Steal Application Access Token (version 1.1)

**Cloud Service Label: IaaS, PaaS**

Description

Adversaries may use application access tokens to bypass the typical authentication process and access restricted accounts, information, or services on remote systems. These tokens are typically stolen from compromised users and used in lieu of login credentials. Application credentials may also be stolen from publicly available cloud software repositories where developmental code may still have credentials hardcoded credentials for testing. This is actually quite common and automated tools exist that will scan publicly available repositories for such tokens.

Application access tokens are used to make authorized API requests on behalf of a user and are commonly used as a way to access resources in cloud-based applications and software-as-a-service (SaaS). OAuth is one commonly implemented framework that issues tokens to users for access to systems. These frameworks are used collaboratively to verify the user and determine what actions the user is allowed to perform. Once identity is established, the token allows actions to be authorized, without passing the actual credentials of the user. Therefore, compromise of the token can grant the adversary access to resources of other sites through a malicious application. For example, with a cloud-based email service once an OAuth access token is granted to a malicious application, it can potentially gain long-term access to features of the user account if a "refresh" token enabling background access is awarded. With an OAuth access token an adversary can use the user-granted REST API to perform functions such as email searching and contact enumeration.

Compromised access tokens may be used as an initial step in compromising other services. For example, if a token grants access to a victim’s primary email, the adversary may be able to extend access to all other services which the target subscribes by triggering forgotten password routines.

Direct API access through a token negates the effectiveness of a second authentication factor and may be immune to intuitive countermeasures like changing passwords. Access abuse over an API channel can be difficult to detect even from the service provider end, as the access can still align well with a legitimate workflow.

Examples

|  |  |
| --- | --- |
| **Name** | **Description** |
| Black Direct | Published Oauth2 vulnerability in certain first party Microsoft web sites that could enable the compromise of Azure credentials if exploited. |
| NetSPI | Documents Numerous opportunities to collect JWT tokens from within Azure applications. |
| APT28 | APT28 has used several malicious applications that abused OAuth access tokens to gain access to target email accounts, including Gmail and Yahoo Mail. |
| Azure AD Pass through Authentication (PTA) | Un-hashed credentials are passed to the connector to validate against Active Directory if Azure AD is configured with PTA. Adam Chester, aka XPN, from MDSec created a PoC leveraging DLL injection into the Azure AD Sync process, specifically the Win32 API *LogonUserW,* to parse and store credentials. |

Mitigations

|  |  |
| --- | --- |
| **Mitigation** | **Description** |
| Logging | Administrators can set up a variety of logs and leverage audit tools to monitor actions that can be conducted as a result of OAuth 2.0 access. For instance, audit reports enable admins to identify privilege escalation actions such as role creations or policy modifications, which could be actions performed after initial access. In Azure you can look at all authentications to Azure AD through the “sign in” blade under the Active Directory heading to determine if access has been obtained. |
| Security Services | Azure now offers security services focused specifically on identifying inappropriate use of credentials (PIM). These services include machine learning and preapproved policies that can be used to limit access to suspicious access requests. |
| Update Corporate Policies | Update corporate policies to restrict what types of third-party applications may be added to any online service or tool that is linked to the company's information, accounts or network (example: Google, Microsoft, Dropbox, Basecamp, GitHub). However, rather than providing high-level guidance on this, be extremely specific—include a list of pre-approved applications and deny all others not on the list. Administrators may also block end-user consent through administrative portals, such as by using policies within Azure, disabling users from authorizing third-party apps through OAuth and forcing administrative consent. |

Detection

Monitor access token activity for abnormal use and permissions granted to unusual or suspicious applications. Administrators can set up a variety of logs and leverage audit tools to monitor actions that can be conducted as a result of OAuth 2.0 access. For instance, audit reports enable admins to identify privilege escalation actions such as role creations or policy modifications, which could be actions performed after initial access.

|  |  |
| --- | --- |
| **Detection of activities after exploitation** | **Description** |
| Create Log Metric Filters and Alarms for AWS | To create a metric filter and alarm:   1. Create a metric filter that checks for IAM policy changes and the *<cloudtrail\_log\_group\_name>* 2. Create an SNS topic 3. Create an SNS subscription to the above topic 4. Create an alarm associated with the filter and SNS topic created in steps 1 and 2 respectively |
| Monitor Activity in AWS Account | Various services in AWS offer logging features that allow for detection capabilities. These include CloudFront, CloudTrail, CloudWatch, Config, and S3. |
| Monitor for Suspicious Activity in Azure | Azure AD can generate anomaly reports than can be run on a daily basis. Azure AD Identity Protection show current risks in its dashboard and provides daily email summary notifications. Policies can also be configured to alert to specific issues. |
| Create Log Metric Filters and Alarms for CloudTrail | To create a metric filter and alarm:   1. Create a filter that checks for CloudTrail changes and the specific *<cloudtrail\_log\_group\_name>* 2. Create an SNS topic that the alarm will notify 3. Create an SNS subscription to the above topic 4. Create an alarm associated with the filter from step 1 and SNS topic in step 2 |
| Create Activity Log Alerts in Azure | To create log activity alerts for deletion in the Azure Console:   1. Navigate to *Monitor’ / ‘Alerts* 2. Select *Manage alert rules* 3. Click on the Alert *Name* where Condition contains *operationName equals Microsoft.Network/networkSecurityGroups/securityRules/delete* 4. Hover a mouse over *Condition* to ensure it is set to *Whenever the Administrative Activity Log “Delete Security Rule (networkSecurityGroups/securityRules)” has “any” level with “any” status and event is initiated by “any*” |
| Create, View, and Manage Activity Alerts in Azure Monitor | To create a log alert in the Azure portal:   1. Select **Monitor -> Alerts** 2. Select **New alert rule** of the **Alerts** window 3. Provide information in **Define alert condition** 4. Provide details in **Define alert details** 5. Specify action group for new alert rule under **Action group**, or create a new action group with + **New group** 6. Select **Yes** for the **Enable rule upon creation** option 7. Select **Create alert rule**   To view and manage alerts:   1. Select **Monitor -> Alerts -> Manage alert rules** 2. Select the rule you want to modify and double-click to edit the rule options 3. Click **Save** |
| Enable CloudTrail across all regions in AWS | To enable CloudTrail across all regions:   1. Sign into the AWS Management Console and open the CloudTrail console 2. Click on *Trails* 3. Set necessary Trails to All option in the I column 4. Click on a trail via the link *Name* column 5. Set *Logging* to *ON* 6. Set *Apply trail to all regions* to *Yes* |
| Configure log profile to capture activity logs for all regions in Azure | To set up activity logs for all regions:   1. Navigate to Azure console 2. Go to *Activity log* 3. Select *Export* 4. Select *Subscription* 5. Check *Select all* in *Regions* 6. Select *Save* |

References

1. https://auth0.com/blog/why-should-use-accesstokens-to-secure-an-api/#:~:text=It%20enables%20you%20to%20authorize,verifying%20who%20the%20user%20is). Accessed September 12, 2019
2. <https://docs.microsoft.com/en-us/azure/security/fundamentals/steps-secure-> identity. Accessed Feb 2, 2020
3. <https://www.cyberark.com/threat-research-blog/blackdirect-microsoft-azure-account-takeover/>. Accessed Feb 28,2020
4. <https://blog.netspi.com/gathering-bearer-tokens-azure/>. Accessed June 8, 2020
5. <https://blog.xpnsec.com/azuread-connect-for-redteam/>. Accessed July 20, 2020
6. <https://www.varonis.com/blog/azure-skeleton-key/>. Accessed July 20, 2020