SSS Pentesting

Cloud

Penetration Test Findings Report

Date: July 10th, 2024

Table of Contents

Confidentiality Statement

This document is the exclusive property of Example Corporation. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of Example Corporation or SSS Pentesting.

Example Corporation may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. SSS Pentesting prioritized the assessment to identify the weakest security controls an attacker would exploit. SSS recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

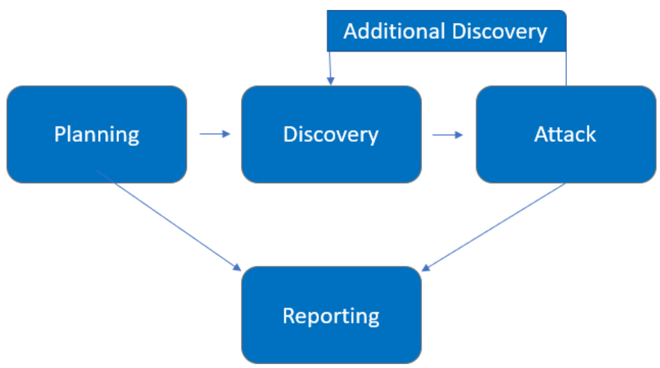
Name	Title	Contact Info
SSS Pentesting		
Sam Shepherd	Penetration Tester	sam@mail.com
Example Corporation		
Bob Bobson	Chief Information Security Officer	bob@example.com

Assessment Overview

From June 12th, 2024, to July 10th, 2024, Example Corporation engaged SSS Pentesting to evaluate the security posture of its cloud infrastructure compared to the current industry's best practices.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered, and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

Cloud Penetration Test

A cloud penetration test is a simulated attack on an organization's cloud-based environment conducted by a security professional or team to identify vulnerabilities that could be exploited by malicious actors.

Finding Severity Ratings

The following table defines severity levels and their corresponding CVSS score ranges, which are used throughout this document. These levels help assess risk by evaluating the likelihood and impact of each vulnerability.

Severity	CVSS V4 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Medium	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
Cloud Penetration Test	Example.onmicrosoft.com

Scope Exclusions

Per client request, SSS Pentesting did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Example Corporation.

Client Allowances

Example Corporation provided SSS Pentesting the following allowances:

- Authorization of tests on the Azure cloud environment.

Executive Summary

SSS Pentesting conducted a cloud penetration test of Example Corporation from June 12th to July 10th, 2024, to evaluate its cloud security. The assessment identified multiple critical vulnerabilities, including credentials exposed in the website's source code (Finding INT-001). This led to initial access to the cloud environment where it was later discovered that sensitive data was exposed in plain text. This is to include credit card information (Finding INT-002), access tokens (Finding INT-006), and credentials stored in Runbooks (Finding INT-005). Additionally, it was discovered that several users were using weak passwords (Finding INT-004) which can increase the risk of lateral movement and/or privilege escalation within the cloud environment. This report provides a high-level overview of these findings, their impact, and remediation strategies. For further details, refer to the Technical Findings section.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place. Cloud penetration testing was permitted for twenty-one (21) business days.

Tester Notes and Recommendations

The penetration test results indicate that Example Corporation had undergone its first penetration test. During testing, a recurring theme was misconfiguration of sensitive data storage. We recommend Example Corporation implement various methods to secure data which would include using encryption and implementing secret management tools.

On a positive note, example Corporation had strong logging and monitoring configurations. Overall, Example Corporation's cloud infrastructure performed as expected for a first-time penetration test. We recommend that the Example Corporation team thoroughly review the recommendations made in this report, correct the findings, and re-test annually to improve their overall security posture.

Key Strength and Weaknesses

The following identifies a key strength found during this assessment:

1. Strong logging and monitoring configurations.

The following identifies key weaknesses found during this assessment:

1. Source code misconfigurations.
2. Credential and key management issues.
3. Data protection challenges.

Vulnerability Summary & Report Card

The following table categorizes the vulnerabilities found by severity. Remediation recommendations are also provided.

