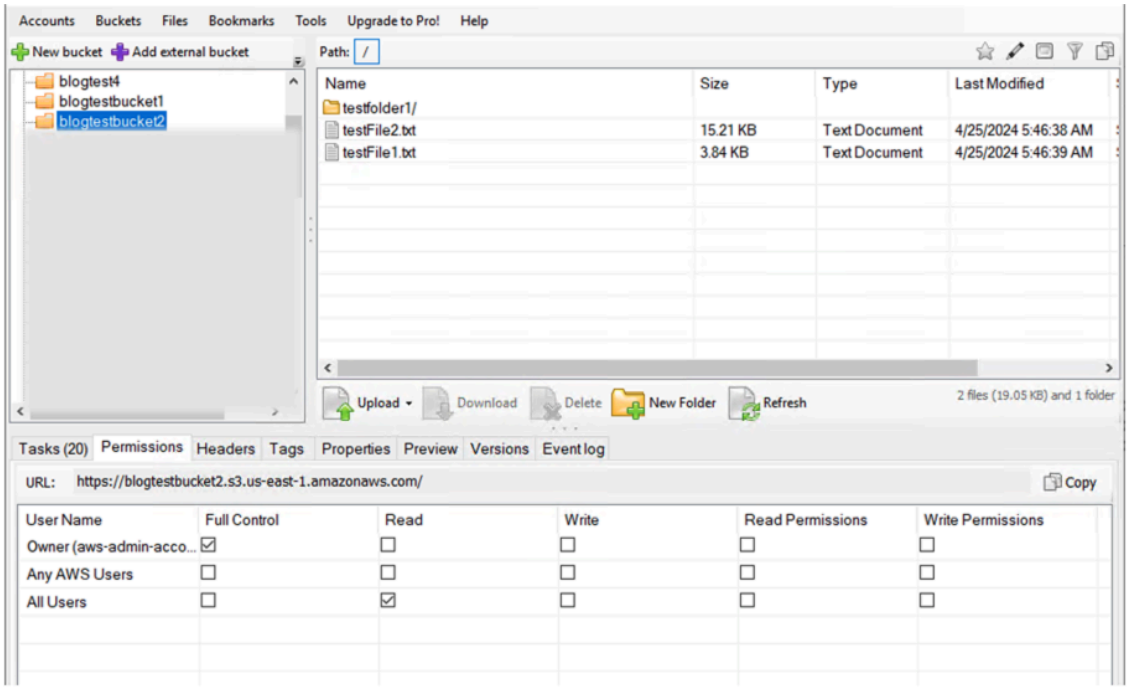


Figure 12. Viewing the permissions of the bucket.

- Viewing the properties (shown in Figure 13) will result in multiple API calls to gather information about all the properties related to the bucket to populate all the information in the Preview window.
    - GetBucketLogging
    - GetBucketObjectLockConfiguration
    - GetBucketReplication
    - GetBucketVersioning
    - GetAccelerateConfiguration*
    - GetBucketRequestPayment

