Check Point
SOFTWARE TECHNOLOGIES LTD

Microsoft

Poisoned RDP

Offense and Defense

@EyalItkin

@dana_baril

Who Are We?

Eyal Itkin

Security Researcher

Check Point Research cp<r>

@EyalItkin

Dana Baril

Security Researcher

Microsoft Defender ATP

@dana_baril

Motivation

- “Lazy” Lateral Movement
- “Ambush” privileged users
- IT Staff
  - Gain credentials
- Malware Researchers
  - Escape isolated virtual machines

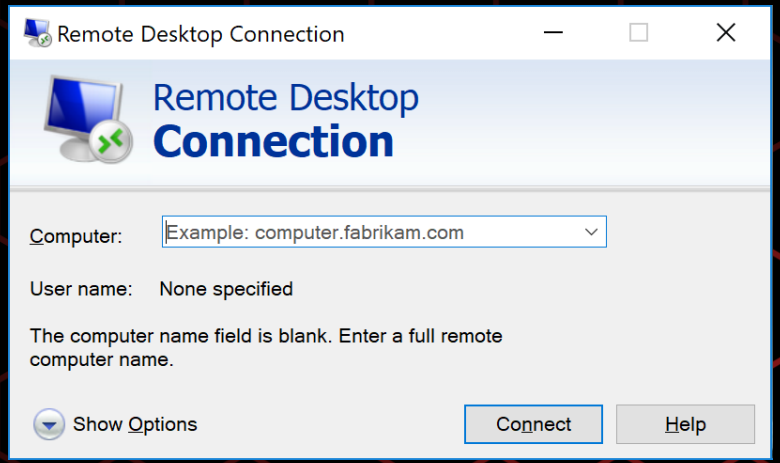
Poisoned RDP ?

- Collect credentials from the victim
- Attack & Take over the victim’s computer

Remote Desktop Protocol (RDP)

“Client” **“Server”**

- Connects to a remote Windows Machine
  - Remote corporate PC / Server
  - Local / Remote Virtual Machine
- Attack doesn’t require “Admin”



Break `mstsc.exe` ?

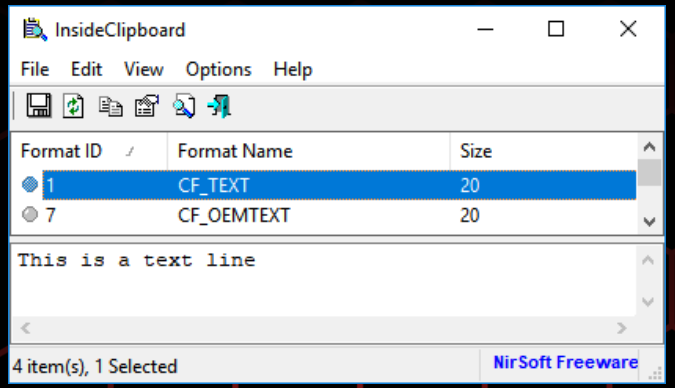
- PoCs from previous targets failed ☹
- The code is robust
  - Smart buffers check for parsing errors
- Includes many more features
- Where should we go now?

Back to the D~~rawing~~ Clipboard

- Until now, the clipboard shared text:
  - CF_TEXT
  - CF_UNICODETEXT
- It seems like Microsoft supports many more formats now
- Let’s dig into the clipboard

Clipboard 101

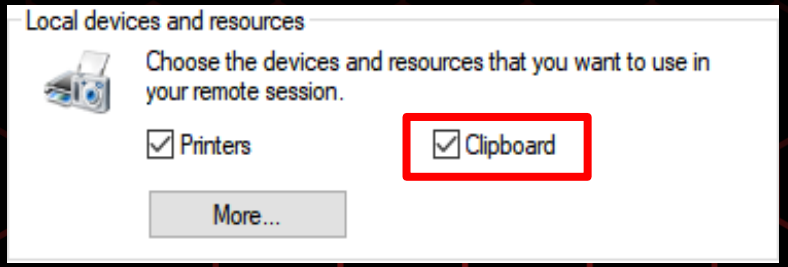
- A kernel data structure that stores data
  - One clipboard per session (“connection”)
  - Shared between processes



