zscaler™

ANATOMY OF A CLOUD BREACH:

How 100 Million Credit Card Numbers Were Exposed.

The US-based major financial institute data breach two years ago—which exposed the personal data of more than 100 million customers—was one of the most devastating data breaches of all time. This major cloud data breach serves as a valuable lesson for any organization storing confidential information in the cloud.

Below were the steps taken by the attackers to compromise the data.

STEP 01

Identified and exploited misconfigured WAF

The attacker identified a misconfigured WAF that enabled accessing the corresponding AWS EC2 instance/ECS task *metadata* using Server-side Request Forgery (SSRF) and called the metadata service endpoint using: http://169.254.169.254/iam/security-credentials command.

The endpoint returned a role (according to the indictment '*****-WAF-Role'). The EC2 instance was configured with Metadata Service v1 which is a weaker version.

STEP 02

Gained temporary credentials

Using the role name, the attacker then queried the specific endpoint to gain access to temporary credentials.

The role in place allowed the EC2 instance to access S3 buckets.

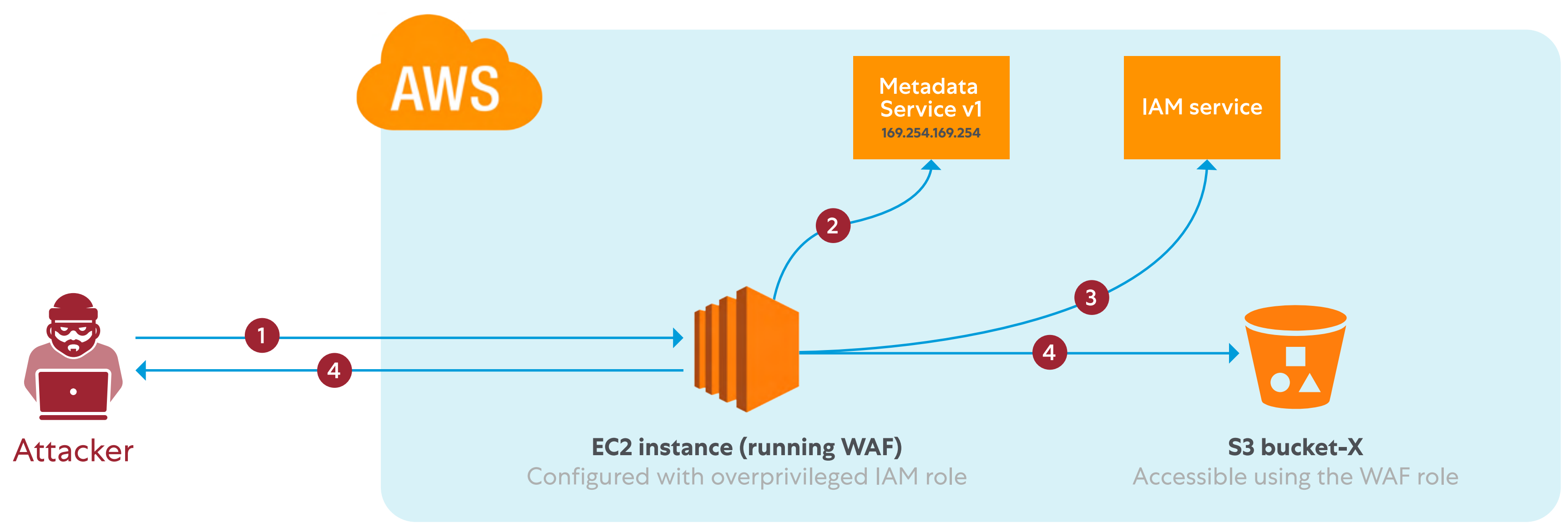
STEP 03

Gained access to S3 buckets

The attacker gained access to S3 buckets by calling the AWS S3 list and Sync CLI commands:
$ aws s3 ls

The ls command lists all S3 buckets accessible using the assumed IAM role:
$ aws s3 sync s3://somebucket

The sync command would then download all resources from the specified S3 bucket(s).


AWS
Metadata Service v1
169.254.169.254
IAM service
Attacker
1
2
3
4
4
EC2 instance (running WAF)
Configured with overprivileged IAM role
S3 bucket-X
Accessible using the WAF role

2 Invokes Metadata Service v1 service to get security credentials for a specific role (***WAF role)

```
$ curl http://169.254.169.251/iam/security-credentials/(specific role ***WAF role)
returns credential {AccessKey1, Secret_passcode_1}
```

