MAGIC DOT

A Hacker's Magic Show of Disappearing Dots and Spaces

Or Yair

MAGIC DOT

- Security Research Team Lead at SafeBreach
- 6+ years in security research
- Linux, embedded and some Android research
- 3 years Windows research
- Creator of Aikido Wiper, DoubleDrive

Agenda

Windows Known Issue Introduction

Research Goals

Post-Exploitation Techniques

Vulnerabilities

CVEs + Fixes

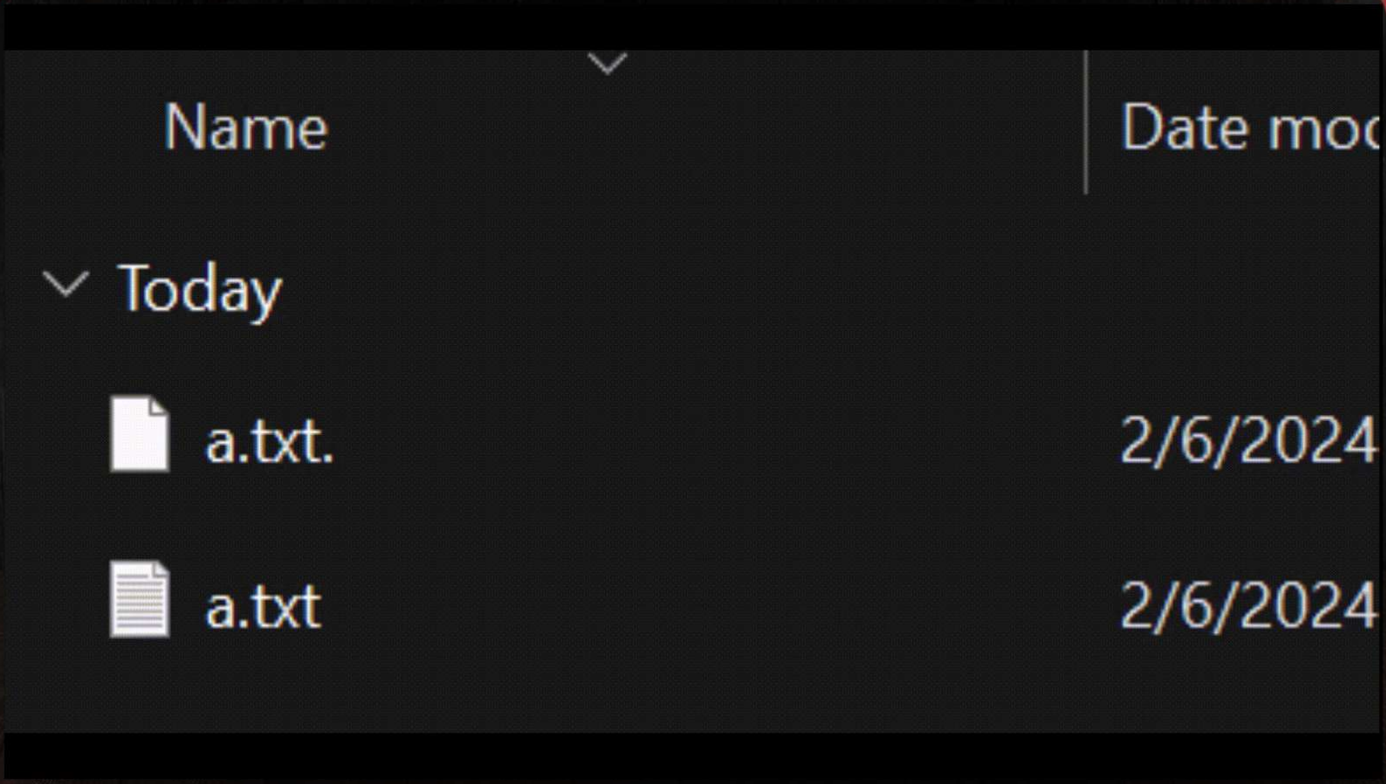
Takeaways

GitHub + Q&A

Windows Backwards Compatibility

More than **1.4 billion** active devices

My first encounter with “Magic"



Name | Date moc

Today

a.txt. 2/6/2024

a.txt 2/6/2024

Microsoft’s Documentation

Do not end a file or directory name with a space or a period. Although the underlying file system may support such names, the Windows shell and user interface does not. However, it is acceptable to specify a period as the first character of a name. For example, “.temp”.

Normal (DOS) to NT Path Conversion

Win32 APIs path arguments are normal DOS paths. Conversion is needed.

`RtlpDosPathNameToRelativeNtPathName()`

```
C:\Users\User\Documents\example.txt
```

↓

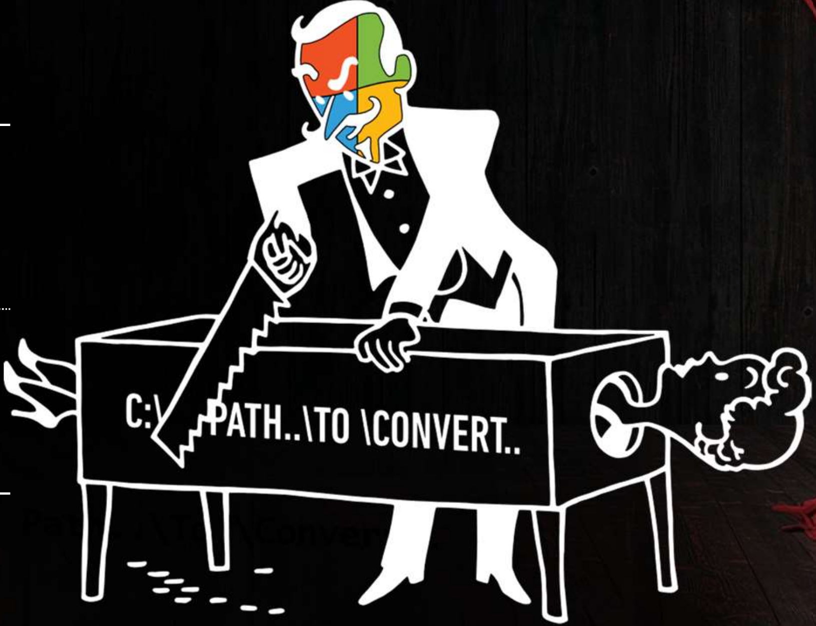
```
\??\C:\Users\User\Documents\example.txt
```

Normal (DOS) to NT Path Conversion

`RtlpDosPathNameToRelativeNtPathName()`

Removes:

- Trailing dots from any path element
- Trailing spaces from the last path element



MAGIC DOT

DOS Path	NT Path
C:\example\example.	\??\C:\example\example
C:\example\example...	\??\C:\example\example
C:\example\example**<space>**	\??\C:\example\example
C:\example\example**<space><space>**	\??\C:\example\example
C:\example.\example	\??\C:\example\example
C:\example**<space>**\example	\??\C:\example**<space>**\example

“The Definitive Guide on Win32 to NT Path Conversion”

MAGIC DOT

by James Forshaw with
Google Project Zero

https://googleprojectzero.blogspot.com/2016/02/the-definitive-guide-on-win32-to-nt.html

#1 Research Goal

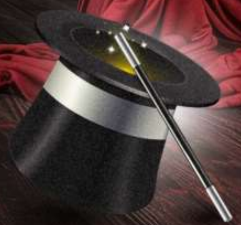
Rootkit-like abilities

Utilize the issue for concealments

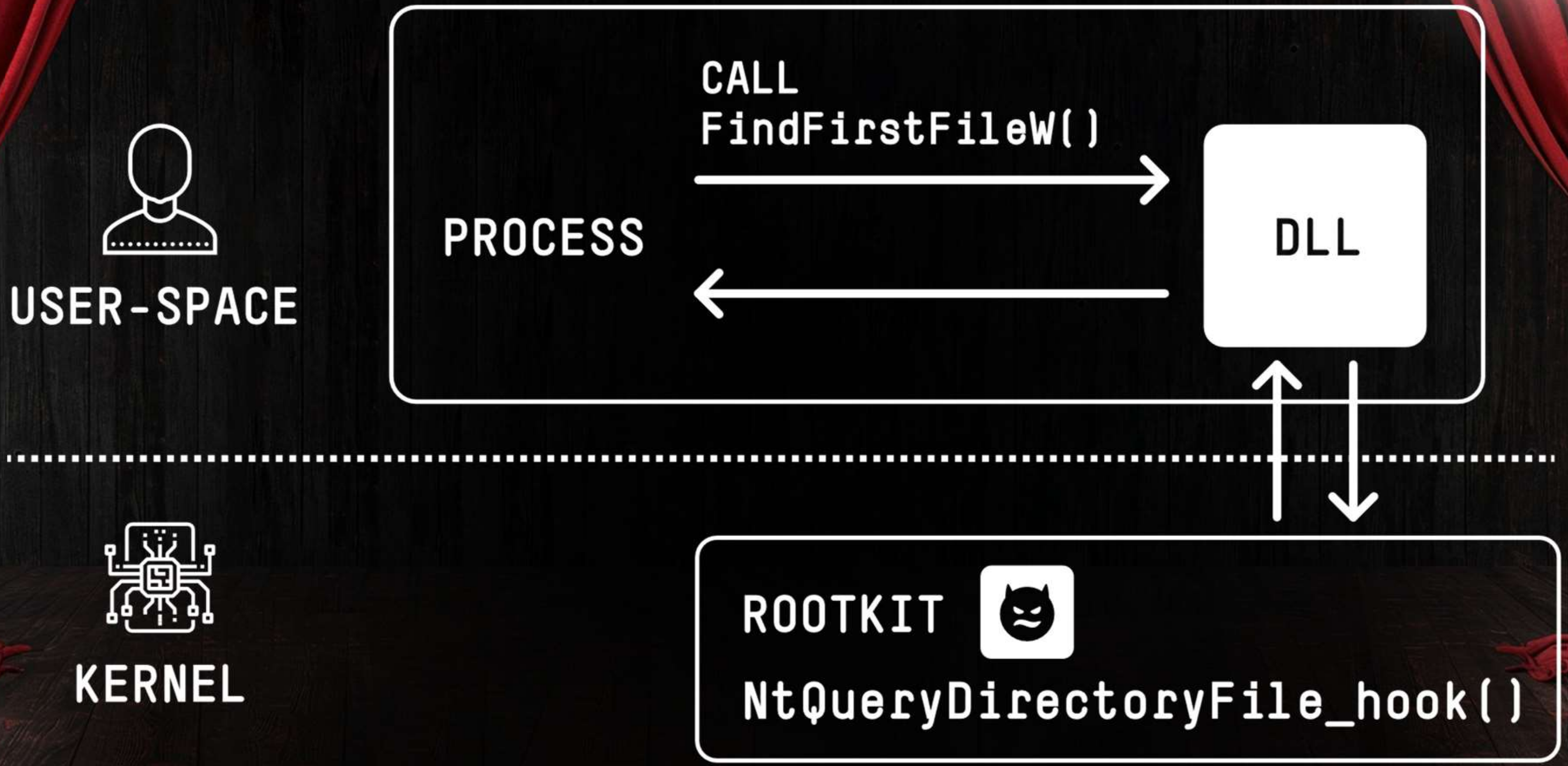
Typical Rootkits

Primary Goal – Concealments

Types
User-Space
Kernel



Kernel Rootkit



Kernel Rootkit Requirements

MAGIC DOT

Ability to run in the kernel:

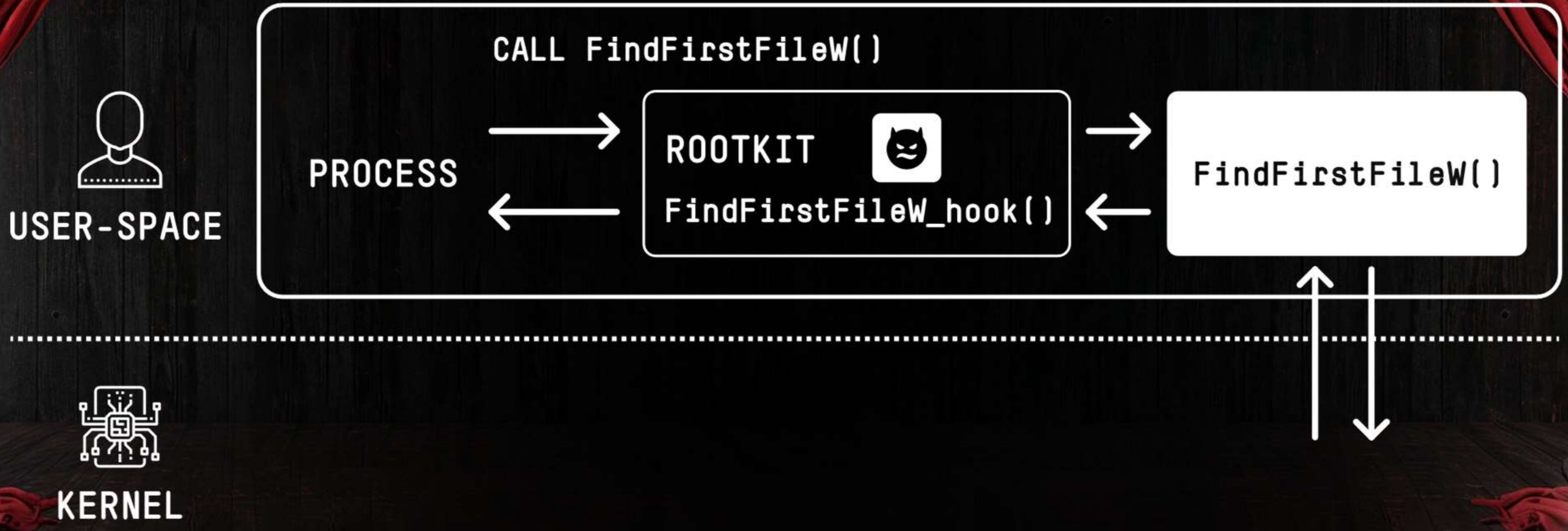
Admin Privileges + Handle Obstacles:

Driver Signature Enforcement

Driver Block List

HVCI

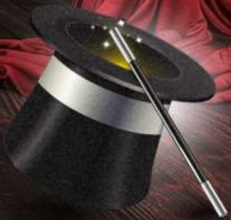
User-Space Rootkit



User-Space Rootkit Requirements

Ability to write or run code in all processes:

Admin Privileges

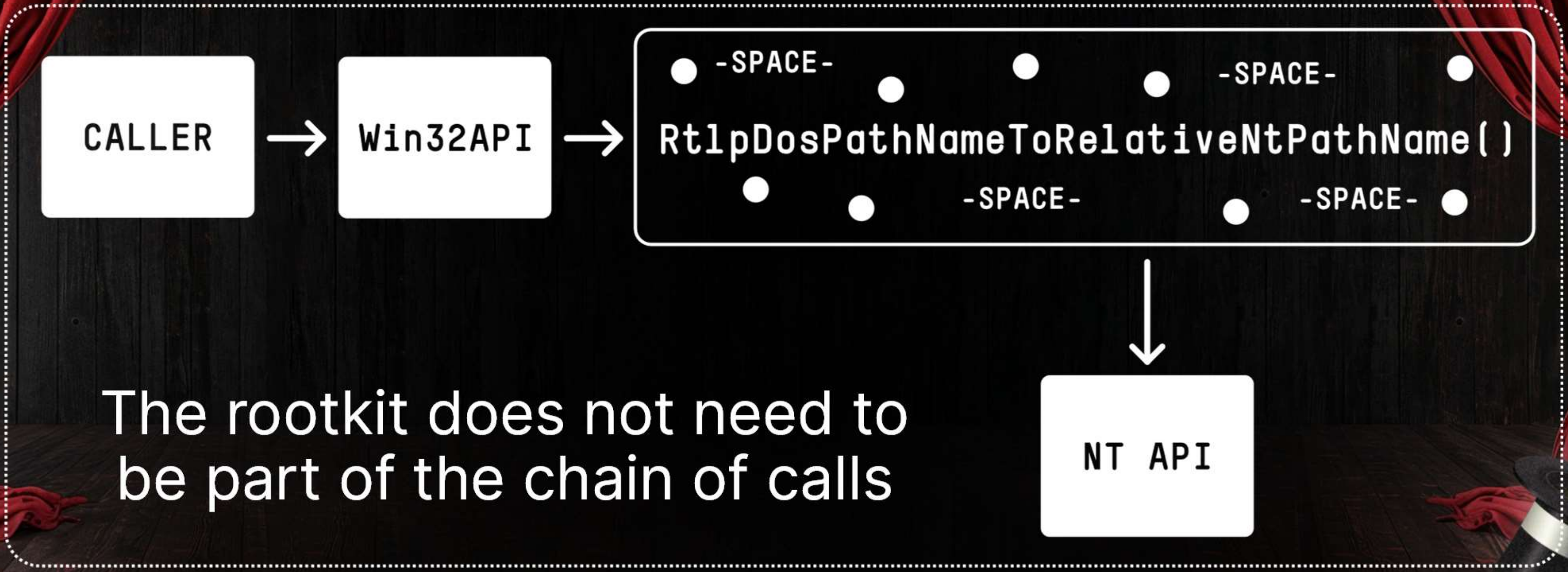


Something is Missing

How can unprivileged malwares conceal themselves?

Do they must have a 0-day PE?

New - Unprivileged Rootkit



CALLER → Win32API → RtlpDosPathNameToRelativeNtPathName() → NT API

-SPACE- -SPACE- -SPACE- -SPACE-

The rootkit does not need to be part of the chain of calls

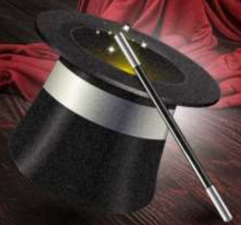
#1 Research Goal

Rootkit-like abilities

Utilize the issue for concealments

+

No special required privileges



#2 Research Goal

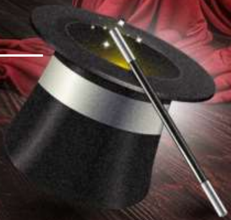
Prove that an unfixed known issue is a security risk:

Find vulnerabilities caused by the known issue.

Files and Directories Concealments

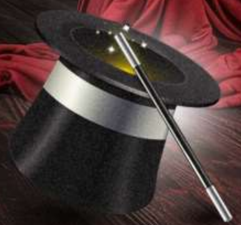
Concealing Files and Directories

Inoperable File/Directory:	Result:
Name File/Directory "..." or "blabla..." (using NT path)	Directory can't be listed, deleted, added with files
	File can't be deleted, written, or read



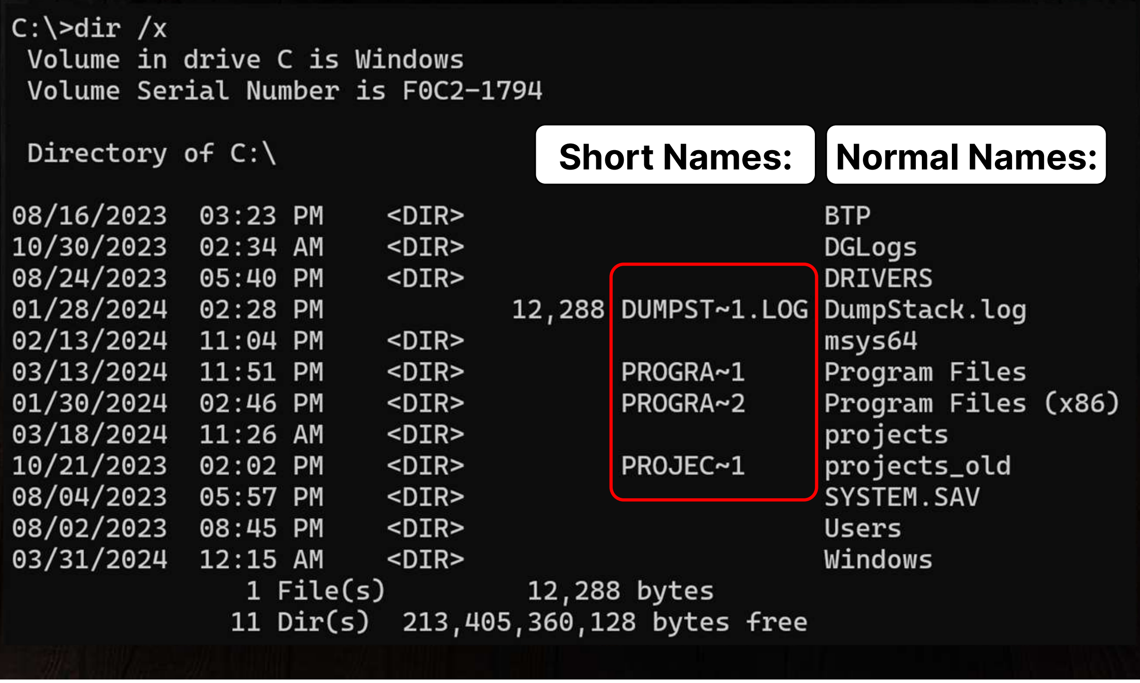
Concealing Files and Directories

Impersonated Directory/File:	Result:
Name a file **“benign.”** (using NT path)	File operations on **“benign.”** affect **“benign”** instead.