- **“Caution:** Clipboard data is not trusted. Parse the data carefully before using it in your application.” (MSDN)

Clipboard Over RDP

- Everything in the clipboard is synchronized automatically
- Black Lists instead of White Lists
  - Some formats are discarded by ID
  - Some formats are discarded by Name
- To avoid syncing “heavy” content, all content is subject to “delayed rendering”



Drag & Drop

- Internally, copying files is called “Drag & Drop”
- Copying files uses multiple formats
  - `CF_HDROP` – lists the file names
  - `FileGroupDescriptorW` – full metadata
  - Many more…
- Let’s see how it works in practice

Drag & Drop In Action – Ctrl+C

Drag & Drop In Action – Ctrl+C

RDP Server

RDP client

Drag & Drop In Action – Ctrl+V

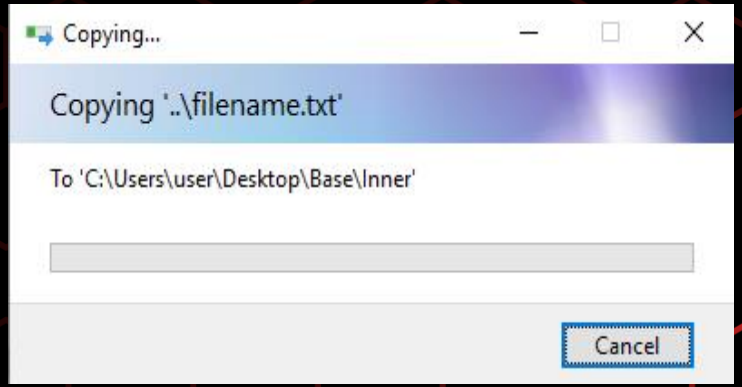
Drag & Drop In Action – Ctrl+V

FileGroupDescriptorW

- Proprietary blob structure
- Contains a list of file records
  - Meta data (timestamps)
  - File path – filename
- Client passes it directly to the clipboard

Path Canonicalization

@GullOmer: “try to find where they sanitize the path”



Path Traversal Over RDP

- We received a CVE from Microsoft: CVE 2019-0887
- When using “Copy & Paste” a malicious server can:
  - Drop arbitrary files to arbitrary locations
- Drop your script in the Startup folder and that’s it

Taking it one step further

- The clipboards are **fully** synchronized
  - Ctrl+C updates the clipboard
  - Each update sends a `CLIPRDR_FORMAT_LIST`
  - The receiver updates his clipboard accordingly
- What does it mean?

Scenario #1 - Eavesdropping

- When the client copies a password we get it too
- This is a **feature** of the synced clipboard
- We know in advance when the client is going to copy a file on **his** computer

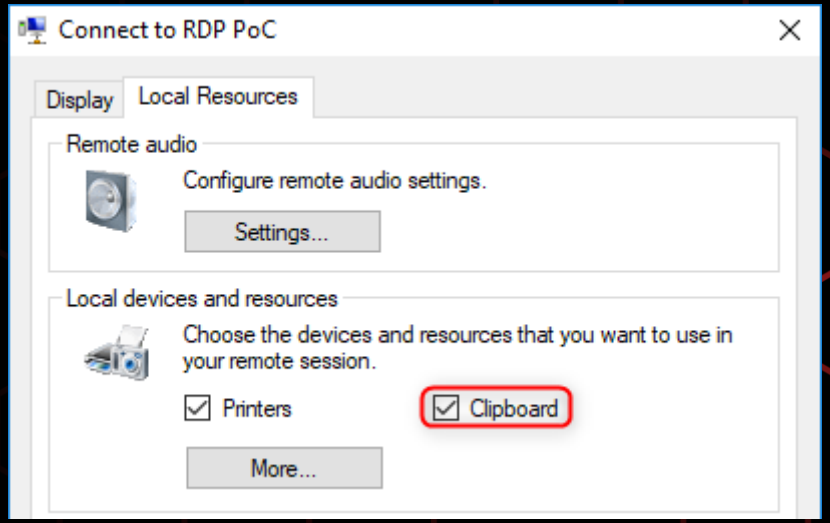
Scenario #2 – Ctrl+V Only Attack

- Once again, ambush the client
- When he copies a file, start the attack
- Send an update message and switch his clipboard to a malicious FGDw
- His Ctrl+V will trigger the path traversal

Did we break them all ?

