Token tactics: How to prevent, detect, and respond to cloud token theft

By Microsoft Defender Experts, Microsoft Incident Response

Published: 2022-11-16 · Archived: 2026-04-05 21:36:12 UTC

As organizations increase their coverage of multifactor authentication (MFA), threat actors have begun to move to more sophisticated techniques to allow them to compromise corporate resources without needing to satisfy MFA. Recently, the Microsoft Detection and Response Team (DART) has seen an increase in attackers utilizing token theft for this purpose. By compromising and replaying a token issued to an identity that has already completed multifactor authentication, the threat actor satisfies the validation of MFA and access is granted to organizational resources accordingly. This poses to be a concerning tactic for defenders because the expertise needed to compromise a token is very low, is hard to detect, and few organizations have token theft mitigations in their incident response plan.

Why it matters

In the new world of hybrid work, users may be accessing corporate resources from personally owned or unmanaged devices which increases the risk of token theft occurring. These unmanaged devices likely have weaker security controls than those that are managed by organizations, and most importantly, are not visible to corporate IT. Users on these devices may be signed into both personal websites and corporate applications at the same time, allowing attackers to compromise tokens belonging to both.

As far as mitigations go, publicly available open-source tools for exploiting token theft already exist, and commodity credential theft malware has already been adapted to include this technique in their arsenal. Detecting token theft can be difficult without the proper safeguards and visibility into authentication endpoints. Microsoft DART aims to provide defenders with the knowledge and strategies necessary to mitigate this tactic until permanent solutions become available.

Tokens are at the center of OAuth 2.0 identity platforms, such as Azure Active Directory (Azure AD). To access a resource (for example, a web application protected by Azure AD), a user must present a valid token. To obtain that token, the user must sign into Azure AD using their credentials. At that point, depending on policy, they may be required to complete MFA. The user then presents that token to the web application, which validates the token and allows the user access.

