(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2012 년 11 월 1 일 (01.11.2012)

WIPO | PCT

(10) 국제공개번호
WO 2012/148080 A2

(51) 국제특허분류:
G06F 9/44 (2006.01)

(21) 국제출원번호: PCT/KR2012/001775

(22) 국제출원일: 2012 년 3 월 12 일 (12.03.2012)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2011-0039901 2011 년 4 월 28 일 (28.04.2011) KR

(71) 출원인 (US 을(를) 제외한 모든 지정국에 대하여): **주식회사 파수닷컴 (FASOO. COM CO., LTD)** [KR/KR]; 서울시 마포구 상암동 1605 누리꿈스퀘어 비즈니스센타 17 층, 121-270 Seoul (KR).

(72) 발명자; 겸
(75) 발명자/출원인 (US 에 한하여): **이종일 (LEE, Jong-Il)** [KR/KR]; 경기도 고양시 일산서구 주엽동 강선마을 709-303, 411-370 Gyeonggi-Do (KR). **이남수 (YI, Nam-Su)** [KR/KR]; 서울시 성북구 하월곡동 228 번지 꿈의 숲푸르지오 102 동 1304 호, 136-130 Seoul (KR).

(74) 대리인: **송경근 (SONG, Kyeong-Keun)**; 서울시 서초구 서초동 1621-25 웰스빌딩 3 층, 137-878 Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

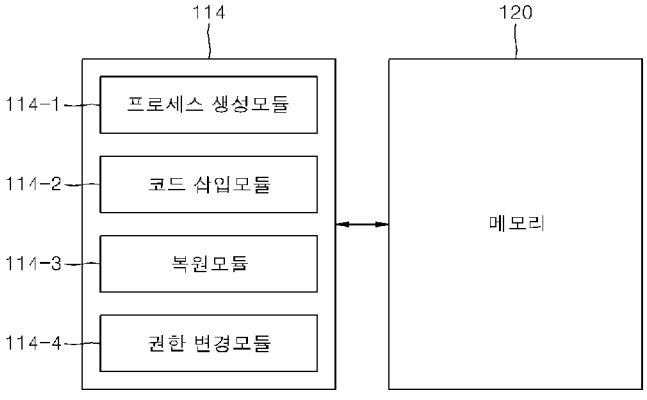
공개:

— 국제조사보고서 없이 공개하며 보고서 접수 후 이를 별도 공개함 (규칙 48.2(g))

(54) Title: COMPUTING DEVICE HAVING A DLL INJECTION FUNCTION, AND DLL INJECTION METHOD

(54) 발명의 명칭 : D L L 인젝션 기능을 구비한 컴퓨팅 장치 및 D L L 인젝션 방법

[Fig. 2]



114-1 ... Process-generating module
114-2 ... Code-inserting module
114-3 ...Restoration module
114-4 ... Permission-changing module
120 ... Memory

(57) **Abstract**: Disclosed are a computing device having a DLL injection function, and a DLL injection method. Upon a notification from an operating system relating to whether or not a target process corresponding to a program that a user commands to be executed has been generated, a process-generating module of an injection program executes a launcher process in order to generate said target process as a child process of the launcher process and set the target process using a stop mode. A code-inserting module receives an allocated memory area loaded with the target process using a process handle of the target process, and inserts a DLL file, which is to be injected into the target process, into a code being executed. A restoration module releases the stop mode of the target process in order to execute the target process. According to the present invention, by inserting a DLL execution code into a memory area of the target process generated with the stop mode, DLL injection may be performed stably without conflicting with another DLL injection technique.

(57) 요약서: DLL 인젝션 기능을 구비한 컴퓨팅 장치 및

[다음 쪽 계속]

WO 2012/148080 A2

DLL 인젝션 방법이 개시된다. 운영 체제로부터 사용자가 실행을 명령한 프로그램에 대응하는 대상 프로세스의 생성 여부가 통지되면, 인젝션 프로그램의 프로세스 생성모듈은 런처 프로세스를 실행하여 런처 프로세스의 자식 프로세스로서 상기 대상 프로세스를 생성하고, 대상 프로세스를 중지 모드로 설정한다. 코드 삽입모듈은 대상 프로세스의 프로세스 핸들을 이용하여 대상 프로세스가 적재된 메모리 영역을 할당받아 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입한다. 복원모듈은 대상 프로세스의 중지 모드를 해제하여 대상 프로세스가 실행되도록 한다. 본 발명에 따르면, 중지 모드로 생성된 대상 프로세스의 메모리 영역에 DLL 실행 코드를 삽입함으로써, 다른 DLL 인젝션 기법과 충돌하지 않고 안정적으로 DLL 인젝션을 수행할 수 있다.