2	2	2	0	0
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
Cloud Penetration Test		
INT-001: Credentials Found in Source Code	Critical	Disable old versions of programs that are no longer in use
INT-002: Sensitive Data Stored in Plain Text	Critical	Encrypt data at rest
INT-003: Dormant Accounts on the Cloud	High	Remove and disable user accounts that are no longer in use
INT-004: Weak Password Policy	High	Enforce a password policy to require a minimum character length of 12 and to include mixed character types
INT-005: Hard Coded Credentials Found in Runbooks	Medium	Secure client secrets in other secret management solutions
INT-006: Access Tokens Available in Clear Text	Medium	When done using Az PowerShell, use the command: Disconnect-Az Account to disconnect

Cloud Penetration Test Findings

Finding INT-001: Credentials Found in Source Code (Critical)

Description:	An inspection of the website’s source code revealed outdated Microsoft storage blobs containing hard-coded credentials.
Risk:	Likelihood: Very High – The source code is publicly available and therefore can be enumerated and exploited by potentially anyone who visits the site. Impact: Critical - This could result in initial access to the cloud environment.
System:	Example.onmicrosoft.com
Tools Used:	Kali Linux
References:	[Keep passwords out of source code — why and how | by Falk Tandetzky | NEW IT Engineering | Medium]

Evidence:

Figure 1.1: Shows a zip file that is accessible through the source code from an old version of microsoft blobs.

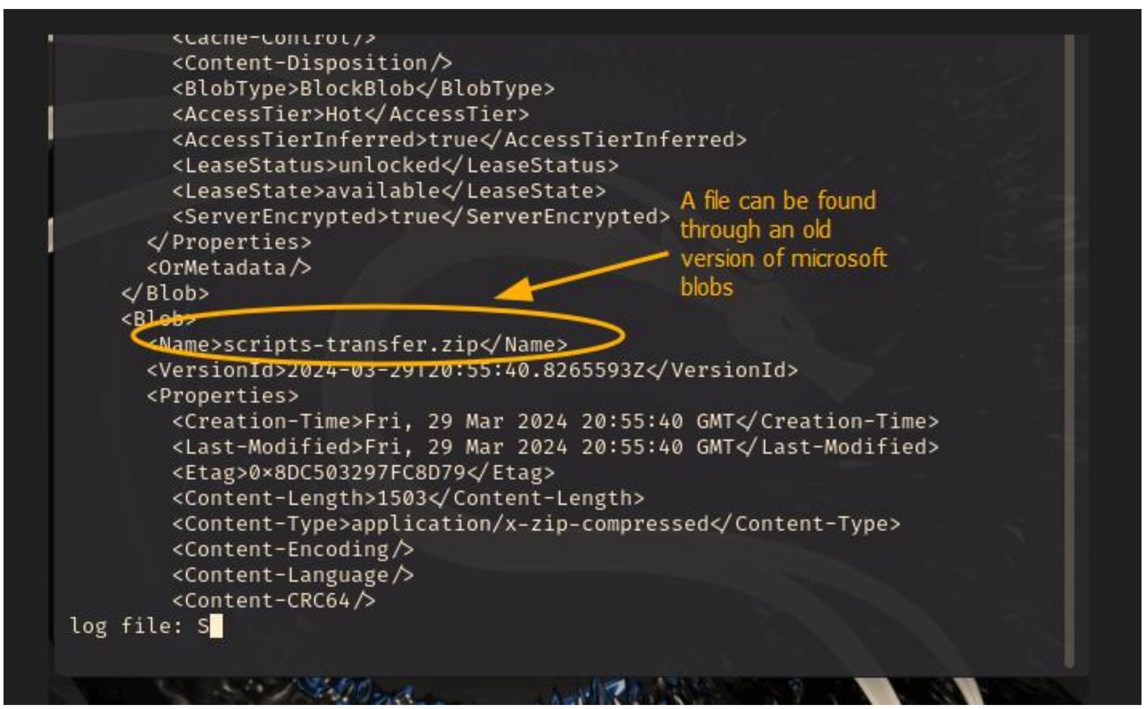
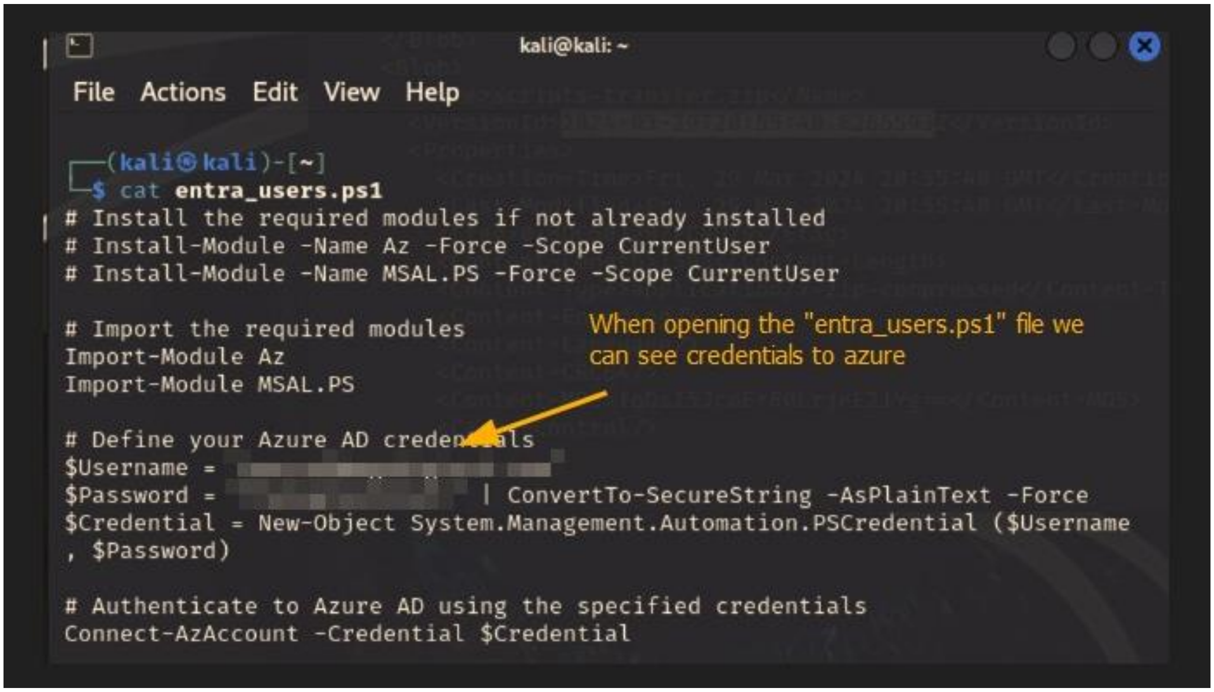


