Never Trust, Always Verify: Analysis of Zero Trust Best Practices for Conditional Access

Author: Glenn Andal

Abstract

The rise in advanced persistent threats (APTs) requires more robust security measures to protect an organization's sensitive data. This study examines the effectiveness of Microsoft Entra's Conditional Access policies in thwarting adversarial bypass attempts, particularly in light of vulnerabilities exposed by recent high-profile breaches, such as those involving the Lapsus$ hacking group. These incidents have highlighted the need for more resilient security frameworks, especially concerning traditional multifactor authentication (MFA) systems. Through comprehensive testing, this research evaluates the strengths and limitations of various Conditional Access policies in addressing MFA attacks. It identifies potential enhancements to improve their robustness within a Zero Trust framework. The findings provide practical insights into the current capabilities of Microsoft Entra Conditional Access and offer recommendations for organizations seeking to strengthen their identity and access management (IAM) strategies against evolving cyber threats.

**1. Introduction**

In today’s cybersecurity landscape, defending against advanced persistent threats (APTs) is more crucial than ever. Conditional access is a critical element of zero trust architectures that help protect access to data by enforcing access policies based on the conditions of the request. Microsoft Entra Conditional Access is a cloud-native solution for Azure environments, providing enhanced security to strengthen an organization’s cloud security posture.

Recent analyses by MITRE have shed light on the threat of advanced persistent threats. Their articles, such as "Advanced Cyber Threats Impact Even the Most Prepared" (MITRE, 2023a), "Technical Deep Dive: Understanding the Anatomy of a Cyber Intrusion" (MITRE, 2023b), and "Infiltrating Defenses: Abusing VMware in MITRE's Cyber Intrusion" (MITRE, 2023c), reveal the sophisticated methods used to breach welldefended environments protected with Multifactor Authentication (MFA).

Microsoft experienced a breach that exploited MFA fatigue attacks. In this attack, the hacker group Lapsus$ used stolen credentials to bombard users with MFA push notifications, hoping the users would eventually approve one out of frustration or confusion. This social engineering attack is known as MFA bombing or push notification spam, which preys on people's tendency to seek relief from persistent notifications. Once the user approves the malicious request, the attacker gains access to the account (BleepingComputer, 2023; SecurityWeek, 2022).

Conditional access can play a significant role in preventing such attacks by ensuring that only verified users access critical resources under secure conditions, taking into account a variety of signals that may include user location, device, time, and behavior. Microsoft Entra Conditional Access policies enforce continuous authentication and integrate remote attestation, enhancing security measures. These policies can dynamically adjust access controls based on real-time risk assessments, ensuring that only trusted devices and users can access sensitive resources.

However, setting up and configuring these policies is not straightforward and is susceptible to misconfiguration or needs to be appropriately enforced (Colley, B, 2022). They are not enabled by default, and cloud administrators must be well-versed in designing, testing, and deploying accurate and secure conditional access policies. Microsoft Azure and the Center for Internet Security (CIS) offer conditional access policy templates and guidance to help organizations improve their cloud security posture. While these templates are valuable, there are areas to increase security and enable automation of these policy configurations to tackle the complex tactics, techniques, and procedures (TTPs) of APTs.

The research will evaluate the effectiveness of Microsoft Entra Conditional Access policies against various attack scenarios. The aim is to uncover any gaps in the template policies and suggest enhancements for a more robust Zero Trust security framework. This study contributes to the broader discussion on IAM best practices and the continuous improvement needed to address cyber threats.

**2. Research Method**

The primary research question for this study is: What is the effectiveness of Microsoft Entra's (formerly Azure Active Directory) Conditional Access in detecting and preventing adversarial bypass attempts under various attack scenarios? The hypothesis assumes that Microsoft’s Conditional Access effectiveness can be validated through a comprehensive approach involving simulated attack scenarios, policy variation testing, and iterative refinement.

**2.1. Testing Environment**

A controlled testing environment was established and configured to assess Microsoft Entra Conditional Access's effectiveness accurately. Figure 1 below provides an overview of the test environment.