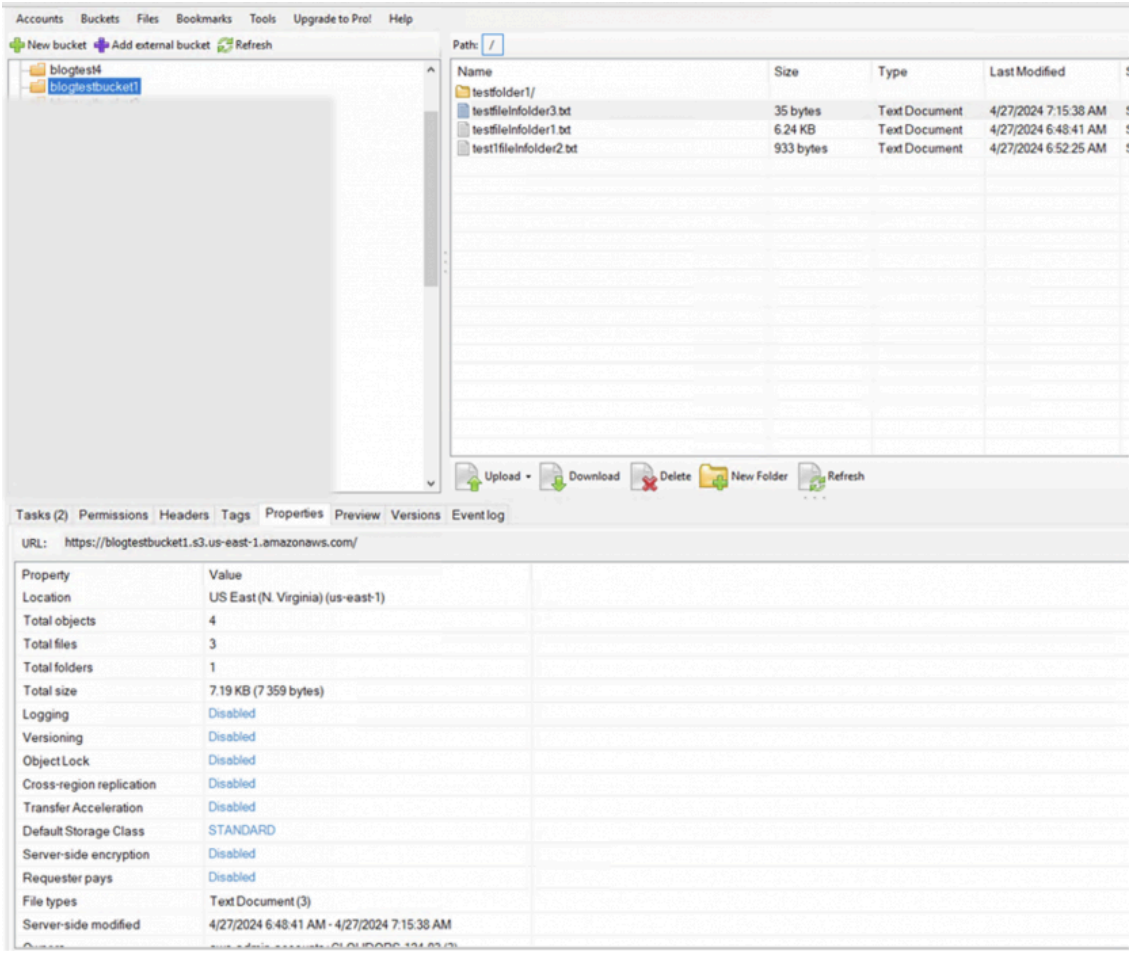


Figure 13. Viewing properties of a bucket in S3 Browser.

- Attempting to enable and disable versioning for the bucket (shown in Figure 14) results in the API calls below, which first gather the versioning state and then set it to the required value.
    - GetBucketVersioning
    - PutBucketVersioning with request parameter showing the status as shown in Figure 15

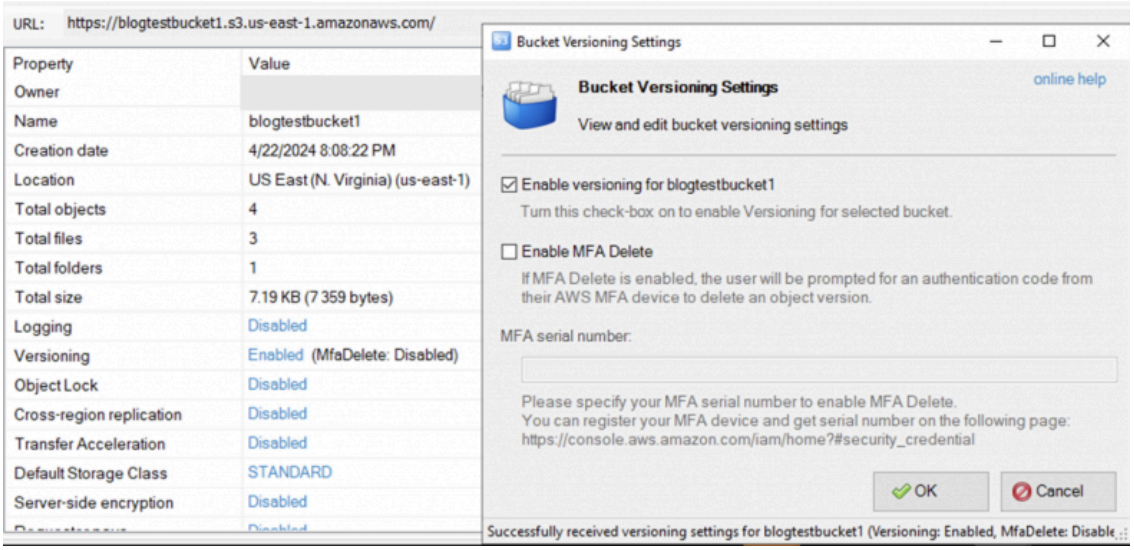


Figure 14. Attempting to modify the versioning properties.