Figure 1.2: Shows credentials from unzipping the file.



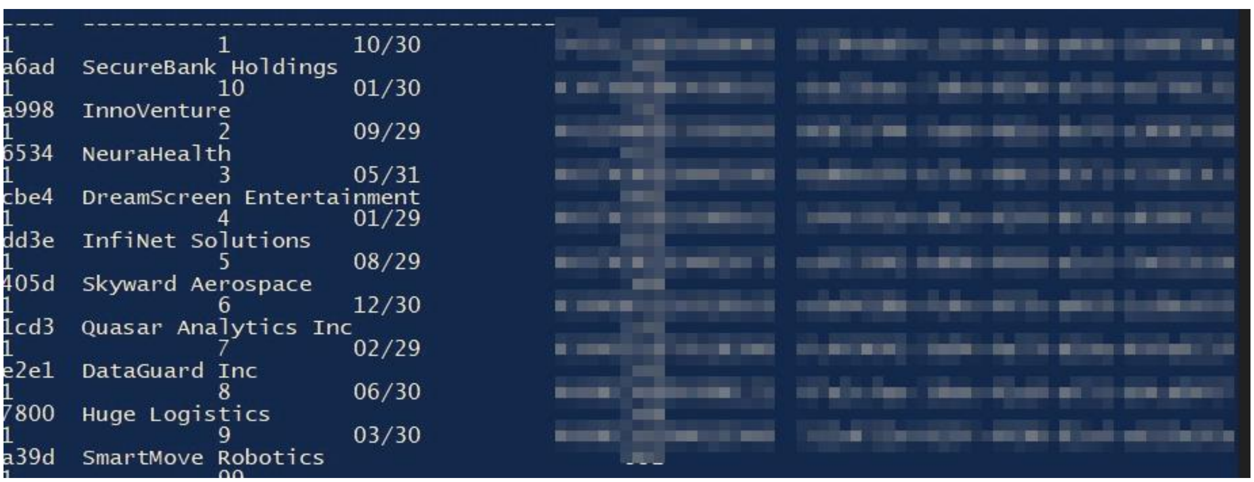
Remediation: Permanently disable old versions of programs that are no longer in use.