Short Names (8.3 filename)

An old filename convention.
Backwards Compatibility (Again).
Used by old versions of DOS & Windows.



```
C:\>dir /x
 Volume in drive C is Windows
 Volume Serial Number is F0C2-1794

 Directory of C:\

08/16/2023  03:23 PM    <DIR>                       BTP
10/30/2023  02:34 AM    <DIR>                       DGLogs
08/24/2023  05:40 PM    <DIR>                       DRIVERS
01/28/2024  02:28 PM            12,288 DUMPST~1.LOG DumpStack.log
02/13/2024  11:04 PM    <DIR>                       msys64
03/13/2024  11:51 PM    <DIR>          PROGRA~1     Program Files
01/30/2024  02:46 PM    <DIR>          PROGRA~2     Program Files (x86)
03/18/2024  11:26 AM    <DIR>                       projects
10/21/2023  02:02 PM    <DIR>          PROJEC~1     projects_old
08/04/2023  05:57 PM    <DIR>                       SYSTEM.SAV
08/02/2023  08:45 PM    <DIR>                       Users
03/31/2024  12:15 AM    <DIR>                       Windows
               1 File(s)         12,288 bytes
              11 Dir(s)  213,405,360,128 bytes free
```

Concealing Files and Directories

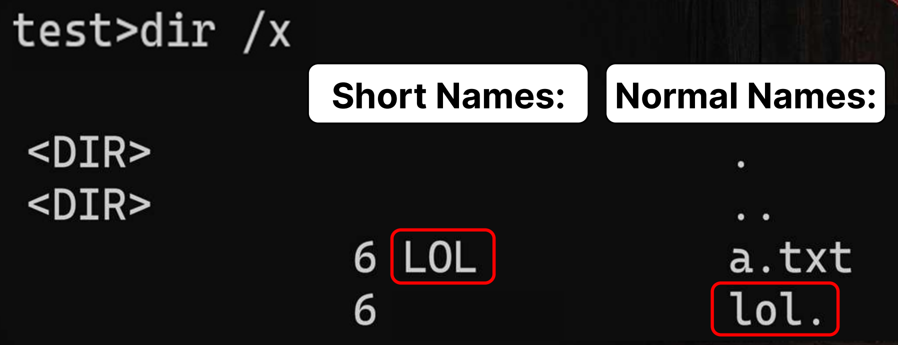
Improved Impersonated File/Directory

Name a file/directory

“lol.” (using NT path)

Result

File operations on **“lol.”** affect a file with the short name **“LOL”** instead.



```
test>dir /x

 <DIR>                                  .
 <DIR>                                  ..
                  6 LOL                 a.txt
                  6                     lol.
```

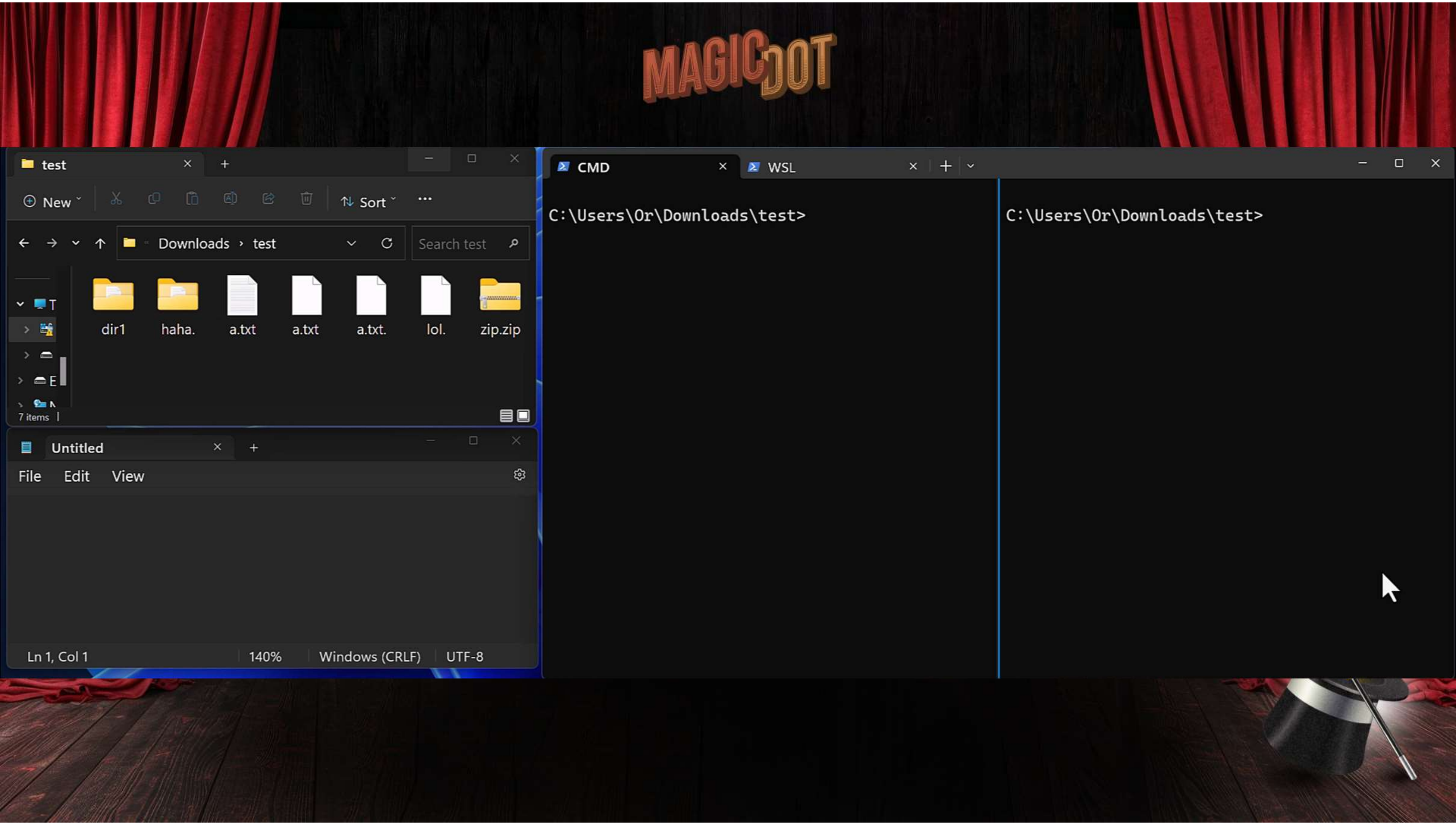
Concealing Files and Directories

ZIP Hidden Files:

End a file name in a ZIP archive with a dot

Result:

Listing the archive with File Explorer does not show the file



```
C:\Users\Or\Downloads\test>
```

```
C:\Users\Or\Downloads\test>
```

Processes
Concealments

Concealing Processes

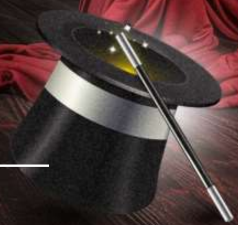
Untraceable Process:

NtCreateUserProcess - “\??\C:\Windows**.**\blabla\blabla.exe”

Result:

Executable cannot be accessed

Executable’s properties cannot be viewed from Task Manager / ProcExp…



Concealing Processes

Impersonated Process:

NtCreateUserProcess -
“\??\C:\Windows**.**\System32\svchost.exe”

Result:

File operations on the executable affect
the original svchost.exe

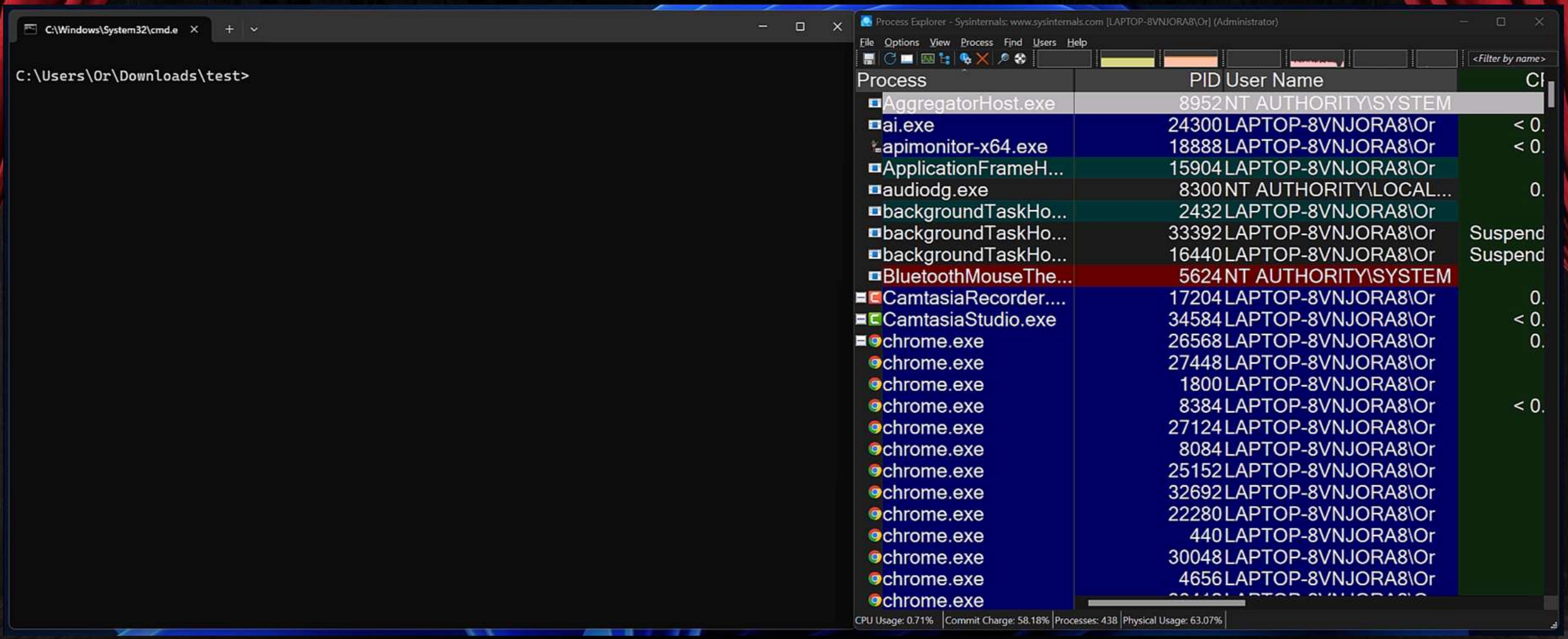
Concealing Processes

Also:

Task Manager, ProcExp show that the executable is verified and signed by Microsoft

Prefetch analysis tools show details about the original svchost.exe

MAGIC DOT



C:\Windows\System32\cmd.e
Process Explorer - Sysinternals: www.sysinternals.com [LAPTOP-8VNJORA8\Or] (Administrator)
File
Options
View
Process
Find
Users
Help
<Filter by name>

