Cloud Ransomware: Trends & Defense Strategies

Archived: 2026-04-05 13:20:23 UTC

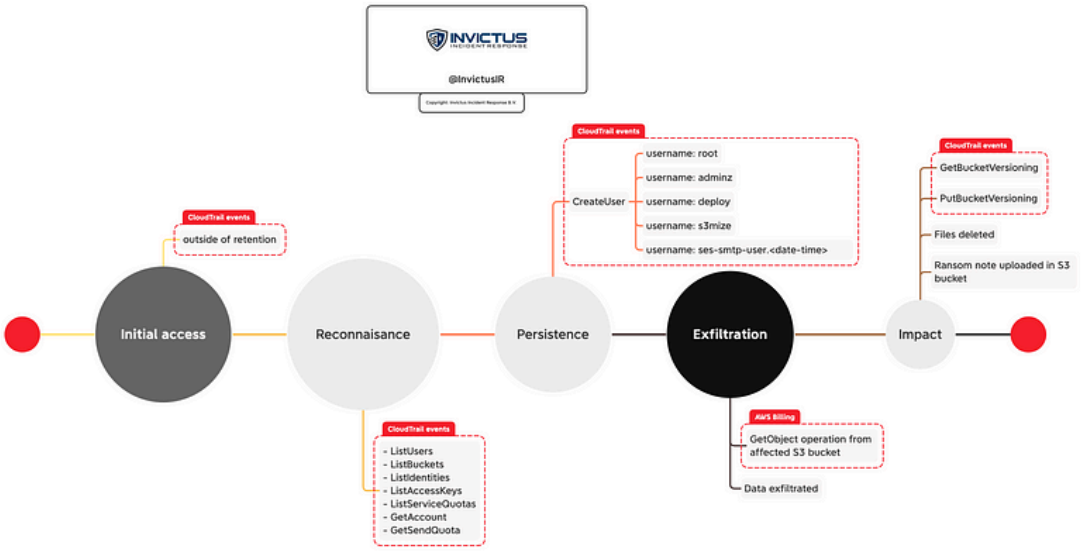
Background

Recently we were engaged by a company after they were targeted by a ransomware attack in their AWS environment. In this blog we want to show you what happened and how we were able to piece together the picture based on available logging.

Due to confidentiality we will be using censored screenshots to protect our client's information. They approved the publication of this blog, to prevent other companies from becoming a victim to a similar attack.

Attack overview

The overall attack activity is mapped to the MITRE ATT&CK steps as shown in the figure below:



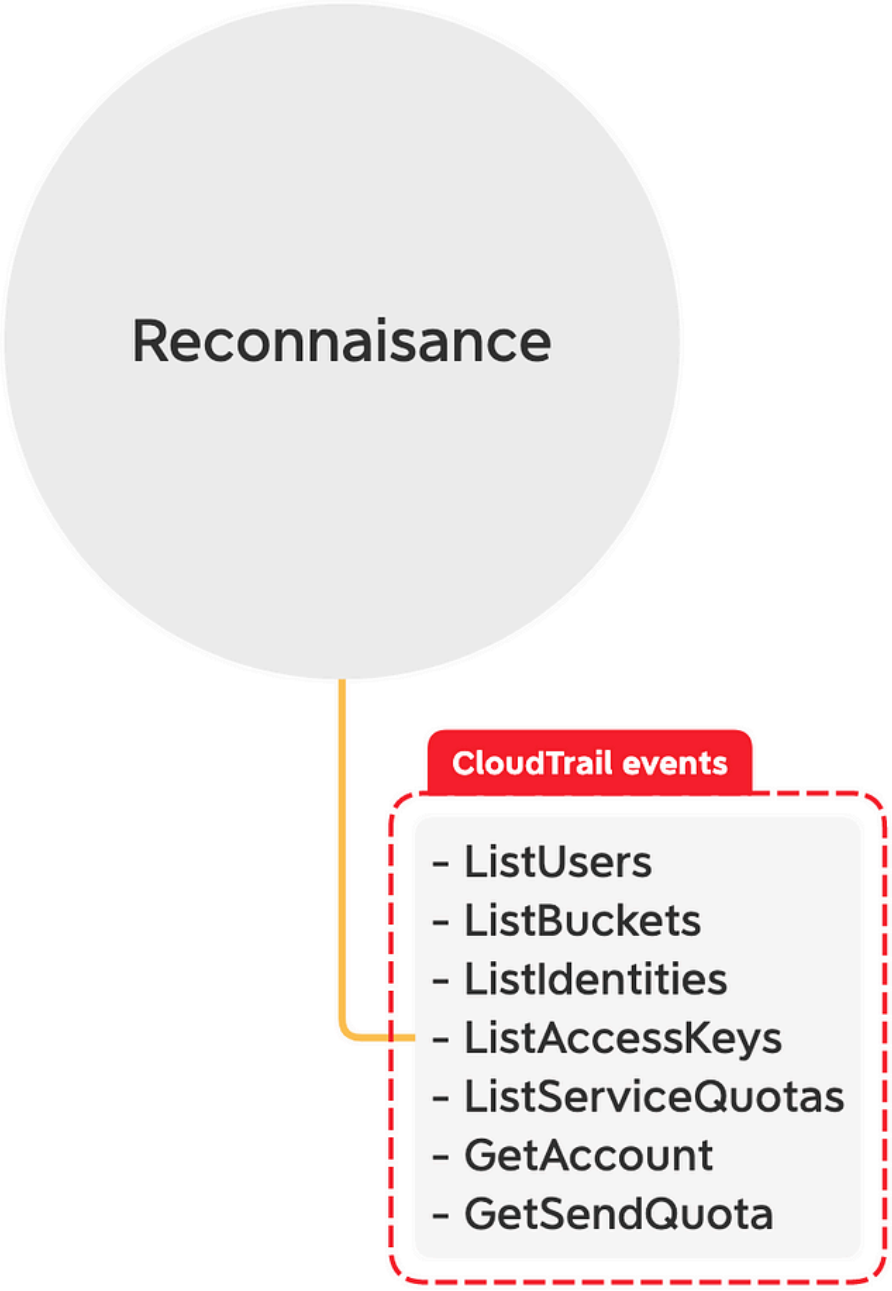
Incident timeline

Initial Access

The threat actor was able to get into the environment due to accidentally exposed long-term credentials. The first malicious activity happened outside of the 90-day retention period of CloudTrail. However, based on analysis of subsequent events and open-source analysis we were able to determine that a specific access key was used which was publicly exposed. Luckily the access key was for an account that only had rights to a specific S3 bucket.