명세서

발명의 명칭: DLL 인젝션 기능을 구비한 컴퓨팅 장치 및 DLL 인젝션 방법

기술분야

[1] 본 발명은 DLL 인젝션 기능을 구비한 컴퓨팅 장치 및 DLL 인젝션 방법에 관한 것으로, 보다 상세하게는, 런처가 실행되어 자식 프로세스로서 대상 프로세스가 실행될 때 DLL 인젝션을 수행하는 장치 및 방법에 관한 것이다.

배경기술

[2] DLL 인젝션(injection)이란 다른 프로세스(process)에 특정 DLL을 강제로 삽입시키는 것을 말한다. 즉, 다른 프로세스가 실행되는 과정에서 강제로 삽입한 DLL이 로드되어 처리되도록 하는 것이다. DLL 인젝션은 루트킷(rootkit)을 비롯하여 바이러스 및 악성코드 등 여러 분야에서 사용되며, 강제로 삽입된 DLL은 해당 프로세스의 메모리에 대한 접근 권한을 가지게 되므로 후킹(hooking) 및 기능 추가 등을 통해 프로세스를 쉽게 제어할 수 있다.

[3] 기존의 대표적인 DLL 인젝션 기술로는 SetWindowsHookEx() 함수를 사용하는 방법, CreateRemoteThread() 함수를 사용하는 방법 및 레지스트리(registry)의 값을 조작하는 방법 등이 있다. 이와 같은 기존의 DLL 인젝션 방법에서는 사용자가 프로세스를 시작하면 해당 프로세스가 메모리에 적재되며, 적재된 프로세스가 실행되면서 DLL 인젝션이 수행된다.

[4] 이상과 같은 기존의 DLL 인젝션 기법들은 동일한 프로세스에 대하여 동시에 적용됨에 따라 충돌이 발생하는 경우가 있으며, DLL 인젝션 시점이 동작 환경의 영향을 받아 달라지거나, 동일한 환경이라도 동작 순간마다 달라질 수 있으므로 실패할 가능성이 있다.

발명의 상세한 설명

기술적 과제

[5] 본 발명이 이루고자 하는 기술적 과제는, 기존의 DLL 인젝션 기법에 비해 안정성이 높으며, 다른 DLL 인젝션 기법과 충돌할 우려가 없는 DLL 인젝션 기능을 구비한 컴퓨팅 장치 및 DLL 인젝션 방법을 제공하는 데 있다.

[6] 본 발명이 이루고자 하는 다른 기술적 과제는, 기존의 DLL 인젝션 기법에 비해 안정성이 높으며, 다른 DLL 인젝션 기법과 충돌할 우려가 없는 DLL 인젝션 기능을 구비한 컴퓨팅 장치 및 DLL 인젝션 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는 데 있다.

과제 해결 수단

[7] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 DLL 인젝션 기능을 구비한 컴퓨팅 장치는, 대상 프로세스에 DLL 인젝션을 수행하기 위한 인젝션 프로그램 및 운영 체제를 실행하는 프로세서; 및 상기 운영 체제, 상기 인젝션

프로그램 및 상기 대상 프로세스가 저장되는 메모리;를 구비하며, 상기 인젝션 프로그램은, 런처 프로세스를 실행하여 상기 런처 프로세스의 자식 프로세스로서 상기 대상 프로세스를 생성하고, 상기 대상 프로세스를 중지(suspend) 모드로 설정하는 프로세스 생성모듈; 상기 대상 프로세스의 프로세스 핸들을 이용하여 상기 대상 프로세스가 적재된 메모리 영역을 할당받아 상기 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입하는 코드 삽입모듈; 및 상기 대상 프로세스의 중지 모드를 해제하여 상기 대상 프로세스가 실행되도록 하는 복원모듈;을 포함한다.

[8] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 DLL 인젝션 방법은, (a) 운영 체제로부터 사용자가 실행을 명령한 프로그램에 대응하는 대상 프로세스의 생성 여부를 통지받는 단계; (b) 런처 프로세스를 실행하여 상기 런처 프로세스의 자식 프로세스로서 상기 대상 프로세스를 생성하고, 상기 대상 프로세스를 중지 모드로 설정하는 단계; (c) 상기 대상 프로세스의 프로세스 핸들을 이용하여 상기 대상 프로세스가 적재된 메모리 영역을 할당받아 상기 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입하는 단계; 및 (d) 상기 대상 프로세스의 중지 모드를 해제하여 상기 대상 프로세스가 실행되도록 하는 단계;를 갖는다.

발명의 효과

[9] 본 발명에 따른 DLL 인젝션 기능을 구비한 컴퓨팅 장치 및 DLL 인젝션 방법에 의하면, 중지 모드로 생성된 대상 프로세스의 메모리 영역에 DLL 실행 코드를 삽입함으로써, 다른 DLL 인젝션 기법과 충돌하지 않고 안정적으로 DLL 인젝션을 수행할 수 있다. 또한 대상 프로세스를 시스템 권한으로 실행함으로써 DLL 인젝션을 수행하는 다른 프로세스가 대상 프로세스에 대해 실행되는 것을 방지할 수 있다.