Figure 15. Request Parameters of the API call when enabling the versioning for bucket.

- Attempting to disable public block access for the bucket (shown in Figure 16) results in the API calls below, which first gather the public access block configuration and then attempts to remove it.
    - GetBucketPublicAccessBlock
    - DeleteBucketPublicAccessBlock
    - PutBucketAcl

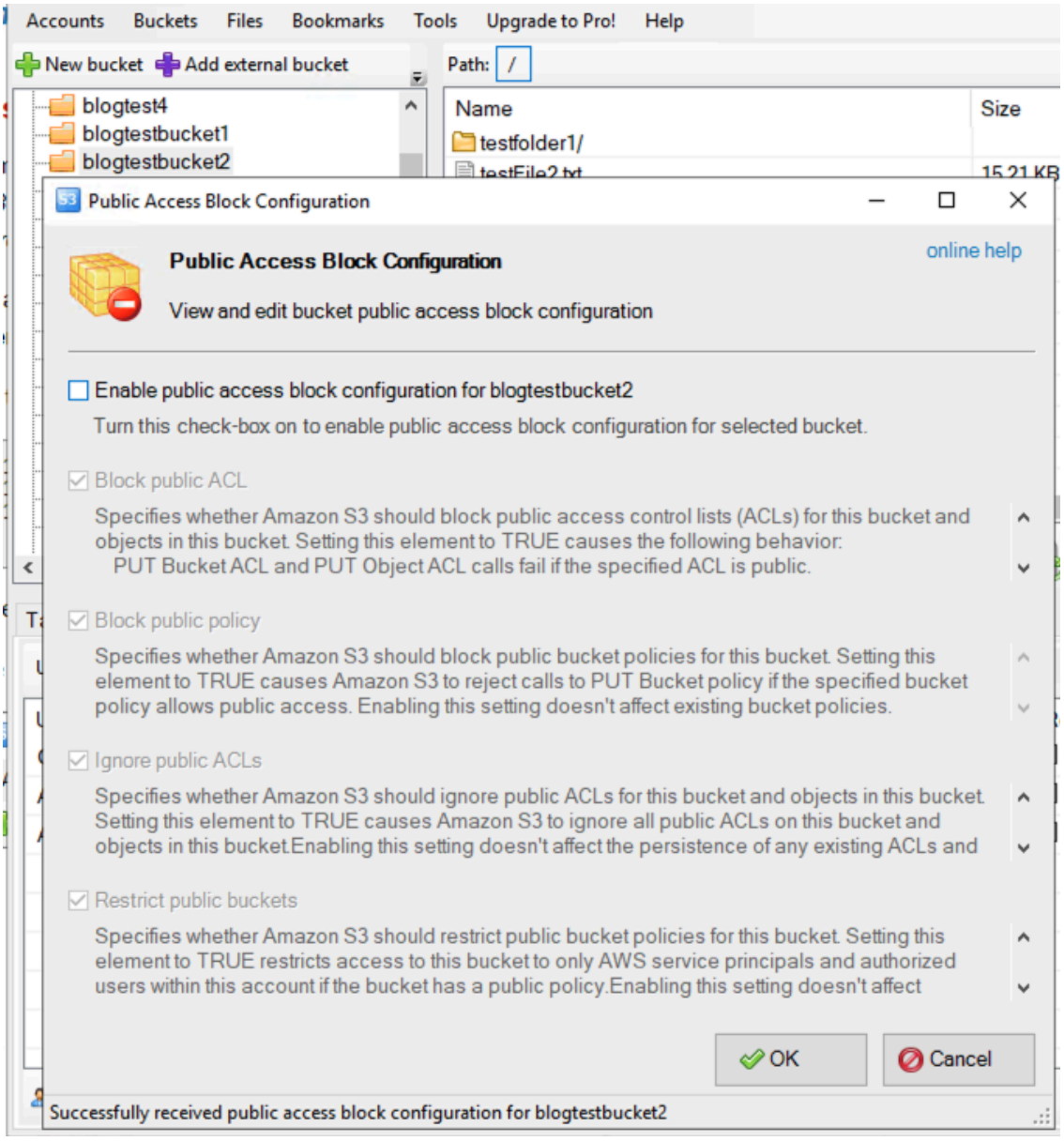


Figure 16. Attempting to modify the public access block configuration.