The test environment is configured mostly using Microsoft Entra ID. Three test users are created to be placed in three different roles: cloud admin, developer, and guest user. To enable Entra’s conditional access feature, Microsoft Entra ID P2 licenses were required and procured (Microsoft, 2024a). Entra’s conditional access policies were manually created and exported in code templates to enable the ability to re-import and delete to change conditional access policy sets quickly.

Entra's diagnostics settings were enabled to monitor and validate the success and failures of tests, with sign-in logs sent to an Azure Log Analytics workspace to be stored and visualized using Azure’s built-in dashboards. An Azure subscription was provisioned, along with a resource group to host the log analytics workspace.

The test workstation was configured to enable the quick provision of conditional access policy sets and the testing of conditional access policies with appropriate tools. The following tools were required:

• PowerShell

• PowerShell scripts to export, import, and delete conditional access policy code

templates

• Firefox web browser

• Visual Studio Code

• Windscribe VPN client

• Wireshark

**2.2. Testing Variables**

The tests manipulated and measured specific variables to determine the efficacy of conditional access policies. The manipulated variables are the type of conditional access policy sets applied. Table 1 provides an overview of the conditional access policy sets provided and recommended by Microsoft and CIS. For this research, analysis will be conducted on Microsoft’s Security Default setting and Microsoft’s Azure Zero Trust recommendation. In addition, CIS Microsoft Azure Foundation Benchmark v2.0.0 has a set of recommended conditional access policies that will be tested (Center for Internet Security, 2022).

The measured variable is the effectiveness of these policies and policy sets in mitigating or preventing simulated adversarial attacks against a Microsoft Azure environment.

**2.3. Policy Configuration**

Configuring conditional access policies in Azure requires the using a designated cloud admin account. The detailed steps were:

1. Log into the Azure portal with the cloud admin account.

2. Navigate to the Microsoft Entra Conditional Access section.

3. Apply predefined policies, such as Microsoft-recommended baseline policies,

Security Defaults, and CIS Benchmark-recommended policies.

4. Export and document each conditional access policy configuration using

infrastructure-as-code (e.g., ARM templates, Terraform scripts) (Microsoft, 2023a;

Tajran, 2023).

5. Leverage PowerShell scripts to quickly export, delete, and import conditional access

policies, enhancing consistency and reproducibility in testing (HashiCorp, 2021;

Tajran, 2023).

**2.4. Evaluating Conditional Access Policies**

The study conducted manual and automated tests to emulate conditional access bypass attempts. Manual testing involves accessing resources under various user roles and conditions expected to trigger conditional access policies. Automated testing involves developing automated scripts to simulate adversarial attack scenarios, such as brute force, phishing, and token theft (O’Reilly, 2019). The system’s response from both manual and automated tests will be documented.

Each conditional access policy will be tested to confirm that each policy functions as intended for accepting and denying logins based on the condition. Once validated, the conditional policies will be deployed to follow CIS Azure Benchmark and Microsoft’s Zero Trust recommended policies. Each policy set will be tested to ensure that the combination of policies functions as intended and enables the exploration of methods to bypass and circumvent the policy set.

**2.4.1. Require MFA for Admins**

**Description:** Only administrators must use Multifactor Authentication (MFA) to access the Azure portal.

**Procedure Steps:**

1. Use two user accounts in Entra ID: one with admin privileges and one without admin

privileges.

2. Attempt to log in as Admin without MFA.

• Log in to the Azure portal with an admin account and complete the MFA process.

Verify that access is denied.

3. Attempt to log in as Admin with MFA:

• Log in to the Azure portal with an admin account and complete the MFA process.

Verify that access is granted.

4. Attempt Non-Admin Login:

• Log in with a non-admin account without MFA. Verify that access is denied.

5. Confirm and document login attempts on Microsoft Entra’s sign-in logs.

**2.4.2. Require MFA for All Users**

**Description:** All users must use MFA to access the Azure portal.

**Procedure Steps:**

1. Use User Account in Entra ID.

2. Attempt Login without MFA:

• Attempt to log in to the Azure portal without completing MFA. Verify that access

is denied.

3. Attempt Login with MFA:

• Log in to the Azure portal and complete the MFA process. Verify that access is

granted.

4. Confirm and document login attempts on Microsoft Entra’s sign-in logs.

**2.4.3. Require MFA for Guest Access**

**Description:** Block access to the Azure portal for all guest users. **Procedure Steps:**

1. Use a guest user account.

2. Attempt to log in with a guest user account.

• Log in to the Azure portal with the guest account. Verify that access is denied.

