Windows 관리자 권한 획득을 위한 시스템 실행 파일의 DLL Hijacking 취약점 분석

배재건, 공성현, 석병진, 이창훈†
서울과학기술대학교 컴퓨터공학과
e-mail:{jgbae, gongsh, sbj7534, chlee}@seoultech.ac.kr

Analysis of DLL Hijacking Vulnerability in System Executable Files for Administrator Privileges of Windows

Jaegeon Bae, Seonghyeon Gong, Byoungjin Seok, Changhoon Lee†
Dept. of Computer Science and Engineering, Seoul National University of Science and Technology

요 약

Windows는 UAC(User Account Control)를 통해 사용자의 동의를 얻은 프로세스에게만 관리자 권한을 부여한다. 관리자 권한을 부여받은 프로세스는 시스템 파일 변경, 환경 변수 변경 등 표준 권한을 가진 프로세스가 수행하지 못하는 작업을 수행할 수 있다. 일부 악성코드들은 사용자 동의 없이 관리자 권한을 획득하기 위해 UAC Bypass 기법을 이용한다. 그러나 UACMe에 공개된 56개의 UAC Bypass 기법 중 20개의 기법에 대한 보안 패치가 현재까지 이루어지지 않고 있다. 따라서 본 논문에서는 현재 Windows 시스템의 UAC Bypass에 대한 안전성 수준을 분석하기 위해 시스템 디렉터리 내부 82개의 프로그램을 대상으로 UAC Bypass가 가능한 DLL Hijacking 취약점을 분석한다. 또한 UAC Bypass에 악용 가능한 50개의 신규 취약점을 발견하고 악용 시나리오에 따른 공격가능성을 보인다.

1. 서론

Desktop OS 시장에서 가장 높은 점유율을 보이는 Microsoft의 Windows는 Vista부터 도입된 UAC를 통하여 프로세스가 관리자 권한을 요구할 경우 다이얼로그를 통해 사용자에게 동의를 구한다[1]. 또한, 프로세스의 권한에 따라 접근할 수 있는 리소스를 제한하여 보안 수준을 높인다[2]. 하지만 UAC의 목적과 달리 자동 권한 상승이 이루어지는 시스템 실행 파일을 악용해 UAC를 무력화 할 수 있는 다양한 방법들이 발견되었다[3][4]. 공격자의 프로그램이 UAC를 우회하여 관리자 권한을 획득할 경우 레지스트리 변조, 시스템 파일 변조 등 더욱 치명적인 공격을 수행 할 수 있다. UAC를 우회하는 기법인 UAC Bypass는 2018년 초 한국 등을 대상으로 이루어진 Honeybee 공격[5] 등 일부 악성코드에 적용되어 UAC를 무력화시켰다. 일부 UAC Bypass 기법은 보안 패치로 인하여 더 이상 악용할 수 없다. 하지만 UACMe[3]에 공개된 56개 기법 중 20개 기법은 현재까지도 패치가 되지 않았으며 일부 기법은 Windows 7부터 Windows 10까지 폭넓게 적용가능하다. UAC Bypass 기법을 비교한 최근 연구에 따르면 49개 UAC Bypass 기법 중 시스템 디렉터리 내부의 DLL Hijacking을 악용하여 수행한 경우는 23개 기법이다[4]. 특정 기법은 시스템 실행 파일이 로드하는 DLL자체가 존재하지 않는 원리를 이용한다. 간단한 취약점이지만 아직 패치가 이루어지지 않은 기법이 존재하며 조사되지 않은 기법이 존재할 수 있다. 이에 따라 Windows 시스템 디렉터리 내부 프로그램에 대해, 관리자 권한 탈취에 악용 가능한 DLL Hijacking 취약점 존재 여부를 조사하였다.

본 논문의 2장에서는 UAC에 관해 기술하고 3장에서는 관련 연구에 관해 기술한다. 4장에서는 UAC Bypass에 적용 가능한 DLL Hijacking에 대해 분석한다. 5장에서는 실험 방법 및 결과에 대해 기술하며 마지막으로 6장에서 결론 및 향후 연구 방향 제시를 통해 마무리하고자 한다.