도면의 간단한 설명

[10] 도 1은 본 발명에 따른 DLL 인젝션 기능을 구비한 컴퓨팅 장치에 대한 바람직한 실시예의 구성을 도시한 블록도,

[11] 도 2는 인젝션 프로그램의 구성을 도시한 도면,

[12] 도 3은 코드 삽입모듈에 의해 수행되는 인젝션 코드의 삽입 과정을 도시한 흐름도,

[13] 도 4는 PE 구조의 구성을 도시한 도면,

[14] 도 5는 권한 변경모듈에 의해 대상 프로그램이 시스템 권한으로 실행되도록 하는 과정을 도시한 흐름도, 그리고,

[15] 도 6은 본 발명에 따른 DLL 인젝션 방법에 대한 바람직한 실시예의 수행과정을 도시한 흐름도이다.

발명의 실시를 위한 형태

[16] 이하에서 첨부된 도면들을 참조하여 본 발명에 따른 DLL 인젝션 기능을

구비한 컴퓨팅 장치 및 DLL 인젝션 방법의 바람직한 실시예에 대해 상세하게 설명한다.

[17] 도 1은 본 발명에 따른 DLL 인젝션 기능을 구비한 컴퓨팅 장치에 대한 바람직한 실시예의 구성을 도시한 블록도이다.

[18] 도 1을 참조하면, 본 발명에 따른 DLL 인젝션 기능을 구비한 컴퓨팅 장치는 프로세서(110) 및 메모리(120)를 구비한다.

[19] 프로세서(110)는 특히 메모리(120)에 저장된 소프트웨어를 실행하기 위한 하드웨어 디바이스이다. 또한 프로세서(110)는 임의의 주문 제작 또는 상용으로 입수할 수 있는 프로세서, 중앙 처리 장치(CPU), 컴퓨터와 연관된 몇몇의 프로세서 중 보조 프로세서, 반도체 기반 마이크로 프로세서, 매크로 프로세서 또는 일반적으로 소프트웨어 명령어를 실행하기 위한 임의의 디바이스일 수 있다.

[20] 도 1에 도시된 바와 같이 프로세서(110)는 운영 체제(112)를 실행하며, 운영 체제(112)는 메모리(120)에 저장될 수 있다. 또한 프로세서(110)는 메모리(120)에 저장되어 있는 프로그램 어플리케이션(122)으로부터 컴퓨터 프로그램 지시사항을 검색하고 실행하며, 인젝션 프로그램(114)을 실행한다. 이때 인젝션 프로그램(114)은 드라이버 형태로 구현되어 레지스트리 설정을 통해 운영 체제(112)의 부팅시 적절한 시점에 자동으로 실행될 수 있으며, 공통 프로그램 어플리케이션의 일 요소로서 구현될 수도 있다.

[21] 메모리(120)는 휘발성 메모리 요소, 예를 들면, RAM(DRAM, SRAM 및 SDRAM 등 및 비휘발성 메모리 요소, 예를 들면, ROM, 소거 가능 프로그램 가능한 읽기 전용 메모리(EPROM), 전기적 소거가능 읽기 전용 메모리(EEPROM), 프로그램 가능한 읽기 전용 메모리(PROM), 테이프, 컴팩트 디스크 읽기 전용 메모리(CD-ROM), 디스크, 디스켓, 카트리지, 카세트 등 중의 어느 하나 또는 이들의 결합을 포함할 수 있다. 나아가 메모리(120)는 전자적, 자기적, 광학적 및/또는 다른 타입의 저장 매체를 채택할 수 있다. 메모리(120)는 다양한 컴포넌트가 서로로부터 멀리 떨어져 위치한 분산 아키텍처를 가질 수 있으나, 프로세서(110)에 의해 액세스될 수 있다.

[22] 또한 도 1에 도시된 바와 같이 메모리(120)에는 문서, 멀티미디어 파일 및 데이터 파일 등 프로그램 데이터(124)가 저장되며, 나아가 메모리(120)의 일부 공간을 인젝션 프로그램(114)이 사용할 수 있다. 이와 같이 메모리(120)에 저장된 소프트웨어는 하나 이상의 개별 프로그램을 포함할 수 있고, 각각의 프로그램은 논리적 기능을 구현하기 위한 실행 가능 명령어의 정렬된 목록을 포함한다.