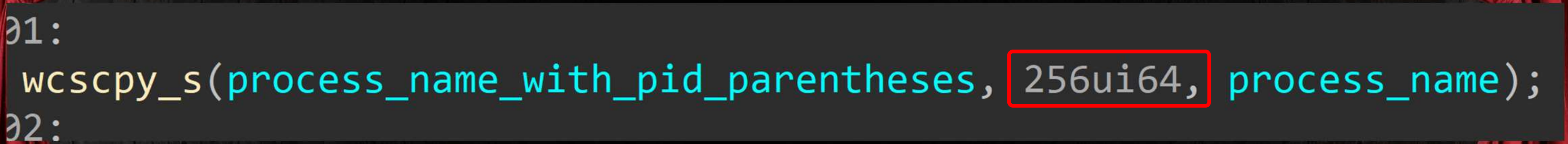
```
C:\Users\Or\Downloads\test>
```

```
Process                     PID User Name                 CI
AggregatorHost.exe         8952 NT AUTHORITY\SYSTEM
ai.exe                    24300 LAPTOP-8VNJORA8\Or        < 0.
apimonitor-x64.exe        18888 LAPTOP-8VNJORA8\Or        < 0.
ApplicationFrameH...      15904 LAPTOP-8VNJORA8\Or
audiodg.exe                8300 NT AUTHORITY\LOCAL...      0.
backgroundTaskHo...        2432 LAPTOP-8VNJORA8\Or
backgroundTaskHo...       33392 LAPTOP-8VNJORA8\Or     Suspend
backgroundTaskHo...       16440 LAPTOP-8VNJORA8\Or     Suspend
BluetoothMouseThe...       5624 NT AUTHORITY\SYSTEM
CamtasiaRecorder....      17204 LAPTOP-8VNJORA8\Or         0.
CamtasiaStudio.exe        34584 LAPTOP-8VNJORA8\Or        < 0.
chrome.exe                26568 LAPTOP-8VNJORA8\Or         0.
chrome.exe                27448 LAPTOP-8VNJORA8\Or
chrome.exe                 1800 LAPTOP-8VNJORA8\Or
chrome.exe                 8384 LAPTOP-8VNJORA8\Or        < 0.
chrome.exe                27124 LAPTOP-8VNJORA8\Or
chrome.exe                 8084 LAPTOP-8VNJORA8\Or
chrome.exe                25152 LAPTOP-8VNJORA8\Or
chrome.exe                32692 LAPTOP-8VNJORA8\Or
chrome.exe                22280 LAPTOP-8VNJORA8\Or
chrome.exe                  440 LAPTOP-8VNJORA8\Or
chrome.exe                30048 LAPTOP-8VNJORA8\Or
chrome.exe                 4656 LAPTOP-8VNJORA8\Or
chrome.exe

CPU Usage: 0.71% | Commit Charge: 58.18% | Processes: 438 | Physical Usage: 63.07%
```

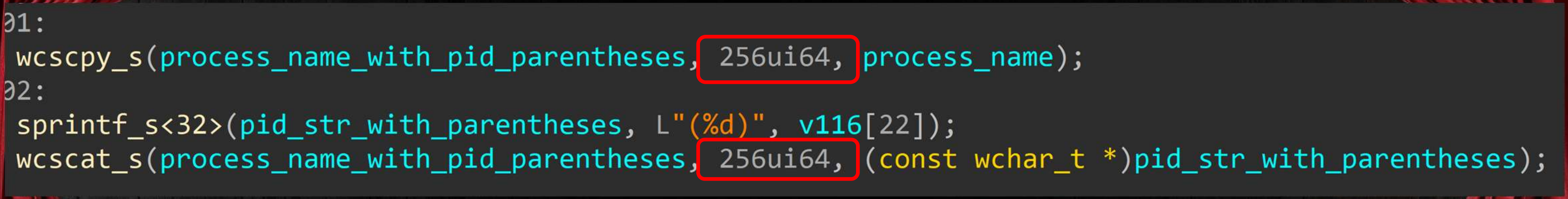
Anti Analysis

ProcExp DoS –
A Built In “Safe” Feature



```
wcscpy_s(process_name_with_pid_parentheses, 256ui64, process_name);
```

ProcExp DoS –
A Built In “Safe” Feature



```
01:
 wcscpy_s(process_name_with_pid_parentheses, 256ui64, process_name);
02:
 sprintf_s<32>(pid_str_with_parentheses, L"(%d)", v116[22]);
 wcscat_s(process_name_with_pid_parentheses, 256ui64, (const wchar_t *)pid_str_with_parentheses);
```

ProcExp DoS –
A Built In “Safe” Feature

https://learn.microsoft.com/en-us/cpp/c-runtime-library/security-enhanced-versions-of-crt-functions

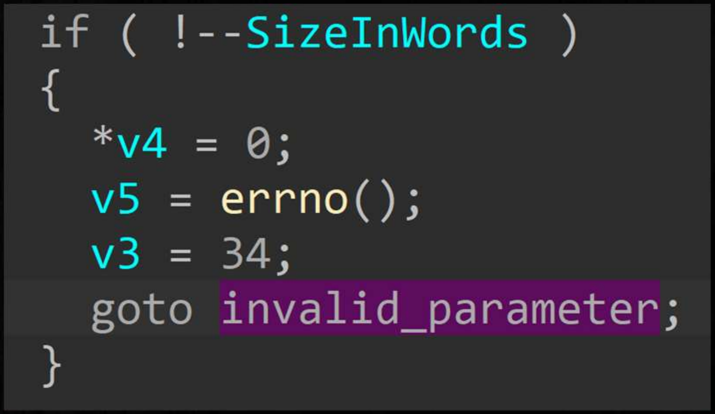
Safe C-Runtime Functions:

The more secure versions

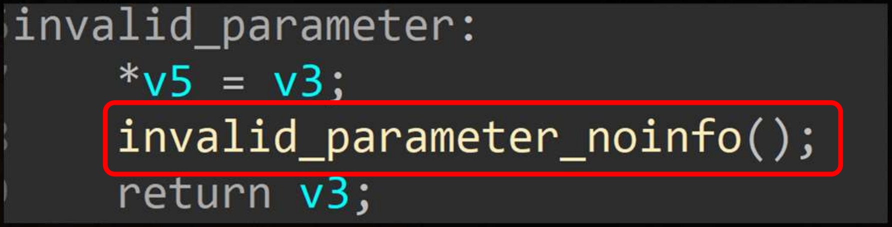
Microsoft’s docs – “If there's an error, they invoke an error handler.”

ProcExp DoS –
A Built In “Safe” Feature

wcscat_s:



```
if ( !--SizeInWords )
{
  *v4 = 0;
  v5 = errno();
  v3 = 34;
  goto invalid_parameter;
}
```



```
invalid_parameter:
    *v5 = v3;
    invalid_parameter_noinfo();
    return v3;
```

ProcExp DoS –
A Built In “Safe” Feature

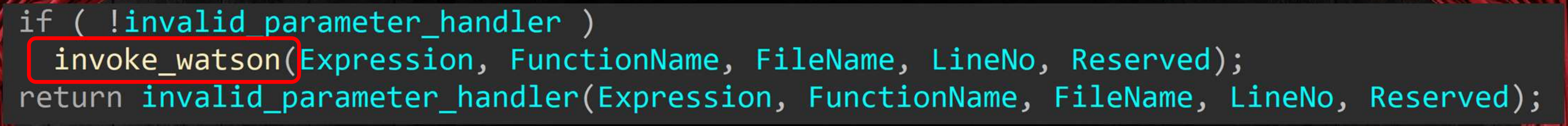
https://learn.microsoft.com/en-us/cpp/c-runtime-library/parameter-validation

“The invalid parameter handler dispatch function calls the currently assigned invalid parameter handler.

By default, the invalid parameter calls _invoke_watson, which causes the application to close and generate a mini-dump.”

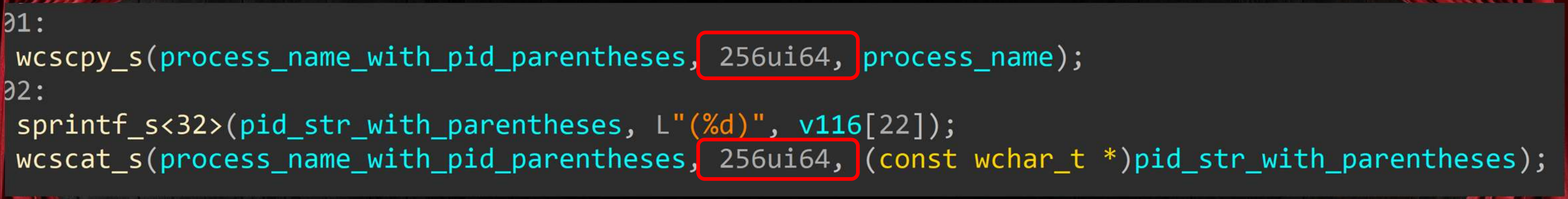
ProcExp DoS –
A Built In “Safe” Feature

invalid_parameter_noinfo():



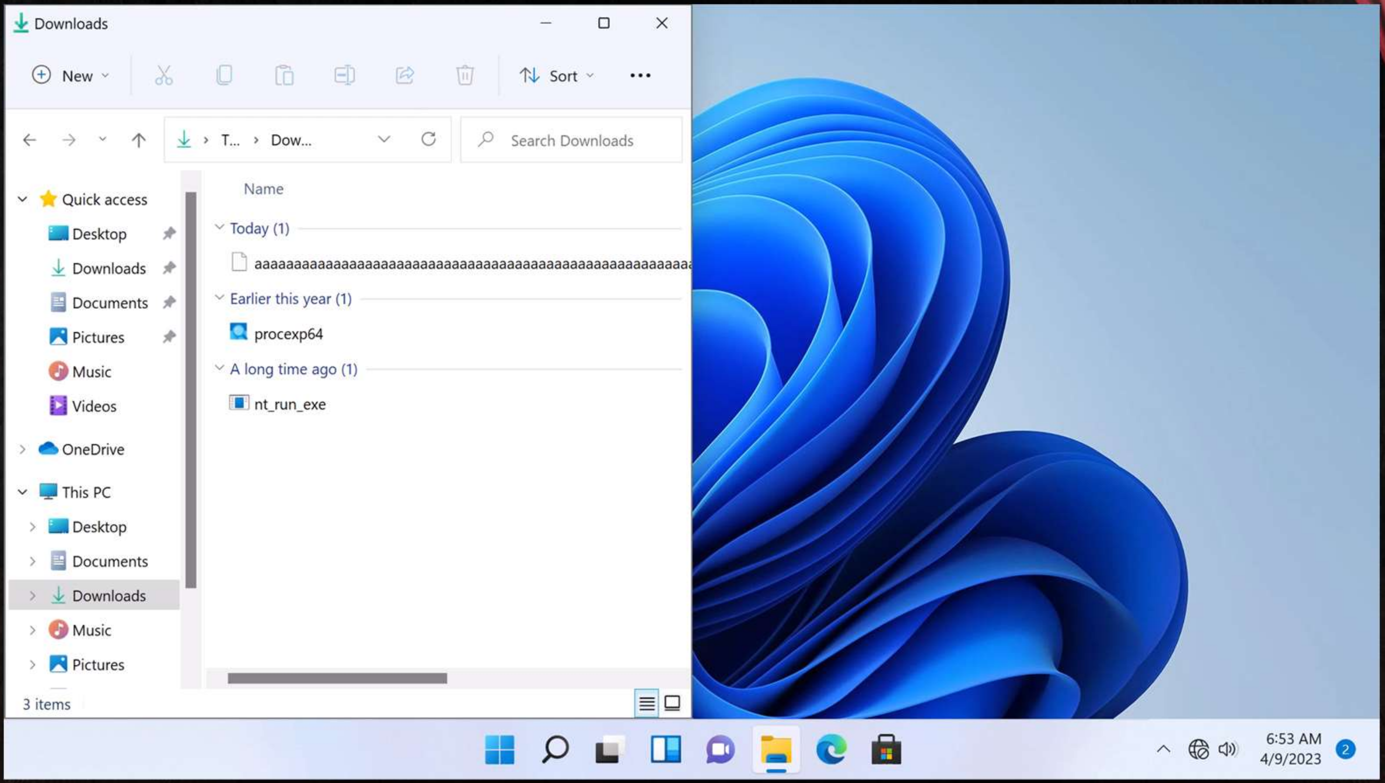
```
if ( !invalid_parameter_handler )
  invoke_watson(Expression, FunctionName, FileName, LineNo, Reserved);
return invalid_parameter_handler(Expression, FunctionName, FileName, LineNo, Reserved);
```

