A Decade After Stuxnet's Printer Vulnerability

Printing is still the Stairway to Heaven

Peleg Hadar Senior Security Researcher **&** **Tomer Bar** Research Team Lead | SafeBreach LABS

SafeBreachLABS

Peleg Hadar

Senior Security Researcher

- 7+ years in InfoSec
- Senior Security Researcher @ SafeBreach Labs
- Main focus in Windows internals and vulnerability research
- @peleghd

SafeBreachLABS

Tomer Bar

Research Team Lead

- 15+ years in Cyber Security
- Research Team Lead @ SafeBreach Labs
- Main focus in APT and vulnerability research
- Past publications:
  - Prince of Persia - Terminating 10 Years Campaign For Fun And Profit
  - Infy Malware Active In Decade Of Targeted Attacks
  - KasperAgent and Micropsia - Targeted Attacks In The Middle East
  - Ride The Lightning With Foudre
  - Double Edge Sword Attack - Exploiting Quasar Rat Command and Control
  - BadPatch (APT-C-23)

Agenda Is Stuxnet 2.0 possible?

- **Analysis of Stuxnet's propagation capabilities (vulnerabilities)**
  - Root Cause
  - Patch
  - Re-Exploitation / Equivalent newer vulnerability in the same component
- **Our Research**
  - How did we re-exploited a patched 10 years old MS Windows vulnerability
  - Demonstration of 2 unpatched 0-day vulnerabilities (Pre-coordinated with Microsoft)
- **Mitigations and Suggestions**
  - Better Patch
  - Better real-time prevention for an entire bug class

Agenda two main takeaways

Stuxnet 2.0

Is it possible to re-occur?

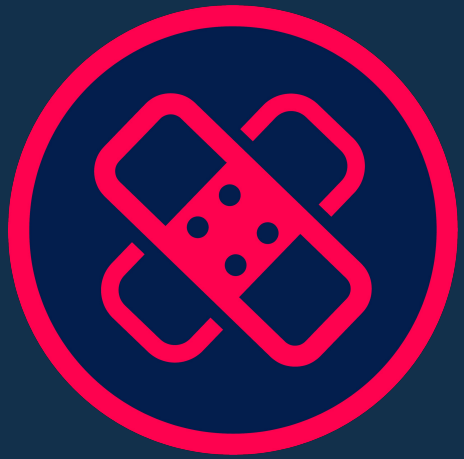
Patch effectiveness

Is it possible to abuse patched vulnerabilities?

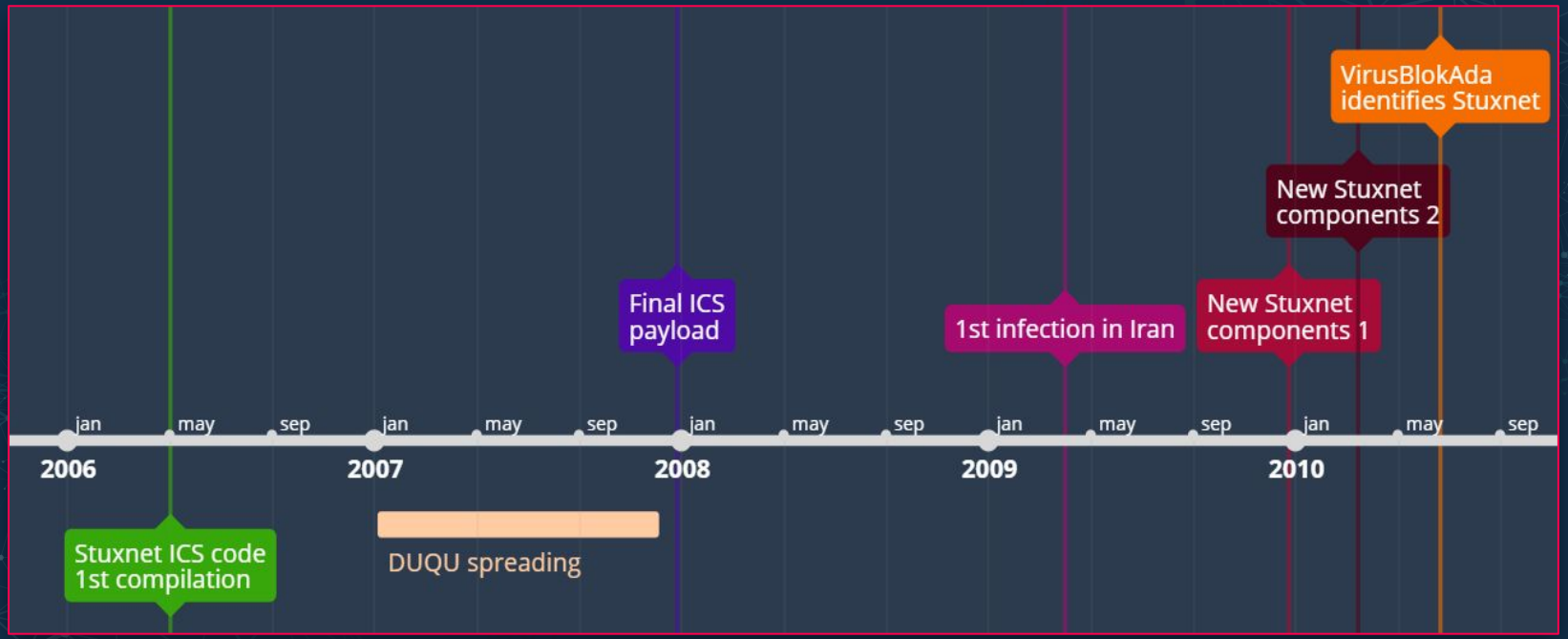
Terminology

Narrow Patch

Patch



Stuxnet Recap & Timeline



Stuxnet As Seen in “0 Days”

Stuxnet Main Building Blocks

Propagation Capabilities

5 Vulnerabilities

3 RCE | 2 LPE

Evasion Capabilities

Rootkit

Stolen Certificate

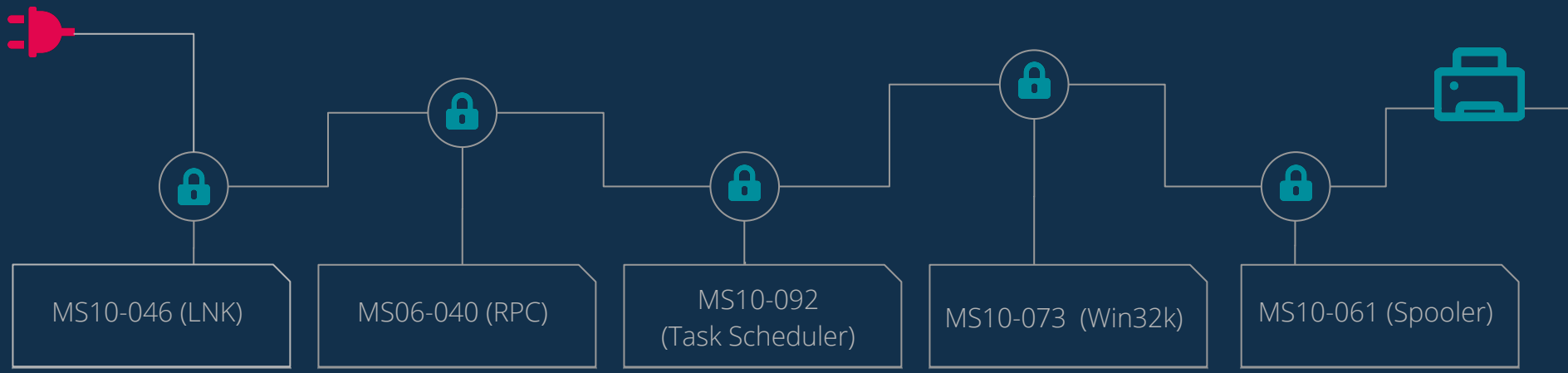
ICS Capabilities

ICS Target Detection

Siemens Related Actions

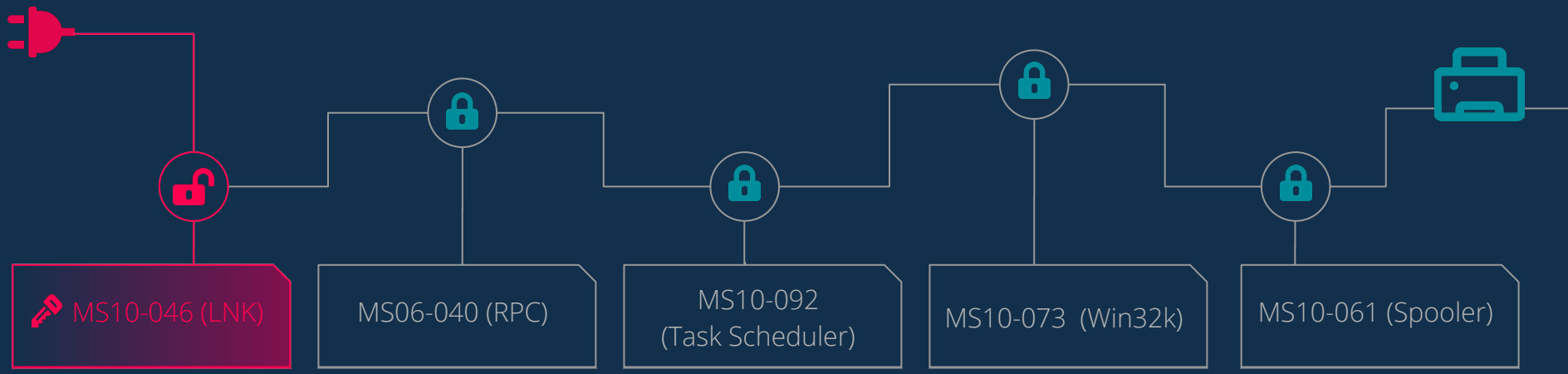
Final Payload

Spooler Propagation Capabilities

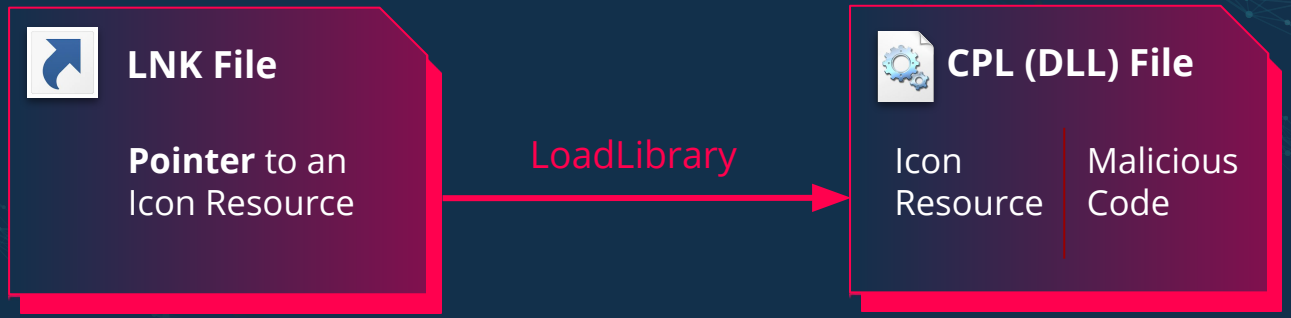


"Now, over 22 million pieces of malware use that blueprint to attack organizations and states..." *-regdox.com*