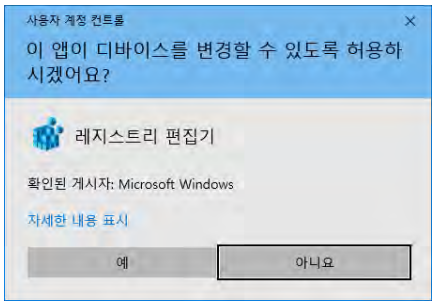
2. User Account Control

UAC는 Windows의 기본 보안 요소 중 하나이다. UAC는 “Least Privilege Principle”에 따라 프로세스는 동작에 필요한 최소한의 권한만 부여한다[6]. Windows에서 사용자는 표준 계정으로 로그온 할 경우 표준 액세스 토큰을 가지며 관리자 계정으로 로그온할 경우 표준 액세스 토큰과 관리자 액세스 토큰을 부여받는다. 일반적으로 사용자가 응용 프로그램을 실행할 때 표준 액세스 토큰을

※ 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2017-0-00158, 국가 차원의 침해사고 대응을 위한 사이버 위협 인텔리전스 분석(CTI) 및 정보 공유 기술 개발)
† 교신저자, chlee@seoultech.ac.kr(Corresponding author)

이용하여 실행된다. 관리자 액세스 토큰이 필요한 응용 프로그램의 경우 UAC 다이얼로그를 통해 사용자의 동의를 얻는다. 표준 액세스 토큰을 통해 실행된 프로세스는 표준 권한으로 실행되며 관리자 액세스 토큰을 통해 실행된 프로세스는 관리자 권한으로 실행된다. 표준 권한으로 실행된 프로세스는 접근할 수 있는 리소스 및 작업이 일부 제한된다[2].

프로세스가 자식 프로세스를 생성할 경우 자식 프로세스는 부모 프로세스의 권한을 상속받는다. 사용자에 의해 실행된 프로세스는 explorer.exe의 자식 프로세스로 생성되며 explorer.exe는 표준 권한으로 동작하기 때문에 일반적으로 사용자에 의해 실행된 프로세스는 표준 권한으로 동작한다. 관리자 권한의 프로세스가 자식 프로세스를 생성할 경우 자식 프로세스는 관리자 권한으로 실행된다. 프로세스가 로드한 DLL 또한 로드한 프로세스의 권한을 상속받는다[2][4].



(그림 1) UAC Dialog

사용자는 사용자 계정 컨트롤 설정을 통해 UAC 다이얼로그가 활성화되는 조건을 설정할 수 있다. 기본값은 “앱에서 사용자 모르게 컴퓨터를 변경하려는 경우에만 알림”이며 항상 알림 등 알림 조건 설정이 가능하다.

3. 관련 연구

DLL Hijacking은 LoadLibrary, LoadLibraryEx 함수를 사용할 때 DLL의 경로를 명시하지 않은 경우 DLL Search Order에 따라 DLL을 탐색한다[7]. LoadLibrary 함수의 경우 ①실행 파일이 로드된 디렉터리 ②시스템 디렉터리 ③16비트 시스템 디렉터리 ④Windows 디렉터리 ⑤현재 작업 중인 디렉터리 ⑥환경 변수에 나열된 디렉터리 순서로 탐색한다. 이러한 탐색 순서를 고려해 공격자가 더 높은 탐색 우선순위를 가진 경로에 악의적인 DLL을 생성할 경우 원본 DLL 대신 공격자의 DLL이 로드되어 공격자의 코드가 실행된다.