Finding INT-002: Sensitive Data Stored in Plain Text (Critical)

Description:	Storing data in plain text leaves it unprotected, allowing an attacker to access sensitive information if compromised.
Risk:	Likelihood: High – Data stored in plain text can be easily exfiltrated by an attacker if an attacker is able to access the data. Impact: Critical – This can result in direct financial loss, reputational damage, and regulatory fines.
System:	Example.onmicrosoft.com
Tools Used:	Mg-graph Az Cli
References:	Top 5 PCI DSS Encryption Requirements - Sprinto

Evidence:

Figure 2.1 Shows a database of credit card numbers stored in plain text.



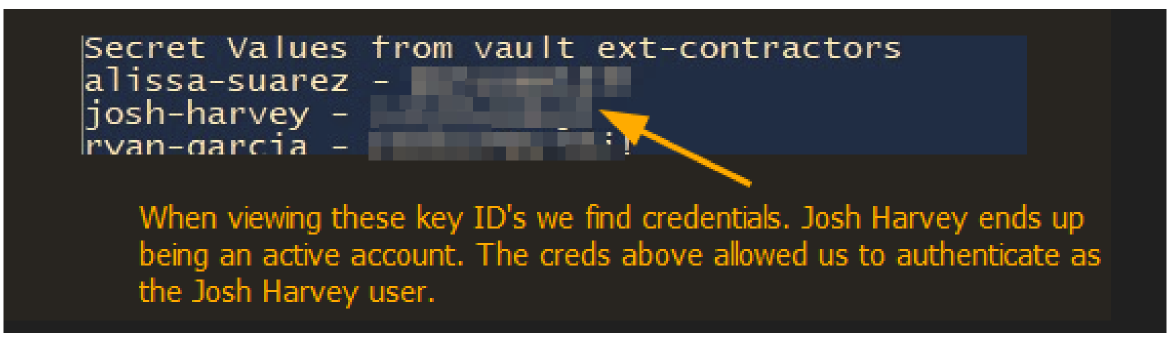
Remediation: Encrypt data at rest and enforce encryption policies for cloud storage.

Finding INT-003: Dormant Accounts on the Cloud ([High]())

Description:	Dormant accounts on the cloud can have credentials, permissions, and access to cloud resources that can be exploited by attackers.
Risk:	Likelihood: High – If dormant accounts are not removed, they could be exploited by attackers to gain unauthorized access. Impact: High – This can result in data exfiltration, privilege escalation, and lateral movement.
System:	Example.onmicrosoft.com
Tools Used:	Mg-graph Az Cli
References:	[Fix user creation and deletion issues in Microsoft Entra ID - Azure | Microsoft Learn]()

Evidence:

Figure 3.1 Shows secrets to various external contractors.



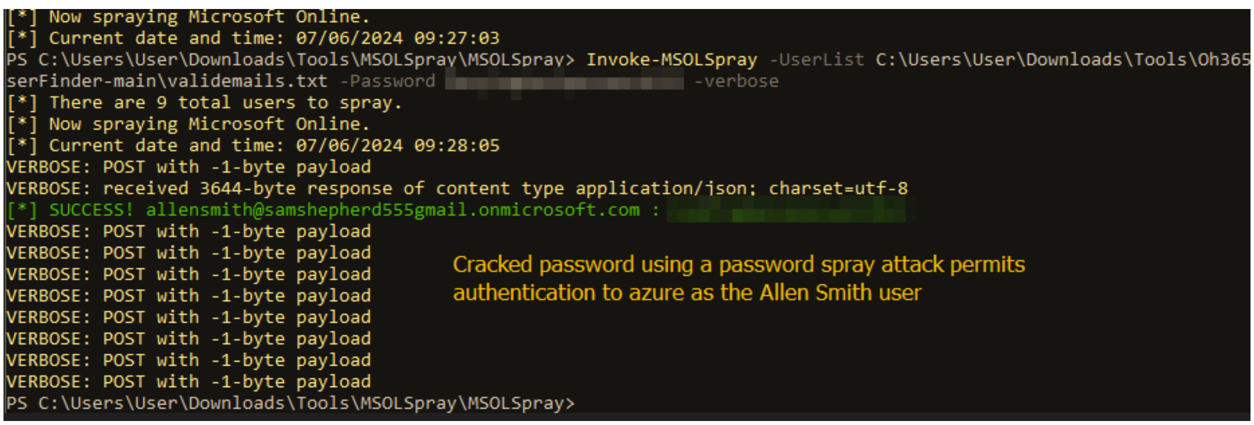
Remediation: Disable / de-activate accounts that are no longer in use.