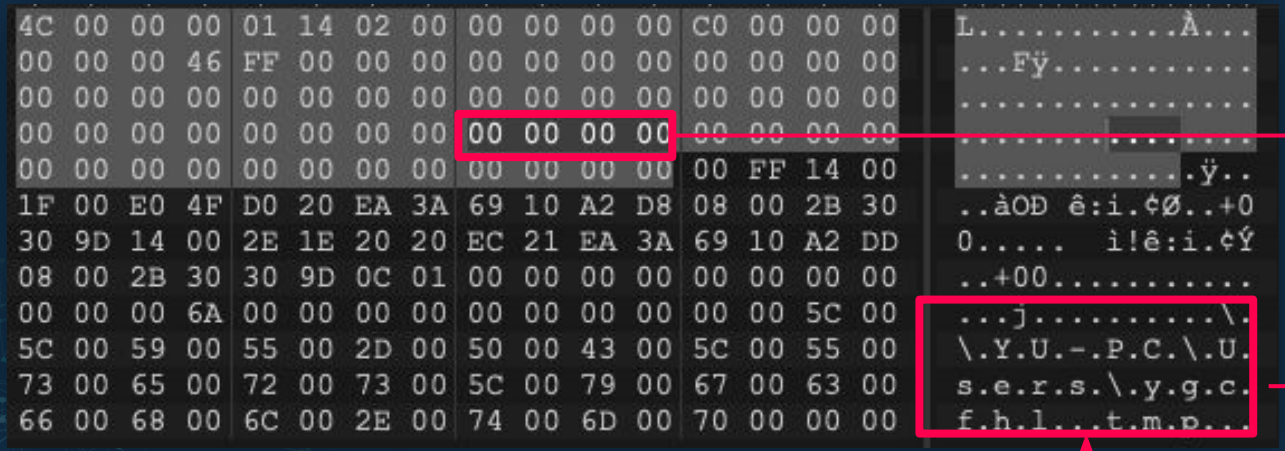
Spooler Propagation Capabilities



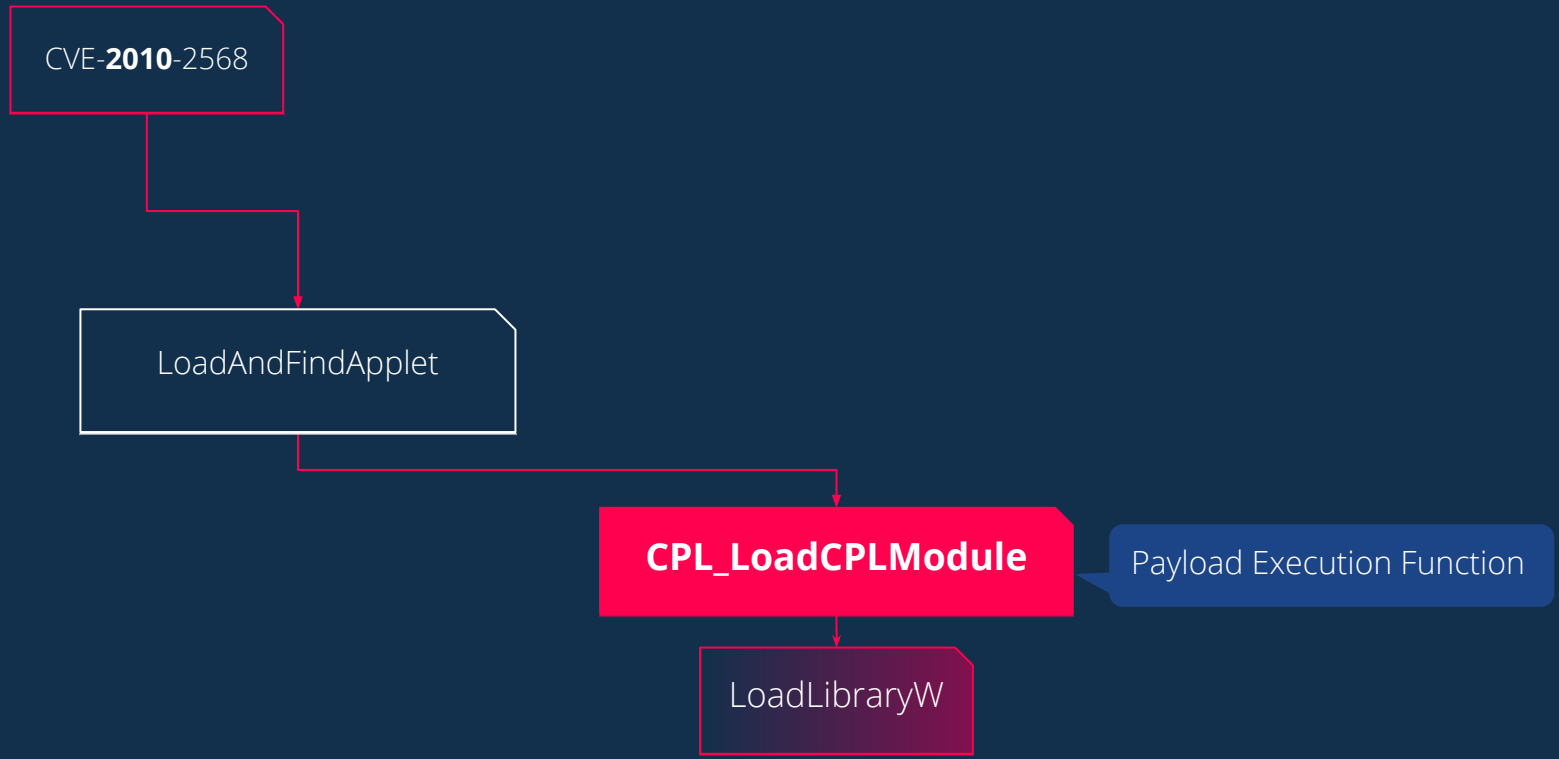
LNK Stuxnet's 0-day - Root Cause



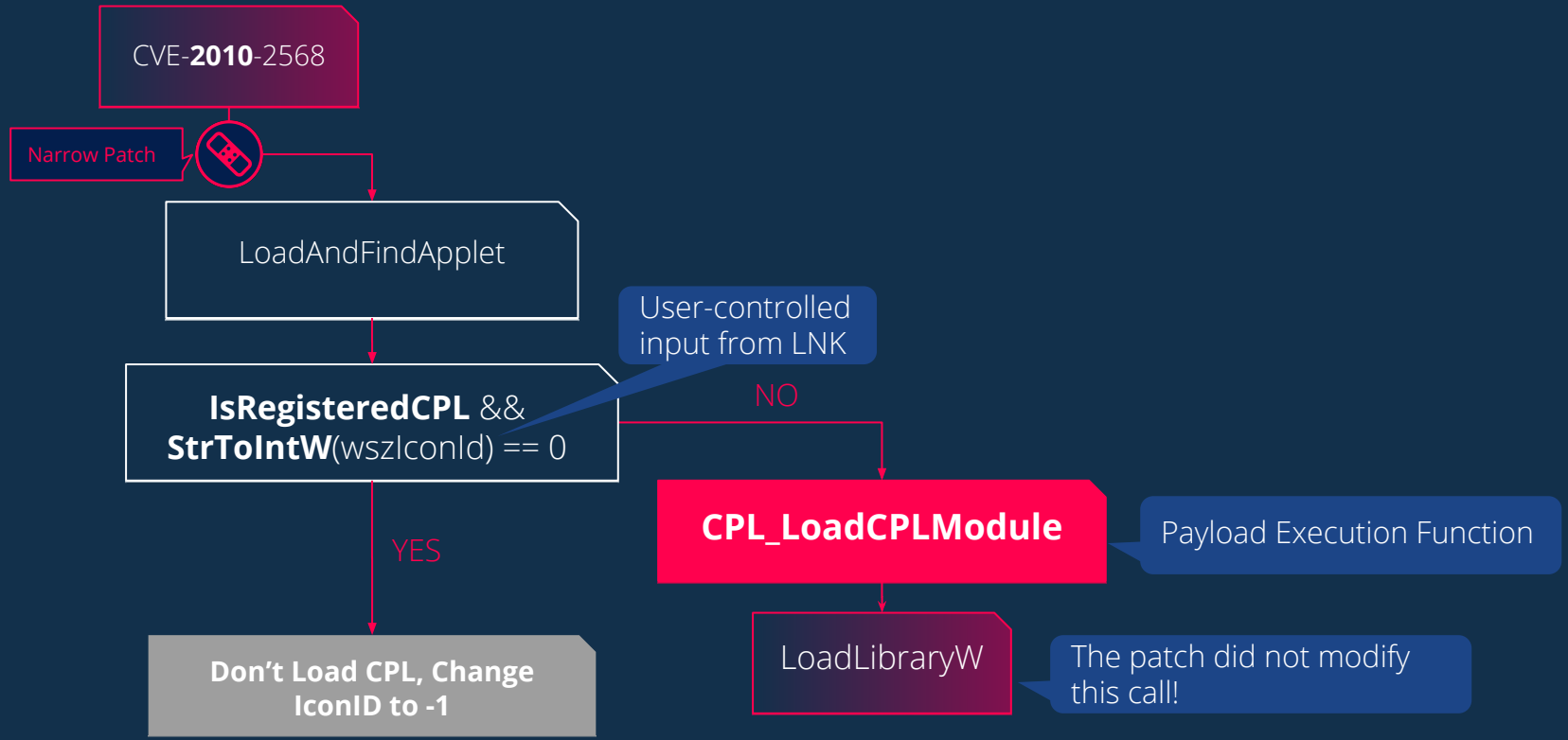
LNK Stuxnet's 0-day - Exploitation



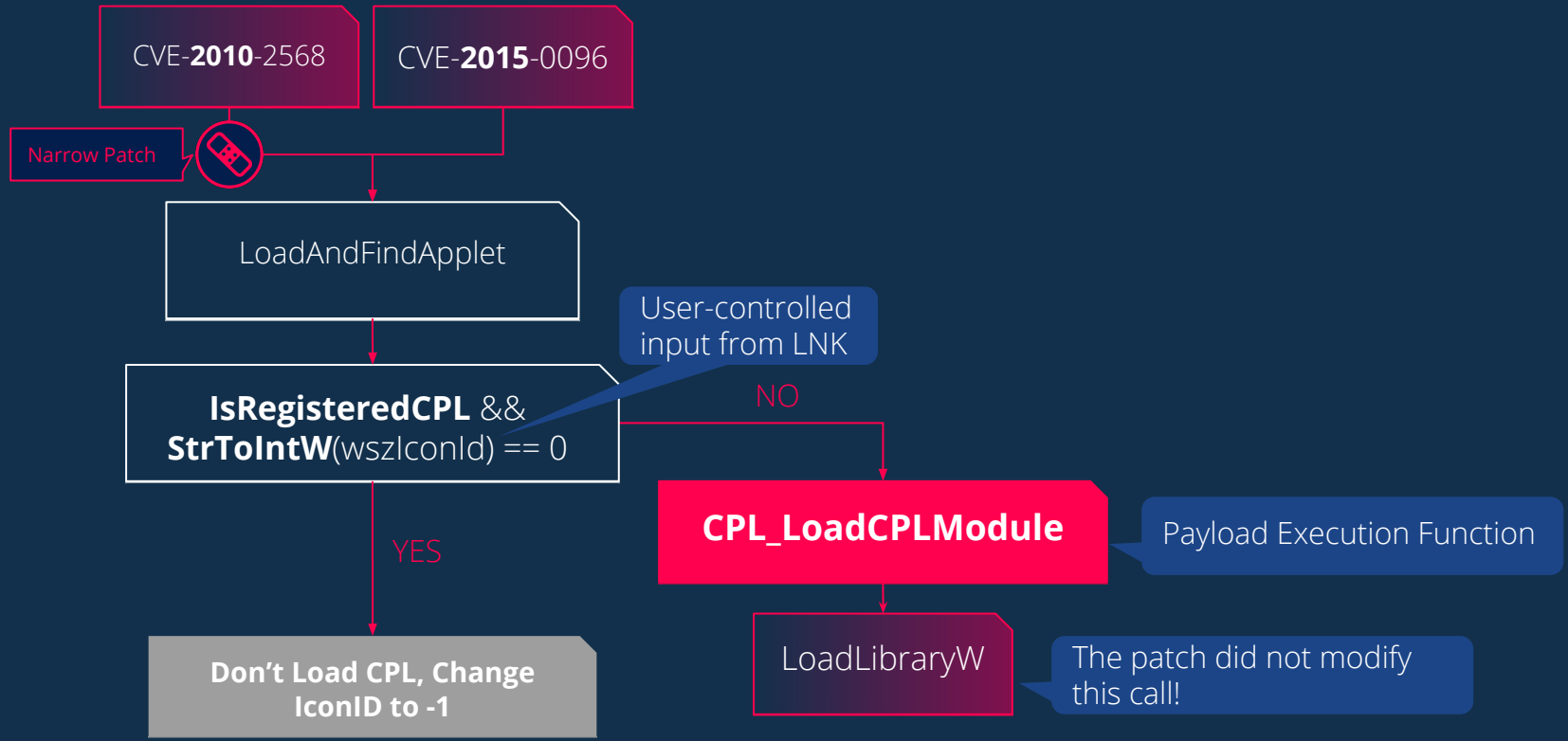
LNK 0-Day Exploitation Paths Overview



LNK MS10-046 Patch



LNK 0-Day Exploitation Paths Overview



CVE-2015-0096 Patch Bypass

Truncated to 260 Wide Chars

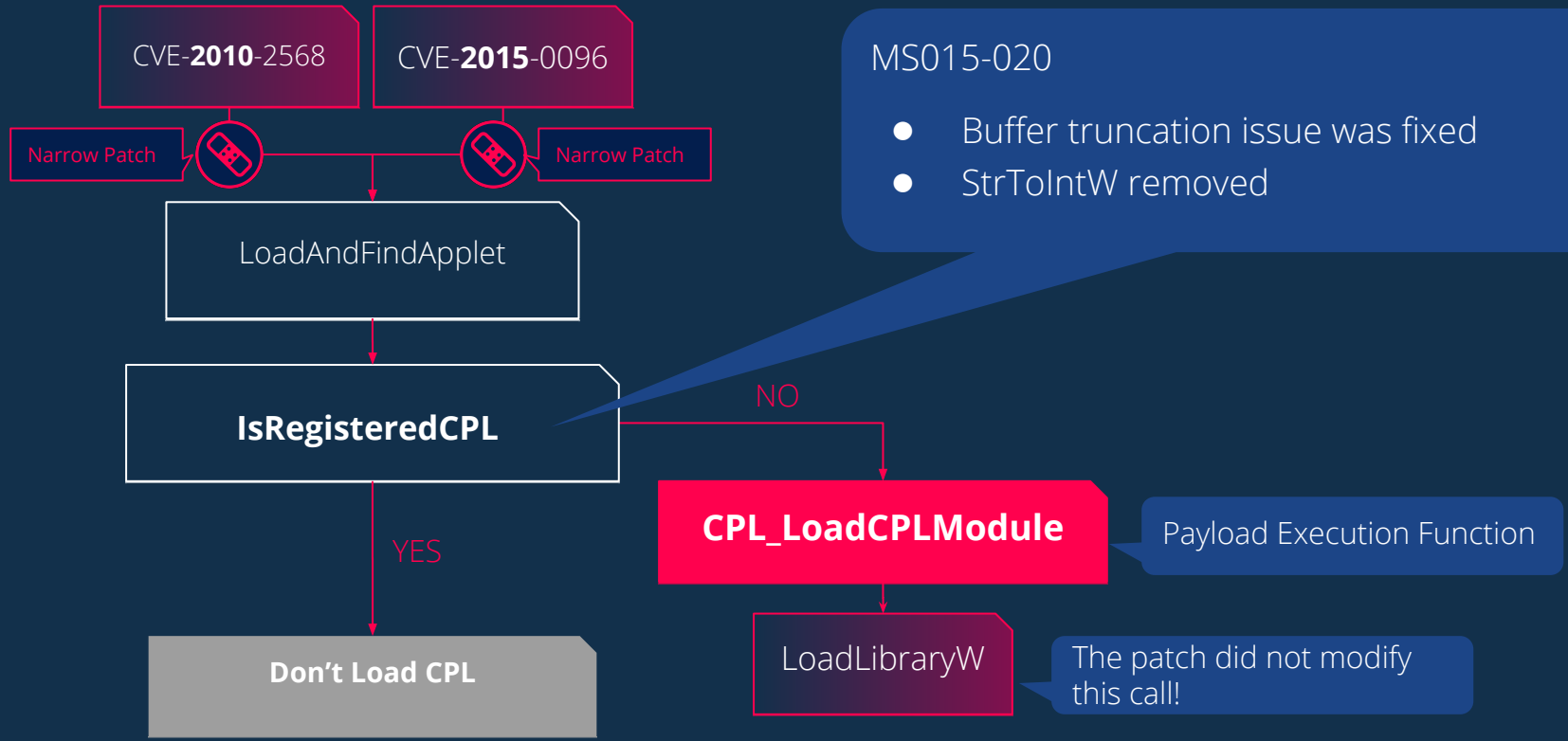
[c:\Ma.dll,-1,AA...AAA\0]