WinSCP

Windows Secure Copy, commonly known as WinSCP, is a popular file transfer application primarily used for transferring files between local and remote systems using various protocols such as SFTP, SCP, FTPS and FTP. While not specifically designed for Amazon S3, WinSCP can be configured to interact with S3 buckets. While WinSCP provides versatile file transfer capabilities, it does not offer dedicated features for managing Amazon S3 storage, such as advanced bucket and object management functionalities found in S3 Browser.

Unit 42 tested to understand what API calls are automatically made when performing specific actions. WinSCP API calls can be tracked via the user agent field in the CloudTrail logs: WinSCP/<version>.

These test activities took place against a bucket we created with data events logging enabled in CloudTrail to compare the visibility in management versus data events.

- The first connection to the S3 storage service after entering the access key credentials (shown in Figure 17) results in the API call below to list all the buckets in the GUI:
    - ListBuckets

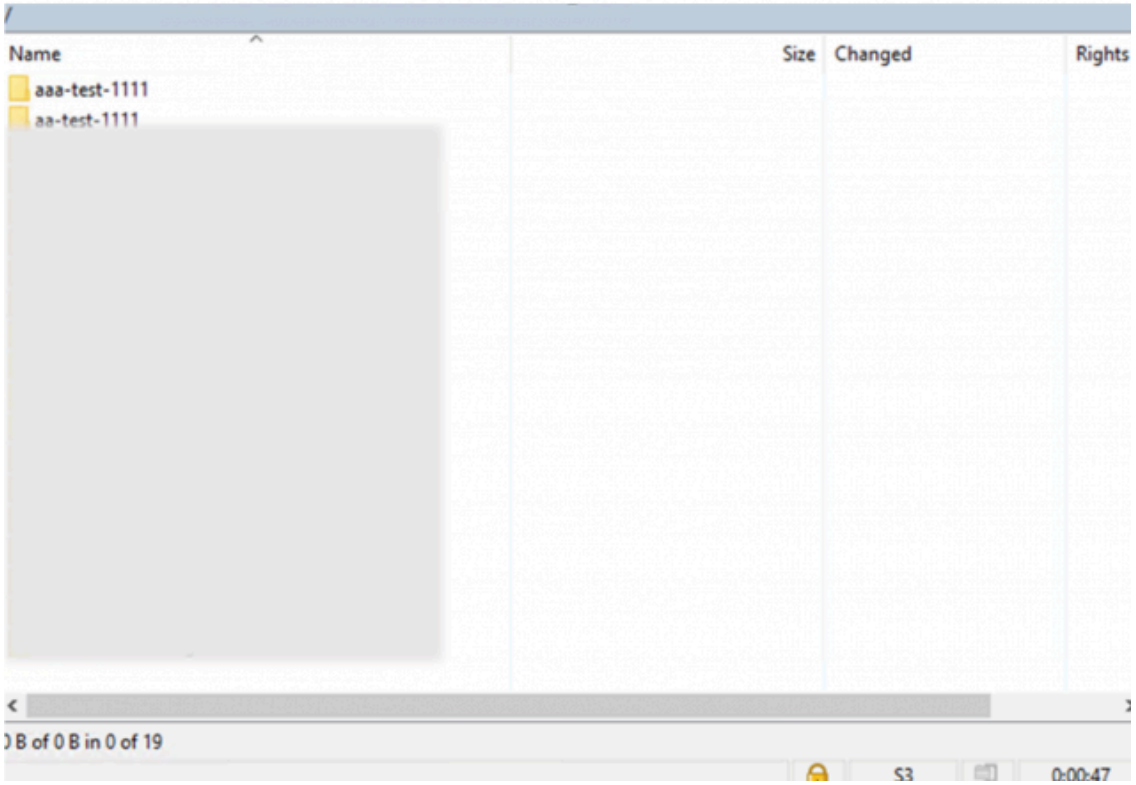


Figure 17. Connection to S3 using WinSCP.