3. Confirm and document login attempts on Microsoft Entra’s sign-in logs.

**2.4.4. Block Access for Unknown or Unsupported Device Platforms Description:** Block access to the Azure portal from unknown or unsupported device platforms.

**Procedure Steps:**

1. Identify supported devices/platforms on Microsoft Entra:

• Authorized platforms are Windows, macOS

• Unauthorized platforms are Android, iOS, Windows Phone, Linux

2. Simulate Access from Supported Platforms:

• Use Firefox browser with a user agent switcher extension to simulate a MacOS

user agent. Attempt to log in to the Azure portal and verify that access is granted.

3. Simulate Access from Unsupported Platforms:

• Change the user agent to an Ubuntu user agent. Attempt to log in to the Azure

portal and verify that access is denied.

4. Confirm and document login attempts on Microsoft Entra’s sign-in logs.

**2.4.5. Ensure that an Exclusionary Geographic Access Policy is Considered**

**Description:** Only allow access to the Azure portal from approved geographic locations. **Procedure Steps:**

1. Define trusted locations and IP addresses/ranges in Microsoft Entra ID and create a

conditional access policy with the trusted location as part of the conditions.

2. Simulate access from a trusted location (United States).

• Attempt to log in to the Azure portal and verify that access is granted.

3. Simulate access from a trusted IP address/range.

• Attempt to log in to the Azure portal and verify that access is granted.

4. Simulate access from a non-trusted location and IP address.

• Use a VPN client, Windscribe, to VPN to a server to simulate access from the

United Kingdom (UK). Attempt to log in to the Azure portal and verify that

access is denied.

5. Confirm and document login attempts on Microsoft Entra’s sign-in logs.

**2.5. Data Collection and Analysis**

The responses from the simulated attack scenarios are recorded and crossvalidated with conditional access logs in Microsoft Entra ID. This high-level procedure is as follows:

1. Collect logs from the Azure portal, focused on conditional access and sign-in

logs.

2. Analyze the logs to determine the effectiveness of each policy in preventing

unauthorized access and policy violations.

3. Compare results against the baseline to identify improvements or gaps in the

conditional access policies (Splunk, 2022)

**2.6. Additional Considerations**

The study will be performed using iterative rounds of testing, where the refinement of conditional access policies will be based on initial findings and observed gaps in coverage. The test procedures, scripts, and infrastructure-as-code configurations will be made available to the security community to enable repeatability and further study. In addition to the test procedures, scripts, and IaC, comprehensive documentation of all configurations, test scenarios, and results will be maintained to ensure transparency (National Institute of Standards and Technology, 2020).

**3. Findings and Discussion**

This section presents a detailed analysis of the efficacy of various Microsoft Entra Conditional Access policies when subjected to adversarial attack simulations. The testing was conducted using a controlled environment where different conditional access policies were configured, manipulated, and evaluated for their ability to enforce security measures and prevent unauthorized access.

The findings show the strengths and vulnerabilities of these policies, specifically in scenarios where sophisticated tactics, techniques, and procedures (TTP) were employed to circumvent security controls. This section assesses the current state of Microsoft's Conditional Access solutions and provides insights into potential improvements that could enhance their resilience against emerging threats within a Zero Trust framework.

**3.1. Entra Conditional Access Policies Findings**

The table below provides an overview of the conditional access policies that were individually tested. The test involves a conditional access policy configured correctly, where the policy will block or accept login as expected based on the condition. The test also includes using a tool to circumvent policy, allowing the adversary to bypass the policy by manipulating values or performing a social engineering attack.

**3.1.1. Circumventing MFA conditional access policies**

Social engineering was necessary to circumvent the conditional access policies requiring MFA for a particular user or to manage Azure, where the adversary had to communicate via phone, email, or text to convince the victim to provide authentication details from the Microsoft Authenticator App. In the interest of time, the assumption is that the adversary can access the victim's username and password from a prior attack, such as spear phishing. With the victim's credentials, the adversary attempts to log in as Peter Parker, a guest user for this Azure environment. The Microsoft Authenticator App prompts the victim to enter the number using the login prompt.