554 Wide Chars

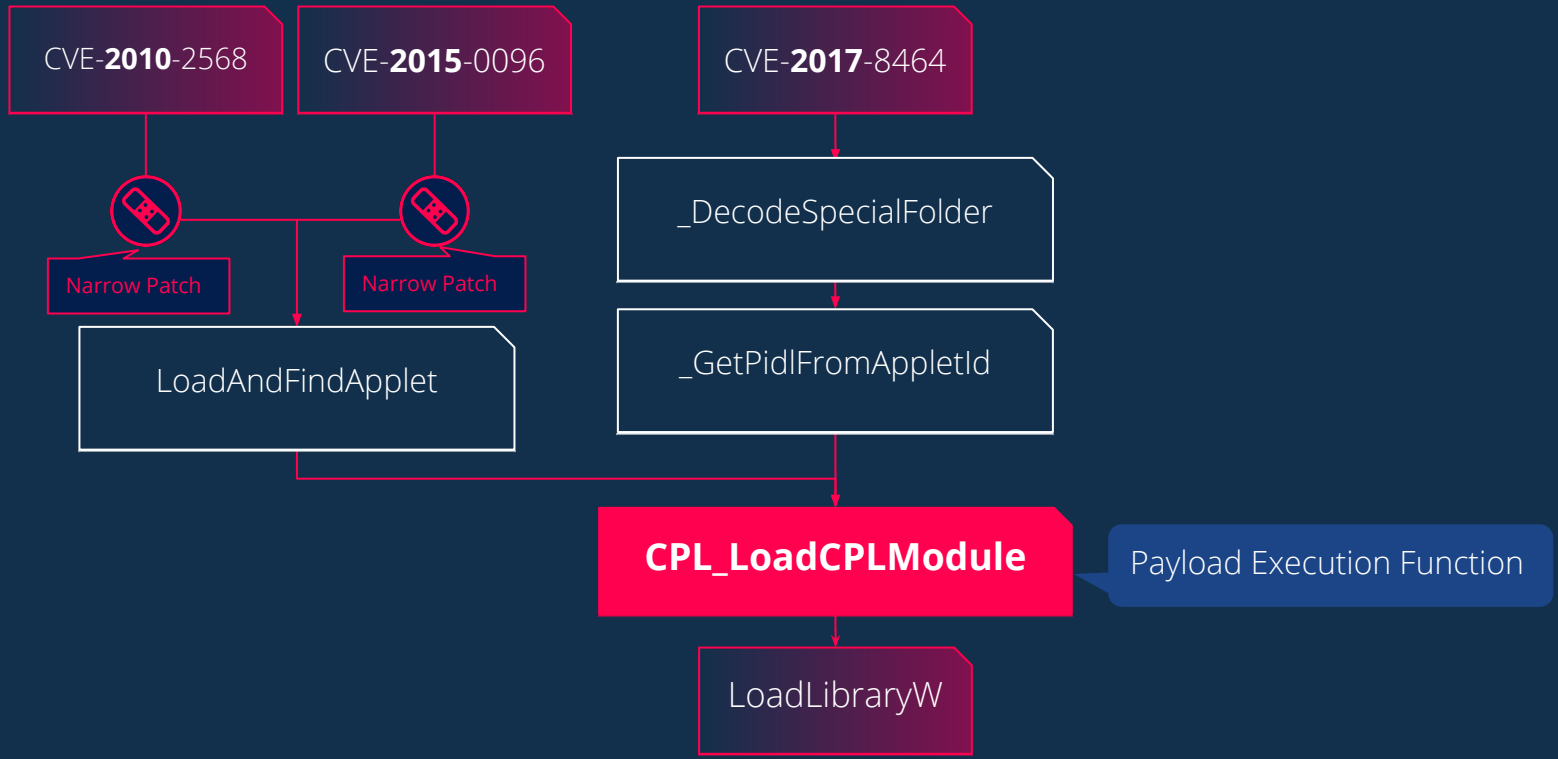
[c:\Ma.dll,-\0]

int dwIconId = StrToIntW(L"**-**")
dwIconId will be 0

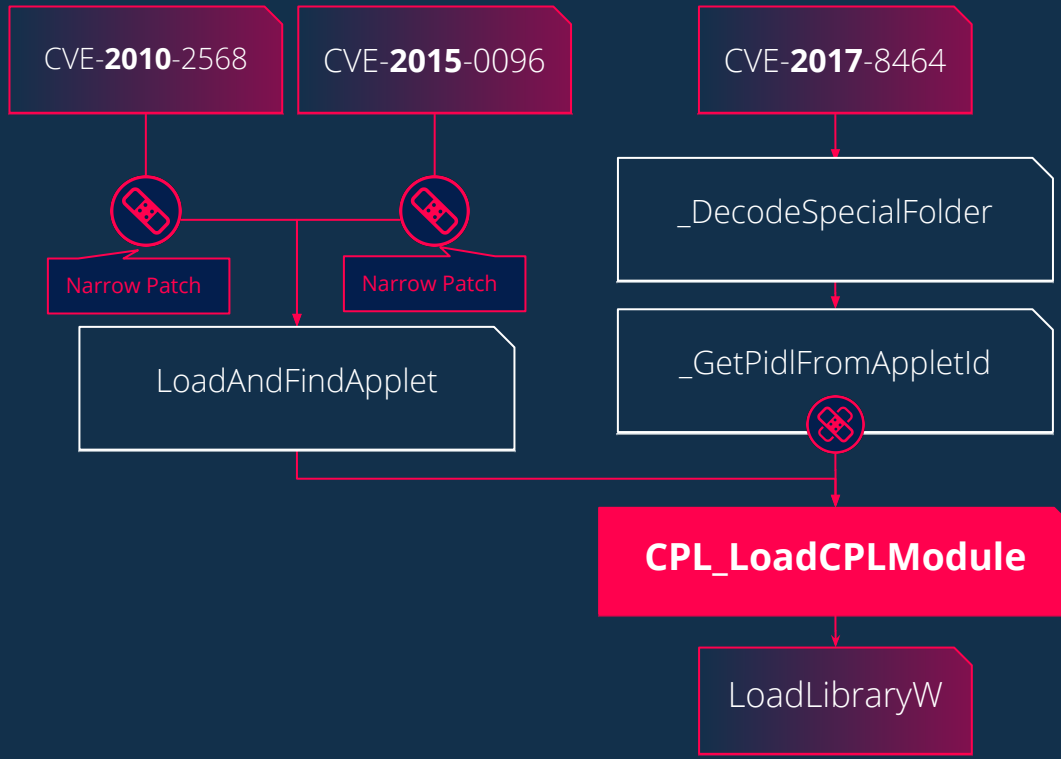
LNK 0-Day Exploitation Paths Overview



LNK 0-Day Exploitation Paths Overview



LNK 0-Day Exploitation Paths Overview

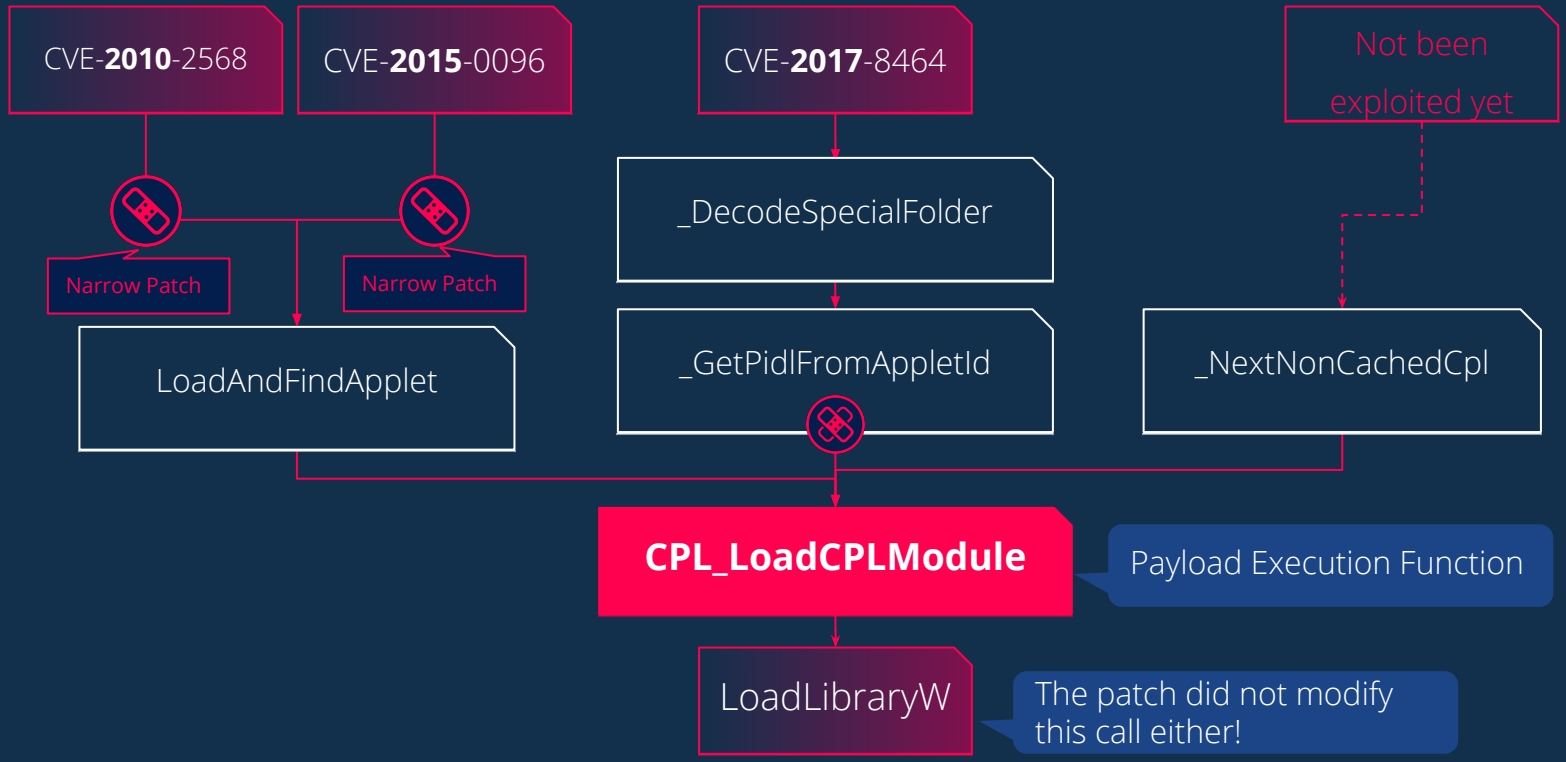


CVE-2017-8464 - Patch

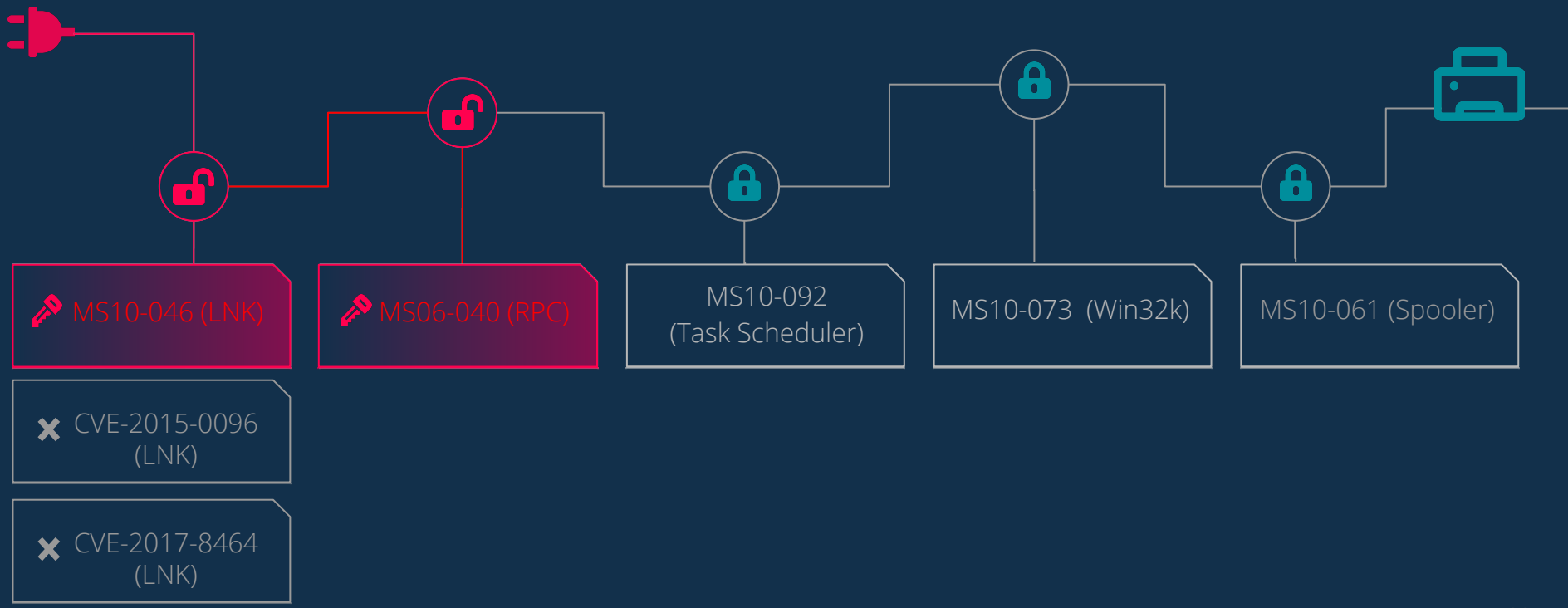
- Added previous validation to validate if CPL is registered

