

PrintNightmare

CVE-2021-1675

발생 원리와
Print Spooler 함수 흐름 분석

넬카말카 세미나 발표 | 제현승 | 2026.07.08

목차

01 PrintNightmare 개요

02 핵심 개념

03 발생 원리

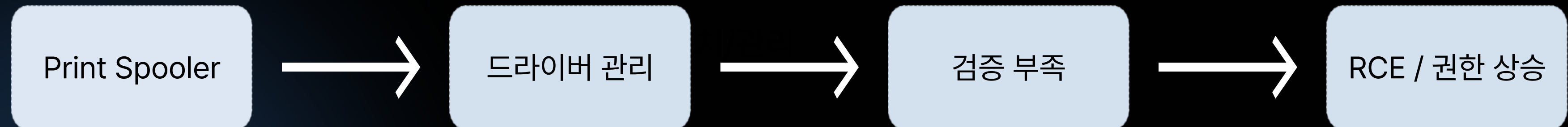
04 IDA 함수 흐름 분석

05 대응 방법 및 한계점

PrintNightmare 개요

- Windows Print Spooler에서 발생한 취약점
- 프린터 드라이버 설치/관리 과정과 관련
- 악용 시 **RCE** 또는 권한 상승 가능

핵심 흐름



왜 위험한가?

- **Windows** 기본 서비스와 관련됨
- 높은 권한으로 동작할 수 있음
- 악용 시 시스템 전체에 영향 가능

프린터를 사용하지 않아도
서비스가 활성화되어 있으면 위험할 수 있음

기본 탑재 서비스

높은 권한과 연결

RCE / 권한 상승

Print Spooler란?

Windows의 인쇄 작업 관리자

사용자의 인쇄 요청을 받아 대기열에 저장하고
프린터가 순서대로 출력하도록 관리하는 서비스



PrintNightmare의 핵심이 되는 Windows 서비스

프린터 드라이버와 DLL

프린터 드라이버

Windows가 프린터를 사용할 수 있게 해주는 연결 소프트웨어

DLL

프로그램이 실행 중 필요한 기능을 불러오는 코드 파일

드라이버 설치 과정에서 DLL이 사용될 수 있어
파일 검증이 중요함

RCE와 권한 상승

악용 시 발생할 수 있는 두 가지 영향

01 RCE

공격자가 만든 코드가
대상 컴퓨터에서 실행되는 것

02 권한 상승

낮은 권한에서 관리자 **/SYSTEM**
권한으로 올라가는 것

PrintNightmare는 두 영향 모두와 연결될 수 있어 위험함

RPC란?