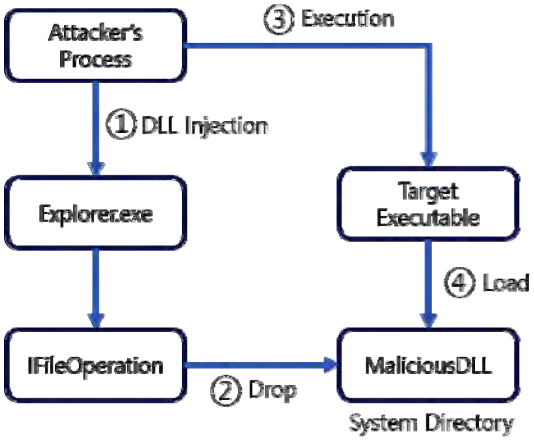
UACMe Project[3]는 GitHub를 통해 UAC Bypass 기법에 대해 각 기법을 저자, 타입, 방법, 대상, 요소 등 7가지 항목으로 정리했다. 2019년 3월 기준 56개의 UAC Bypass 기법이 존재하며 20개의 기법은 현재도 적용 가능하다. 26개 기법은 DLL Hijack 타입을 가지며 다른 타입을 가진 일부 기법 또한 수행 과정에서 DLL Hijacking을 이용한다.

Zejin Zhu등 2명[4]은 Windows UAC에 대한 취약성 연구를 수행했다. 연구 결과에 따르면 49개 UAC Bypass 기법에 대해 분류한 결과 29개 기법은 DLL Hijacking을 이용한 기법이다. 또한, 29개 기법 중 23개의 기법이 시스템 디렉터리 내부에서 일어난 DLL Hijacking이며 Windows Side by Side 매커니즘을 악용하거나 COM 인터페이스를 악용한 DLL Hijacking 기법 등이 존재한다. 공격자의 악용을 막기 위해 부모 프로세스의 PID를 검사하는 등 몇 가지 UAC 매커니즘의 개선 방안을 제안했다.

David Wells[8]는 실행 파일의 자동 권한 상승 과정에 대해 연구를 수행했다. 자동 권한 상승을 위해서는 3단계를 거친다. 먼저 실행된 파일의 경로가 C:\Windows\System32인지 검사한다. 다음으로 매니페스트의 Auto-Elevate 속성을 검사한다. 마지막으로 파일의 디지털 서명이 Microsoft인지 검사한다. 3단계를 모두 통과한 실행 파일만 UAC 다이얼로그 없이 관리자 권한으로 실행된다.

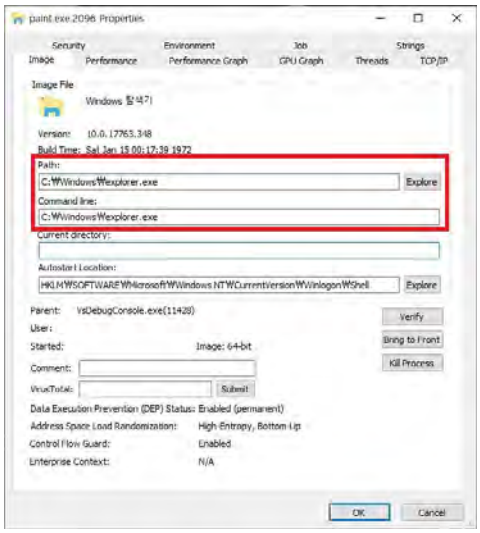
4. DLL Hijacking 기법 분석

UACMe에 공개된 56개의 기법 중 26개 기법의 수행 방식은 DLL Hijacking이다. 이 중 18개 기법은 DLL Hijacking을 위한 DLL을 시스템 디렉터리에 생성하기 위해 IFileOperation 오브젝트를 사용한다. IFileOperation 오브젝트는 파일 복사, 이동 및 삭제 등 명령을 수행한다. IFileOperation 오브젝트는 Microsoft의 서명이 되어있는 프로세스에서 자동으로 권한이 상승되어 동작한다. 공격자는 주로 explorer.exe를 대상으로 IFileOperation 오브젝트를 악용한다. explorer.exe는 표준 권한으로 실행되기 때문에 표준 권한으로 동작하는 다른 프로세스에서 DLL Injection이 가능하다. explorer.exe에 DLL Injection을 수행하여 DLL에 구현된 IFileOperation을 통해 다이얼로그 없이 관리자 권한으로 파일 복사, 이동, 삭제 등의 명령을 수행할 수 있다[4][9].