Payload Execution Function

LNK 0-Day Exploitation Paths Overview



Spooler Printing our Way to SYSTEM



RPC

2006

MSRC - 1st Vulnerability - **Limited Scope**
"Very limited, targeted attacks"

> As a reminder, Microsoft is aware of very limited, targeted attacks that exploited the vulnerability prior to the release of the update, but we're not currently seeing broad attacks that use this newly posted exploit code
>
> *~Microsoft Security Response Center*

2009

Wide spread - The same vulnerable dll was exploited By Stuxnet & Conficker Worm

Conficker HeatMap

http://mapscroll.blogspot.com/2009/04/mapping-conficker-worm.html

RPC Path Canonical path

Path Canonization

absolute path: canonical path:
C:\xxx\..\abc\file.txt ----> C:\abc\file.txt

It allows textual comparison of two different representation of the same canonical path

C:\xxx\..\abc\xxx\..\file.txt == C:\xxx\..\abc\file.txt == C:\abc\file.txt

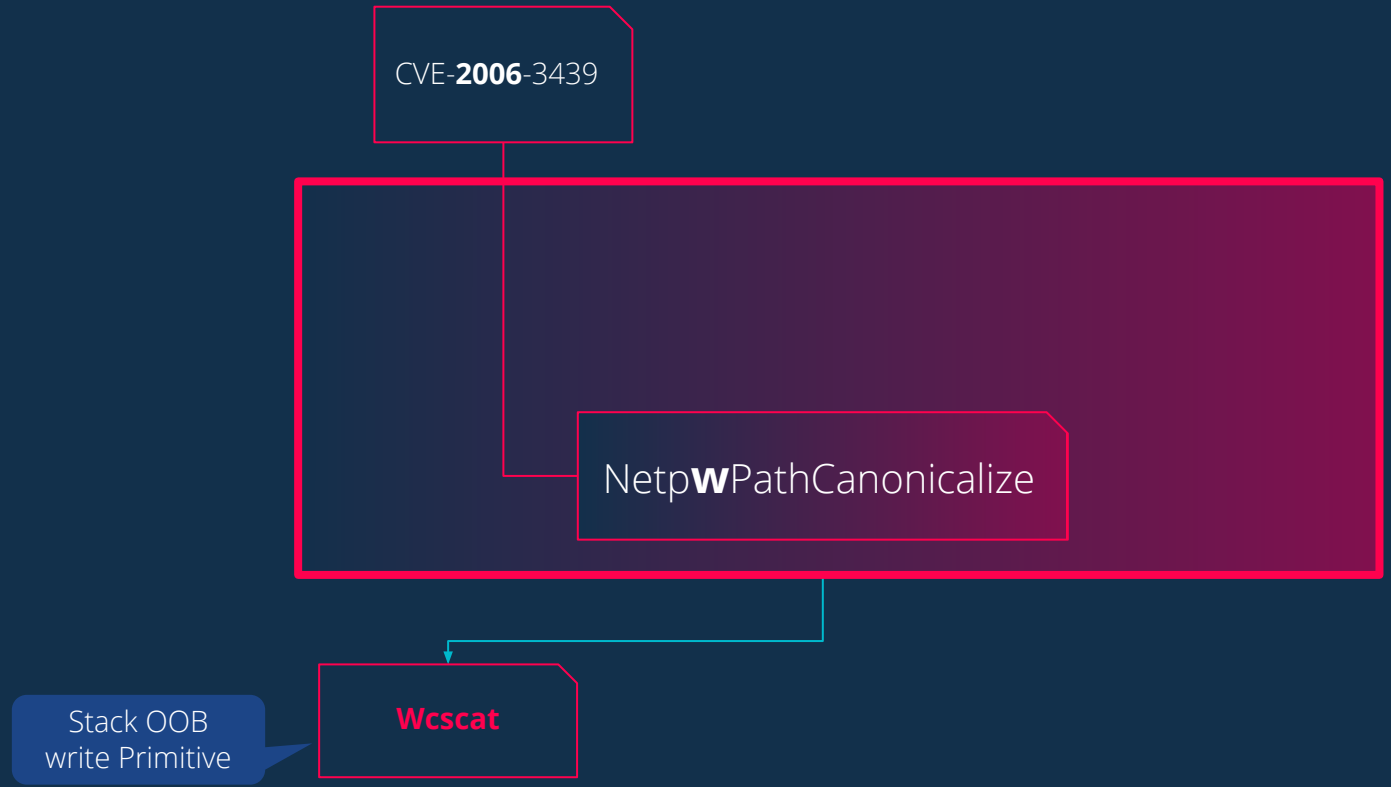
RPC Root Cause - CVE-2006-3439

CVE-2006-3439 - Old school stack based buffer overflow

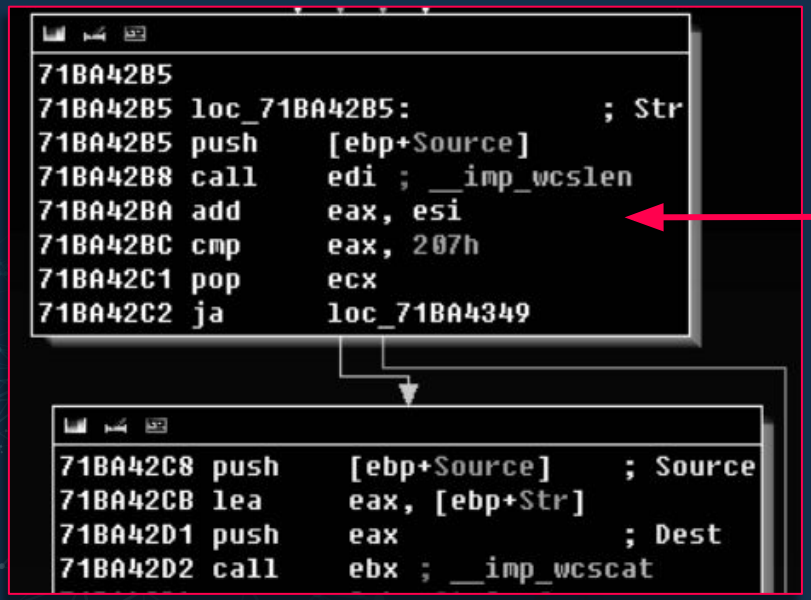
The vulnerable function allocates 0x414 bytes of space, **but** limits the length of the Path to 0x411 **Unicode** chars (0x822 bytes).


```
netpwpathcanonicalize(
_in_ DWORD Unicode_path_ptr_second_half,
_out_ DWORD lpwidecharstr,
_in_ DWORD Size,,
_in_ DWORD Unicode_path_ptr_first_half,
_in_out_ DWORD long_ptr_ptr,
_in_ DWORD flag_bit
);
```