Remote Procedure Call

다른 프로세스나 서비스에
기능 실행을 요청하는 방식

Print Spooler 같은 **Windows** 서비스 기능도
RPC를 통해 호출될 수 있음



드라이버 설치 과정의 검증

드라이버 설치
시스템에 파일이 추가되고 로드될 수 있는 작업

- ✓ 권한 확인
- ✓ 파일 확인
- ✓ 경로 확인

검증이 부족하면 위험한 파일이 로드될 수 있음

PrintNightmare 발생 흐름

Print Spooler



드라이버 설치/관리 요청



검증 부족



위험한 DLL 로드 가능



RCE / 권한 상승

핵심은 드라이버 설치/관리 과정의 검증 부족

분석 방향

드라이버 설치/관리 과정에 집중

PrintNightmare의 핵심을

드라이버 설치/관리 과정의 검증 문제로 보고 분석

spoolsv.exe

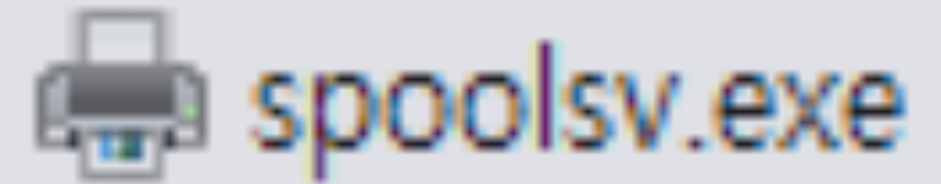
Driver 문자열

함수 흐름

분석 준비

spoolsv.exe

Print Spooler 서비스와 관련된 실행 파일



Driver 관련 문자열을 먼저 확인

Driver 문자열 검색

IDA Strings에서 Driver 검색

여러 Driver 관련 문자열 중
RpcAddPrinterDriverEx 중심으로 확인

Address	Length	Type	String
.rdata:00000000...	00000023	C	IsDriverOperationAllowedForTesting
.rdata:00000000...	00000014	C	RpcAddPrinterDriver
.rdata:00000000...	00000016	C	RpcAddPrinterDriverEx
.rdata:00000000...	00000016	C	RpcEnumPrinterDrivers
.rdata:00000000...	00000014	C	RpcGetPrinterDriver
.rdata:00000000...	0000001D	C	RpcGetPrinterDriverDirectory
.rdata:00000000...	00000017	C	RpcDeletePrinterDriver
.rdata:00000000...	00000019	C	RpcDeletePrinterDriverEx
.rdata:00000000...	00000015	C	RpcGetPrinterDriver2
.rdata:00000000...	00000023	C	RpcInstallPrinterDriverFromPackage
.rdata:00000000...	0000001E	C	RpcUploadPrinterDriverPackage
.rdata:00000000...	0000001E	C	RpcDeletePrinterDriverPackage
.rdata:00000000...	0000002D	C	RpcInstallPrinterDriverPackageFromConnection
.rdata:00000000...	0000002C	C	YRestrictDriverInstallationToAdministrators
.rdata:00000000...	00000012	C	YAddPrinterDriver
.rdata:00000000...	00000014	C	YAddPrinterDriverEx
.rdata:00000000...	00000014	C	YEnumPrinterDrivers
.rdata:00000000...	00000012	C	YGetPrinterDriver
.rdata:00000000...	0000001B	C	YGetPrinterDriverDirectory
.rdata:00000000...	00000015	C	YDeletePrinterDriver
.rdata:00000000...	00000017	C	YDeletePrinterDriverEx
.rdata:00000000...	00000013	C	YGetPrinterDriver2

RpcAddPrinterDriverEx 함수 확인

```
22     || !IsDriverOperationAllowedForTesting(  
23         *(const unsigned __int16 **)(v7 + 8),  
24         *(const unsigned __int16 **)(v7 + 24)) )  
25     {  
26         SpoolerServiceTelemetry::WriteDbgTraceError(  
27             "RpcAddPrinterDriverEx",  
28             L"Trying to add printer driver in Windows Protected Print mode.");  
29         return 50;  
30     }  
31 }  
32 if ( !RpcServerInqBindingHandle((RPC_BINDING_HANDLE *)&Binding) && TlsSetValue(gdwTlsBindingHandle, Binding) )  
33 {  
34     if ( (unsigned int)RpcApiValidSecurityLevel(Binding) )  
35     {  
36         SpoolerServiceTelemetry::WriteDbgTraceError("RpcAddPrinterDriverEx", L"Rpc call is access denied.");  
37         return 5;  
38     }
```

드라이버 추가 로그

RPC 보안 수준 검사

ACCESS DENIED 반환

드라이버 추가 전

보안 검사 흐름 확인

권한 검증 흐름

```
if ( !(unsigned int)YImpersonateClient(1) )  
    return GetLastError();  
HIDWORD(v12) = YIsElevated();  
v15 = YIsInAdministratorGroup();  
v11 = v15;  
((void (__fastcall *)(char *, __int64 *, int *, unsigned int *))SpoolerServiceTeleme  
    (char *)&v12 + 4,  
    0, 0, 0, 0)
```

요청 사용자 권한 기준 검사

Elevated 권한 여부 확인

관리자 그룹 여부 확인

드라이버 설치 전 사용자 권한을
확인하는 흐름

YAddPrinterDriverEx 호출 흐름

```
&v14);  
if ( v9 && !v11 )  
    v4 &= ~0x8000u;  
YRevertToSelf(1);  
if ( !v10 || v11 )  
{  
    v3 = YAddPrinterDriverEx(a1, a2, v4, 1, v12);  
}  
else  
{
```

조건 통과 후

YAddPrinterDriverEx 호출

드라이버 추가 처리와 관련된
함수로 판단

권한 검증 이후

드라이버 추가 관련 함수 호출 확인

확인한 함수 흐름

RpcAddPrinterDriverEx



RPC 보안 수준 검사



사용자 권한 확인



조건 통과 여부 확인



YAddPrinterDriverEx 호출

IDA에서 확인한 함수명, 로그 문자열, 조건문을 바탕으로 정리

취약점과 연결되는 지점

드라이버 추가 요청

→ Print Spooler 내부에서 처리

권한·보안 검증

→ 중요한 확인 지점

검증 부족

→ 위험한 드라이버/DLL 로드 가능

함수 흐름 분석 → PrintNightmare 원리와 연결

대응 방법

보안 업그레이드

취약점 패치 적용

서비스 비활성화

불필요하면 Print Spooler 중지

권한 제한

드라이버 설치 권한 관리

대응의 핵심은 패치 적용과 권한 제한

대응 방법의 한계

미패치 시스템

업데이트 누락 시 위험 지속

서비스 사용 환경

업무상 Print Spooler가 필요할 수 있음

권한 설정 문제

잘못된 권한 설정은 재위험 요소

패치 이후에도 서비스와 권한 관리가 필요함

Q&A

감사합니다