(그림 2) Abusing IFileOperation

프로세스의 PEB(Process Environment Block)을 조작하여 프로세스를 explorer.exe로 위장한 뒤 IFileOpeartion 오브젝트를 사용하는 기법 또한 존재한다. (그림 3)은 임의로 생성한 프로세스인 paint.exe의 PEB를 변조하여 Process Explorer[10]가 paint.exe를 Windows 탐색기로 인식하는 것을 나타낸다.



(그림 3) PEB가 변조된 프로세스

PEB는 명령 줄, 실행 파일 위치 등 프로세스에 대한 정보를 담고 있는 구조체이다. 공격자는 공격자가 제작한 임의의 프로세스의 PEB를 획득한다. 이후 해당 프로세스의 명령 줄, 실행 파일 경로 그리고 LDR 테이블의 프로세스 이름을 정상 explorer.exe와 동일하게 변조한다. 변조에 성공한 뒤 공격자의 프로세스는 IFileOperation을 이용하여 시스템 디렉터리에 악성 DLL을 생성 할 수 있다[4][9][11].

이러한 기법을 이용하여 시스템 실행 파일이 호출하는 DLL이 존재하지 않는 경로가 시스템 디렉터리 일지라도 DLL Hijacking을 통해 관리자 권한 획득이 가능하다.

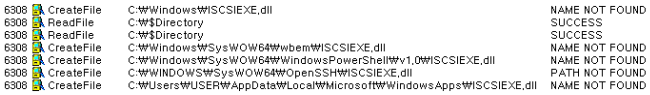
5. 실험 및 결과

실험은 Windows 10 64bit Pro 버전 1809 OS 빌드 17763.379 환경에서 수행한다. 2019년 3월 기준 최신 빌드 버전이며 최초 설치 환경에서 실험한다.

대상 폴더는 C:\Windows\System32 폴더와 C:\Windows\SysWOW64이며 해당 폴더에 존재하는 실행 파일 중 매니페스트 정보를 검사하여 Auto-Elevate 속성의 값이 True인 실행 파일을 대상으로 한다. 또한 Appinfo.dll의 Reference String에 존재하는 실행 파일 또한 포함한다. 해당 실행 파일들은 호환성의 문제로 하드 코딩된 리스트이며, 관리자 권한으로 실행시켜도 UAC 다이얼로그가 활성화되지 않는다[4].

Process Monitor[12]를 이용해 응용 프로그램의 로그를 추적하며 DLL 파일을 대상으로 CreateFile의 결과가 NAME NOT FOUND인 경우 DLL Hijacking이 가능한지 실험한다. DLL을 시스템 디렉터리 내부에 생성하여야 할 경우는 IFileOperation을 통해 DLL을 시스템 디렉터리로 복제한다. 환경 변수에 생성하여야 할 경우는 시스템 기본 환경 변수인 %USERPROFILE%\Appdata\Local\Microsoft\WindowsApps에 DLL을 복제하여 실험하였다.

System32 폴더 내 54개 실행 파일 및 SysWOW64폴더 내 28개 실행 파일을 대상으로 실험한 결과 DLL Hijacking을 통해 관리자 권한 탈취가 가능한 50개의 신규 취약점을 발견하였다. 발견된 취약점은 모두 SysWOW64에서 발견되었다. 1개 취약점은 iscsicpl.exe에서 발견되었으며 해당 실행 파일이 iscsiexe.dll을 탐색하지만, DLL Search Order 전체에 해당 DLL이 존재하지 않는다. 환경 변수에 설정된 경로 중 임의의 경로에 iscsiexe.dll을 생성하면 해당 DLL을 로드하며 관리자 권한으로 실행된다.