RPC - Exploitation Paths Overview

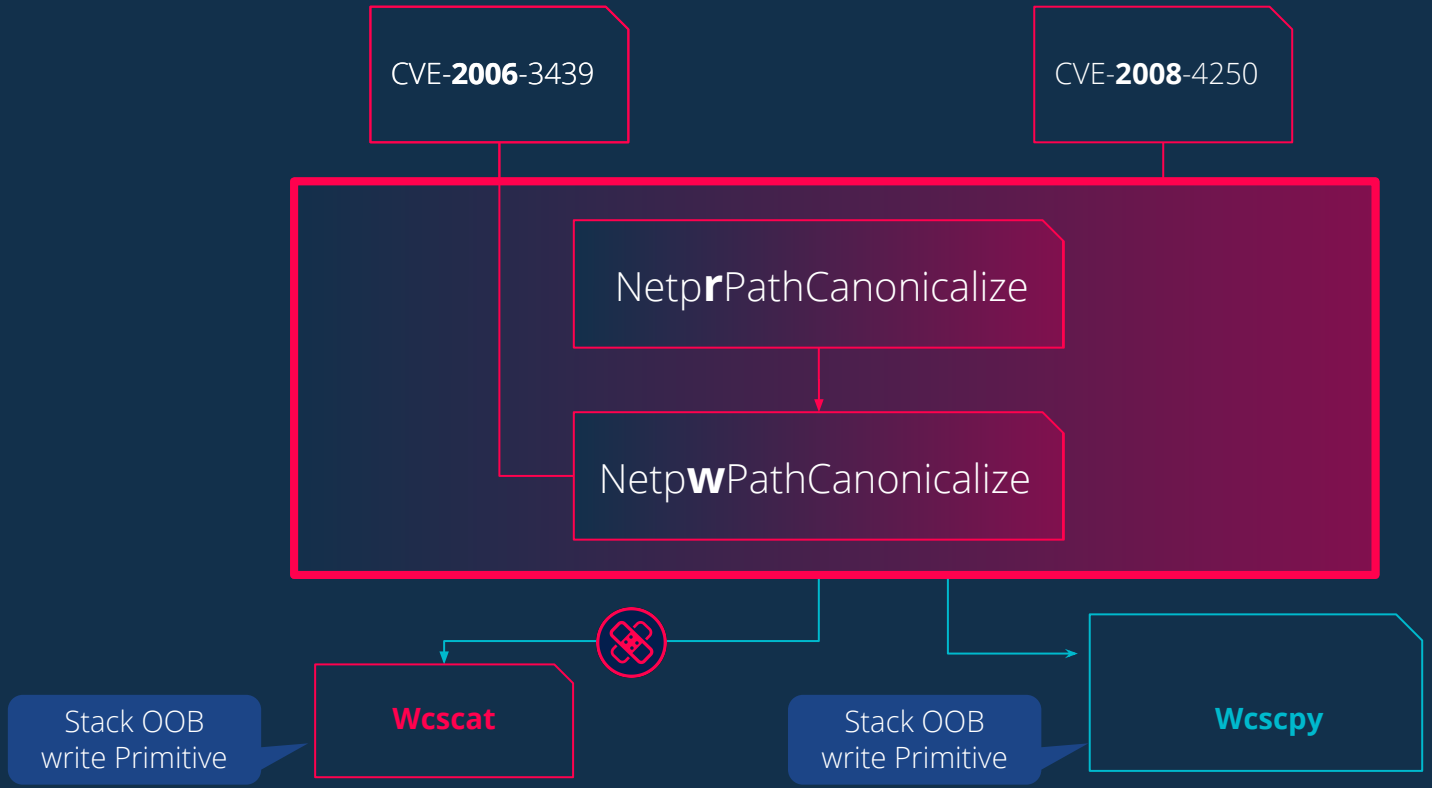


MS06-040 Patch

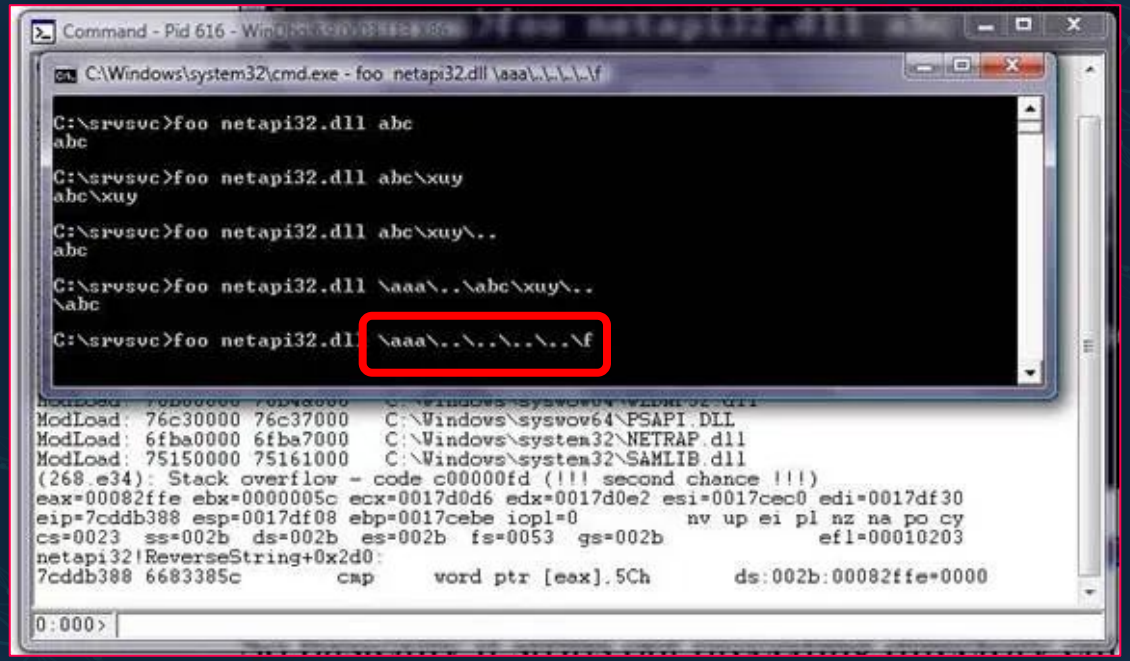


1. Check if **path** length is more than 0x207
2. Omit the **wcscat** function call

RPC Exploitation Paths Overview

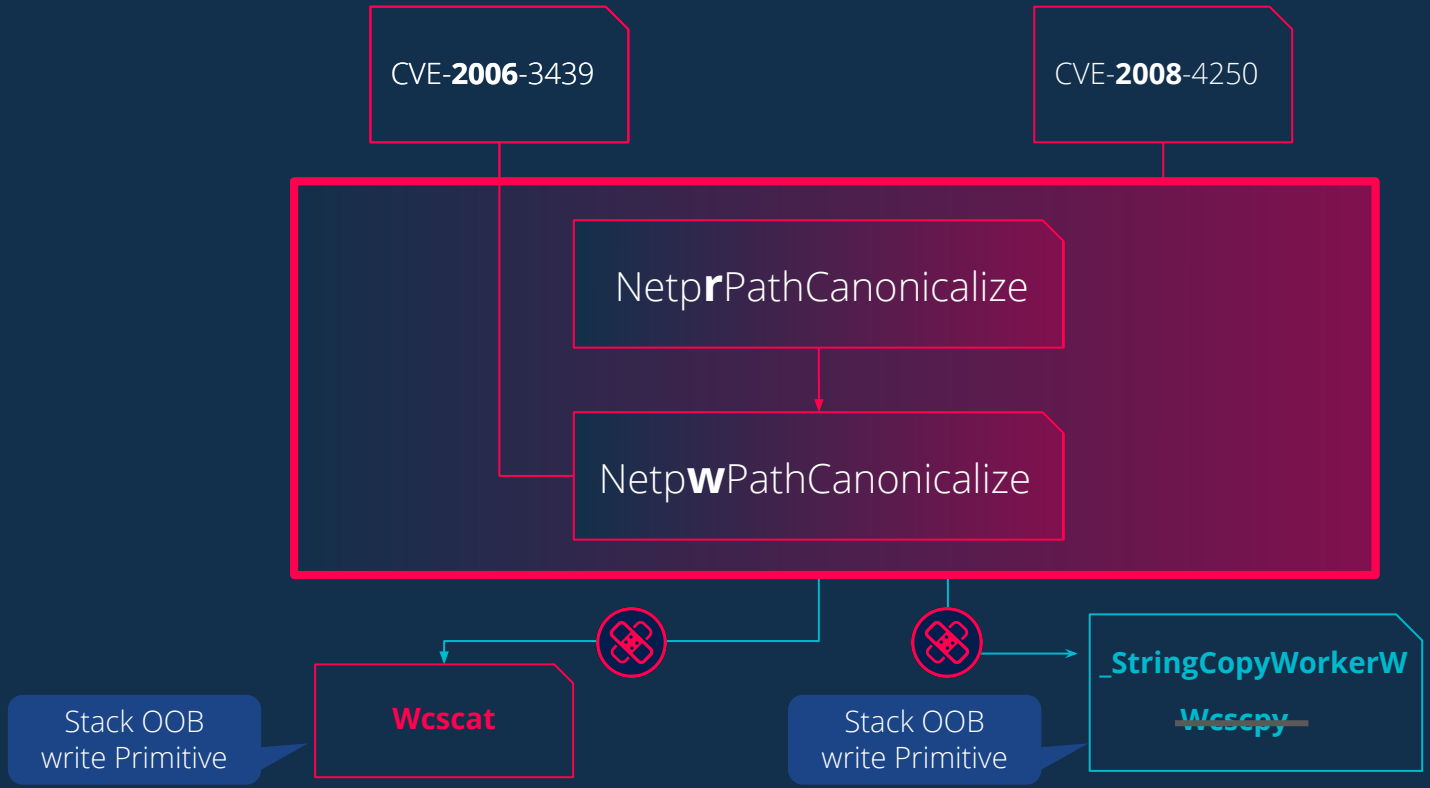


RPC Exploitation Paths Overview



https://dontstuffbeansupyournose.com/2008/10/23/looking-at-ms08-067/

RPC The Patch - MS08-067



Task Scheduler LPE - CVE-2010-3338 - Root Cause

The Patch - MS10-092

Microsoft has implemented a 2nd integrity check SHA-256 using ComputeHash function.

A registered job

```
<Principals>
  <Principal id="LocalSystem">
    <UserId>S-1-5-18</UserId>
    <RunLevel>HighestAvailable</RunLevel>
  </Principal>
</Principals>
<Actions Context="LocalSystem">
  <Exec>
    <Command>C:\WINDOWS\NOTEPAD.EXE</Command>
    <Arguments></Arguments>
  </Exec>
</Actions>
```

A crafted job with a forged CRC32

```
- <Principals>
  - <Principal id="LocalSystem">
      <UserId>S-1-5-18</UserId>
      <RunLevel>HighestAvailable</RunLevel>
    </Principal>
  </Principals>
- <Actions Context="LocalSystem">
  - <Exec>
      <Command>C:\WINDOWS\NOTEPAD.EXE</Command>
      <Arguments />
    </Exec>
  </Actions>
</Task>
<!-- 陀结  -->
```

MALICIOUS.EXE

The xml command is modified to execute the malicious code

Added bytes will change back the CRC32 value to bypass the integrity check

Source: https://aroundcyber.files.wordpress.com/2012/11/stuxnet_under_the_microscope.pdf

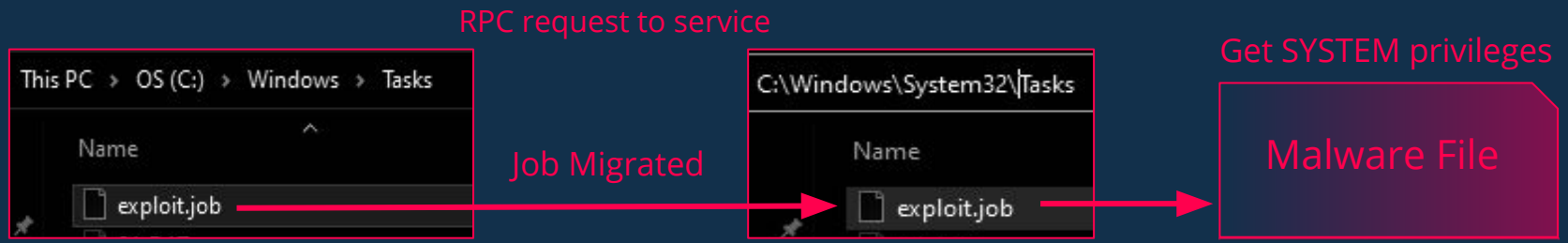
Task Scheduler LPE - CVE-2019-1069

CVE-2019-1069 - new Task Scheduler LPE

Task Scheduler stores tasks as files in two separate locations:
C:***Windows***\Tasks < ----(legacy location).

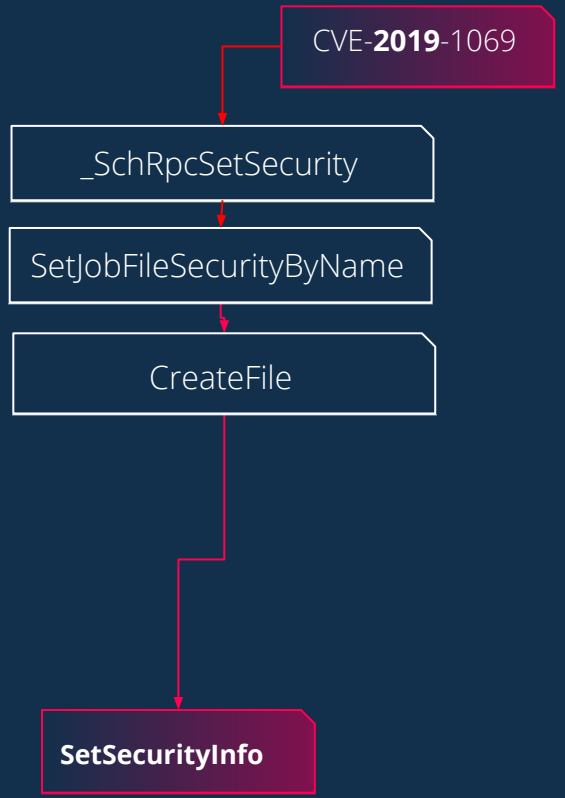
C:***Windows\System32***\Tasks

Sending an RPC request to the service for modifying a legacy-located task will migrate it to the preferred location of C:\Windows\System32\Tasks.

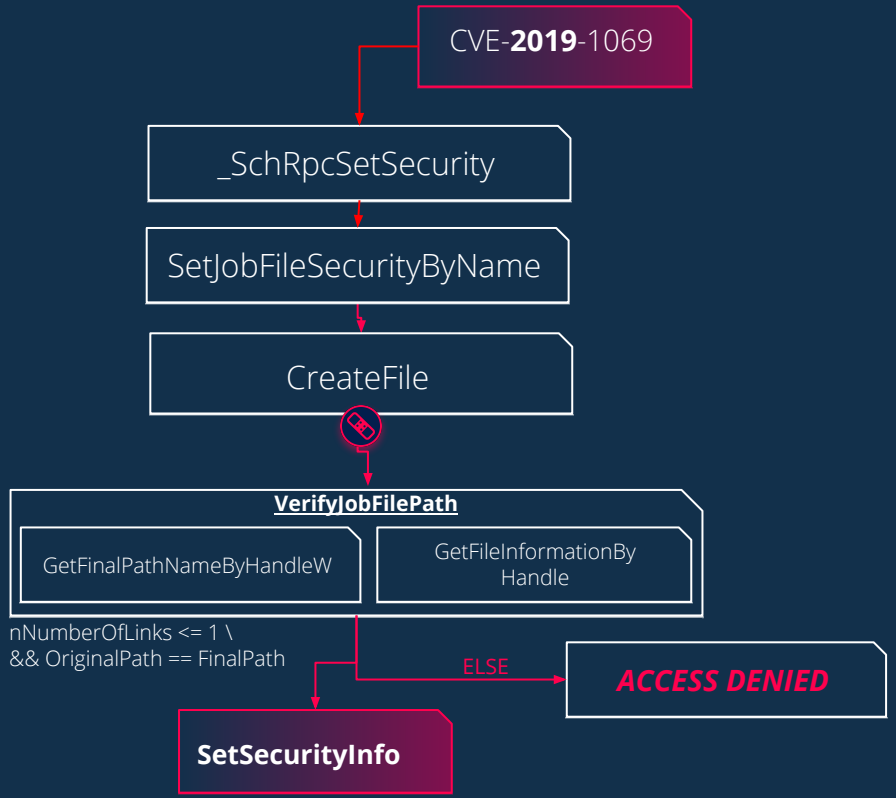


https://www.zerodayinitiative.com/blog/2019/6/11/exploiting-the-windows-task-scheduler-through-cve-2019-1069

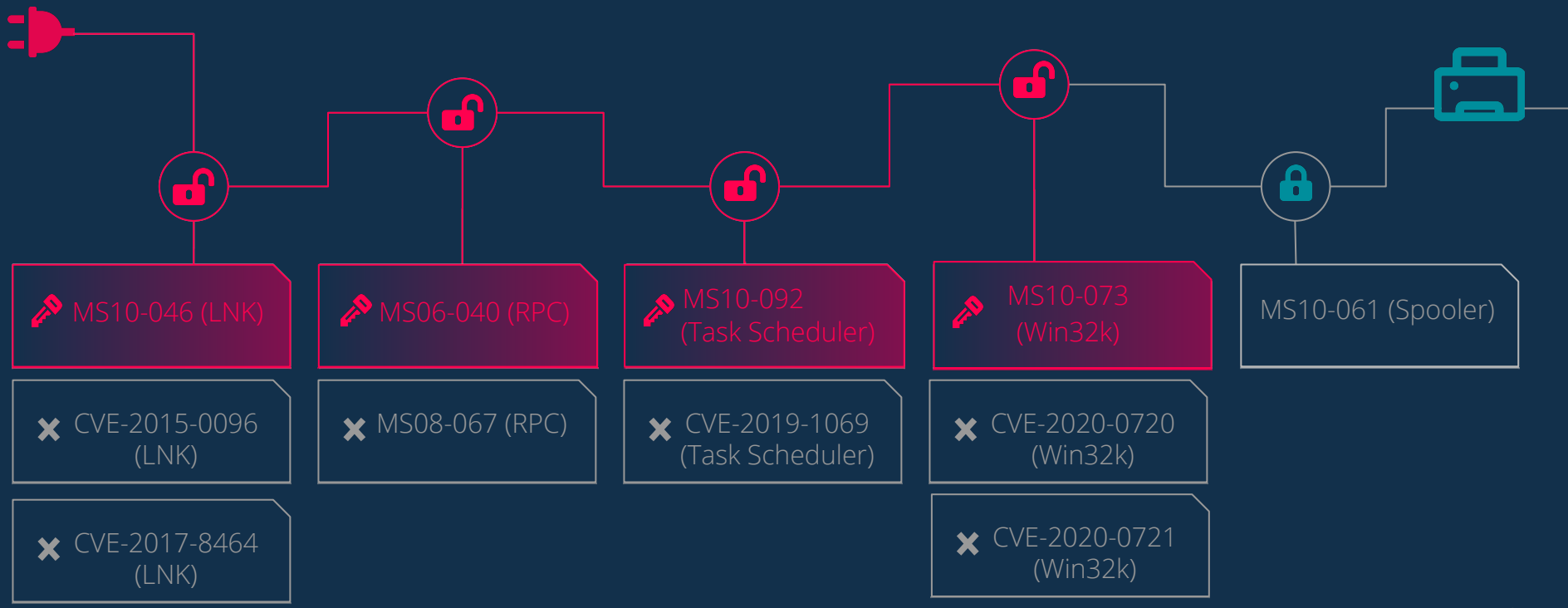
Task Scheduler 0-Day Exploitation Paths Overview