Hyper-V

- Never used it till now
- Installed a Hyper-V machine, and



Hyper-V? RDP!

- Microsoft uses RDP for accessing virtualized machines
- The GUI connection to the VM is transferred over RDP!
- Our PoC worked on the first attempt
- We just found a Guest-to-Host VM Escape ☺

Hyper-V Demo

https://youtu.be/nSGlMJqQEh0

Note on WDAG and friends

- Windows Defender Application Guard
  - Browsing “risky” sites with a virtualized Edge browser
- Uses hvsirdpclient.exe instead of mstsc.exe
- This time, MS uses White-lists instead of Black-Lists
  - Clipboard is off by default
  - The clipboard permits only 2 format types: Text & Images
- The White list blocks our vulnerability, good job ☺

Defense

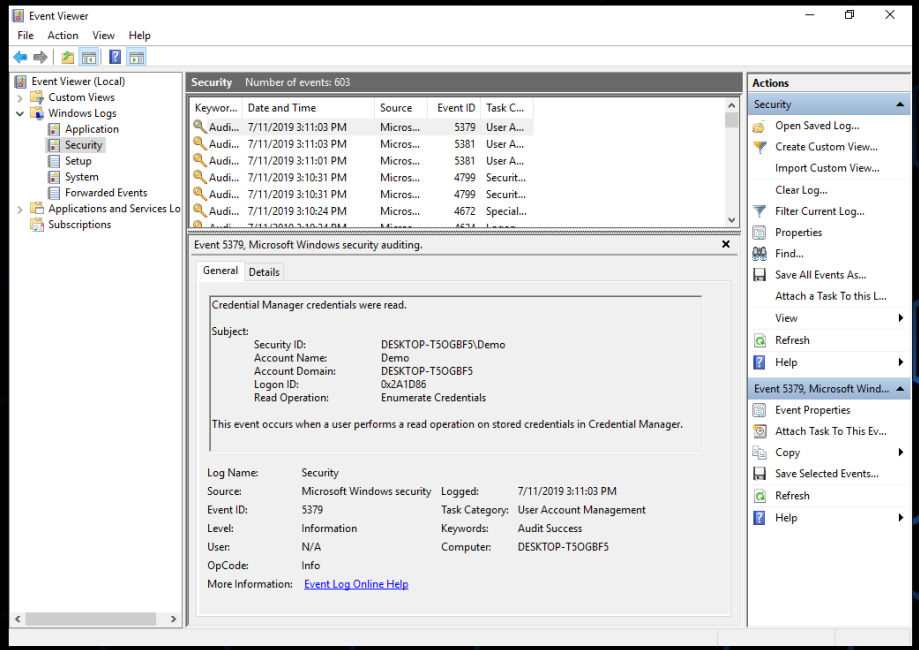
VULNERABILITY DETECTED
PATCH EVERYTHING NOW!
imgflip.com

A patch is not enough

- Users remain vulnerable until they install patch
- Detect using existing telemetry
- Detection must be implemented on “victim” machine
- RDP anomaly detection won’t cut it

Event Tracing for Windows (ETW)

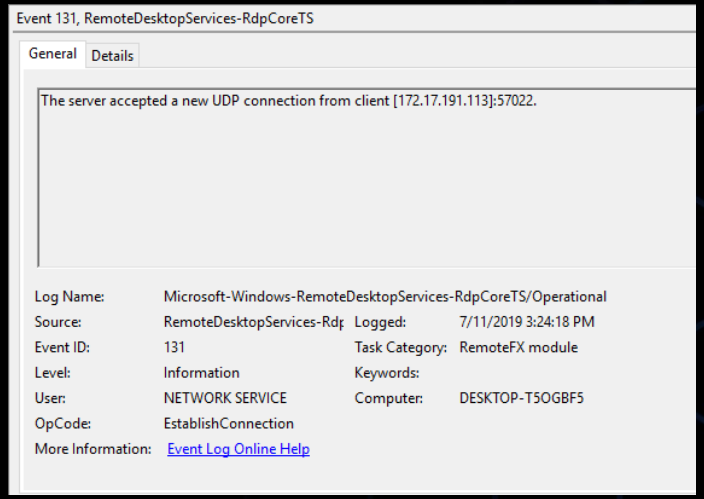
- Kernel-level tracing facility that lets you log kernel or application-defined events