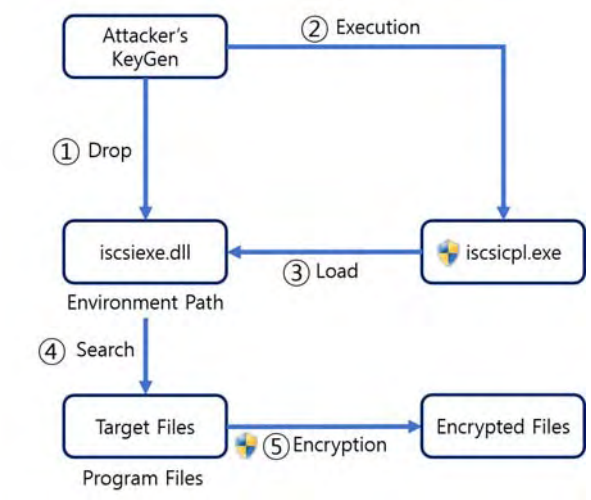
(그림 4) DLL 탐색 실패 로그

다른 1개 취약점은 ComputerDefaults.exe에서 발견하였으며 SysWOW64 폴더의 windows.ui.appdefaults.dll 탐색에 실패한다. 나머지 48개 취약점은 dccw.exe, eudcedit.exe 등 17개 실행 파일에 존재한다. 48개 취약점은 시스템 실행 파일이 아닌 SysWOW64\IME 폴더 내부의 imkrtip.dll 및 imetip.dll이 실행 파일에 로드된 뒤 로드 된 2개 DLL이 propsys.dll 등 3개의 DLL을 탐색하며 발생했다. 이는 Process Monitor를 통한 로그 분석과 LoadLibray 함수와 Process Explorer를 이용하여 imkrtip.dll 및 imetip.dll을 로드한 직후 propsys.dll 등 3개 DLL이 로드되는 것을 통해 확인했다. 프로세스가 imkrtip.dll을 로드하기 전 이미 propsys.dll을 로드한 경우 및 imetip.dll을 로드하기 전 이미 oleacc.dll을 로드한 경우는 해당 dll을 탐색하지 않는다. <표 1>의 DLL 경로는 C:\Windows\SysWOW64폴더를 기준으로 한다.

<표 1> 실행 파일 별 탐색 실패 DLL

유형	대상 DLL	시스템 실행 파일
실행 파일이 직접 탐색	▪ iscsiexe.dll	iscsicpl.exe
	▪ windows.ui.appdefaults.dll	ComputerDefaults.exe
실행 파일이 로드한 모듈에서 탐색	▪ IME\IMEKR\propsys.dll ▪ IME\shared\oleacc.dll ▪ IME\shared\version.dll	dccw.exe
		eudcedit.exe
		Netplwiz.exe
		odbcad32.exe
		PkgMgr.exe
		printui.exe
		shrpubw.exe
		SystemProperties*.exe (7종)
		Taskmgr.exe
		tcmsetup.exe
		wusa.exe

취약점이 발견된 실행 파일 중 iscsicpl.exe를 통해 공격자가 UAC Bypass를 통해 관리자 권한을 취득하여 악성 행위를 할 수 있는지 실험했다. 공격자가 불법 복제 프로그램용 가짜 KeyGen 프로그램을 가장한 랜섬웨어를 배포하는 시나리오를 가정한다. 암호화 대상은 파일 수정에 관리자 권한이 필요한 Program Files 폴더 내 특정 파일을 대상으로 한다. 사용자는 공격자의 KeyGen 프로그램을 실행시켜 KeyGen 버튼을 클릭한다. 버튼을 클릭한 직후 기본 환경 변수인 %USERPROFILE%\Appdata\Local\Microsoft\WindowsApps\경로에 악성 DLL인 iscsiexe.dll을 생성한다. 이후 ShellExecuteEx 함수를 이용해 관리자 권한으로 iscsicpl.exe를 실행시킨다. 실행된 iscsicpl.exe는 공격자가 생성한 iscsiexe.dll을 로드한 뒤 iscsiexe.dll은 Program Files 폴더 내의 특정 파일을 암호화한다.