Task Scheduler CVE-2019-1069 - Patch



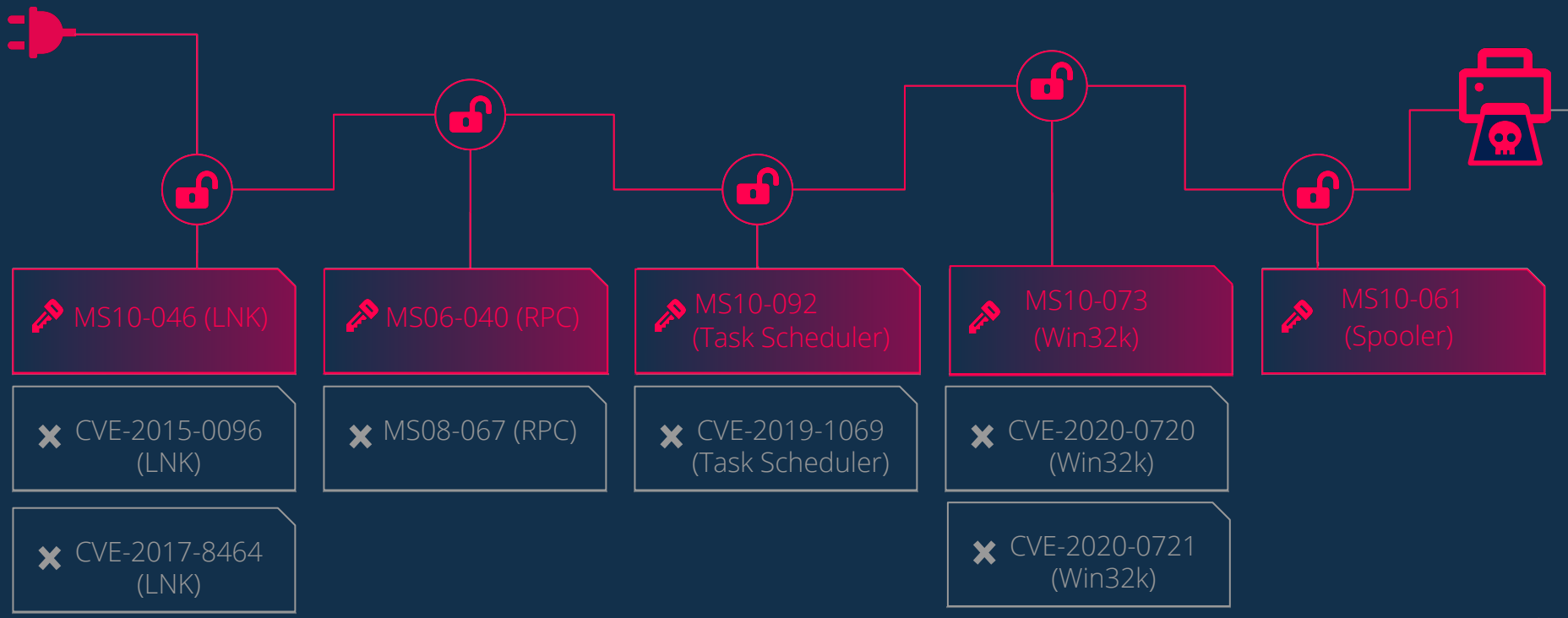
Spooler Propagation Capabilities



Win32k Vulnerabilities - 2020 List (Partial)

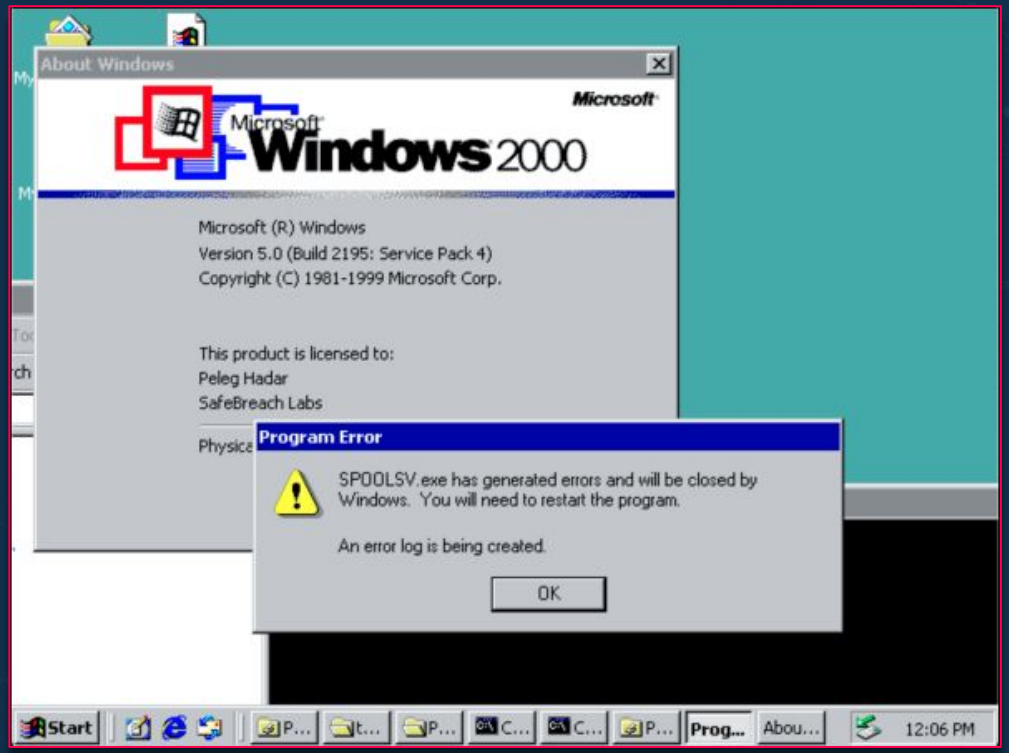
CVE-2020-0720	Win32k Elevation of Privilege Vulnerability
CVE-2020-0721	Win32k Elevation of Privilege Vulnerability
CVE-2020-0722	Win32k Elevation of Privilege Vulnerability
CVE-2020-0723	Win32k Elevation of Privilege Vulnerability
CVE-2020-0725	Win32k Elevation of Privilege Vulnerability
CVE-2020-0726	Win32k Elevation of Privilege Vulnerability
CVE-2020-0731	Win32k Elevation of Privilege Vulnerability
CVE-2020-0719	Win32k Elevation of Privilege Vulnerability

Spooler Propagation Capabilities

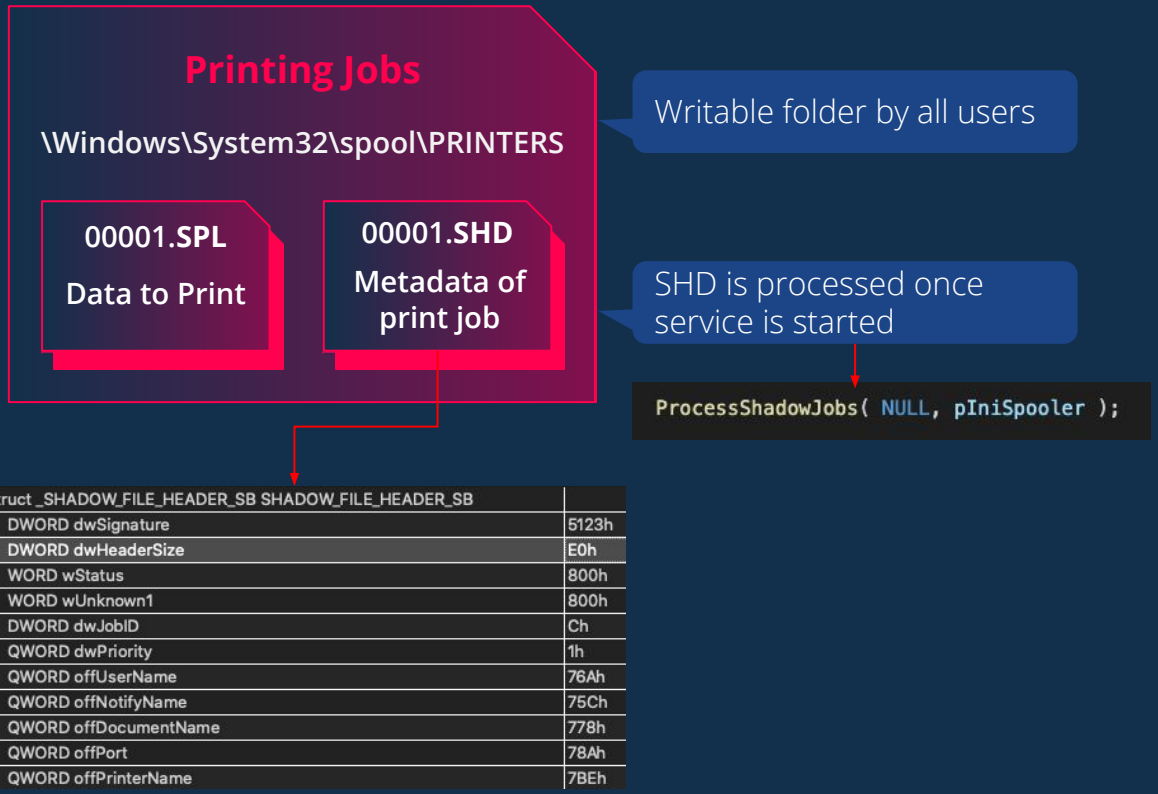


Our Research

20+ Year-old Bug in 20 Minutes of Fuzzing



Spooler SHD and SPL files



struct _SHADOW_FILE_HEADER_SB SHADOW_FILE_HEADER_SB	
DWORD dwSignature	5123h
DWORD dwHeaderSize	E0h
WORD wStatus	800h
WORD wUnknown1	800h
DWORD dwJobID	Ch
QWORD dwPriority	1h
QWORD offUserName	76Ah
QWORD offNotifyName	75Ch
QWORD offDocumentName	778h
QWORD offPort	78Ah
QWORD offPrinterName	7BEh

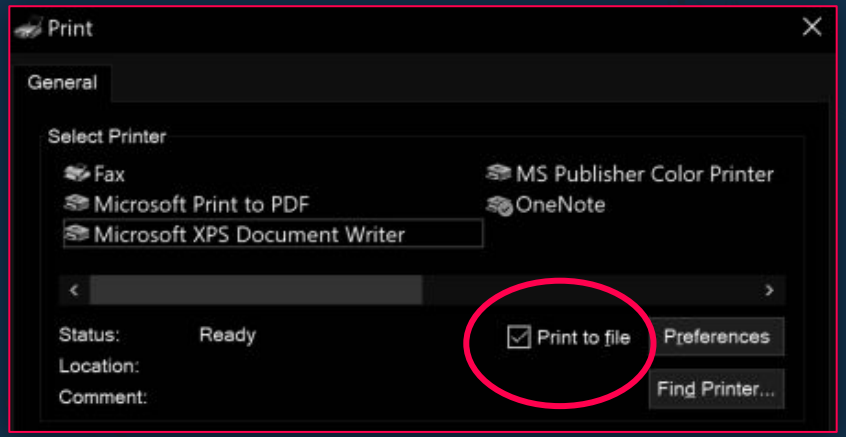
Spooler Fuzzing in the Shadow (File)

After 20 minutes…

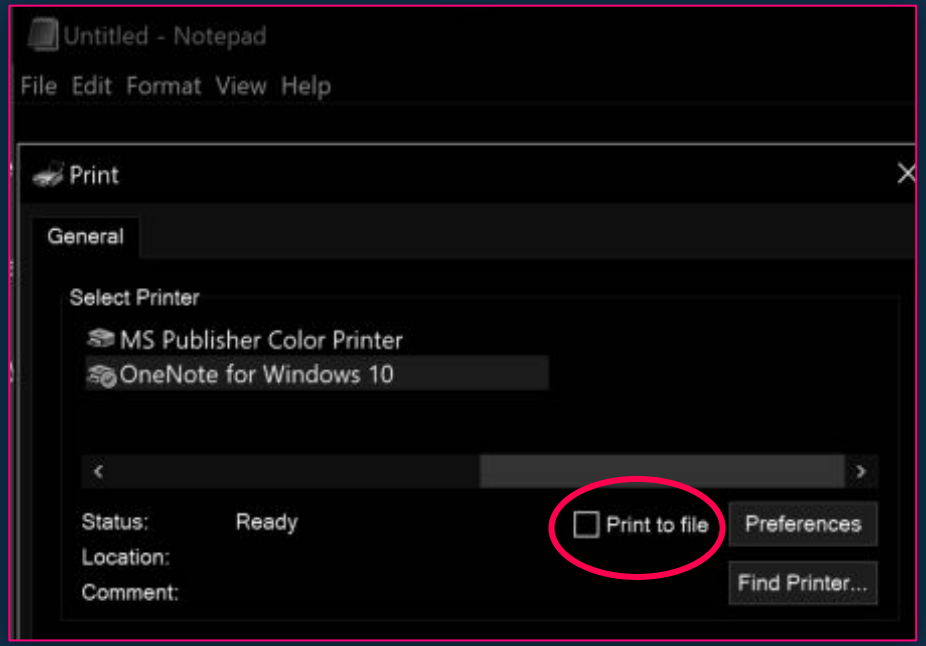
```
ntdll!RtlLengthSecurityDescriptor+0x60:
00007fff`881cbe70 0fb64001        movzx   eax,byte ptr [rax+1] ds:8c040001`00000001=??
Resetting default scope

EXCEPTION_RECORD:  (.exr -1)
ExceptionAddress: 00007fff881cbe70 (ntdll!RtlLengthSecurityDescriptor+0x0000000000000060)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: ffffffffffffffff
Attempt to read from address ffffffffffffffff
```