ProcExp DoS –
A Built In “Safe” Feature



```
01:
 wcscpy_s(process_name_with_pid_parentheses, 256ui64, process_name);
02:
 sprintf_s<32>(pid_str_with_parentheses, L"(%d)", v116[22]);
 wcscat_s(process_name_with_pid_parentheses, 256ui64, (const wchar_t *)pid_str_with_parentheses);
```

ProcExp DoS –
A Built In “Safe” Feature



Name

Today (1)

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Earlier this year (1)

procexp64

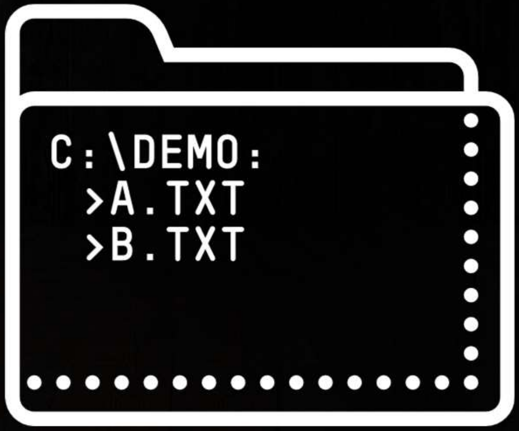
A long time ago (1)

nt_run_exe

Vulnerabilities

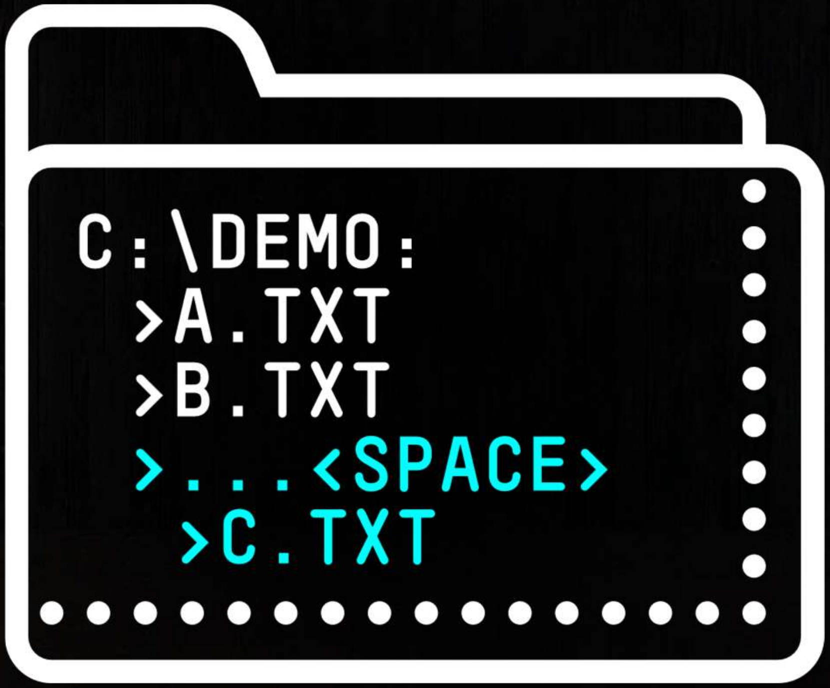
EoP Deletion Vuln – The disappearing act

- ❌ Permissions for a.txt and b.txt
- ✅ Permissions to write into C:\demo



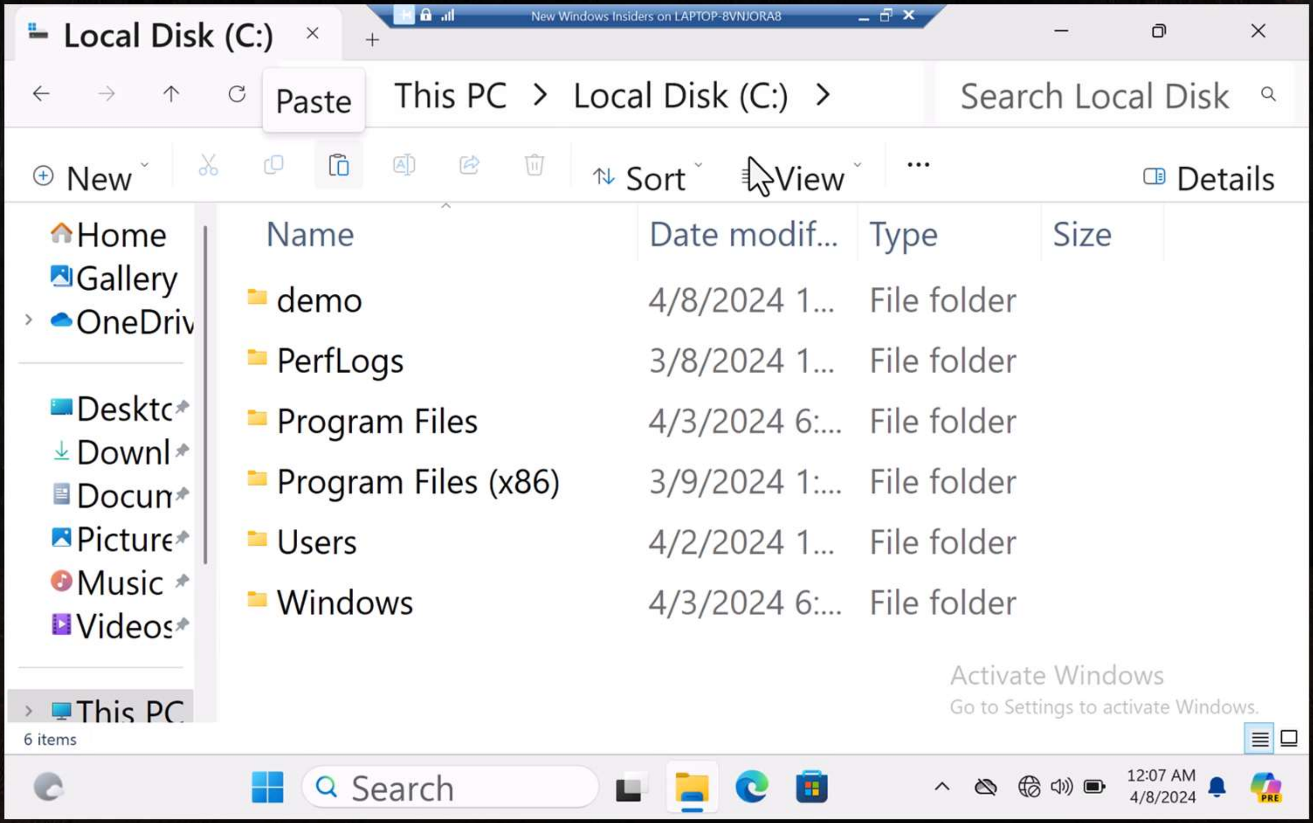
```
C:\DEMO:
  >A.TXT
  >B.TXT
```

EoP Deletion Vuln – The disappearing act



```
C:\DEMO:
  >A.TXT
  >B.TXT
  >...<SPACE>
   >C.TXT
```

EoP Deletion Vuln – The disappearing act

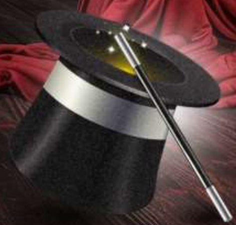


Name	Date modif...	Type	Size
demo	4/8/2024 1...	File folder	
PerfLogs	3/8/2024 1...	File folder	
Program Files	4/3/2024 6:...	File folder	
Program Files (x86)	3/9/2024 1:...	File folder	
Users	4/2/2024 1...	File folder	
Windows	4/3/2024 6:...	File folder	

EoP Deletion Vuln – The disappearing act

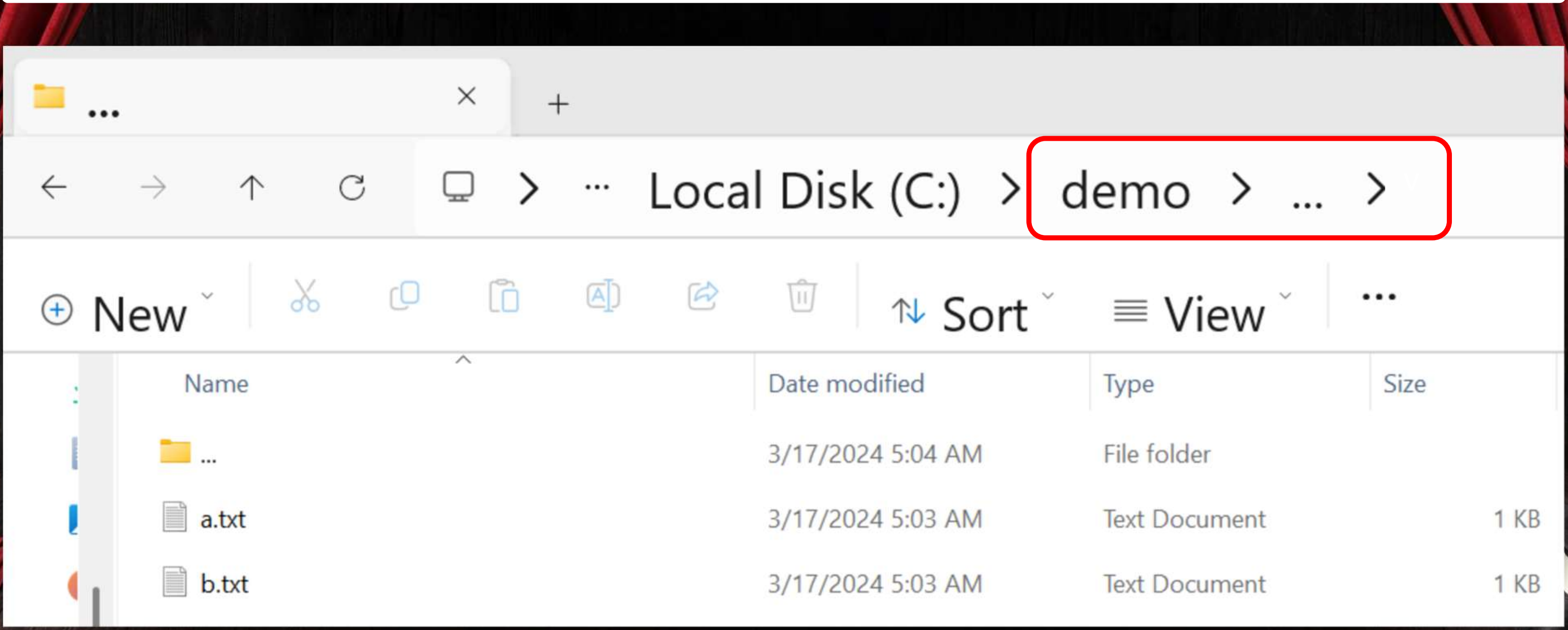
Deleting “C:\demo\...<space>”:

1. List all files inside “...<space>”



MAGIC DOT

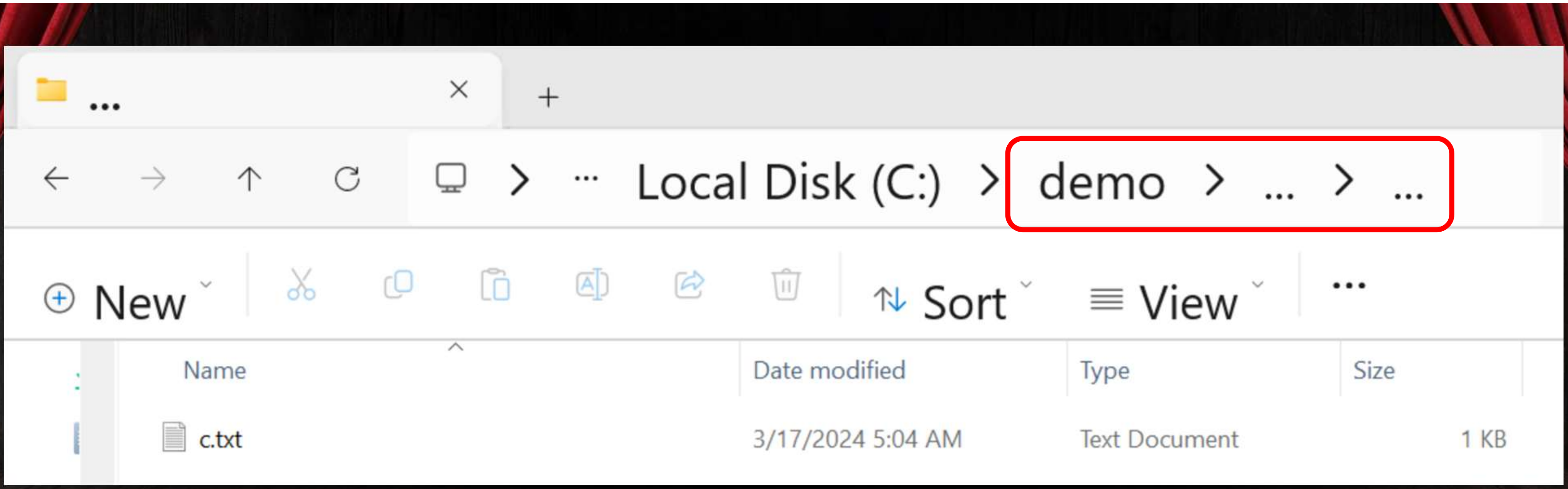
“C:\demo\...<space>\” == “C:\demo\”



Name	Date modified	Type	Size
...	3/17/2024 5:04 AM	File folder	
a.txt	3/17/2024 5:03 AM	Text Document	1 KB
b.txt	3/17/2024 5:03 AM	Text Document	1 KB

MAGICDOT

“C:\demo\...<space>\...<space>” == “C:\demo\...<space>”



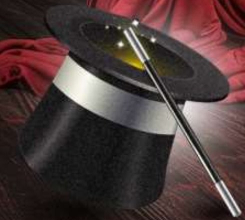
Name	Date modified	Type	Size
c.txt	3/17/2024 5:04 AM	Text Document	1 KB

EoP Deletion Vuln – The disappearing act

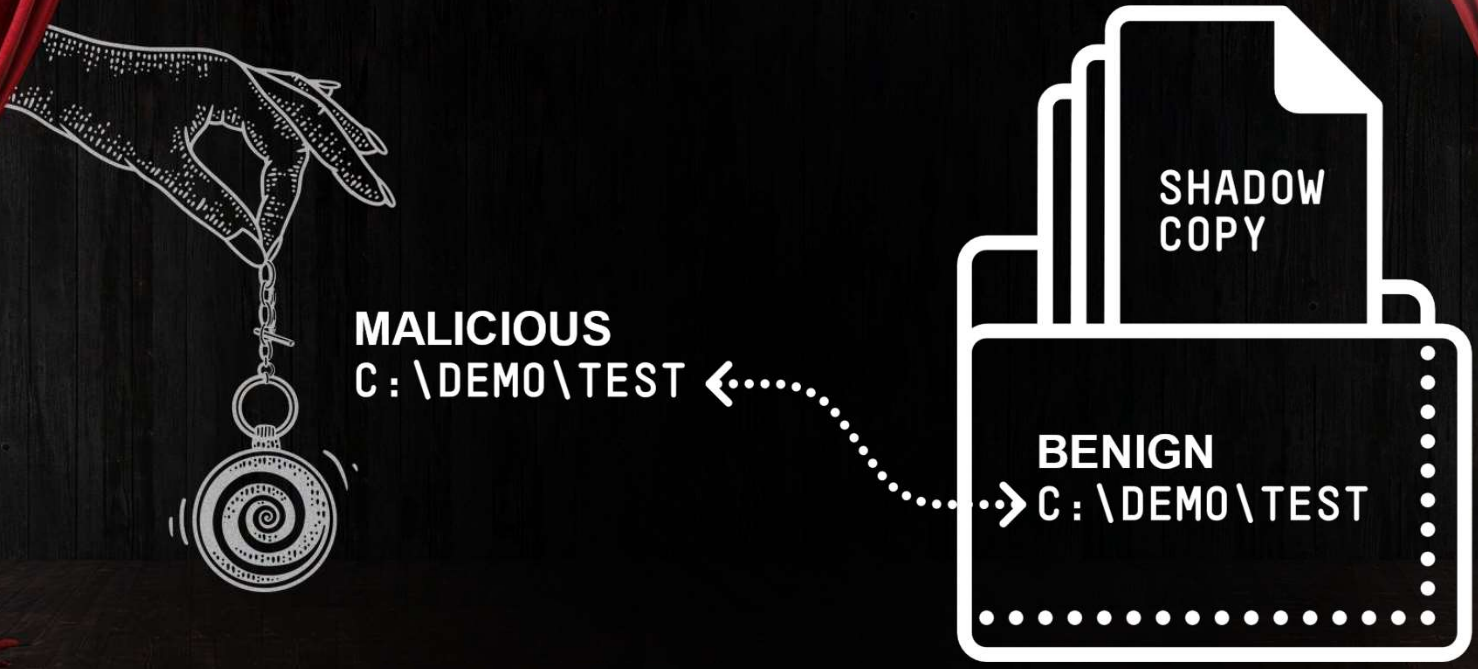
2. Delete all listed files

3. Delete the top directory:

“C:\demo\...<space>\” == “C:\demo\”