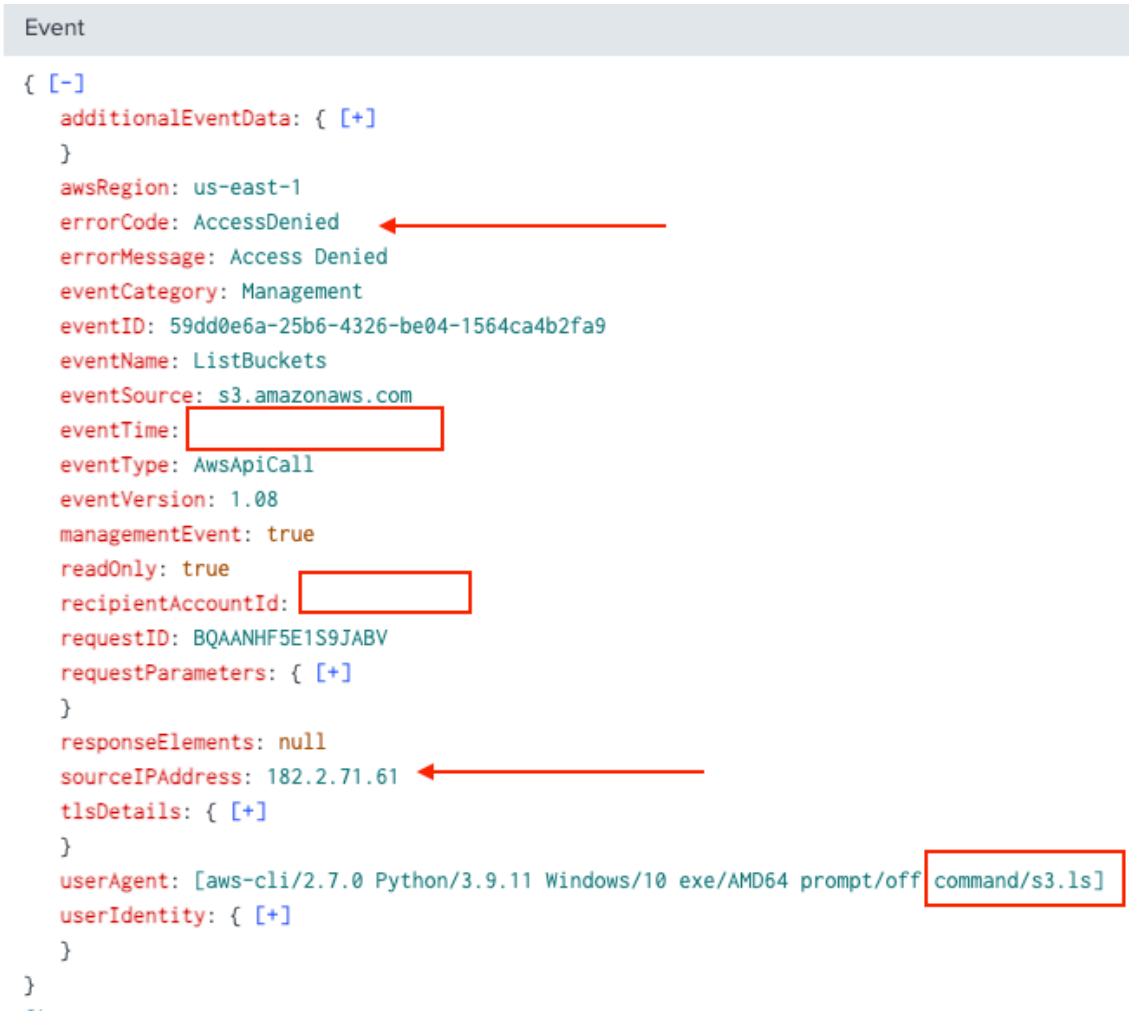
Reconnaissance

Following the initial access the threat actor performed the following activities for reconnaissance.



Most of the activities are self-explanatory and they are attempts to list other users, buckets and any available access keys.

The more interesting calls are around Quotas the `ListServiceQuotas` and `GetSendQuota` events are related to the AWS Simple Email Solution (SES) service. We've seen that SES is an interesting attack vector for threat actors, because they can leverage SES for spamming and (spear)phishing campaigns. Because the access keys that were used for making these calls had limited permissions these calls all resulted in AccessDenied events as shown in an example below.



In the above example we can see that the treat actor also uses the AWS CLI package on a Windows host and used the `aws s3` ls command. (reference)

Persistence

The threat actor attempted to create a number of additional users which is captured in CloudTrail under the `CreateUser` event name. The following user creations were attempted:

- root
- adminz
- deploy
- s3mize
- ses-smtp-user.<date-time-of-creation>

All of the above actions were again denied. Note that the SES user has a default naming convention which will tell you the (attempted) creation date and time. Also the attempt to create a `root` user is interesting as each AWS account by default has a root user. By creating an IAM user with the name root it might be a method to stay under the radar. We were unable to find any interesting leads on the other user accounts in public code repositories, but maybe someone else knows more.