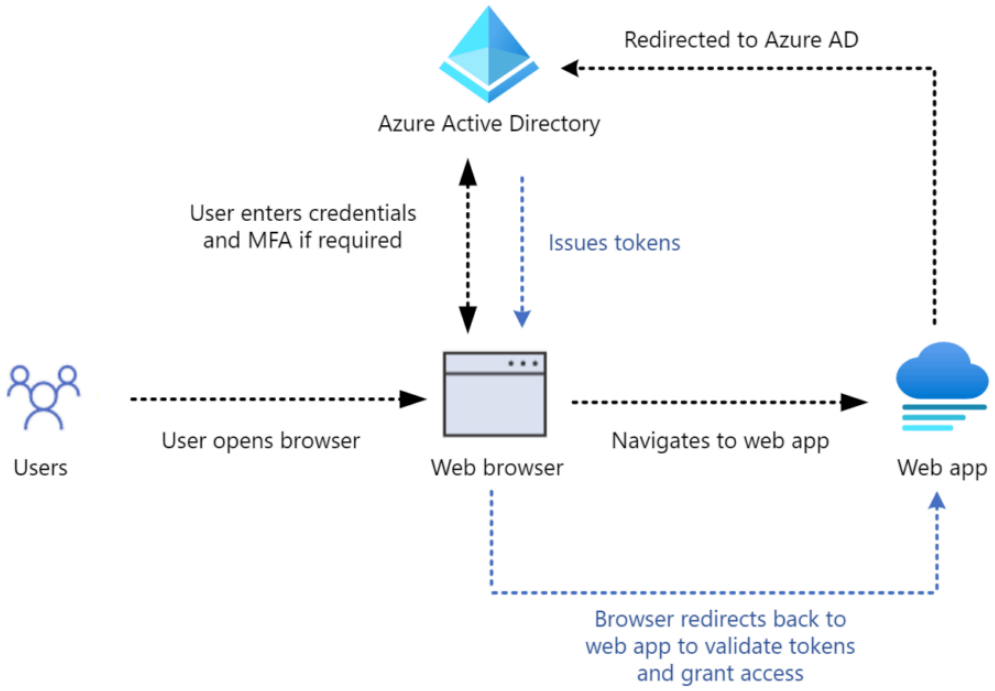

Figure 1. OAuth Token flow chart

When Azure AD issues a token, it contains information (claims) such as the username, source IP address, MFA, and more. It also includes any privilege a user has in Azure AD. If you sign in as a Global Administrator to your Azure AD tenant, then the token will reflect that. Two of the most common token theft techniques DART has observed have been through adversary-in-the-middle (AitM) frameworks or the utilization of commodity malware (which enables a 'pass-the-cookie' scenario).

With traditional credential phishing, the attacker may use the credentials they have compromised to try and sign in to Azure AD. If the security policy requires MFA, the attacker is halted from being able to successfully sign in. Though the users' credentials were compromised in this attack, the threat actor is prevented from accessing organizational resources.

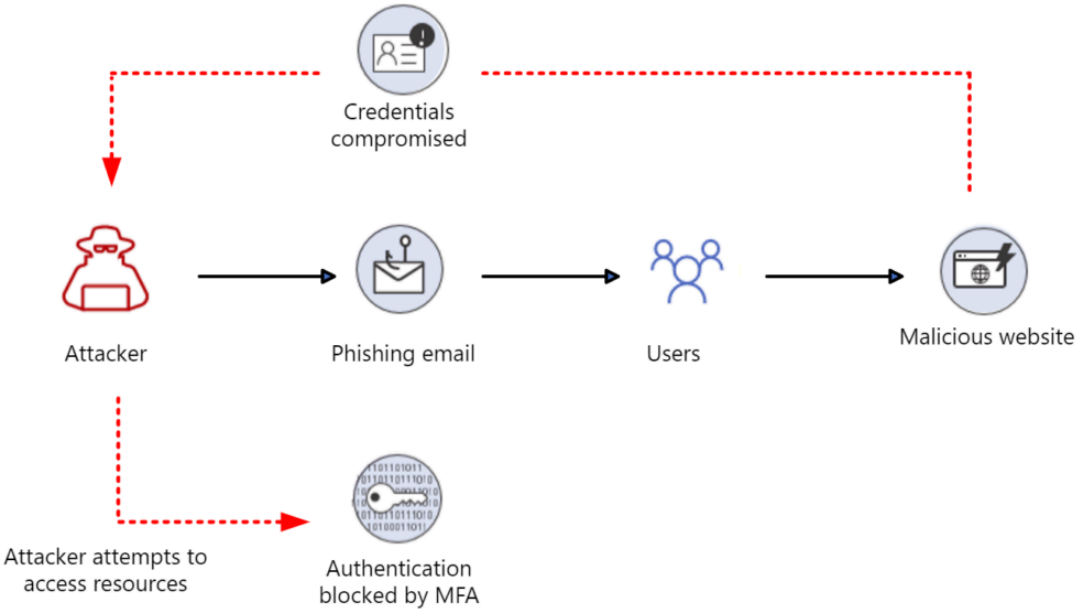

Figure 2. Common credential phishing attack mitigated by MFA

Adversary-in-the-middle (AitM) phishing attack

Attacker methodologies are always evolving, and to that end DART has seen an increase in attackers using AitM techniques to steal tokens instead of passwords. Frameworks like Evilginx2 go far beyond credential phishing, by inserting malicious infrastructure between the user and the legitimate application the user is trying to access. When the user is phished, the malicious infrastructure captures both the credentials of the user, and the token.