EoP Write Vuln – Changing your memories



MALICIOUS
C:\DEMO\TEST

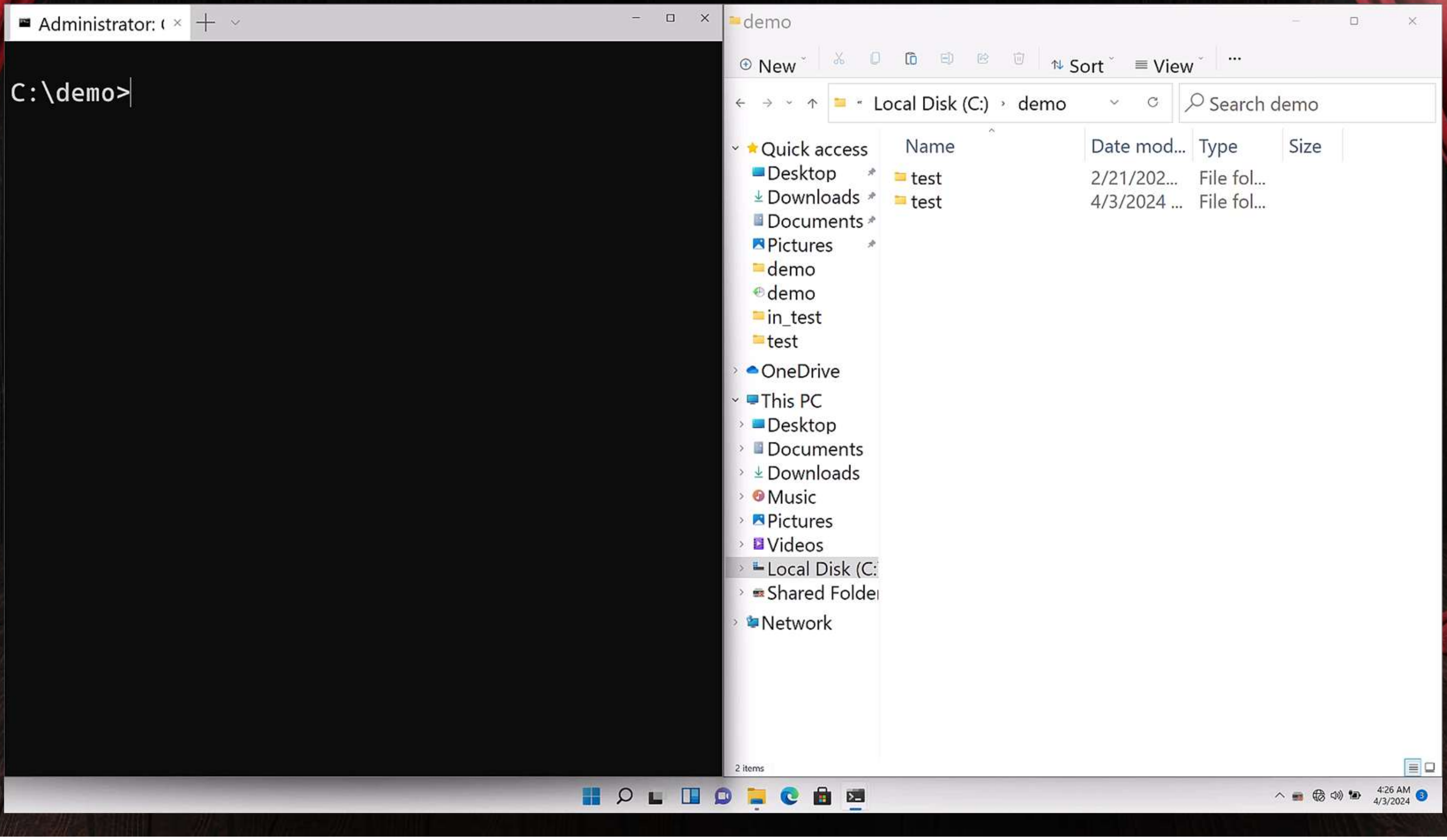
SHADOW
COPY

BENIGN
C:\DEMO\TEST

EoP Write Vuln – Changing your memories

```
C:\DEMO:
  >TEST
  >TEST<SPACE>
```

EoP Write Vuln – Changing your memories



```
C:\demo>
```

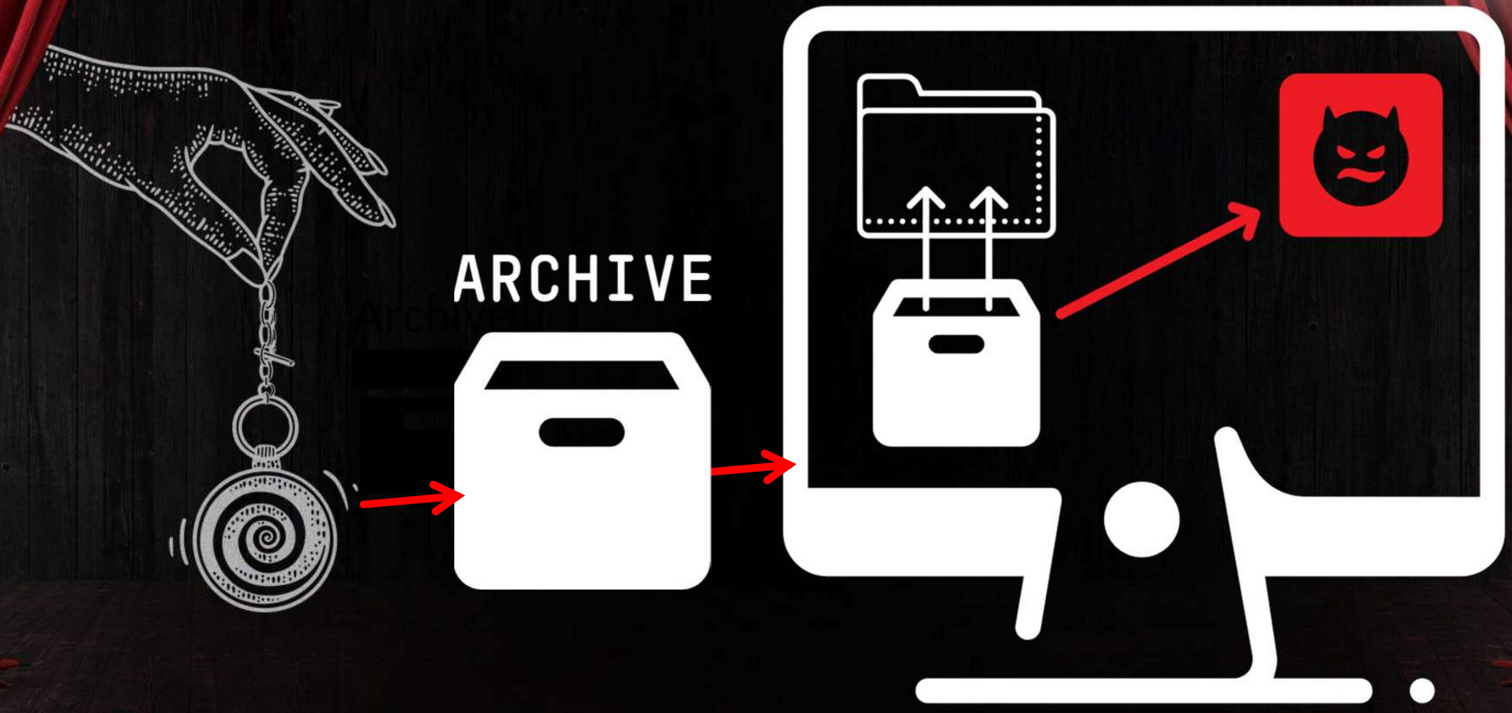
demo

Local Disk (C:) › demo

Name	Date mod...	Type	Size
test	2/21/202...	File fol...	
test	4/3/2024 ...	File fol...	

2 items

RCE Vuln – Hypnotizing Remote Computers

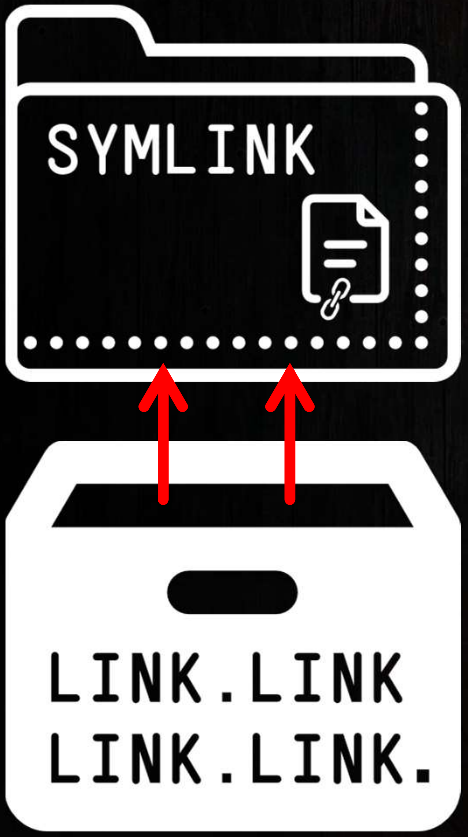


Windows 11 New Archive Types

.rar	.tar.bz2	.tbz2
.7z	.tar.zst	.tzst
.tar	.tar.xz	.txz
.tar.gz	.tgz	

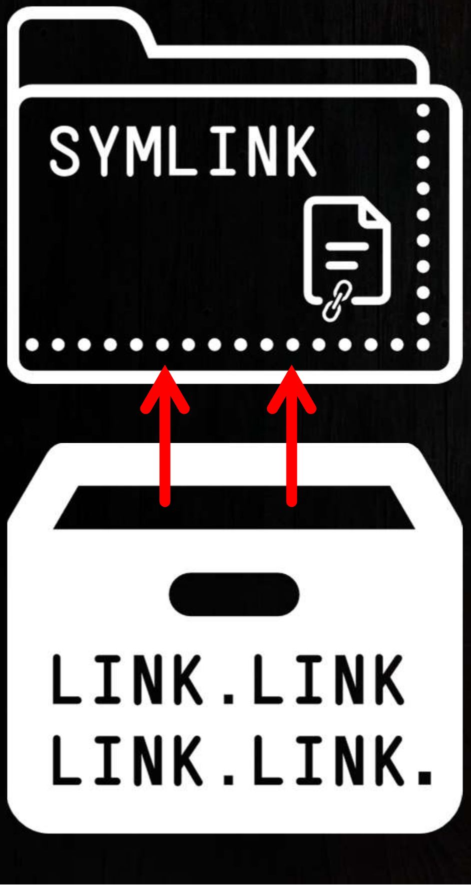
Symlinks – Extraction Vulnerabilities Lead

Is it dangerous?



Symlinks – Extraction Vulnerabilities Lead

Not really, because writing to the symlink's target is not a feature



SYMLINK

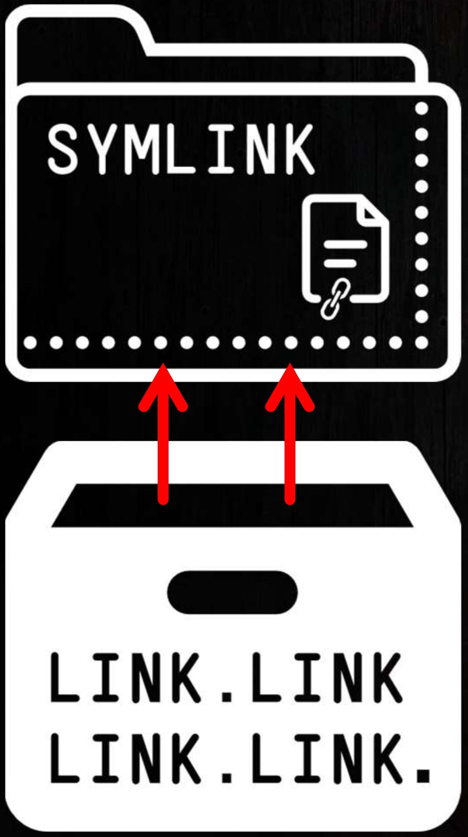
LINK.LINK
LINK.LINK.