Exfiltration

Whilst most of the attempted activity failed, using the access key the threat actor did have full access to a S3 bucket. The threat actor used this access to exfiltrate data. We were able to confirm this by using the AWS billing information which contained an entry for a `GetObject` Operation which is recorded when a file is downloaded from a S3 bucket. The AWS billing can be a useful tool for detecting data exfiltration, as any data leaving the AWS buckets incurs egress costs that are recorded in the bill.

If you want to investigate this yourself, use the following process:

- Go to AWS billing, then select Cost & Usage Reports;
- Select the Create a Usage Report link under AWS Usage Report and select the time period of the incident and the S3 service;
- Open the CSV file in something like Excel and sort by the `UsageValue` column from large to small;
- Filter on DataTransfer-Out or C3DataTransfer-Out Bytes in the `UsageTypecolumn`;
- Search for the `GetObjectOperation` which contains file downloads;
- You'll get one or more entries with aggregated data for a period of 1-hour related to outgoing data transfers.

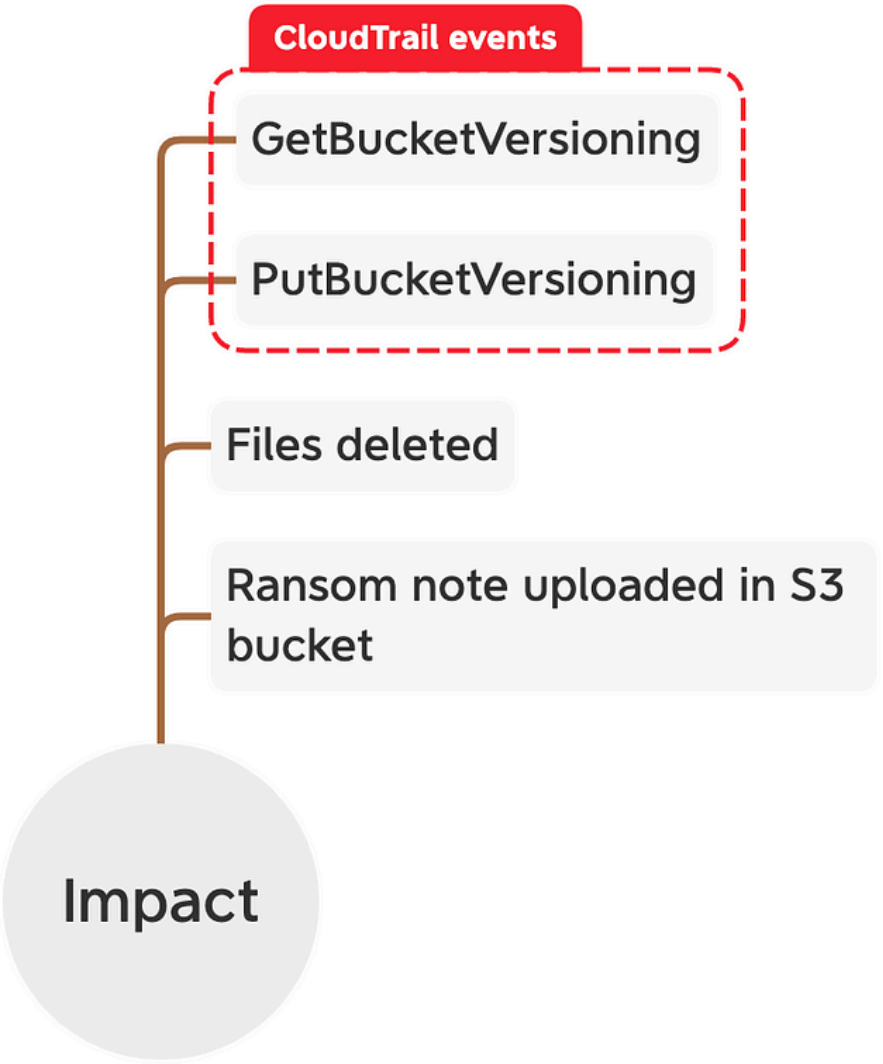
In case of data being exfiltrated from a bucket you could see something like this in the billing report:

	Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue	Gigabyte
148	Amazon53	GetObject	EU-DataTransfer-Out-Bytes	invictus-test-bucket	04/01/2023 10:00	04/01/2023 11:00	13333333377	12,4176344

In the above example, data was transferred out of the EU region from a bucket named `Invictus-test-bucket`. With this information, we know the time period during which the transfer occurred and the number of bytes that were transferred. The billing information doesn't show where the transfer went, but it can help figure out if data was exfiltrated, especially if there's no record of it in CloudTrail.

Impact

After the exfiltration of the data the threat actor, disabled bucket versioning, deleted data from several buckets and left behind a ransom note.



As mentioned earlier data events were not audited in CloudTrail for this environment, so we can't see the individual delete events, but the data from the bucket was deleted. It's also interesting to see what the threat actor did before file deletion. The following events were recorded in CloudTrailGetBucketVersioning and PutBucketVersioning.

> Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket. You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets. With versioning you can recover more easily from both unintended user actions and application failures. Source

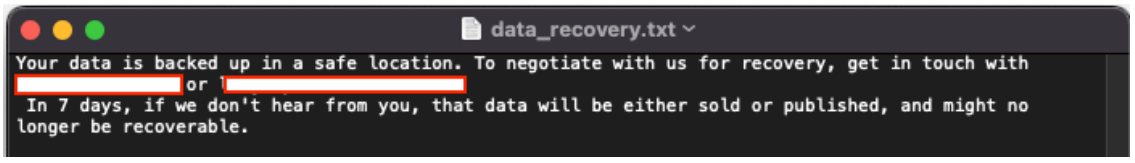
The threat actor used the GetBucketVersioningcall to establish whether versioning was enabled for the S3 bucket. Versioning would've allowed our client to easily restore data if it was deleted. In this case the versioning was enabled, the next call was to change the versioning settings with PutBucketVersioning as shown in the picture below.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": [redacted]
        "arn": [redacted]
        "accountId": [redacted]
        "accessKeyId": [redacted]
        "userName": [redacted]
    },
    "eventTime": "[redacted]
    "eventSource": "s3.amazonaws.com",
    "eventName": "PutBucketVersioning",
    "awsRegion": [redacted]
    "sourceIPAddress": "64.226.75.246",
    "userAgent": "[Boto3/1.13.14 Python/3.9.2 Linux/5.10.0-21-amd64 Botocore/1.20.0 Resource]",
    "requestParameters": {
        "bucketName": [redacted]
        "Host": [redacted]
        "versioning": "",
        "VersioningConfiguration": {
            "Status": "Suspended",
            "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/"
        }
    },
```


We've censored some confidential details such as the bucket and user that performed the action. Important to know is that the userName field will contain the user responsible for this action and the bucketName will contain the bucket for which the versioning was changed. Under Status we can see that the versioning was 'Suspended'.

Ransomware note

The following ransomware note was left behind in the affected buckets.



CloudTrail events

The below table highlights the CloudTrail events we've come across for this case. You can use these events for manual alerting in CloudWatch or setup rules in your SIEM. We've added a note on the usability of the events to detect threats based on our experience, it might be different for your environment.

	eventName	Phase	Note