- Viewing the properties of an object will result in an API call that returns metadata about the object such as location, size, owner, ACL as shown in Figure 18.
    - GetObjectAcl*

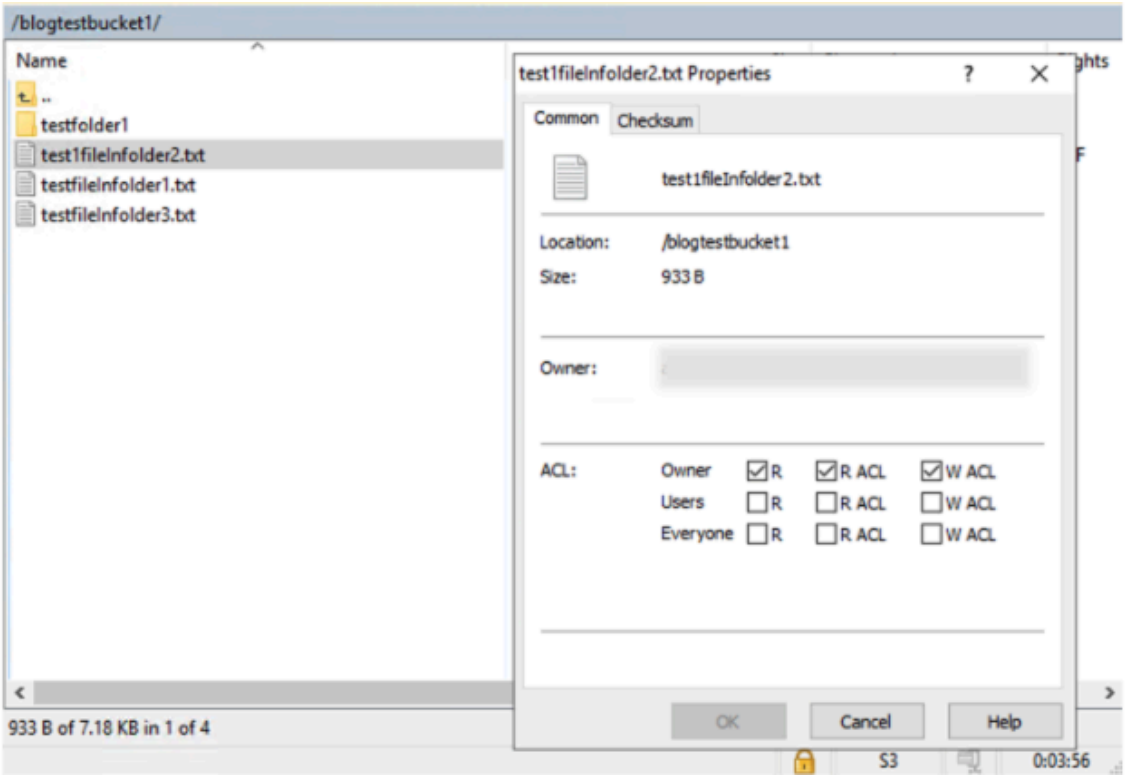


Figure 18. Viewing the properties of an object in WinSCP.

- Downloading, deleting and creating objects and buckets generated the API calls below.
    - GetObject*
    - DeleteObject*
    - PutObject*
    - CreateBucket
    - DeleteBucket

When comparing the two tools, S3 Browser generates a lot more API calls that automatically appear in the CloudTrail logs based on user interaction, compared to WinSCP. The difference in the events generated in the CloudTrail logs comes down to the purpose of the two tools. Being a cloud native tool, S3 Browser takes advantage of more AWS features that generate additional API calls versus WinSCP, which works for more file transfer types than solely S3. Figure 19 shows the entire attack broken down based on API calls to display the various results from each tool used.