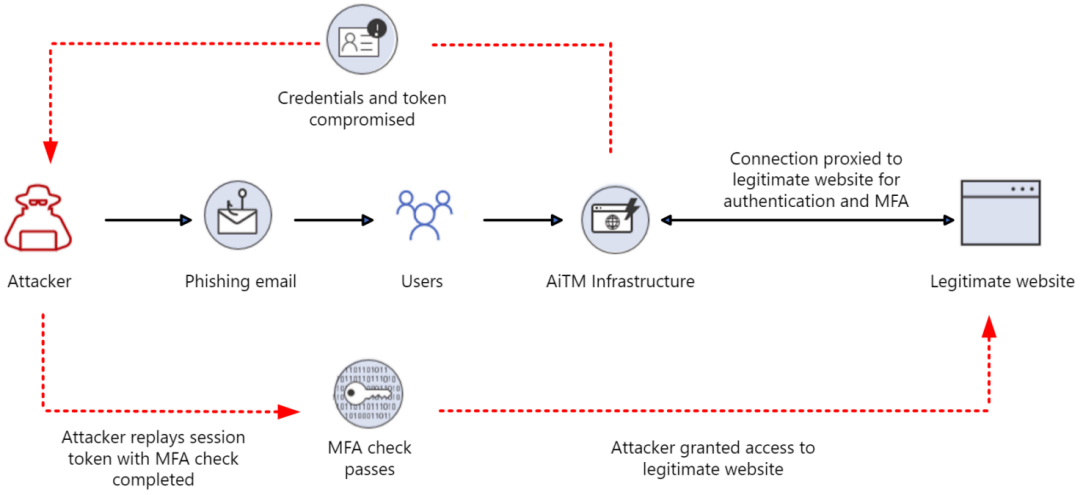

Figure 3. Adversary-in-the-middle (AitM) attack flowchart

If a regular user is phished and their token stolen, the attacker may attempt business email compromise (BEC) for financial gain. If a token with Global Administrator privilege is stolen, then they may attempt to take over the Azure AD tenant entirely, resulting in loss of administrative control and total tenant compromise.

Pass-the-cookie attack

A “pass-the-cookie” attack is a type of attack where an attacker can bypass authentication controls by compromising browser cookies. At a high level, browser cookies allow web applications to store user authentication information. This allows a website to keep you signed in and not constantly prompt for credentials every time you click a new page.

“Pass-the-cookie” is like pass-the-hash or pass-the-ticket attacks in Active Directory. After authentication to Azure AD via a browser, a cookie is created and stored for that session. If an attacker can compromise a device and extract the browser cookies, they could pass that cookie into a separate web browser on another system, bypassing security checkpoints along the way. Users who are accessing corporate resources on personal devices are especially at risk. Personal devices often have weaker security controls than corporate-managed devices and IT staff lack visibility to those devices to determine compromise. They also have additional attack vectors, such as personal email addresses or social media accounts users may access on the same device. Attackers can compromise these systems and steal the authentication cookies associated with both personal accounts and the users’ corporate credentials.