Symlinks – Extraction Vulnerabilities Lead

“Create Symbolic Links” user right

or

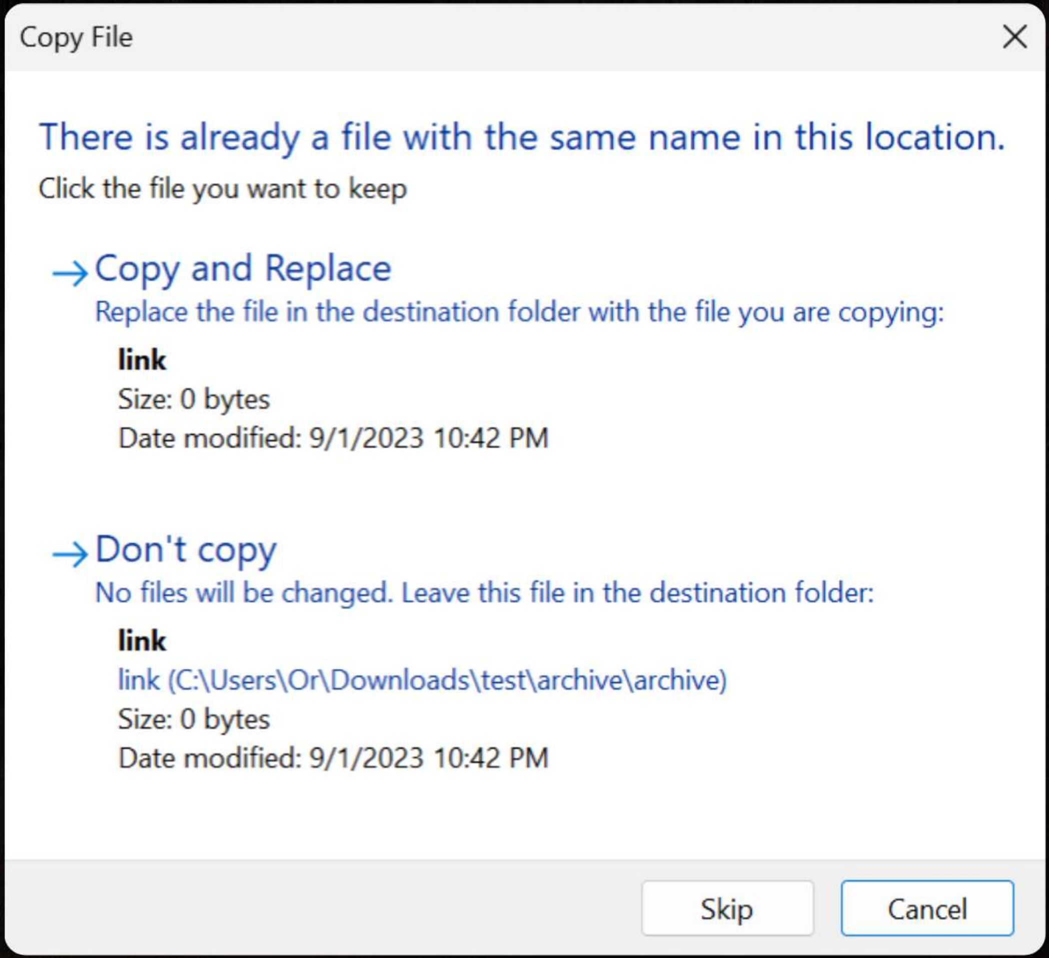
“Developer Mode”



Symlinks – Extraction Vulnerabilities Lead

LINK.LINK
LINK.LINK.

MAGIC DOT



There is already a file with the same name in this location.

Click the file you want to keep

→ Copy and Replace

Replace the file in the destination folder with the file you are copying:

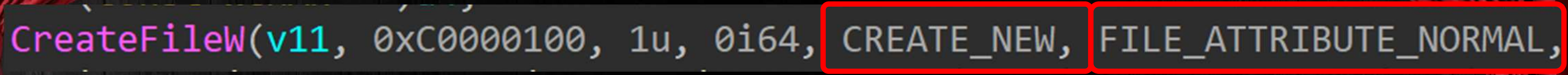
link
Size: 0 bytes
Date modified: 9/1/2023 10:42 PM

→ Don't copy

No files will be changed. Leave this file in the destination folder:

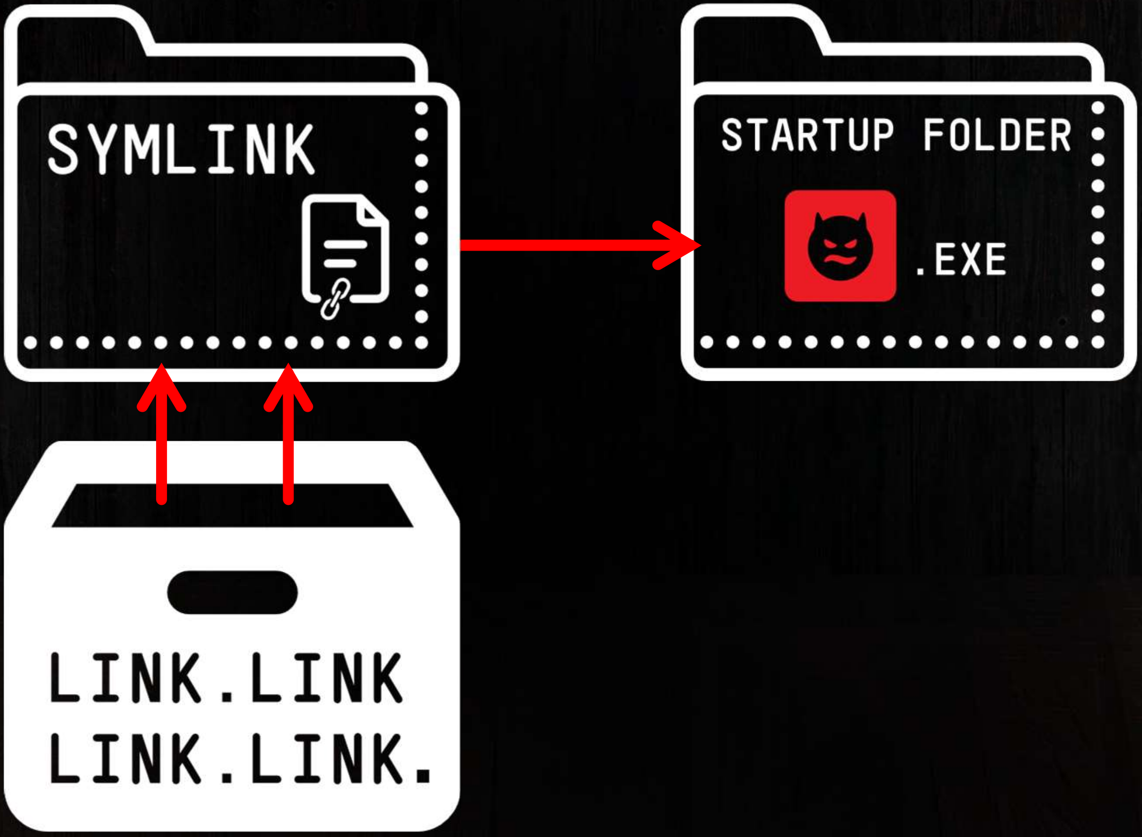
link
link (C:\Users\Or\Downloads\test\archive\archive)
Size: 0 bytes
Date modified: 9/1/2023 10:42 PM

MAGIC DOT

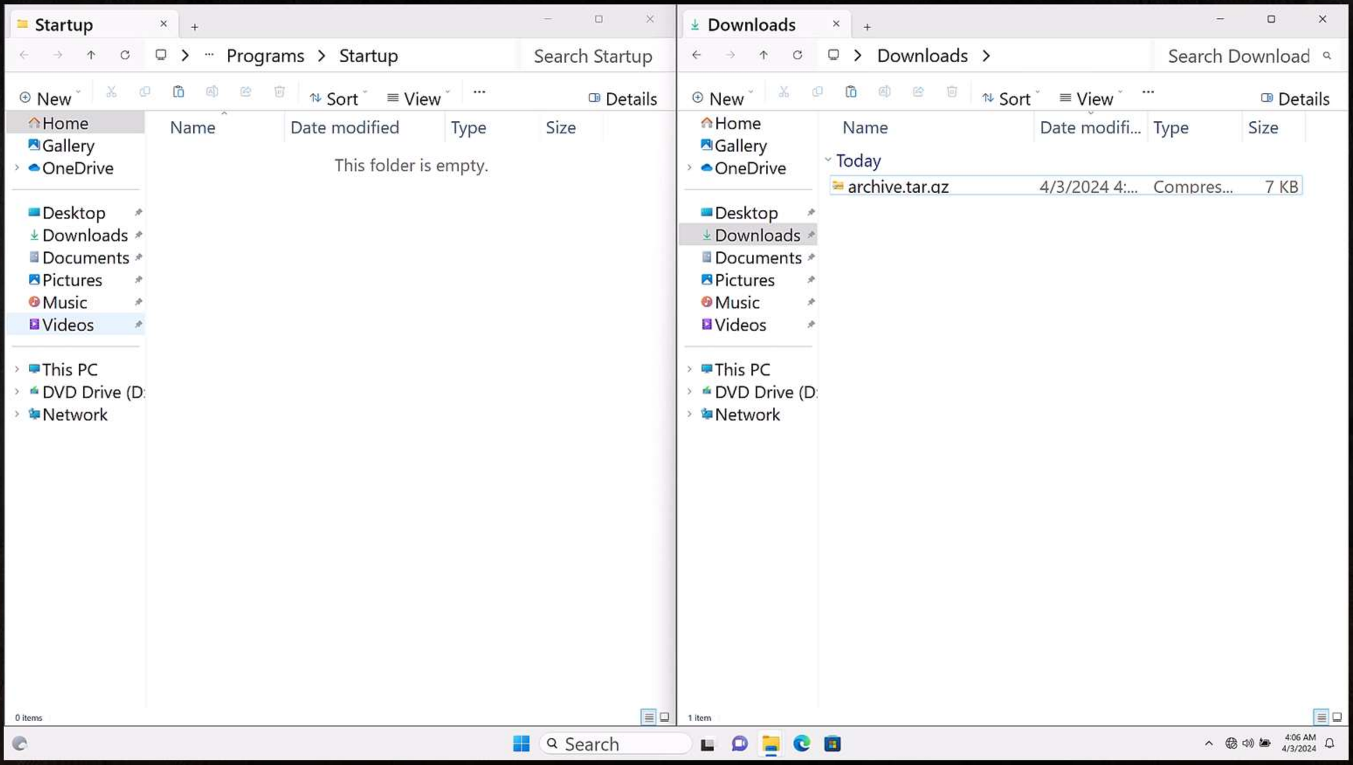


```
CreateFileW(v11, 0xC0000100, 1u, 0i64, CREATE_NEW, FILE_ATTRIBUTE_NORMAL,
```

RCE Vuln – Hypnotizing Remote Computers



RCE Demo



This folder is empty.

Today

```
archive.tar.gz    4/3/2024 4:...    Compres...    7 KB
```

CVEs and Responses

CVEs (Fixed)

Extraction RCE	CVE-2023-36396, CVSS: 7.8
Shadow Copy EoP	CVE-2023-32054, CVSS: 7.3
Process Explorer DoS	CVE-2023-42757 (Reserved)

Unfixed

Deletion EoP

“Thank you again for submitting this issue to Microsoft. We determined that this issue does not require immediate security service but did reveal unexpected behavior. A fix for this issue will be considered in a future version of this product or service.”

MagicDot Post-Exploitation Techniques

“We have assessed this issue as not a security vulnerability. One reason for that is that no security boundary is crossed. This issue is a post exploitation technique an attacker might leverage once they have already compromised the target machine.”

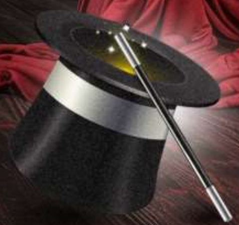
Takeaways

Backwards compatibility & known issues create security risks

Malware can be completely hidden without admin privileges

More DOS-to-NT path conversion vulnerabilities

Use NT paths instead of DOS paths



MagicDot GitHub + Q&A

@oryair1999

https://www.linkedin.com/in/or-yair

or.yair@safebreach.com

https://github.com/SafeBreach-Labs/MagicDot