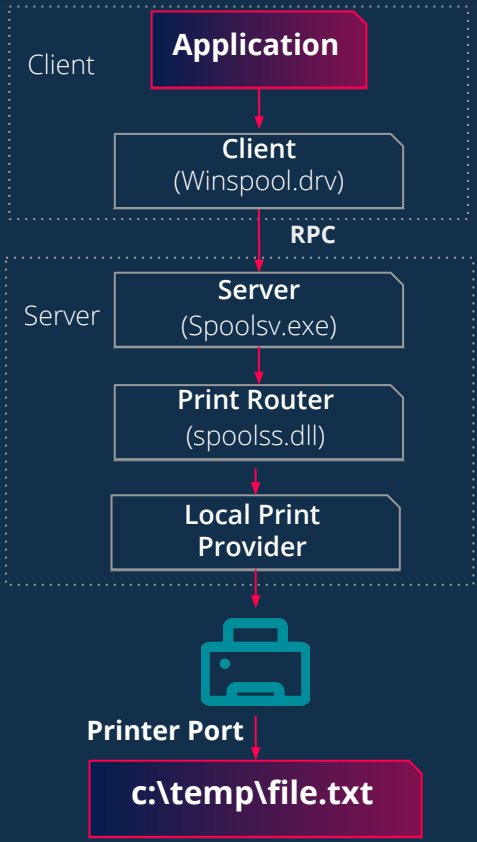
Spooler Crash Demo

Print Spooler (Printing to a File)

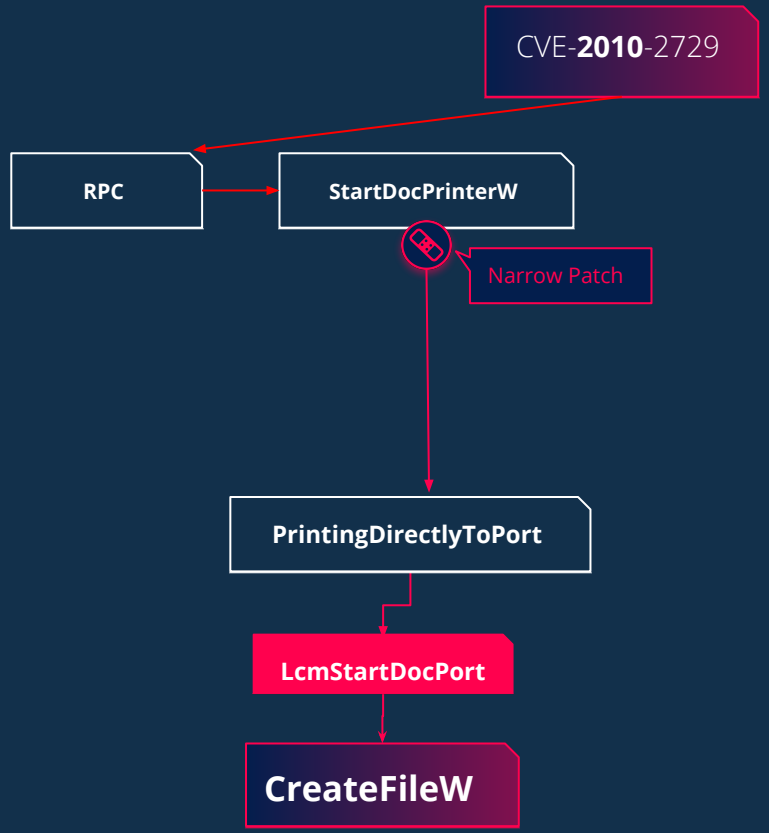


Print Spooler (Printing to a File)

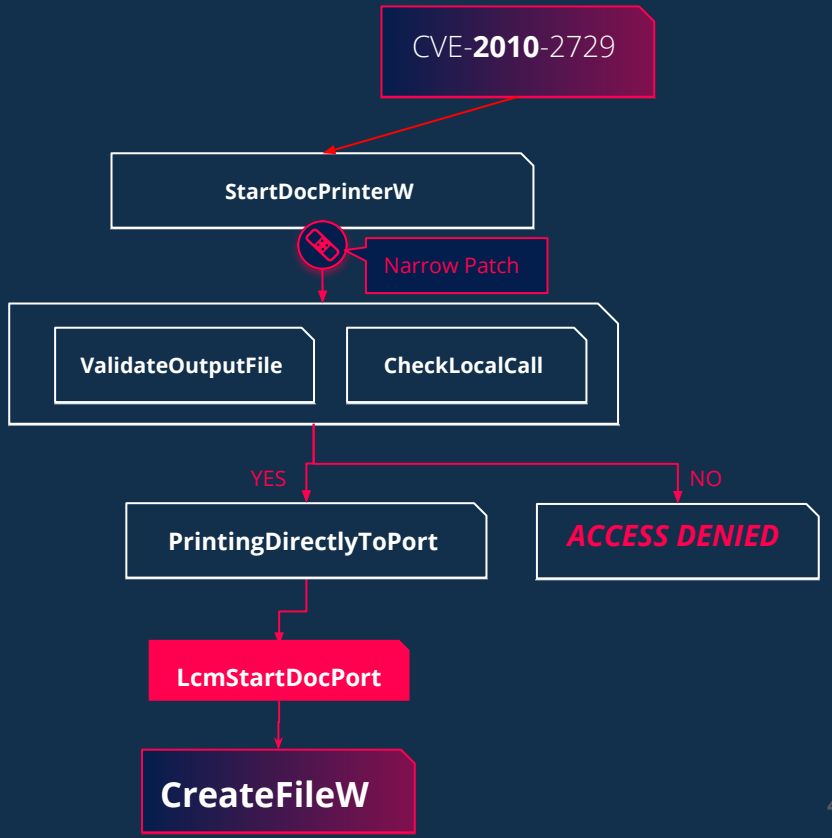




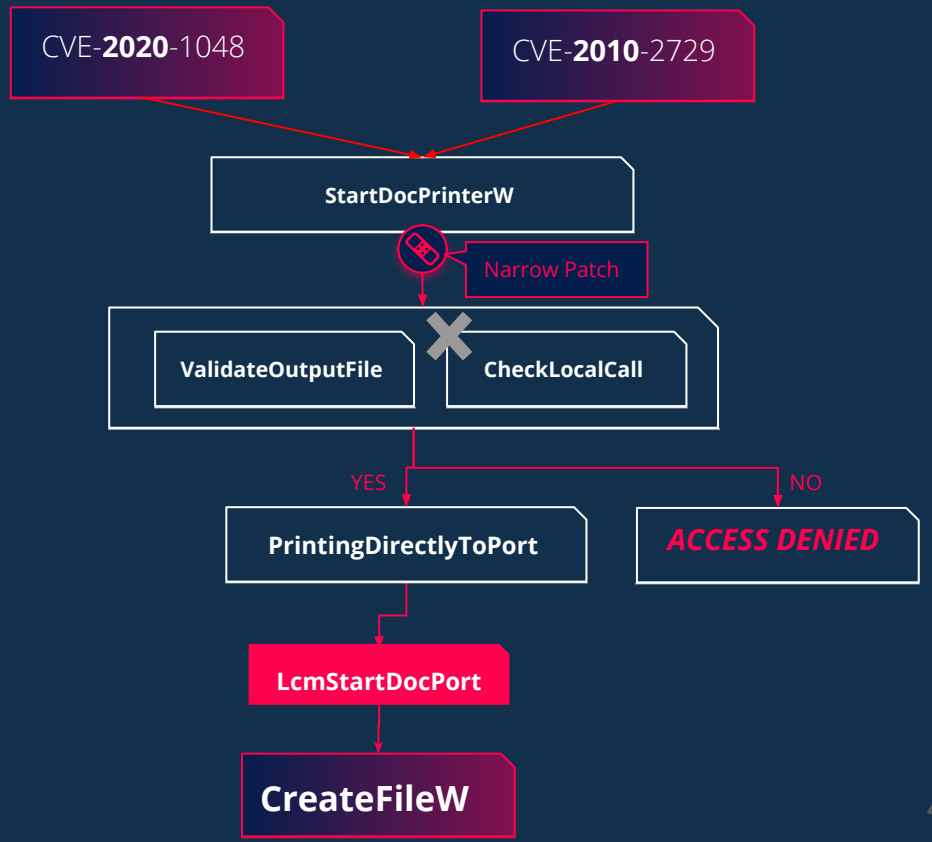
Spooler 0-Day Exploitation Paths Overview



Spooler MS10-061 Patch



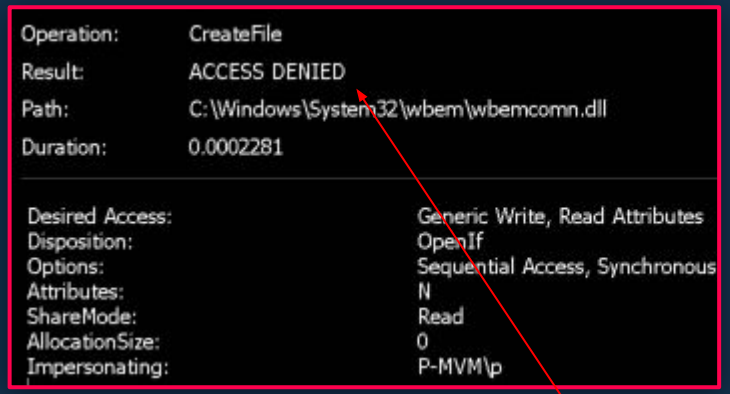
Spooler MS10-061 Patch Bypass #1

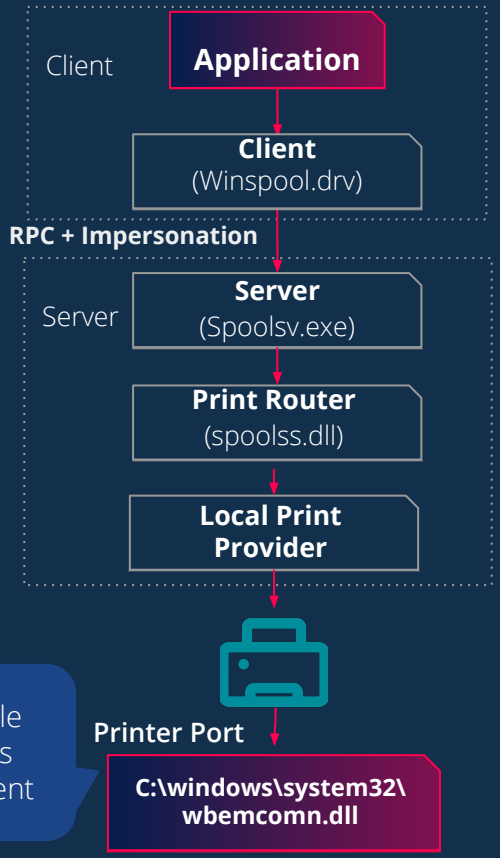


Spooler Arbitrary Printer Port Creation

```
PS C:\Users\Johnny> Add-PrinterPort c:\windows\system32\wbem\wbemcomn.dll
PS C:\Users\Johnny> Add-Printer "MS Publisher Color Printer" -DriverName
-PortName "c:\windows\system32\wbem\wbemcomn.dll"
```