	ListUsers	Reconnaissance	Useful usage in your environment might vary due to applications requesting this information
	ListBuckets	Reconnaissance	Useful usage in your environment might vary due to applications requesting this information

	eventName	Phase	Note
	ListIdentities	Reconnaissance	Can be noisy due to applications requesting this information
	ListAccessKeys	Reconnaissance	Can be noisy due to applications requesting this information
	ListServiceQuotas	Reconnaissance	Useful limited to AWS SES so should be rarely used
	GetAccount	Reconnaissance	Can be noisy due to applications requesting this information
	GetSendQuota	Reconnaissance	Useful limited to AWS SES so should be rarely used
	CreateUser	Persistence	Highly useful you should know who is able to create users
	GetBucketVersioning	Impact	Highly useful should be limited usage in your environment
	PutBucketVersioning	Impact	Highly useful should be limited usage in your environment

Recommendations

The following recommendations are tailored to the prevention, detection, response and recovery of a ransomware incident in AWS.

- Enable a trail in CloudTrail to store data in a S3 bucket which allows for longer data retention;
- Enabled CloudTrail for data events, this can generate a lot of events and comes with an added cost, prioritize based on where your most important data is stored;
- Limit the usage of long-term access key, where possible use IAM roles. E.g. use an IAM role for an application hosted on EC2 that needs to store data in an S3 bucket and not an access key;
- Protect your access keys by regularly rotating them and monitoring for abuse, follow best practices guide by AWS;
- Enable bucket versioning with MFA delete, this will limit the ability to change bucket versioning settings, because MFA is required;
- Use AWS Backup for immutable backups, excellent blog by AWS here.

Conclusion

Ransomware is a threat for all organizations not just limited to on-premise environments. Threat actors will follow you to the cloud and a misconfiguration in the cloud is very easy to make with some very serious consequences as highlighted in this case.

Indicators Of Compromise (IOCs)

Please find below a list of IOCs we encountered during this attack.

	Indicator Type	Indicator	
	IP address	139.99.120.65	
	IP address	139.99.123.180	
	IP address	139.99.68.31	
	IP address	164.90.200.10	
	IP address	193.148.18.59	
	IP address	198.135.55.189	
	IP address	198.98.183.38	
	IP address	212.102.33.34	
	IP address	64.226.75.246	
	IP address	95.142.120.45	
	IP address	95.142.120.58	
	IP address	95.142.120.75	
	IP address	114.10.26.42	
	IP address	114.125.126.25	
	IP address	120.188.39.58	
	IP address	125.162.111.178	
	IP address	140.213.132.181	
	IP address	140.213.138.160	
	IP address	140.213.138.35	
	IP address	158.140.163.5	
	IP address	180.242.79.14	
	IP address	180.244.166.155	
	IP address	182.1.119.185	
	IP address	182.1.122.169	
	IP address	182.2.71.61	

	Indicator Type	Indicator	
	IP address	36.85.32.120	
	IP address	36.85.32.129	
	IP address	36.85.35.148	
	IP address	36.85.36.50	
	IP address	36.85.37.4	
	IP address	36.85.38.180	
	IP address	36.85.38.45	
	IP address	36.85.39.118	
	IP address	36.85.39.21	
	User-Agent	Boto3/1.13.14 Python/3.9.2 Linux/5.10.0-21-amd64 Botocore/1.20.0	