Finding INT-004: Weak Password Policy ([High](#))

Description:	A weak password policy is when passwords are permitted that lack complexity and length requirements.
Risk:	Likelihood: High – Weak passwords increase the likelihood for password spraying attacks, one of the most common techniques used by attackers to gain initial access. Impact: High – Compromised accounts can permit an attacker initial access and to move laterally in the cloud environment.
System:	Example.onmicrosoft.com
Tools Used:	MSOL Spray
References:	[Create and use strong passwords - Microsoft Support](#)

Evidence:

Figure 4.1:



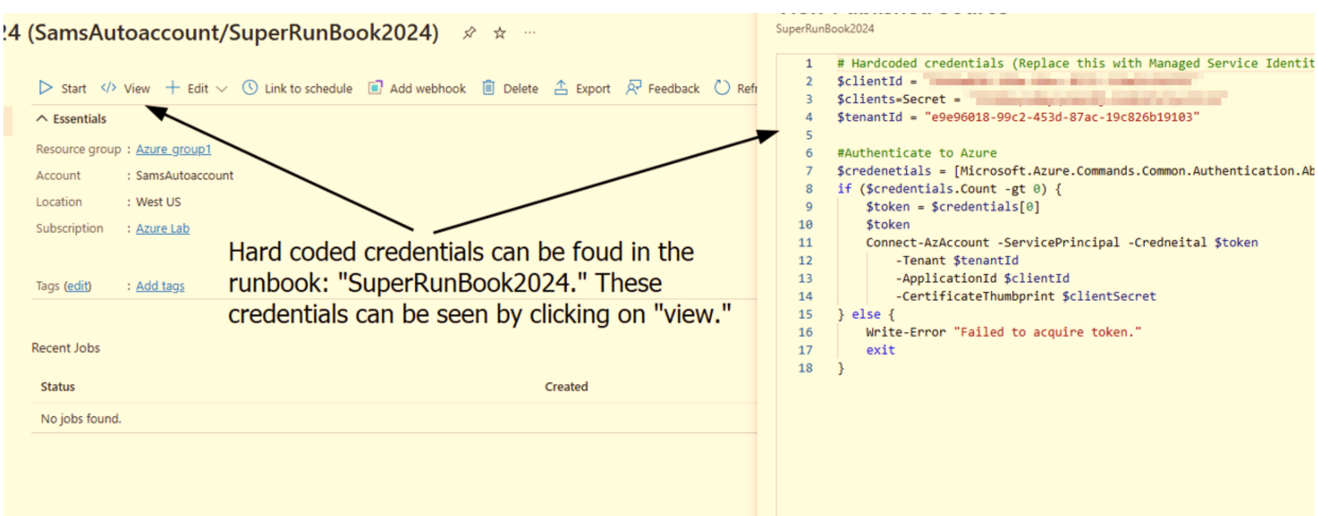
Remediation: Enforce a password policy requiring a minimum of 12 characters, including uppercase letters, lowercase letters, and special characters. Implement multi-factor authentication (MFA) to add an additional layer of security.

Finding INT-005: Hard-Coded Credentials Found in Runbooks (Medium)

Description:	Hard-coded credentials found in run books means that credentials are stored directly in scripts used to automate tasks.
Risk:	Likelihood: High – Hard-coded credentials are static and easy for attackers to extract. Impact: Medium – If these credentials are exposed, it can lead to unauthorized access and lateral movement within the cloud environment.
System:	Example.onmicrosoft.com
Tools Used:	Azure
References:	How to Prevent Hardcoded Passwords? - O360 (offensive360.com)

Evidence:

Figure 5.1: Shows client ID and a client secret available in the run book, SuperRunBook2024.