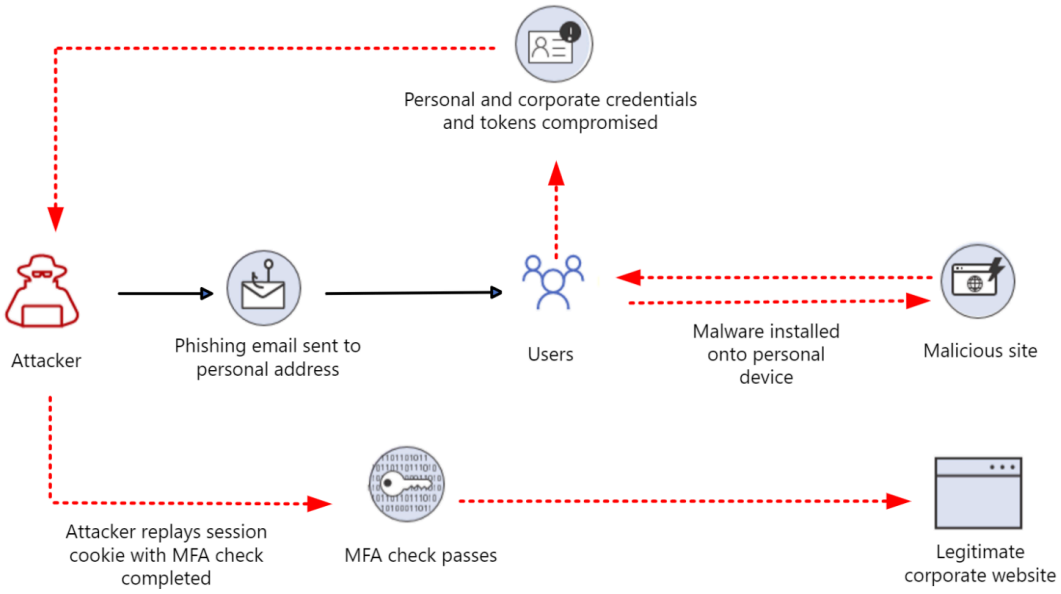

Figure 4. Pass-the-cookie attack flowchart

Commodity credential theft malware like Emotet, Redline, IcedID, and more all have built-in functionality to extract and exfiltrate browser cookies. Additionally, the attacker does not have to know the compromised account password or even the email address for this to work— those details are held within the cookie.

Recommendations

Protect

Organizations can take a significant step toward reducing the risk of token theft by ensuring that they have full visibility of where and how their users are authenticating. To access critical applications like Exchange Online or SharePoint, the device used should be known by the organization. Utilizing compliance tools like Intune in combination with device based conditional access policies can help to keep devices up to date with patches, antivirus definitions, and EDR solutions. Allowing only known devices that adhere to Microsoft’s recommended

security baselines helps mitigate the risk of commodity credential theft malware being able to compromise end user devices.

For those devices that remain unmanaged, consider utilizing session conditional access policies and other compensating controls to reduce the impact of token theft:

- Reducing the lifetime of the session increases the number of times a user is forced to re-authenticate but minimizes the length of time session token is viable.
- Reducing the viable time of a token forces threat actors to increase the frequency of token theft attempts which in turn provides defenders with additional chances at detection.
- Implement Conditional Access App Control in Microsoft Defender for Cloud Apps for users connecting from unmanaged devices.

Protect your users by blocking initial access:

- Plan and implement phishing resistant MFA solutions such as FIDO2 security keys, Windows Hello for Business, or certificate-based authentication for users.
    - While this may not be practical for all users, it should be considered for users of significant privilege like Global Admins or users of high-risk applications.
- Users that hold a high level of privilege in the tenant should have a segregated cloud-only identity for all administrative activities, to reduce the attack surface from on-premises to cloud in the event of on-premises domain compromise and abuse of privilege. These identities should also not have a mailbox attached to them to prevent the likelihood of privileged account compromise via phishing techniques.

We recognize that while it may be recommended for organizations to enforce location, device compliance, and session lifetime controls to all applications it may not always be practical. Decisionmakers should instead focus on deploying these controls to applications and users that have the greatest risk to the organization which may include:

- Highly privileged users like Global Administrators, Service Administrators, Authentication Administrators, and Billing Administrators among others.
- Finance and treasury type applications that are attractive targets for attackers seeking financial gain.
- Human capital management (HCM) applications containing personally identifiable information that may be targeted for exfiltration.
- Control and management plane access to Microsoft 365 Defender, Azure, Office 365 and other cloud app administrative portals.
- Access to Office 365 services (Exchange, SharePoint, and Teams) and productivity-based cloud apps.
- VPN or remote access portals that provide external access to organizational resources.

Detect

When a token is replayed, the sign-in from the threat actor can flag anomalous features and impossible travel alerts. Azure Active Directory Identity Protection and Microsoft Defender for Cloud Apps both alert on these events. Azure AD Identity Protection has a specific detection for anomalous token events. The token anomaly

detection in Azure AD Identity Protection is tuned to incur more noise than other alerts. This helps ensure that genuine token theft events aren't missed.

DART recommends focusing on high severity alerts and focusing on those users who trigger multiple alerts rapidly. Detection rules that map to the MITRE ATT&CK framework can help detect genuine compromise. For example, a risky sign-in followed closely by indicators of persistence techniques, such as mailbox rule creation.