	User-Agent	Boto3/1.24.75 Python/3.9.6 Windows/10 Botocore/1.27.75	
	User-Agent	Boto3/1.24.84 Python/3.10.7 Windows/2012ServerR2 Botocore/1.27.84	
	User-Agent	Boto3/1.24.84 Python/3.9.7 Windows/10 Botocore/1.27.84	
	User-Agent	Boto3/1.25.0 Python/3.10.6 Linux/5.15.0-58-generic Botocore/1.28.0	
	User-Agent	Boto3/1.26.22 Python/3.9.5 Windows/10 Botocore/1.29.22	
	User-Agent	Boto3/1.26.40 Python/3.11.1 Windows/2012ServerR2 Botocore/1.29.40	
	User-Agent	Boto3/1.26.51 Python/3.11.1 Windows/10 exec-env/EC2 Botocore/1.29.51	
	User-Agent	Boto3/1.26.54 Python/3.11.1 Windows/10 exec-env/EC2 Botocore/1.29.54	
	User-Agent	Boto3/1.26.60 Python/3.11.1 Windows/10 Botocore/1.29.60	
	User-Agent	Boto3/1.26.64 Python/3.11.1 Windows/10 Botocore/1.29.64	
	User-Agent	Boto3/1.26.69 Python/3.8.13 Linux/5.15.0-47-generic Botocore/1.29.69	
	User-Agent	Boto3/1.26.71 Python/3.11.2 Windows/10 exec-env/EC2 Botocore/1.29.71	
	User-Agent	Boto3/1.26.76 Python/3.11.2 Windows/10 exec-env/EC2 Botocore/1.29.76	
	User-Agent	Boto3/1.26.87 Python/3.8.13 Linux/5.15.0-47-generic Botocore/1.29.87	
	User-Agent	Boto3/1.26.90 Python/3.8.16 Linux/5.4.0-1097-aws Botocore/1.29.90	

	Indicator Type	Indicator	
	User-Agent	Boto3/1.26.92 Python/3.11.2 Windows/10 exec-env/EC2 Botocore/1.29.92	
	User-Agent	[Boto3/1.13.14 Python/3.9.2 Linux/5.10.0-21-amd64 Botocore/1.20.0 Resource]	
	User-Agent	[aws-cli/1.19.112 Python/2.7.18 Linux/4.4.0-19041-Microsoft botocore/1.20.112]	
	User-Agent	[aws-cli/2.7.0 Python/3.9.11 Windows/10 exe/AMD64	
	User-Agent	aws-cli/1.18.69 Python/3.8.10 Linux/5.4.0-137-generic botocore/1.16.19	
	User-Agent	aws-cli/1.19.1 Python/3.9.2 Linux/5.10.0-21-cloud-amd64 botocore/1.20.0	
	User-Agent	aws-cli/1.19.112 Python/2.7.18 Linux/4.4.0-19041-Microsoft botocore/1.20.112	
	User-Agent	aws-cli/1.22.65 Python/3.8.10 Linux/4.4.0-19041-Microsoft botocore/1.27.20	
	User-Agent	aws-cli/1.24.10 Python/3.6.9 Linux/4.4.0-22621-Microsoft botocore/1.26.10	
	User-Agent	aws-cli/1.25.17 Python/3.9.7 Windows/10 botocore/1.27.26	
	User-Agent	aws-cli/1.27.59 Python/3.8.16 Linux/5.4.0-1085-aws botocore/1.29.59	
	User-Agent	aws-cli/1.27.81 Python/3.8.16 Linux/5.4.0-1085-aws botocore/1.29.81	
	User-Agent	aws-cli/1.27.83 Python/3.8.16 Linux/5.4.0-1085-aws botocore/1.29.83	
	User-Agent	aws-cli/2.10.1 Python/3.9.11 Linux/5.15.90.1-microsoft-standard-WSL2 exe	
	User-Agent	aws-cli/2.10.1 Python/3.9.11 Linux/5.15.90.1-microsoft-standard-WSL2 exe/x86_64.ubuntu.22	
	User-Agent	aws-cli/2.11.2 Python/3.11.2 Windows/10 exec-env/EC2 exe/AMD64 prompt	
	User-Agent	aws-cli/2.7.0 Python/3.9.11 Windows/10 exe/AMD64	
	User-Agent	aws-cli/2.9.19 Python/3.11.1 Linux/4.4.0-19041-Microsoft source/x86_64.kali.2022	
	User-Agent	aws-cli/2.9.20 Python/3.9.11 Linux/4.4.0-19041-Microsoft exe/x86_64.ubuntu.18	
	User-Agent	aws-cli/2.9.23 Python/3.9.11 Windows/10 exec-env/EC2 exe/AMD64	

About Invictus Incident Response

We are an incident response company and we ❤️ the cloud and specialise in supporting organisations facing a cyber attack. We help our clients stay undefeated!

🆘 Incident Response support reach out to cert@invictus-ir.com or go to https://www.invictus-ir.com/247

📧 Questions or suggestions contact us at info@invictus-ir.com

---