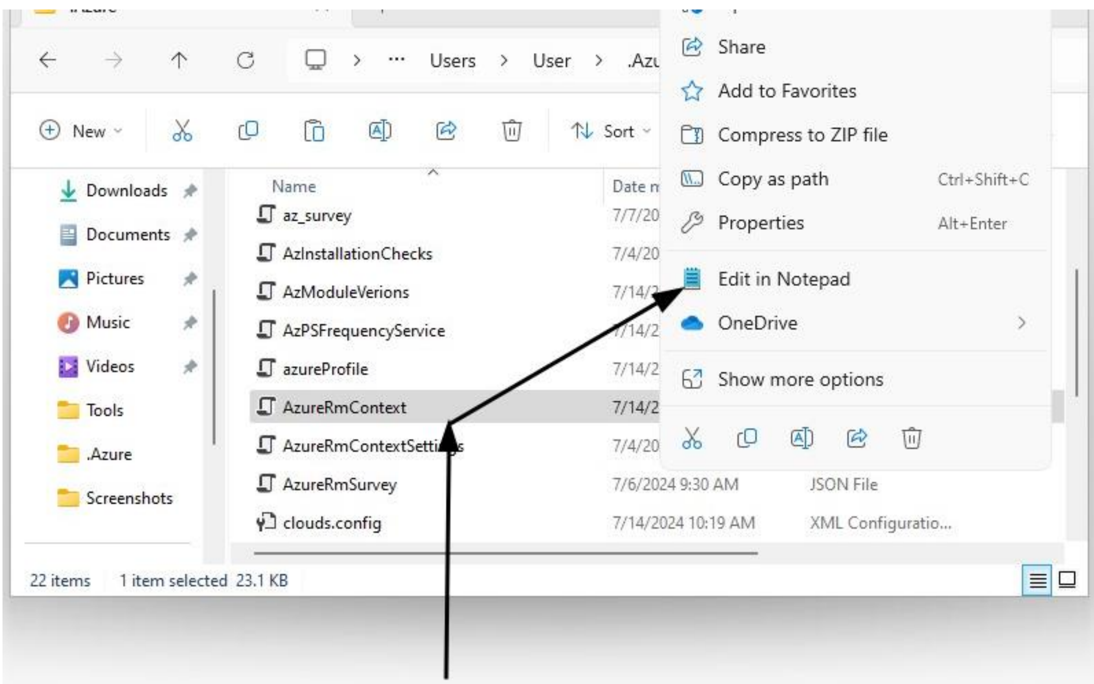
Remediation: Store client secrets in Azure Key Vault.

Finding INT-006: Access Tokens Available in Clear Text ([Medium](#))

Description:	Access tokens found in clear text can permit authentication of a user and further exploitation of other resources in Entra ID.
Risk:	Likelihood: Medium – The risk of token exploitation depends on whether users properly disconnect from Azure PowerShell sessions. Impact: Medium – The impact depends on the level of access granted by the compromised token.
System:	Example.onmicrosoft.com
Tools Used:	Az PowerShell
References:	security - Securely storing an access token - Stack Overflow

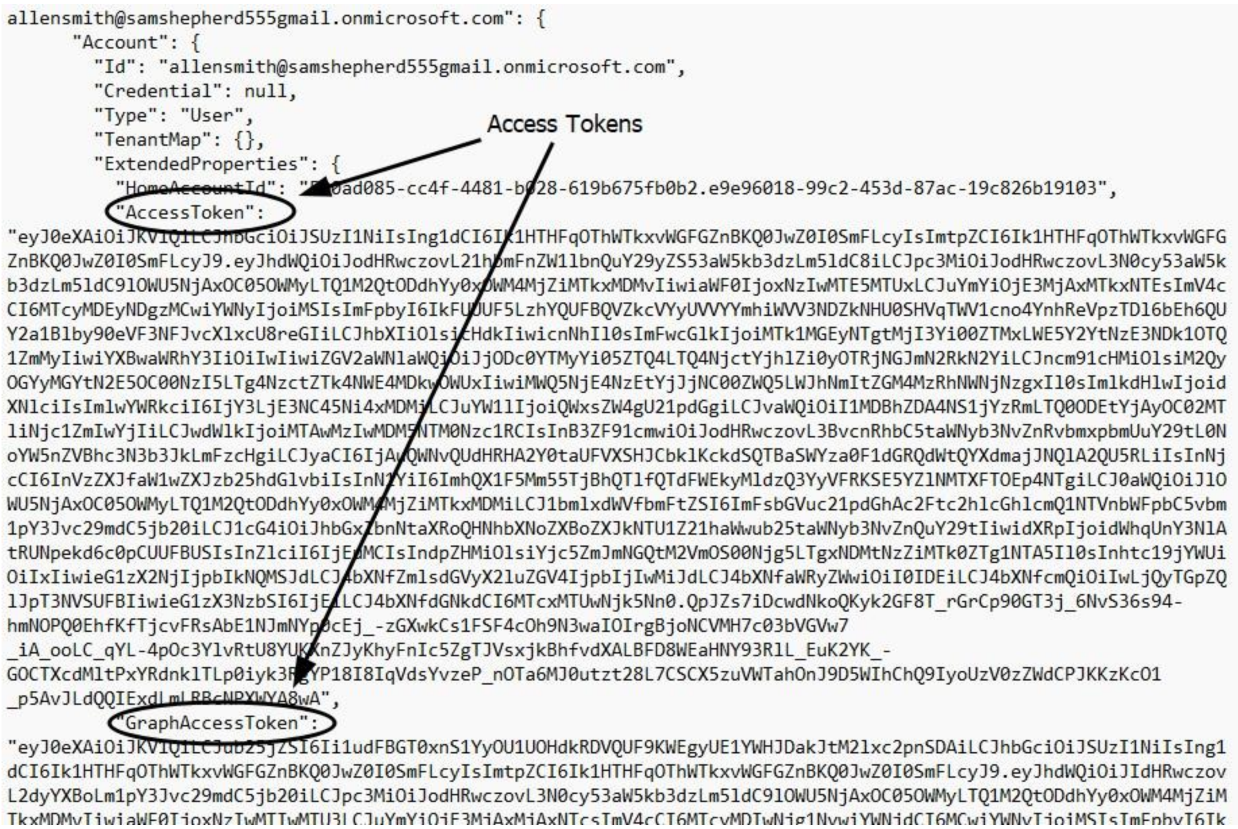
Evidence:

Figure 6.1:



The AzureRmContext JSON file in the .Azure folder contains access tokens. We can view these access tokens when right clicking on the file and then clicking "Edit in Notepad."

__Figure 6.2:__




```
allensmith@samshepherd555gmail.onmicrosoft.com": {
      "Account": {
        "Id": "allensmith@samshepherd555gmail.onmicrosoft.com",
        "Credential": null,
        "Type": "User",
        "TenantMap": {},
        "ExtendedProperties": {
          "HomeAccountId": "f00ad085-cc4f-4481-b028-619b675fb0b2.e9e96018-99c2-453d-87ac-19c826b19103",
        "AccessToken":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1HTHFqOThWTkxvWGFGZnBKQ0JwZ0I0SmFLcyIsImtpZCI6Ik1HTHFqOThWTkxvWGFG
ZnBKQ0JwZ0I0SmFLcyJ9.eyJhdWQiOiJodHRwczovL21hbmFnZW1lbnQuY29yZS53aW5kb3dzLm5ldC8iLCJpc3MiOiJodHRwczovL3N0cy53aW5k
b3dzLm5ldC9lOWU5NjAxOC05OWMyLTQ1M2QtODdhYy0xOWM4MjZiMTkxMDMvIiwiaWF0IjoxNzIwMTE5MTUxLCJuYmYiOjE3MjAxMTkxNTEsImV4c
CI6MTcyMDEyNDgzMCwiYWNyIjoiMSIsImFpbyI6IkFUUUF5LzhYQUFBQVZkcVYyUVVYYmhiWVV3NDZkNHU0SHVqTWV1cno4YnhReVpzTDl6bEh6QU
Y2a1Blby90eVF3NFJvcXlxcU8reGIiLCJhbXIiOlsicHdkIiwicnNhIl0sImFwcGlkIjoiMTk1MGEyNTgtMjI3Yi00ZTMxLWE5Y2YtNzE3NDk1OTQ
1ZmMyIiwiYXBwaWRhY3IiOiIwIiwiZGV2aWNlaWQiOiJjODc0YTMyYi05ZTQ4LTQ4NjctYjhlZi0yOTRjNGJmN2RkN2YiLCJncm91cHMiOlsiM2Qy
OGYyMGYtN2E5OC00NzI5LTg4NzctZTk4NWE4MDkwOWUxIiwiMWQ5NjE4NzEtYjJjNC00ZWQ5LWJhNmItZGM4MzRhNWNjNzgxIl0sImlkdHlwIjoid
XNlciIsImlwYWRkciI6IjY3LjE3NC45Ni4xMDMiLCJuYW1lIjoiQWxsZW4gU21pdGgiLCJvaWQiOiI1MDBhZDA4NS1jYzRmLTQ0ODEtYjAyOC02MT
liNjc1ZmIwYjIiLCJwdWlkIjoiMTAwMzIwMDM5NTM0Nzc1RCIsInB3ZF91cmwiOiJodHRwczovL3BvcnRhbC5taWNyb3NvZnRvbmxpbmUuY29tL0N
oYW5nZVBhc3N3b3JkLmFzcHgiLCJyaCI6IjAuQWNvQUdHRHA2Y0taUFVXSHJCbklKckdSQTBaSWYza0F1dGRQdWtQYXdmajJNQlA2QU5RLiIsInNj