3 Uses credentials and assumes the WAF role that is permitted to access S3 buckets

4 Lists all S3 buckets, picks one of them, and syncs (downloads) its content

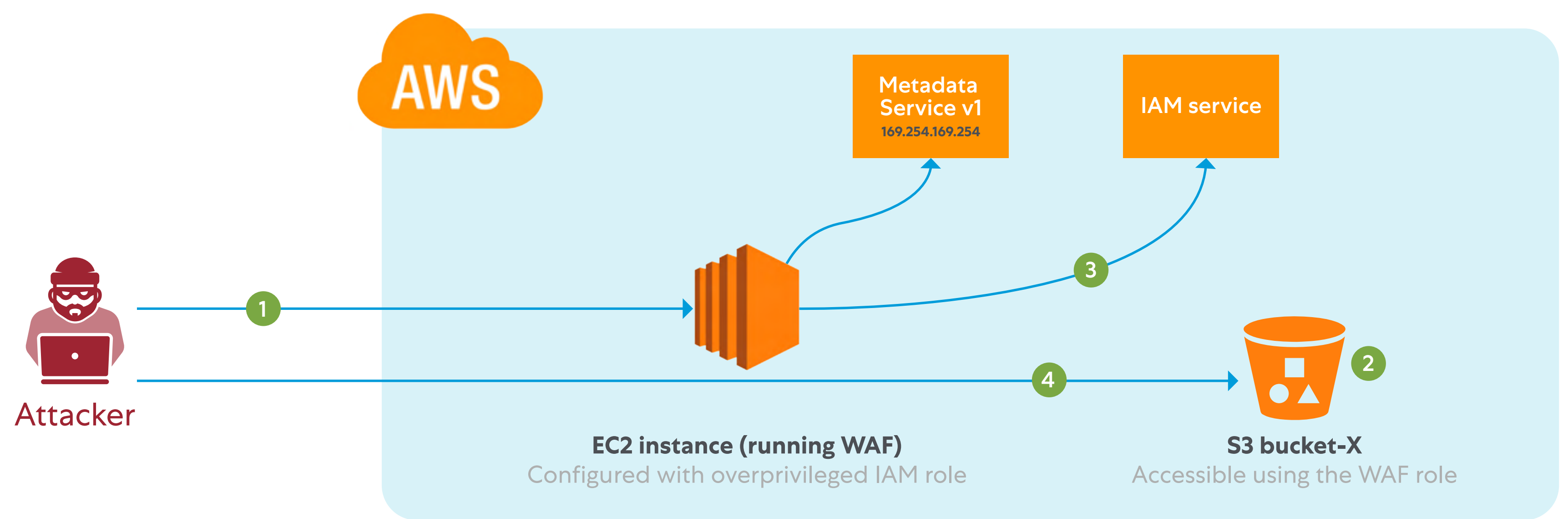
```
$ aws s3 is
$ aws s3 syncs3://somebucket > would download all objects from somebucket
```

SUMMARY

The root cause of the attack

1. **Poor security architecture design:**
   - EC2 instance configured with Metadata Service v1. Metadata Service v2 would have masked/blocked the initial exploit of the WAF.
   - EC2 instance with WAF responsibilities having access to S3 buckets with confidential data.
2. **Exposed S3 buckets to an EC2 instance** which were not supposed to have access to those S3 buckets.
3. **AWS S3 buckets were not exposed** to the internet like many other breaches, an EC2 instance with an excessive IAM role might have been the culprit.

How CSPM and CIEM can help to prevent such attacks


AWS
Metadata Service v1
169.254.169.254
IAM service
Attacker
1
3
4
2
EC2 instance (running WAF)
Configured with overprivileged IAM role
S3 bucket-X
Accessible using the WAF role

Examples

CSPM
Validates infrastructure

1 Ensure all EC2 Instances are configured to exclusively use Instance Metadata Service v2

CIEM
Validates permissions

2 Ensure externally facing instances are not granted access to sensitive data.

3 Ensure least privilege access is set for all instance identities.

4 Validate there are no further anomalous permissions leveraging instance peer group analysis.

Recommendations

→ Identify and remediate violations

- Identify vulnerabilities — cybercriminals proactively look for security vulnerabilities. The original entry point into the financial institution system was through a compromised open-source WAF. Continuous monitoring and risk assessment will help to identify potential "soft spots."
- Monitor and remediate violations — remediate misconfigured resources, IAM permissions, over-privileged identities, and over-exposed data before they are exploited based on benchmarks and industry best practices.

→ Data protection

- Data classification — discover, classify, and protect sensitive data stored in AWS.
- Enable logging, especially for extremely sensitive buckets — ensure that you have a bucket, object, etc. logging turned on.

→ Enforce least privileged IAM controls

- Restrict broader access to all resources for applications on EC2 instances.
- Right-size scope of roles to access limited resources and limit access to specific identities in other accounts.
- Clean up unused cloud resources (especially EC2 instances and S3 buckets) leftover from prior development or production debugging efforts.
- Ensure that all IAM policies follow the least privileged model (despite the complexity of managing numerous IAM policies).

→ Services

- Use CloudTrail, CloudWatch, and/or AWS lambda services to review and automate specific actions taken on S3 resources.

LEARN MORE

How Can Zscaler Workload Posture Help?

Zscaler Workload Posture automates security, entitlements, and compliance in the cloud, delivering continuous visibility and enforcing adherence to the most comprehensive set of security policies, least privileges, and compliance frameworks.

For more information