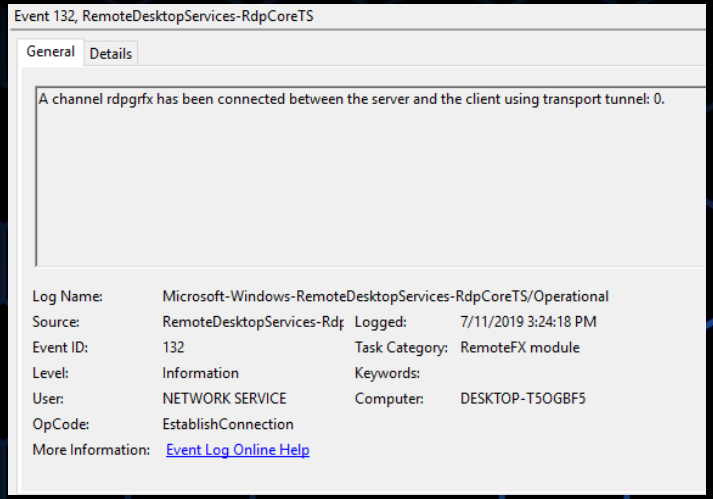
RDP Connections Events

Provider Guid: 1139c61b-b549-4251-8ed3-27250a1edec8

Microsoft-Windows-RemoteDesktopServices-RdpCoreTS



Events 131 – accepting connection



Event 132 – channel connected

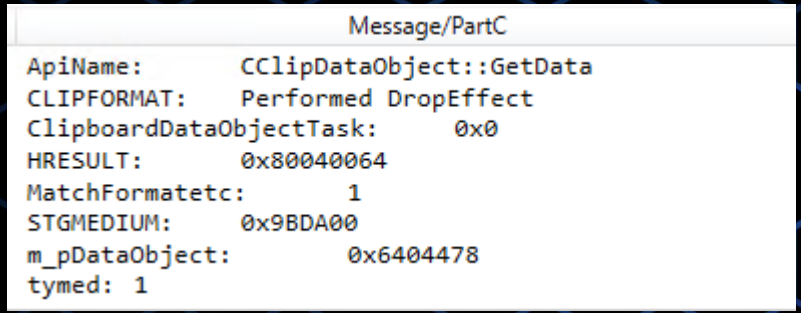
Clipboard Events

Non-manifested provider, tracing clipboard API usages

Provider guid: 3e0e3a92-b00b-4456-9dee-f40aba77f00e

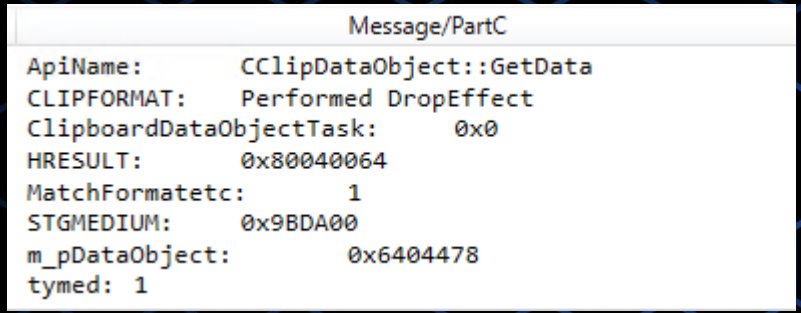
Microsoft.Windows.OLE.Clipboard

Task name: OLE_Clipboard_MethodDiagnostics



Clipboard Events

- Selected properties:
  - ApiName: GetData, SetData
  - CLIPFORMAT: Returned clipboard format (bitmap, text, Unicode text, etc.).
  - HRESULT: Api HResult
  - Tymed: Paste destination medium
- **No clipboard content!**



Telemetry Demo

https://youtu.be/nSGlMJqQEh0

Detection Logic – Basic

- While in RDP:
  1. When multiple files are copied in a short period of time
  2. Triggers a scan

File Creation Events

- In order to overcome the file information gap, we need more data!
- Security products have file creation indications
  - File name
  - File creation time

Detection Logic

- While in RDP:
  1. When multiple files are copied in a short period of time
  2. Correlate file creation with the same timestamps
  3. If the correlated files are in different directories – alert!