[23] 도 1의 인젝션 프로그램(114)은 대상 프로세스에 DLL 인젝션을 수행하기 위한 것으로, 도 2에는 인젝션 프로그램(114)의 구성이 도시되어 있다. 도 2를 참조하면, 인젝션 프로그램(114)은 프로세스 생성모듈(114-1), 코드 삽입모듈(114-2) 및 복원모듈(114-3)로 이루어져 대상 프로세스의 실행이 완료되기 이전에 DLL 인젝션을 수행한다.

[24] 기존의 DLL 인젝션 기법에서는 사용자로부터 프로세스 실행 명령이 입력되면 해당 프로세스가 메모리에 적재된 후 실행되면서 DLL 인젝션이 수행된다. 그러나 앞에서 설명한 바와 같이 이러한 기존의 DLL 인젝션 기법은 실패 또는 다른 DLL 인젝션 기법과의 충돌 가능성이 있다.

[25] 따라서 본 발명에서 제안하는 인젝션 프로그램(114)은 런처(launcher) 프로세스를 실행하여 대상 프로세스를 자식 프로세스로서 실행하되, 대상 프로세스가 적재된 메모리 영역에 DLL 파일의 경로를 나타내는 인젝션 코드를 삽입함으로써 대상 프로세스의 실행 이전에 DLL 인젝션이 이루어지도록 할 수 있다.

[26] 프로세스 생성모듈(114-1)은 런처 프로세스를 실행하여 런처 프로세스의 자식 프로세스로서 대상 프로세스를 생성하고, 생성한 대상 프로세스를 중지(suspend) 모드로 설정한다. 이는 대상 프로세스가 실행을 위해 메모리에 적재된 상태에서 프로세스 실행이 완료되기 이전에 인젝션 코드를 삽입하기 위해서이다. 또한 대상 프로세스는 런처 프로세스의 자식 프로세스로서 생성되므로, 프로세스 생성모듈(114-1)은 '실행 프로세스 경로' 및 '실행 프로세스 파라미터'를 입력하여 자식 프로세스를 생성한다.

[27] 이와 같이 본 발명에서 제안하는 인젝션 프로그램(114)이 중지 모드로서 실행이 중지된 대상 프로세스의 메모리 영역에 인젝션 코드를 삽입한 후 중지 모드를 해제하여 대상 프로세스가 실행되도록 하면, DLL 인젝션의 시점을 조절할 수 있어 DLL 인젝션이 실패하는 것을 방지할 수 있다.

[28] 다음으로 코드 삽입모듈(114-2)은 대상 프로세스의 프로세스 핸들을 이용하여 대상 프로세스가 적재된 메모리 영역을 할당받아 인젝션하고자 하는 DLL 파일의 경로를 나타내는 인젝션 코드(Load DLL Code)를 삽입한다.

[29] 도 3은 코드 삽입모듈(114-2)에 의해 수행되는 인젝션 코드의 삽입 과정을 도시한 흐름도이다.

[30] 프로세스 실행모듈(114-1)에 의해 대상 프로세스가 자식 프로세스로 생성되었으므로, 코드 삽입모듈(114-2)은 대상 프로세스의 메모리 영역에 직접 접근할 수 있는 권한을 가진다. 따라서 코드 삽입모듈(114-2)은 대상 프로세스의 생성 과정에서 얻어진 프로세스 핸들을 이용하여 메모리(120)에서 대상 프로세스가 적재된 위치를 찾고, 해당 위치로부터 실행 코드를 읽어온다(S210).

[31] 코드 삽입모듈(114-2)이 읽어온 실행 코드는 마이크로소프트(Microsoft)에서 지원하는 실행 파일의 형식인 PE(Portable Excutable) 구조를 가진다. 도 4에는 PE 구조의 구성이 도시되어 있다. 코드 삽입부(120)는 실행 코드가 PE 구조인지 확인하기 위해서 도 4에 도시된 구성의 IMAGE_DOS_HEADER의 첫 두 바이트가 IMAGE_DOS_SIGNATURE(문자 MZ)인지 확인한다. 또한 IMAGE_NT_HEADER의 첫 네 바이트가 IMAGE_NT_SIGNATURE(문자 PE00)인지 확인한다. 그 결과 실행 코드가 PE 구조를 가지는 것으로 확인되면 PE 구조로부터 엔트리 포인트(EntryPoint)를 찾는다(S220).

[32] 다음으로 코드 삽입모듈(114-2)은 프로세스 핸들을 이용하여 프로세스 가상 메모리 영역의 일부를 할당받는다(S230). 할당받은 메모리 영역은 추후 삽입될 Load DLL Code가 저장되기 위한 공간이다. 또한 지역 변수에 LoadLibrary를 호출하고 원래의 엔트리 포인트로 리턴하도록 하는 Load DLL Code를 작성하여 기록한다(S240).