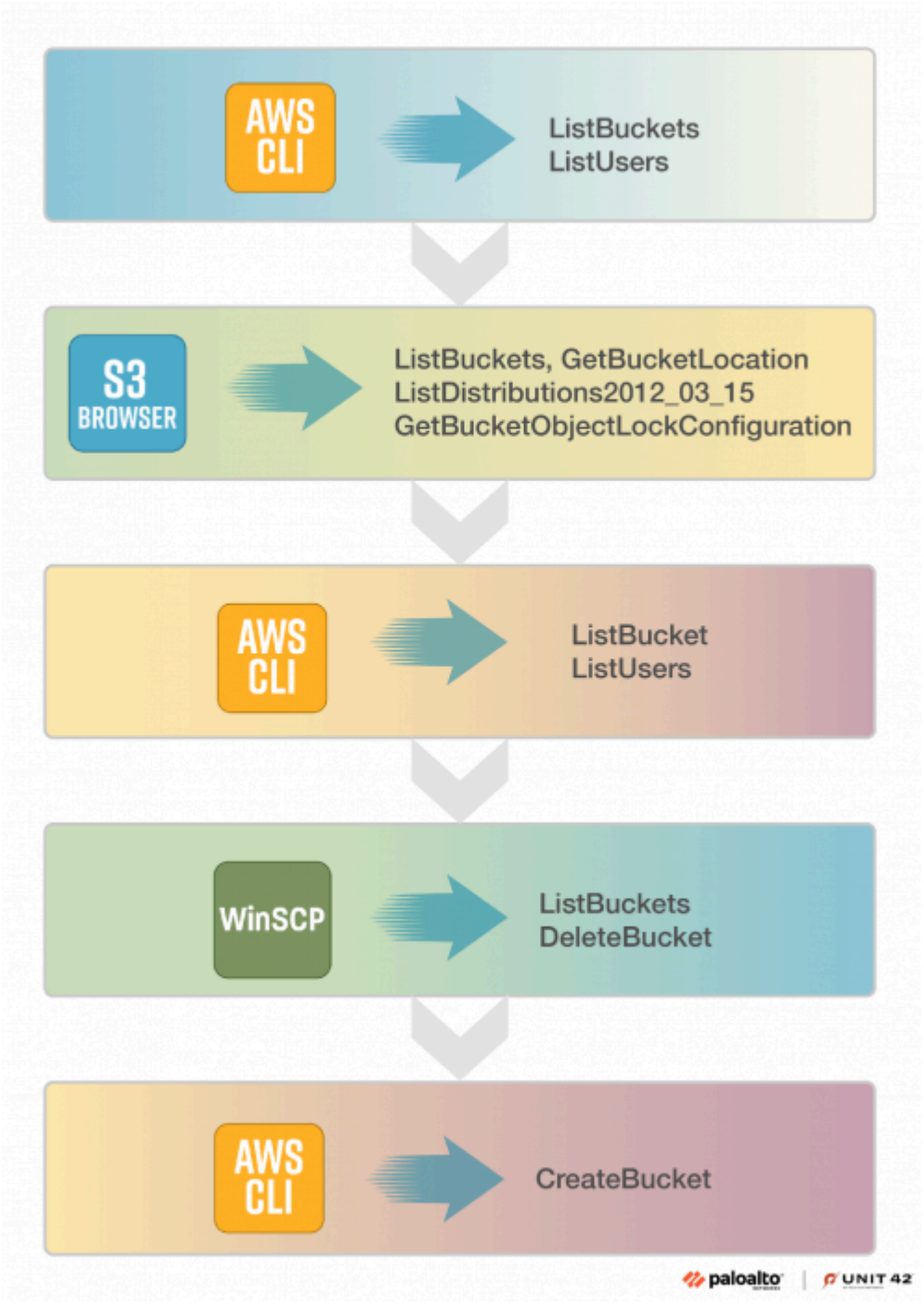


Figure 19. API call-delineated attack chain.

Conclusion

As organizations increasingly migrate their critical operations to the cloud, we continuously witness a concerning trend of overly permissive credentials. Threat actors like Bling Libra have evolved their tactics to exploit misconfigurations and exposed credentials in cloud environments.

Due to limited permissions of the compromised access keys, the breach covered in this post did not have an impact past the S3 buckets. However, in the past, Unit 42 has observed threat actors abusing overly permissive

credentials to create resources for malicious use or modifying the IAM users and permissions to maintain persistence within an environment.

Use IAM Access Analyzer to better identify and manage access risks by analyzing resource policies. Additionally, when using AWS Organizations, AWS Service Control Policies and permission boundaries can be used to help ensure that only permitted actions are allowed, regardless of the individual IAM policies within each account.

Figure 20 shows the complete attack path taken by the threat actor as grouped by MITRE tactics.