Response and investigation

If a user is confirmed compromised and their token stolen, there are several steps DART recommends evicting the threat actor. Azure AD provides the capability to revoke a refresh token. Once a refresh token is revoked, it's no longer valid. When the associated access token expires, the user will be prompted to re-authenticate. The following graphic outlines the methods by which access is terminated entirely:

	Password-based browser session	Password based token	Non-password-based browser session	Non-password-based browser token	Enterprise application token
Password changed/reset or admin center sign-out	Revoked	Revoked	Not revoked	Not revoked	Not revoked
Revoked via AAD portal, PowerShell, Graph	Revoked	Revoked	Revoked	Revoked	Revoked
⚠	Terminating **all** access to resources, especially in response to a compromised account, requires revoking refresh tokens, not just refreshing the password.				

Figure 5. Refresh token revocation by type

It's crucial to use both the Azure AD portal, Microsoft Graph, or Azure AD PowerShell in addition to resetting the users' passwords to complete the revocation process.

Importantly, revoking refresh tokens via the above methods doesn't invalidate the access token immediately, which can still be valid for up to an hour. This means the threat actor may still have access to a compromised user's account until the access token expires. Azure AD now supports continuous access evaluation for Exchange, SharePoint and Teams, allowing access tokens to be revoked in near real time following a 'critical event'. This helps to significantly reduce the up to one hour delay between refresh token revocation and access token expiry.

Microsoft DART also recommends checking the compromised user's account for other signs of persistence. These can include:

- Mailbox rules – threat actors often create specific mailbox rules to forward or hide email. These can include rules to hide emails in folders that are not often used. For example, a threat actor may forward all emails containing the keyword 'invoice' to the Archive folder to hide them from the user or forward them to an external email address.
- Mailbox forwarding – email forwarding may be configured to send a copy of all email to an external email address. This allows the threat actor to silently retrieve a copy of every email the user receives.
- Multifactor authentication modification – DART has detected instances of threat actors registering additional authentication methods against compromised accounts for use with MFA, such as phone numbers or authenticator apps.

- Device enrollment – in some cases, DART has seen threat actors add a device to an Azure AD tenant they control. This is an attempt to bypass conditional access rules with exclusions such as known devices.
- Data exfiltration – threat actors may use the inbuilt sharing functionality in SharePoint and OneDrive to share important or sensitive documents and organizational resources externally.

To strengthen your security posture, you should configure alerts to review high-risk modifications to a tenant. Some examples of this are:

- Modification or creation of security configurations
- Modification or creation of Exchange transport rules
- Modification or creation of privileged users or roles

Incident responders should review any audit logs related to user activity to look for signs of persistence. Logs available in the Unified Audit Log, Microsoft Defender for Cloud Apps, or SIEM solutions like Microsoft Sentinel can aid with investigations.

Conclusion

Although tactics from threat actors are constantly evolving, it is important to note that multifactor authentication, when combined with other basic security hygiene—utilizing antimalware, applying least privilege principals, keeping software up to date and protecting data—still protects against 98% of all attacks.

Fundamentally, it is important to consider the identity trust chain for the organization, spanning both internally and externally. The trust chain includes all systems (such as identity providers, federated identity providers, MFA services, VPN solutions, cloud-service providers, and enterprise applications) that issue access tokens and grant privilege for identities both cloud and on-premises, resulting in implicit trust between them.

In instances of token theft, adversaries insert themselves in the middle of the trust chain and often subsequently circumvent security controls. Having visibility, alerting, insights, and a full understanding of where security controls are enforced is key. Treating both identity providers that generate access tokens and their associated privileged identities as critical assets is strongly encouraged.

Adversaries have and will continue to find ways to evade security controls. The tactics utilized by threat actors to bypass controls and compromise tokens present additional challenges to defenders. However, by implementing the controls presented in this blog DART believes that organizations will be better prepared to detect, mitigate, and respond to threats of this nature moving forward.