[33] 실행 코드의 엔트리 포인트를 변경하여 대상 프로그램의 실행 과정에서 Load DLL Code가 실행되도록 하기 위해서는 엔트리 포인트의 접근 권한을 쓰기 가능(writable)한 상태로 변경하여야 한다(S250). 또한 코드 삽입모듈(114-2)은 Load DLL Code가 실행된 후 원래의 엔트리 포인트로 리턴할 수 있도록 앞에서 생성된 지역 변수의 일부인 BackupEntry에 변경 전 현재의 엔트리 포인트를 저장한다(S260).

[34] 다음으로 코드 삽입모듈(114-2)은 앞에서 할당받은 가상 메모리 영역에 지역 변수에 저장하였던 Load DLL Code를 삽입하고(S270), 쓰기 가능한 상태로 변경된 엔트리 포인트에 Load DLL Code의 위치로 이동하기 위한 점프 코드(Jump Code)를 삽입한다(S280).

[35] 이상에서 설명한 과정에 의해 DLL 인젝션을 위한 Load DLL Code의 삽입 과정이 완료된다. 마지막으로 코드 삽입모듈(114-2)은 엔트리 포인트에 대한 접근 권한을 원상태로 복구시킨다(S290). 이와 같이 대상 프로세스의 메모리 영역에 Load DLL Code가 삽입된 후, 복원모듈(114-3)은 대상 프로세스의 중지 모드를 해제하여 대상 프로세스가 정상적으로 실행되도록 한다. 이때 코드 삽입모듈(114-2)에 의해 대상 프로세스의 메모리 영역에 삽입된 Load DLL Code로 인하여 인젝션된 DLL이 대상 프로세스와 함께 실행된다.

[36] 한편, 인젝션 프로그램(114)이 대상 프로세스를 중지 모드로 실행하여 DLL 인젝션을 수행하는 것은 다른 DLL 인젝션 기법과의 충돌을 막아 안정적으로 DLL 인젝션이 이루어지도록 하기 위해서이다. 이를 위해서는 앞에서 설명한 바와 같이 DLL 인젝션의 시점을 달리할 뿐 아니라 대상 프로세스를 다른 DLL 인젝션 기법으로부터 방어할 필요성이 있다.

[37] 일반적인 DLL 인젝션 방식에서는 사용자 권한으로 대상 프로세스가 실행될 때 동일하게 사용자 권한으로 실행되고 있던 DLL 인젝션을 시도하는 프로세스가 대상 프로세스의 실행을 감지하고 DLL 인젝션을 시도하게 된다. 그에 따라 DLL 인젝션이 성공하고, 대상 프로세스가 실행될 때 인젝션된 DLL이 로드되어 함께 실행된다.

[38] 본 발명에서 제안하는 인젝션 프로그램(114)에는 권한 변경모듈(114-4)이 더 구비되어 사용자 권한으로 실행되는 프로세스에 의한 DLL 인젝션으로부터 대상 프로세스를 방어하기 위하여 대상 프로세스를 시스템 권한으로 실행한다. 이와 같이 함으로써 일반적인 DLL 인젝션 프로세스들은 권한이 높은 프로세스에 대한 DLL 인젝션을 실패하게 되며, 인젝션 프로그램(114)은 다른 DLL 인젝션 기법이 대상 프로세스에 적용되는 것을 막음과 동시에 안정적으로 DLL

인젝션을 수행할 수 있다.

[39] 도 5는 권한 변경모듈(114-4)에 의해 대상 프로그램이 시스템 권한으로 실행되도록 하는 과정을 도시한 흐름도이다.

[40] 도 5를 참조하면, 권한 변경모듈(114-4)은 먼저 모든 윈도우즈 운영체제에서 항상 시스템 권한으로 동작하고 있는 프로세스, 예를 들면 'winlogon.exe'의 프로세스 아이디(ID)를 얻어온다(S410). 프로세스 아이디를 얻어오는 방식으로는 마이크로소프트사에서 프로세스 정보를 얻을 수 있도록 제공하는 Tool Help 라이브러리의 스냅샷(snapshot) 방식을 사용한다.

[41] 다음으로 권한 변경모듈(114-4)은 얻어온 프로세스 아이디를 사용하여 'winlogon.exe'의 프로세스 핸들을 열고(S420), 이를 사용하여 'winlogon.exe'의 토큰(token)을 연다(S430). 하나의 토큰에는 로그온 세션(logon session)에 대한 보안 정보가 포함되어 있으며, 사용자가 운영체제에 로그온할 때 만들어진 하나의 토큰을 복사하여 유저가 실행하는 모든 프로세스가 실행된다.