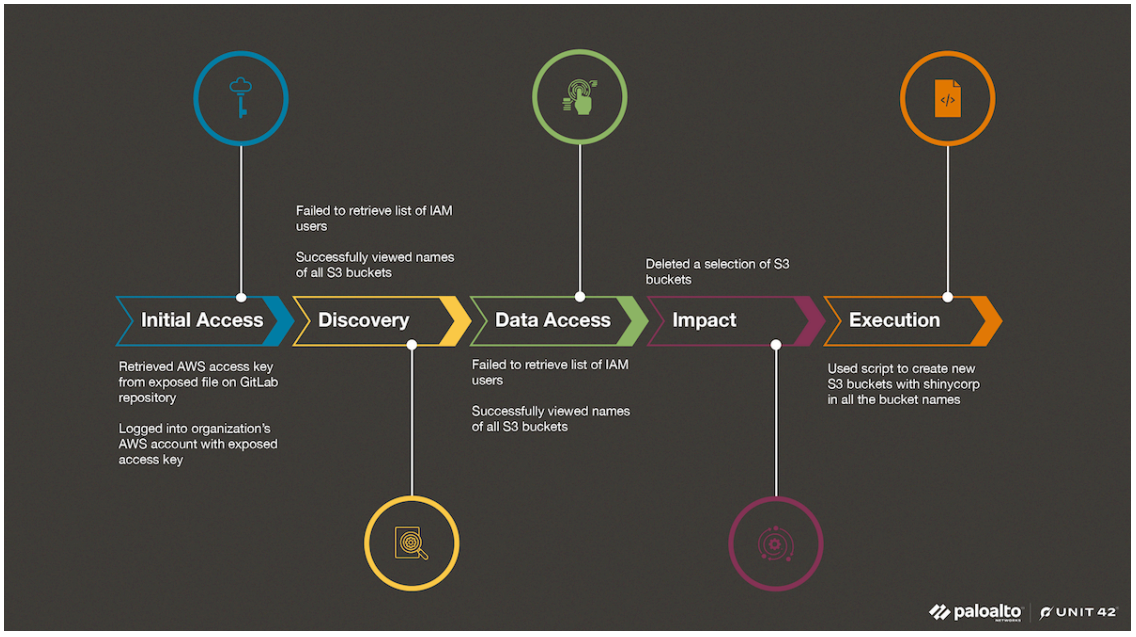


Figure 20. The MITRE timeline details the attack path taken by the threat actor.

Ensuring appropriately configured security configurations lays the groundwork to better protect organizations against potential breaches and data compromises. In this attack, attaching an S3 bucket policy requiring MFA to perform sensitive API calls such as DeleteBucket would have provided additional layers of protection against data loss. For critical data, we recommend replicating the data into another region or account to help ensure availability and resilience against security breaches including data destruction.

Ensuring adequate protection also involves regularly auditing and updating access controls, encryption settings and network configurations to align with best practices and compliance requirements. Leveraging services like AWS Config and AWS Security Hub provides continuous monitoring and assessment of the AWS environment's security posture.

Additionally, comprehending the attack lifecycle and the tactics, techniques and procedures (TTPs) of threat actors allows defenders to build the proper understanding for safeguarding cloud environments. By understanding how adversaries operate and the stages they go through to achieve their objectives, security analysts can proactively configure and monitor the necessary log sources in AWS . This allows defenders to more effectively detect and respond to cloud threats. Additionally, integrating Amazon GuardDuty for threat detection further strengthens security postures.

Palo Alto Networks customers are better protected from the threats discussed above through the following products:

- Cortex XDR for cloud can offer a comprehensive incident story by integrating activity from cloud hosts, cloud traffic and audit logs together with endpoint and network data.
- Palo Alto Networks' customers can also take advantage of Prisma Cloud to monitor posture and maintain compliance across public clouds.
- Palo Alto Networks' Cloud Security Agent (CSA) uses XSIAM to provide improved security posture and further enable detection and runtime monitoring capabilities to critical cloud infrastructure through both Prisma Cloud and Cortex cloud agents.

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

Palo Alto Networks has shared these findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

IoCs

Threat actor email address

- shinycorp@tutonota[.]com

User Agents

(X stands for varying version numbers)

- S3 Browser/X.X.X (https://s3browser.com)
- WinSCP/X.X.X neon/X.X.X
- aws-cli/X.X.X Python/X.X.X Linux/X.X.X-aws botocore/X.X.X
- aws-cli/X.X.X md/Botocore#X.X.X ua/X.X os/linux#X.X.X-aws md/arch#x86_64 lang/python#X.X.X md/pyimpl#CPython cfg/retry-mode#legacy botocore/X.X.X
- aws-cli/X.X.X md/Botocore#X.X.X md/awscrt#X.X.X ua/2.0 os/linux#X.X.X md/arch#x86_64 lang/python#X.X.X md/pyimpl#CPython cfg/retry-mode#legacy botocore/X.X.X