(그림 5) 관리자 권한 탈취 시나리오

6. 결론

Windows는 UAC를 통해 관리자 권한 획득을 위해 사용자 동의를 요구한다. 하지만 현재까지 공개된 다양한 UAC Bypass 기법을 통해 UAC를 무력화 할 수 있다. 56개 UAC Bypass 기법은 UACMe를 통해 공유되고 있지만 20개 기법은 현재까지도 패치가 이루어지지 않았다. 패치가 이루어지지 않아 공격자는 손쉽게 관리자 권한을 탈취할 수 있으며 이는 UAC의 존재를 무의미하게 한다. 공격자는 탈취한 관리자 권한을 통해 접근이 제한된 리소스에 접근이 가능해지며 더욱 치명적인 공격이 가능하다.

본 논문에서는 Windows 관리자 권한 탈취를 목적으로 한 시스템 실행 파일의 DLL Hijacking 취약점 존재 여부를 조사하여 관리자 권한 탈취에 악용 가능한 50개의 신규 취약점을 발견하였다. 또한, 공격자가 불법 복제 프로그램용 가짜 KeyGen 프로그램을 가장한 랜섬웨어를 배포하는 시나리오를 통해 공격자가 관리자 권한 획득을 위해 본 취약점을 악용 가능하며 관리자 권한이 필요한 작업을 수행할 수 있음을 보였다.

본 논문에서는 관리자 권한 탈취를 위해 시스템 실행 파일을 통한 DLL Hijacking을 통한 기법만 조사하였으나 본 논문에서 조사한 기법 외에 레지스트리 변조 등 다양한 기법이 존재한다. 따라서 향후 연구로는 현재까지 연구되지 않은 다른 UAC Bypass 기법 및 구체적인 대응 방안에 대한 연구가 이루어져야 한다.

참고문헌

[1] StatCounter, “Desktop Operating System Market Share Worldwide”, http://gs.statcounter.com/os-market-share/desktop/worldwide/, 2019.02.

[2] Dani Halfin, Justinha and brbrahm, “How User Account Control works”, https://docs.microsoft.com/ko-kr/windows/security/identity-protection/user-account-control/how-user-account-control-works, 2018.11.16.

[3] UACMe Project, https://github.com/hfiref0x/UACME, 2019.

[4] Zhu, Zejin, and Guojun Peng. “An Analysis About the Defects of Windows UAC Mechanism.” Chinese Conference on Trusted Computing and Information Security. Springer, Singapore, 2018.

[5] Ryan Sherstobitoff, “McAfee Uncovers Operation Honeybee, a Malicious Document Campaign Targeting Humanitarian Aid Groups”, https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/mcafee-uncovers-operation-honeybee-malicious-document-campaign-targeting-humanitarian-aid-groups/, 2018.05.02.

[6] Motiee, Sara, Kirstie Hawkey, and Konstantin Beznosov. “Do windows users follow the principle of least privilege?: investigating user account control practices.” Proceedings of the Sixth Symposium on Usable Privacy and Security. ACM, 2010.

[7] Microsoft Support, “Secure loading of libraries to prevent DLL preloading attacks”, https://support.microsoft.com/en-us/help/2389418/secure-loading-of-libraries-to-prevent-dll-preloading-attacks, 2011.

[8] David Wells, “UAC Bypass by Mocking Trusted Directories”, https://medium.com/tenable-techblog/uac-bypass-by-mocking-trusted-directories-24a96675f6e, 2018.11.08.

[9] Parvez, “Bypassing Windows User Account Control (UAC) and ways of mitigation”, 2014.12.24.

[10] Mark Russinovich, “Process Explorer v16.22”, “https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer”, 2018.12.11.

[11] FuzzySecurity, “Anatomy of UAC Attacks”, https://www.fuzzysecurity.com/tutorials/27.html

[12] Mark Russinovich, “Process Monitor v3.50”, https://docs.microsoft.com/en-us/sysinternals/downloads/procmon, 2018.02.13.