[42] 윈도우즈 운영체제에서는 각각의 사용자를 독립적으로 관리하기 위하여 세션으로 권한을 관리하는데, 기본적으로 하나의 사용자가 로그온하면 하나의 세션을 할당받으며, 할당된 세션 내에서 작업을 수행한다. 로그온한 사용자에게 할당되는 세션과 비슷한 방식으로 최상위 권한인 시스템에 대하여도 세션이 할당되고, 운영체제 내부의 중요한 작업이 처리된다.

[43] 따라서 권한 변경모듈(114-4)이 시스템 권한으로 동작하는 'winlogon.exe'의 토큰을 얻어오는 것은 시스템에 할당된 세션에 대상 프로세스를 진입시키기 위한 권한 획득의 의미를 가지게 된다.

[44] 다음으로 권한 변경모듈(114-4)은 얻어온 토큰을 복사하여 새로운 토큰을 생성한다(S450). 복사된 토큰은 'winlogon.exe'의 시스템 권한을 내포하고 있다. 한편, 현재 로컬 시스템을 구분할 수 있는 값인 LUID를 얻어(S440) 복사한 토큰을 사용할 수 있도록 권한을 변경한다(S460).

[45] 마지막으로 권한 변경모듈(114-4)은 프로세스 생성모듈(114-1)이 복사된 토큰을 사용하여 대상 프로세스를 생성하도록 한다(S470). 이때 프로세스 경로 및 프로세스 파라미터를 지정함으로써 원하는 프로세스를 생성하게 된다. 또한 본 발명에서 대상 프로세스가 본 발명에 따른 DLL 인젝션 장치의 자식 프로세스로서 생성된다는 점은 앞에서 설명한 바와 같다. 한편, 프로세스 생성모듈(114-1)은 운영체제에서 시스템 권한으로 동작하는 프로세스의 프로세스 아이디(ID)를 사용하여 대상 프로세스가 아닌 런처 프로세스를 시스템 권한으로 실행할 수 있으며, 그에 따라 생성되는 자식 프로세스인 대상 프로세스 역시 시스템 권한으로 실행된다.

[46] 이후 대상 프로세스는 시스템 권한으로 실행되며, 코드 삽입모듈(114-2)은 외부 프로세스에 의한 DLL 인젝션의 영향을 받지 않고 대상 프로세스가 적재된 메모리 영역에 Load DLL Code를 삽입하여 원하는 DLL이 실행되도록 한다.

[47] 이상에서 설명한 것과 같은 권한 변경모듈(114-4)에 의한 DLL 인젝션 방어

기법은 DLL 인젝션 기법을 분석하여 DLL 인젝션의 발생을 모니터링함으로써 방어하는 기존의 기법에 비하여 원천적으로 대상 프로세스를 DLL 인젝션으로부터 방어할 수 있다는 효과를 가진다.

[48] 도 6은 본 발명에 따른 DLL 인젝션 방법에 대한 바람직한 실시예의 수행과정을 도시한 흐름도이다.

[49] 도 6을 참조하면, 운영 체제(112)로부터 사용자가 실행을 명령한 프로그램에 대응하는 대상 프로세스의 생성 여부가 통지되면(S510), 인젝션 프로그램(114)의 프로세스 생성모듈(114-1)은 런처 프로세스를 실행하여 런처 프로세스의 자식 프로세스로서 대상 프로세스를 생성하고, 생성된 대상 프로세스를 중지 모드로 설정한다(S520). 이때 앞에서 설명한 것과 같이 권한 변경모듈(114-4)이 시스템 권한으로 실행되는 'winlogon.exe'의 토큰을 복사하여 대상 프로세스가 시스템 권한으로 생성되도록 할 수 있다.

[50] 다음으로 코드 삽입모듈(114-2)은 대상 프로세스의 프로세스 핸들을 이용하여 대상 프로세스가 적재된 메모리 영역을 할당받아 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입한다(S530). 코드 삽입모듈(114-2)의 자세한 동작은 앞에서 도 3을 참조하여 설명한 바와 같다.

[51] 코드 삽입모듈(114-2)에 의해 대상 프로세스의 메모리 영역에 DLL 파일을 실행하기 위한 Load DLL Code가 삽입되면, 복원모듈(114-3)은 대상 프로세스의 중지 모드를 해제하여 대상 프로세스가 실행되도록 한다(S540).

[52] 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[53] 이상에서 본 발명의 바람직한 실시예에 대해 도시하고 설명하였으나, 본 발명은 상술한 특정의 바람직한 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능한 것은 물론이고, 그와 같은 변경은 청구범위 기재의 범위 내에 있게 된다.