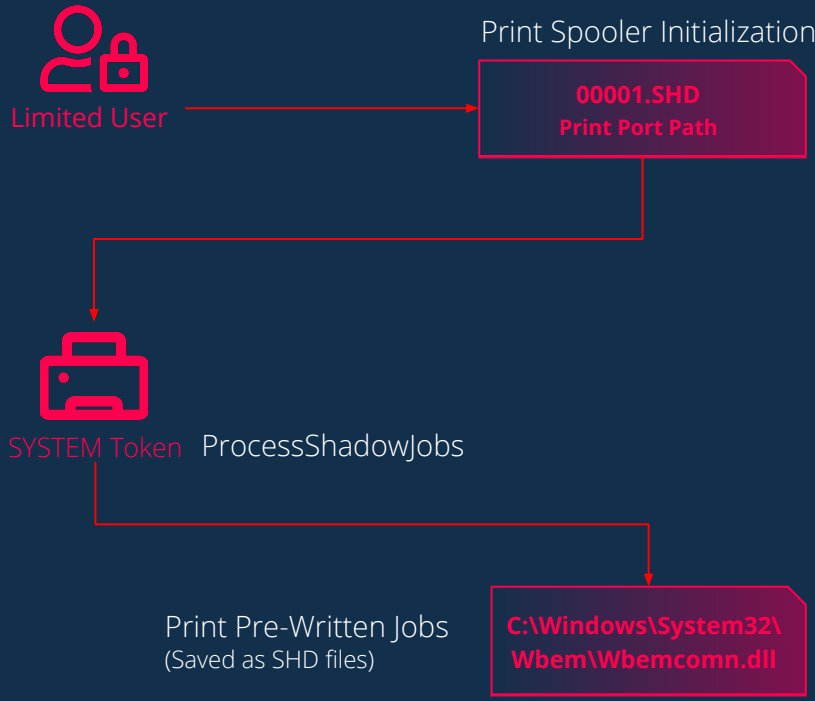
Spooler The Impersonation Barrier





Accessing the file using the access token of the client

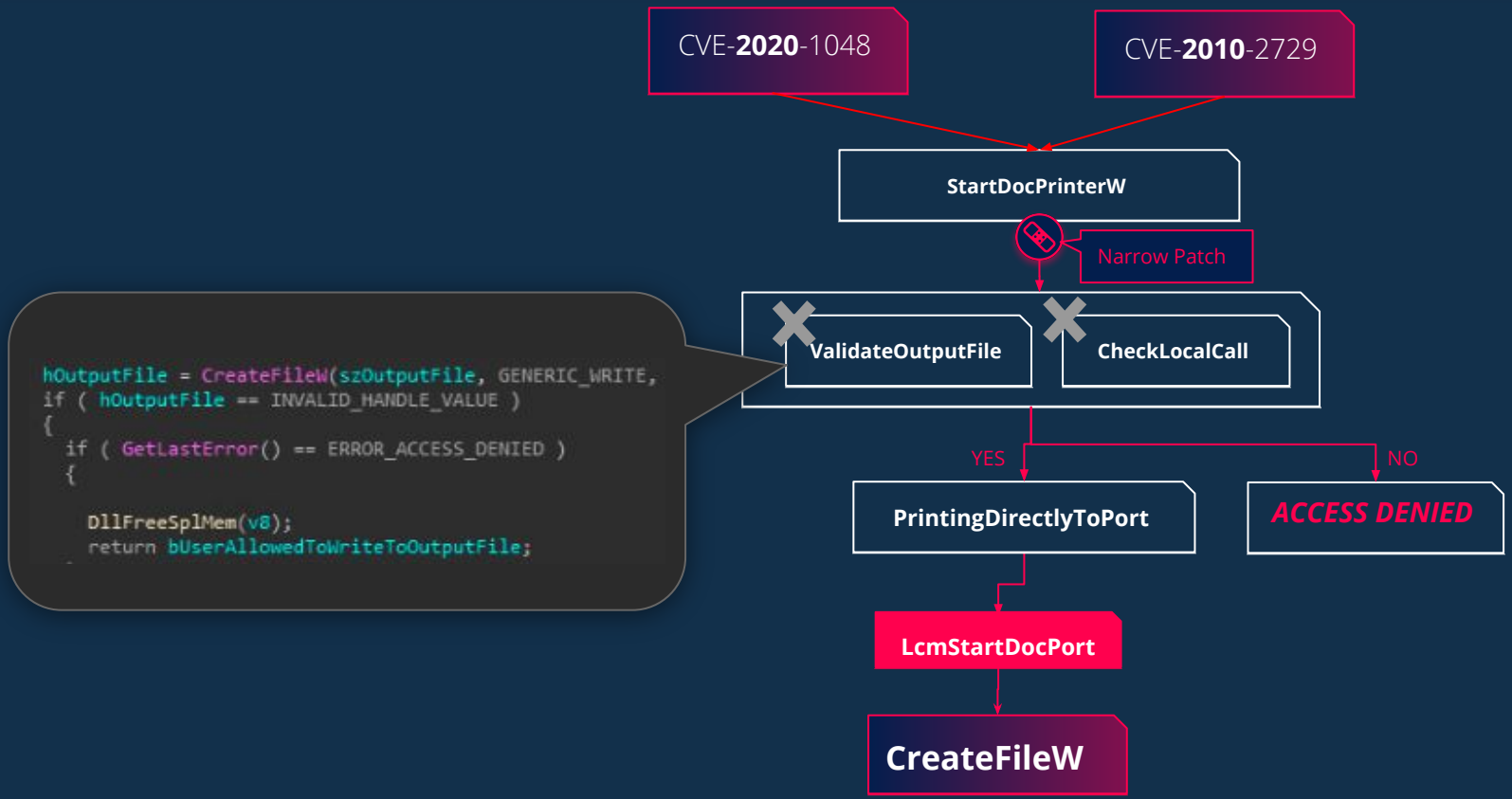
Spooler CVE-2020-1048 Root Cause



Operation:	CreateFile
Result:	REPARSE
Path:	C:\Windows\System32\wbem\wbemcomn.dll
Duration:	0.0000160

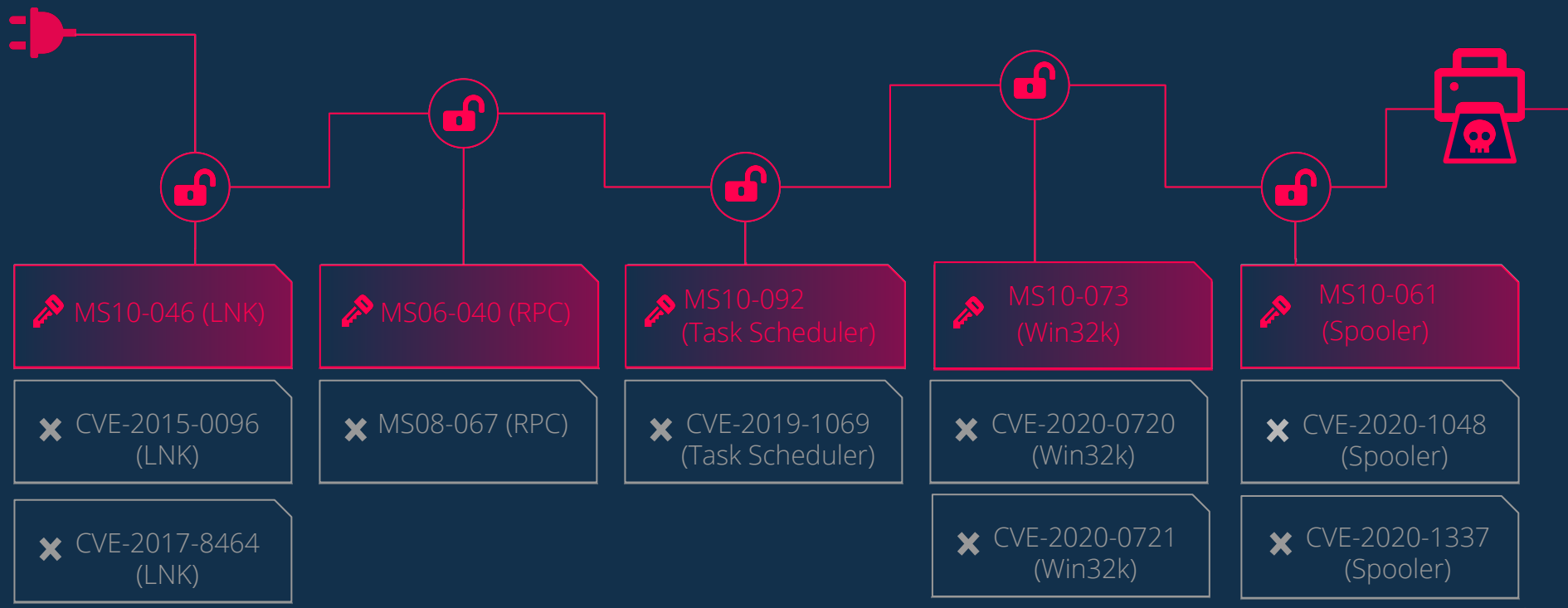
Desired Access:	Generic Write, Read Attribut
Disposition:	OpenIf
Options:	Sequential Access, Synchron
Attributes:	N
ShareMode:	Read
AllocationSize:	0
Impersonating:	NT AUTHORITY\SYSTEM
OpenResult:	<unknown>

Spooler MS10-061 Patch Bypass #2



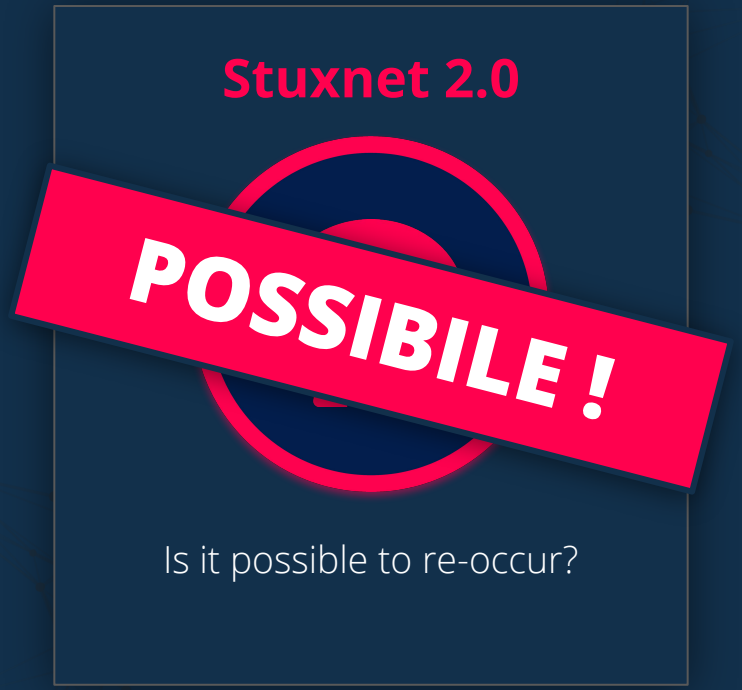
Spooler LPE Demo (1/2)

Spooler Printing our Way to SYSTEM



Spooler Printing our Way to SYSTEM

Spooler Printing our Way to SYSTEM



Spooler 0-day - Patch Bypass - CVE-2020-1337

CVE-**2020**-1337

CVE-**2020**-1048

Narrow Patch

CVE-**2010**-2729

- This is a **0-day** and it will be fixed by Microsoft
- Stay tuned for our exploit blog post which will be released in the next few days (once the vulnerability is fixed)

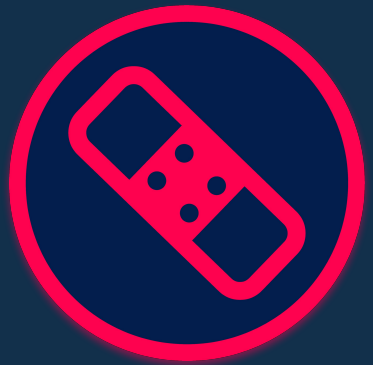
REDACTED

Spooler 0-day Demo - CVE-2020-1337 - REDACTED

Mitigations

Recommended Mitigations

Patch effectiveness



Is it possible to abuse patched vulnerabilities?

Recommended Mitigations Spooler

- Breach and Attack Simulations
- Security Operation Center
- Network Security Controls
- Real Time Detection & Prevention
- OS Patching

Recommended Mitigations Bug Class

A limited user can write to the following paths which leads to multiple vulnerabilities

1. System32\spool\PRINTERS - CVE-2020-1048, CVE-2020-1337, Spooler DoS
2. Spool\drivers\color - CVE-2020-1117 (RCE)
3. System32\tasks - CVE-2019-1069
4. C:\ProgramData\Microsoft\Windows\WER\ReportQueue - CVE-2019-0863
5. c:\windows\debug\WIA
6. c:\windows\PLA - 3 sub directories.

```
C:\>echo "MZmy malicious arbitrary file write" > c:\windows\system32\Safebreach.exe
Access is denied.

C:\>echo "MZmy malicious arbitrary file write" > c:\windows\system32\spool\PRINTERS\Safebreach.exe

C:\>echo "MZmy malicious arbitrary file write" > c:\windows\system32\spool\drivers\color\Safebreach.exe

C:\>echo "MZmy malicious arbitrary file write" > C:\ProgramData\Microsoft\Windows\WER\ReportQueue\Safebreach.exe

C:\>echo "MZmy malicious arbitrary file write" > c:\windows\debug\WIA\Safebreach.exe

C:\>echo "MZmy malicious arbitrary file write" > c:\windows\PLA\reports\Safebreach.exe
```

Recommended Mitigations driver demo

Microsoft Response

Spooler LPE

> The additional vector for CVE-2020-1048 will be addressed in August 2020 as CVE-2020-1337
>
> *~Microsoft Security Response Center*

Spooler DoS

> The technique results in a local Denial of Service; which doesn't meet Microsoft's servicing bar for security updates
>
> *~Microsoft Security Response Center*

Related Work

- **Alex Ionescu & Yarden Shafir** - PrintDemon
- **Dave Weinstein** - Full details on CVE-2015-0096 and the failed MS10-046 Stuxnet fix
- **ITh4cker** - Windows Lnk Vul Analysis:From CVE-2010-2568 to CVE-2017-8464
- **Jeongoh Kyea** - CVE-2020-1770 - Print Spooler EoP Vulnerability

Released Tools

- CVE-2020-1048 - Exploit PoC
- **0-day** Spooler ServiceS DoS - Exploit PoC
- Arbitrary File Write Mitigation - Driver
- *On August 12th* - CVE-2020-1337 - Exploit PoC

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

Q&A

See you next time on -

Peleg Hadar Senior Security Researcher **&** **Tomer Bar** Research Team Leader | SafeBreach LABS

Thank You!

Peleg Hadar Senior Security Researcher **&** **Tomer Bar** Research Team Leader | SafeBreach LABS