cCI6InVzZXJfaW1wZXJzb25hdGlvbiIsInN1YiI6ImhQX1F5Mm55TjBhQTlfQTdFWEkyMldzQ3YyVFRKSE5YZlNMTXFTOEp4NTgiLCJ0aWQiOiJlO
WU5NjAxOC05OWMyLTQ1M2QtODdhYy0xOWM4MjZiMTkxMDMiLCJ1bmlxdWVfbmFtZSI6ImFsbGVuc21pdGhAc2Ftc2hlcGhlcmQ1NTVnbWFpbC5vbm
1pY3Jvc29mdC5jb20iLCJ1cG4iOiJhbGxlbnNtaXRoQHNhbXNoZXBoZXJkNTU1Z21haWwub25taWNyb3NvZnQuY29tIiwidXRpIjoidWhqUnY3NlA
tRUNpekd6c0pCUUFBUSIsInZlciI6IjEuMCIsIndpZHMiOlsiYjc5ZmJmNGQtM2VmOS00Njg5LTgxNDMtNzZiMTk0ZTg1NTA5Il0sInhtc19jYWUi
OiIxIiwieG1zX2NjIjpbIkNQMSJdLCJ4bXNfZmlsdGVyX2luZGV4IjpbIjIwMiJdLCJ4bXNfaWRyZWwiOiI0IDEiLCJ4bXNfcmQiOiIwLjQyTGpZQ
lJpT3NVSUFBIiwieG1zX3NzbSI6IjEiLCJ4bXNfdGNkdCI6MTcxMTUwNjk5Nn0.QpJZs7iDcwdNkoQKyk2GF8T_rGrCp90GT3j_6NvS36s94-
hmNOPQ0EhfKfTjcvFRsAbE1NJmNYp0cEj_-zGXwkCs1FSF4cOh9N3waIOIrgBjoNCVMH7c03bVGVw7
_iA_ooLC_qYL-4pOc3YlvRtU8YUKXnZJyKhyFnIc5ZgTJVsxjkBhfvdXALBFD8WEaHNY93RlL_EuK2YK_-
GOCTXcdMltPxYRdnklTLp0iyk3RgYP18I8IqVdsYvzeP_nOTa6MJ0utzt28L7CSCX5zuVWTahOnJ9D5WIhChQ9IyoUzV0zZWdCPJKKzKcO1
_p5AvJLdQQIExdLmLRBcNPXWYA8wA",
        "GraphAccessToken":
"eyJ0eXAiOiJKV1QiLCJub25jZSI6Ii1udFBGT0xnS1YyOU1UOHdkRDVQUF9KWEgyUE1YWHJDakJtM2lxc2pnSDAiLCJhbGciOiJSUzI1NiIsIng1
dCI6Ik1HTHFqOThWTkxvWGFGZnBKQ0JwZ0I0SmFLcyIsImtpZCI6Ik1HTHFqOThWTkxvWGFGZnBKQ0JwZ0I0SmFLcyJ9.eyJhdWQiOiJIdHRwczov
L2dyYXBoLm1pY3Jvc29mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9lOWU5NjAxOC05OWMyLTQ1M2QtODdhYy0xOWM4MjZiM
TkxMDMvIiwiaWF0IjoxNzIwMTIwMTU3LCJuYmYiOjE3MjAxMjAxNTcsImV4cCI6MTcyMDIwNjg1NywiYWNjdCI6MCwiYWNyIjoiMSIsImFpbyI6Ik
```


__Remediation:__ These JSON files only exist if you have an active session. When done using Az PowerShell, use the command: Disconnect-AzAccount to disconnect. Never use the command: Save-AzContext, as this will save the session and thus the tokens in the .azure folder.