청구범위

[청구항 1] 대상 프로세스에 DLL 인젝션을 수행하기 위한 인젝션 프로그램 및 운영 체제를 실행하는 프로세서; 및
상기 운영 체제, 상기 인젝션 프로그램 및 상기 대상 프로세스가 저장되는 메모리;를 포함하며,
상기 인젝션 프로그램은,
런처 프로세스를 실행하여 상기 런처 프로세스의 자식 프로세스로서 상기 대상 프로세스를 생성하고, 상기 대상 프로세스를 중지(suspend) 모드로 설정하는 프로세스 생성모듈;
상기 대상 프로세스의 프로세스 핸들을 이용하여 상기 대상 프로세스가 적재된 메모리 영역을 할당받아 상기 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입하는 코드 삽입모듈; 및
상기 대상 프로세스의 중지 모드를 해제하여 상기 대상 프로세스가 실행되도록 하는 복원모듈;을 포함하는 것을 특징으로 하는 컴퓨팅 장치.

[청구항 2] 제 1항에 있어서,
상기 인젝션 프로그램은,
상기 운영 체제에서 시스템 권한으로 동작하는 프로세스의 프로세스 아이디(ID)를 사용하여 상기 대상 프로세스가 시스템 권한으로 실행되도록 하는 권한 변경모듈을 더 포함하는 것을 특징으로 하는 컴퓨팅 장치.

[청구항 3] 제 2항에 있어서,
상기 권한 변경모듈은 상기 운영 체제에서 시스템 권한으로 동작하는 프로세스의 프로세스 아이디를 사용하여 얻어진 프로세스 핸들에 의해 토큰을 열고, 상기 토큰을 복사하며,
상기 프로세스 생성모듈은 상기 복사된 토큰을 사용하여 상기 대상 프로세스를 생성하는 것을 특징으로 하는 컴퓨팅 장치.

[청구항 4] 제 1항에 있어서,
상기 프로세스 생성모듈은 상기 운영 체제에서 시스템 권한으로 동작하는 프로세스의 프로세스 아이디(ID)를 사용하여 상기 런처 프로세스가 시스템 권한으로 실행되도록 하는 것을 특징으로 하는 컴퓨팅 장치.

[청구항 5] (a) 운영 체제로부터 사용자가 실행을 명령한 프로그램에 대응하는 대상 프로세스의 생성 여부를 통지받는 단계;
(b) 런처 프로세스를 실행하여 상기 런처 프로세스의 자식 프로세스로서 상기 대상 프로세스를 생성하고, 상기 대상

프로세스를 중지 모드로 설정하는 단계;

(c) 상기 대상 프로세스의 프로세스 핸들을 이용하여 상기 대상 프로세스가 적재된 메모리 영역을 할당받아 상기 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입하는 단계; 및

(d) 상기 대상 프로세스의 중지 모드를 해제하여 상기 대상 프로세스가 실행되도록 하는 단계;를 포함하는 것을 특징으로 하는 DLL 인젝션 방법.

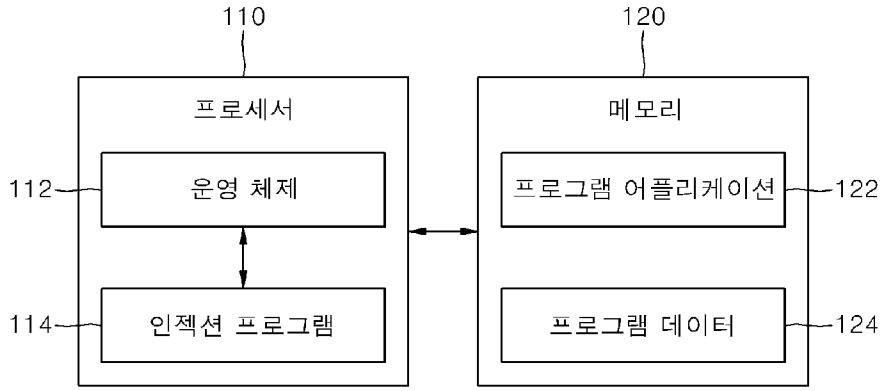
[청구항 6] 제 5항에 있어서,

상기 (a) 단계 및 상기 (b) 단계의 사이에,

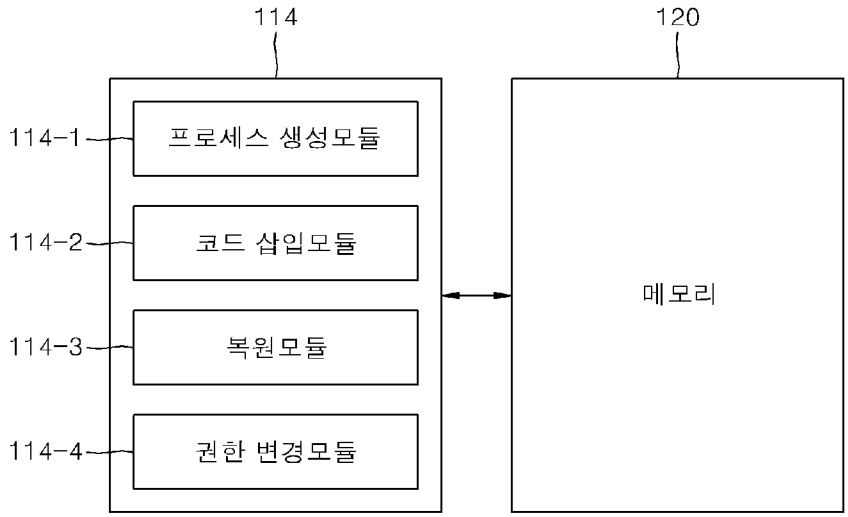
(e) 상기 운영 체제에서 시스템 권한으로 동작하는 프로세스의 프로세스 아이디를 사용하여 상기 대상 프로세스가 시스템 권한으로 실행되도록 하는 단계를 더 포함하는 것을 특징으로 하는 DLL 인젝션 방법.