More Detection Logics

- Startup folder as a destination
  - Anomaly detection
  - Files scanning
- Clipboard as an attack vector
  - File pasting anomaly - number of pasted files or the files directories
- Malicious files dropping
  - File creation anomaly - file path, creation time and file name

OS Patch

- Verify the RDP clipboard: ValidateFilePaths

```
000000016AB893FC   CFormatDataPacker::DecodeFormatData(void * *,ulong,uchar *,ulong)
000000016AB89640   call        ?FileDescriptorW@CClipFormatTypes@@QEAAIXZ
000000016AB89645   xor         b4 r9d, b4 r9d
000000016AB89648   mov         b4 r8d, b4 ebp                                  // unsigned int
000000016AB8964B   cmp         b4 ebx, b4 eax
000000016AB8964D   mov         rdx, r14                                        // unsigned __int8 *
000000016AB89650   lea         rax, ss:[rsp+arg_10]
000000016AB89655   setz        b1 r9b                                          // int
000000016AB89659   mov         ss:[rsp+var_28], rax                            // int *
000000016AB8965E   call        ?ValidateFilePaths@CFormatDataPacker@@AEAAJPEAEKHPEAH@Z
000000016AB89663   mov         b4 ebx, b4 eax
000000016AB89665   test        b4 eax, b4 eax

000000016AB89667   jns         0x16AB896AD
```

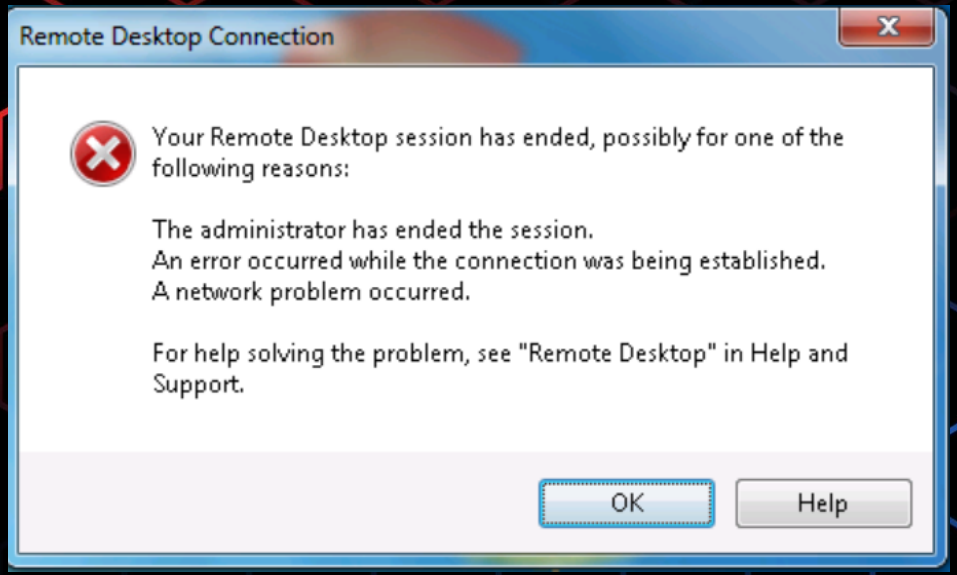
- Verify canonical path before pasting:

```
pszFilename = pCurrentFileRecord->szFilename;
status_code = PathCchCanonicalize(&pszPathOut, 0x104ui64, pszFilename);
if ( (status_code & 0x80000000) != 0 )
{
```

What have we learned ?

- Design lesson: Think twice before connecting different modules
  - Clipboards were designed to be used locally, and therefore trusted
  - When sharing across machines it made sense to enable clipboard sharing
  - However, this exposed machines to clipboards they can no longer trust
- Windows telemetry is an important tool in the defender’s toolbox
- Our industry can benefit from cross-community collaborations

That’s all folks



@Eyalltkin

@dana_baril