GitHub - dafthack/MailSniper: MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for specific terms (passwords, insider intel, network architecture information, etc.). It can be used as a non-administrative user to search their own email, or by an administrator to search the mailboxes of every user in a domain.

By L1ghtn1ng

Archived: 2026-04-05 23:51:46 UTC

MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for specific terms (passwords, insider intel, network architecture information, etc.). It can be used as a non-administrative user to search their own email or by an Exchange administrator to search the mailboxes of every user in a domain.

MailSniper also includes additional modules for password spraying, enumerating users and domains, gathering the Global Address List (GAL) from OWA and EWS and checking mailbox permissions for every Exchange user at an organization.

For more information about the primary MailSniper functionality check out blog post.

For more information about additional MailSniper modules check out:

- GAL & Password Spraying
- Invoke-OpenInboxFinder

Download the MailSniper Field Manual to quickly reference various MailSniper functions.

Quick Start Guide

There are two main functions in MailSniper. These two functions are **Invoke-GlobalMailSearch** and **Invoke-SelfSearch**.

Invoke-GlobalMailSearch is a module that will connect to a Microsoft Exchange server and grant the "ApplicationImpersonation" role to a specified user. Having the "ApplicationImpersonation" role allows that user to search through all other domain user's mailboxes. After this role has been granted, the Invoke-GlobalMailSearch function creates a list of all mailboxes in the Exchange database. It then connects to Exchange Web Services (EWS) using the impersonation role to gather a number of emails from each mailbox and ultimately

searches through them for specific terms. By default, the script searches for "*password*","*creds*","*credentials*"

To search all mailboxes in a domain:

```
Invoke-GlobalMailSearch -ImpersonationAccount current-username -ExchHostname Exch01 -OutputCsv globa
```

This command will connect to the Exchange server located at 'Exch01' and prompt for administrative credentials (i.e. member of "Exchange Organization Administrators" or "Organization Management" group). Once administrative credentials have been entered, a PowerShell remoting session is setup with the Exchange server where the ApplicationImpersonation role is then granted to the "current-username" user. A list of all email addresses in the domain is then gathered, followed by a connection to EWS as "current-username" where by default, 100 of the latest emails from each mailbox will be searched through for the terms "*pass*","*creds*","*credentials*" and output to a CSV file called global-email-search.csv.

Invoke-SelfSearch is a module that will connect to a Microsoft Exchange server using EWS to gather a number of emails from the current user's mailbox. It then searches through them for specific terms. This could potentially assist in privilege escalation after obtaining a user's credentials or assist in locating sensitive data as a non-admin user.

To search the current user's mailbox:

```
Invoke-SelfSearch -Mailbox current-user@domain.com
```

This command will connect to the Exchange server autodiscovered from the email address entered using EWS where by default, 100 of the latest emails from the "Mailbox" will be searched through for the terms "*pass*","*creds*","*credentials*".

Invoke-GlobalO365MailSearch same as Invoke-GlobalMailSearch, with support for single sign-on (SSO) based authentication to O365.

Invoke-GlobalMailSearch Options

```
 ImpersonationAccount  - This user will be granted the ApplicationImpersonation role on the Exchange server.
ExchHostname          - The hostname of the Exchange server to connect to (If $AutoDiscoverEmail is specified th
AutoDiscoverEmail     - A valid email address that will be used to autodiscover where the Exchange server is loc
MailsPerUser          - The total number of emails returned from each mailbox.
Terms                 - Specific search terms used to search through each email subject and body. By default, th
OutputCsv             - Outputs the results of the search to a CSV file.
ExchangeVersion       - Specify the version of Exchange server to connect to. By default the script tries Exchan
AdminUserName         - The username of an Exchange administator (i.e. member of the "Exchange Organization Admi
AdminPassword         - The password to the Exchange administator (i.e. member of the "Exchange Organization Adm
EmailList             - A text file listing email addresses to search (one per line).
Folder                - A specific folder within each mailbox to search. By default, the script only searches th
```

```
Regex                - Use a regular expressions when performing searches. This will override the -Terms flag.
CheckAttachments     - Attempts to search through the contents of email attachements in addition to the default
DownloadDir          - Download files to a specific location.
```

