Targeting Passwords for Managed and Federated Microsoft Accounts

The Basics

With the continual rise in popularity of cloud services, Microsoft launched their Azure cloud infrastructure in early 2010, which eventually went on to support their Virtual Machines, Cloud Services, and Active Directory Domain Services. There are two different ways a Microsoft domain can support cloud authentication; managed and federated. A federated domain is one whose authentication communicates with on-site federation providers such as Active Directory Federation Services (ADFS). These on-site providers communicate with the internal Active Directory domain controllers to determine if a user's username and password are correct. In contrast, managed domains communicate solely with Microsoft's cloud infrastructure and pass the provided username and password to Windows Azure Active Directory to validate authorization. It is worth noting that on premise Active Directory can be synced with Azure, meaning that usernames and passwords have a decent chance of being shared across the two.

Use Case

During external penetration tests, it's common to attempt password guessing against available services to attempt to gain a foothold within an application or environment. This includes testing weak passwords for externally available domain services such as Office365, OWA, and VPN, among others. Being able to quickly and efficiently obtain the correct URI and perform password guessing across domains leaves more time for other fun testing. So I went on a search to find a program or script that could help automated password attempts against these cloud friendly services. My coworker Karl Fosaaen recently released a script and blog on identifying federated and managed domains with PowerShell. This prompted me to write a natural continuation on the subject by adding automated password guessing. The end result is Invoke-ExternalDomainBruteforce.psm1, a password bruteforce tool for managed and federated domains. Features of the script include:

- Automatically identifying managed or federated domains
- Single email or email list password guessing

Get the code here:
https://github.com/NetSPI/PowerShell/blob/master/Invoke-ExternalDomainBruteforce.psm1

Below is an overview for each password guessing scenario. Please note that single email and email lists are supported by both Managed and Federated domains.

Single email targeting against a managed domain

Currently, there is no elegant way to exit from a successful connection to Microsoft's Managed infrastructure in PowerShell. Because of this, the script outputs a warning informing the attacker that any commands run against a Managed domain will be run as the user displayed in the output. A

potential workaround is to use PowerShell sessions, allowing you can successfully create and destroy sessions connected to managed domains. However, in the interest of not requiring local admin, avoiding major state changes to a computer by enabling PowerShell remoting, and simplicity, I decided to simply warn the user to exit their current PowerShell session to avoid any unintended changes within the managed domain.

Email list targeting against a Federated domain

Targeting against a Federated domain will first identify the Authentication URL, then send an ADFSSecurityTokenRequest to the URL with the provided username and password. A valid username and password combination only returns a token value, meaning no active sessions are stored in the current PowerShell session. After all the usernames have been tested, the Authentication URL is also printed out for an attacker to visit the site and manually log in.

Prerequisites

The code to connect to both Federated and Managed domains was taken from Microsoft. Below are the links to download that code:

Federated:

https://blogs.msdn.microsoft.com/besidethepoint/2012/10/17/request-adfs-security-token-with-powershell/

Managed (Azure AD Powershell module):

https://msdn.microsoft.com/en-us/library/jj151815.aspx

Limitations / Future Work

Currently the script can only target one domain at a time. In the near future I'll update the script to target multiple domains at once.

References:

https://blogs.technet.microsoft.com/jeff_stokes/2013/07/08/another-cloud-tipfederated-vs-managed-users/
https://blogs.technet.microsoft.com/canitpro/2014/05/13/step-by-step-syncing-an-on-premise-ad-with-azure-active-directory/