Use Alternate Authentication Material: Application Access Token, Sub-technique T1550.001 - Enterprise

Archived: 2026-04-05 13:33:16 UTC

Adversaries may use stolen 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 users or services and used in lieu of login credentials.

Application access tokens are used to make authorized API requests on behalf of a user or service and are commonly used to access resources in cloud, container-based applications, and software-as-a-service (SaaS).[1]

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.[2]

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.[3] With an OAuth access token an adversary can use the user-granted REST API to perform functions such as email searching and contact enumeration.[4]

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. In AWS and GCP environments, adversaries can trigger a request for a short-lived access token with the privileges of another user account.[5][6] The adversary can then use this token to request data or perform actions the original account could not. If permissions for this feature are misconfigured – for example, by allowing all users to request a token for a particular account - an adversary may be able to gain initial access to a Cloud Account or escalate their privileges.[7]

Direct API access through a token negates the effectiveness of a second authentication factor and may be immune to intuitive countermeasures like changing passwords. For example, in AWS environments, an adversary who compromises a user’s AWS API credentials may be able to use the `sts:GetFederationToken` API call to create a federated user session, which will have the same permissions as the original user but may persist even if the original user credentials are deactivated.[8] Additionally, 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.

Source: https://attack.mitre.org/techniques/T1550/001