Invoke-SelfSearch Options

```
 ExchHostname          - The hostname of the Exchange server to connect to (If $Mailbox is specified the server
Mailbox              - Email address of the current user the PowerShell process is running as.
MailsPerUser         - Number of emails to return.
Terms                - Specific search terms used to search through each email subject and body. By default, th
OutputCsv            - Outputs the results of the search to a CSV file.
ExchangeVersion      - Specify the version of Exchange server to connect to (default Exchange2010).
Remote               - A new credential box will pop up for accessing a remote EWS service from the internet.
Folder               - A specific folder within each mailbox to search. By default, the script only searches th
Regex                - Use a regular expressions when performing searches. This will override the -Terms flag.
CheckAttachments     - Attempts to search through the contents of email attachements in addition to the default
DownloadDir          - Download files to a specific location.
OtherUserMailbox     - Use this flag when attempting to read emails from a different user's mailbox
UsePrt               - Uses the current user's PRT to authenticate.
AccessToken          - Use provided oauth access token to authenticate.
```

Invoke-GlobalO365MailSearch Options

```
 UsePrtImperonsationAccount       - Uses the current user's PRT to authenticate ImperonsationAccount.
AccessTokenImpersonationAccount  - Use provided oauth access token to authenticate ImperonsationAccount.
UsePrtAdminAccount               - Uses the current user's PRT to authenticate AdminAccount.
AccessTokenAdminAccount          - Use provided oauth access token to authenticate ImperonsationAccount.
```

Additional MailSniper Modules

Get-GlobalAddressList will attempt to connect to an Outlook Web Access (OWA) portal and utilize the "FindPeople" method (only available in Exchange2013 and up) of gathering email addresses from the GAL. If this does not succeed the script will attempt to connect to EWS and attempt to gather the GAL.

```
Get-GlobalAddressList -ExchHostname mail.domain.com -UserName domain\username -Password Spring2021 -
```

Get-MailboxFolders will connect to a Microsoft Exchange server using EWS and gather a list of folders from the current user's mailbox.

```
Get-MailboxFolders -Mailbox current-user@domain.com
```

Invoke-PasswordSprayOWA will attempt to connect to an OWA portal and perform a password spraying attack using a userlist and a single password.

```
Invoke-PasswordSprayOWA -ExchHostname mail.domain.com -UserList .\userlist.txt -Password Spring2021
```

Invoke-PasswordSprayEWS will attempt to connect to an EWS portal and perform a password spraying attack using a userlist and a single password.

```
Invoke-PasswordSprayEWS -ExchHostname mail.domain.com -UserList .\userlist.txt -Password Spring2021
```

Invoke-PasswordSprayGmail This module will first attempt to connect to a Gmail Authentication portal and perform a password spraying attack using a userlist and a single password.

```
Invoke-PasswordSprayGmail -UserList .\userlist.txt -Password Fall2016 -Threads 15 -OutFile gmail-spra
```

Invoke-DomainHarvestOWA will attempt to connect to an OWA portal and determine a valid domain name for logging into the portal from the WWW-Authenticate header returned in a web response from the server or based off of small timing differences in login attempts.

```
Invoke-DomainHarvestOWA -ExchHostname mail.domain.com
```

Invoke-UsernameHarvestOWA will attempt to connect to an OWA portal and harvest valid usernames based off of small timing differences in login attempts.

```
Invoke-UsernameHarvestOWA -ExchHostname mail.domain.com -UserList .\userlist.txt -Threads 1 -OutFile
```

Invoke-UsernameHarvestGmail is a module that will attempt to enumerate Google Apps user accounts and potentially identify user accounts that opt-out of implemented 2FA solutions.

```
Invoke-UsernameHarvestGmail -Account
Invoke-UsernameHarvestGmail -UserFile .\emails.txt
Invoke-UsernameHarvestGmail -UserFile .\emails.txt -ProxyHosts 10.0.0.5:8080,10.0.0.6:8080,10.0.0.10
Invoke-UsernameHarvestGmail -UserFile .\emails.txt -Detailed
Get-Content emails.txt | % { Invoke-UsernameHarvestGmail $_ }
```

Invoke-OpenInboxFinder will attempt to determine if the current user has access to the Inbox of each email address in a list of addresses.

```
Invoke-OpenInboxFinder -EmailList email-list.txt
```

Get-ADUsernameFromEWS will attempt to determine the Active Directory username for a single email address or a list of addresses. Use the Get-GlobalAddressList module to harvest a full list of email addresses to use with Get-ADUsernameFromEWS.

```
Get-ADUsernameFromEWS -EmailList email-list.txt
```

Send-EWSEmail will attempt to connect to EWS and send an email.

```
Send-EWSEmail --ExchHostname substrate.office.com -Recipient $targetEmail -Subject "Foo" -EmailBody
```

Source: https://github.com/dafthack/MailSniper