[청구항 7] 제 5항 또는 제 6항에 기재된 DLL 인젝션 방법을 컴퓨터에서 실행하기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

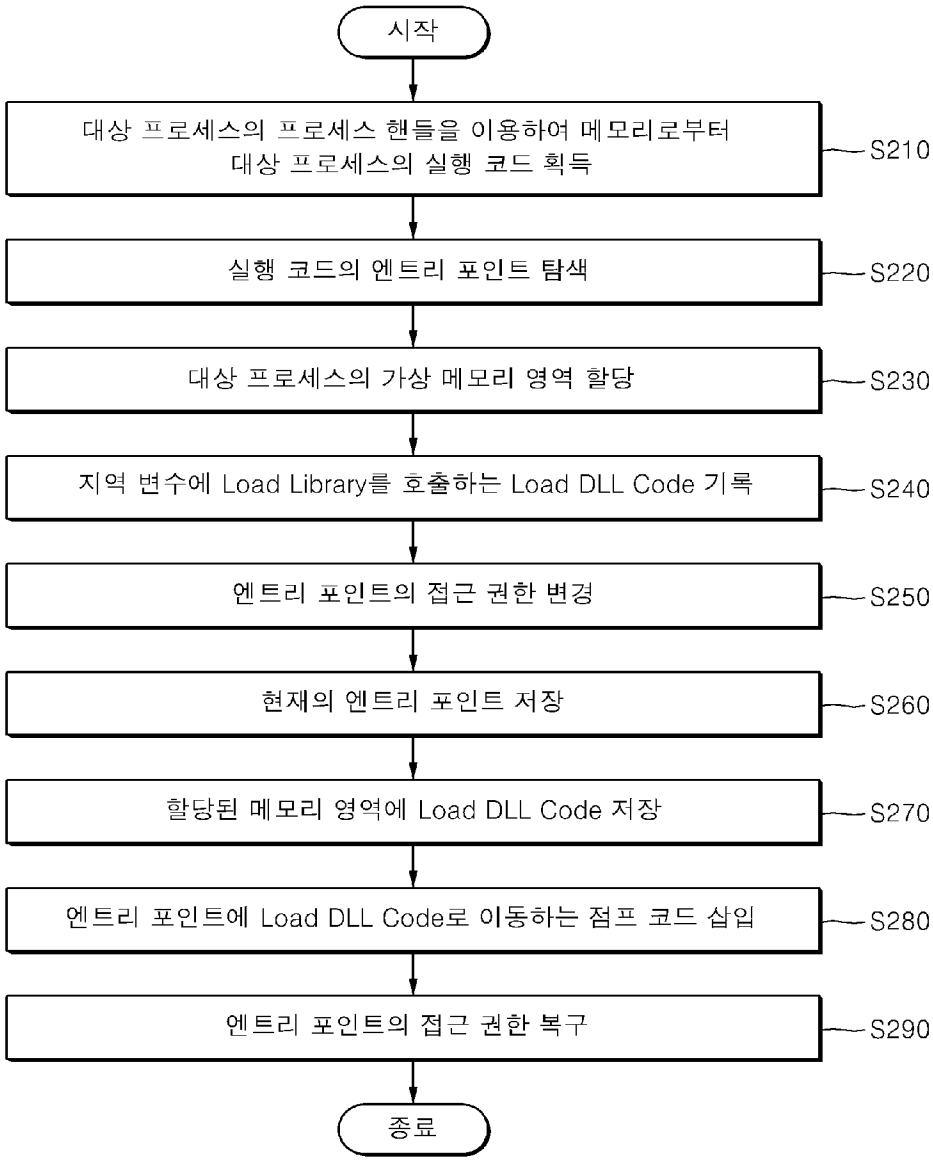
[Fig. 1]


110
120
프로세서
메모리
112
운영 체제
프로그램 어플리케이션
122
114
인젝션 프로그램
프로그램 데이터
124

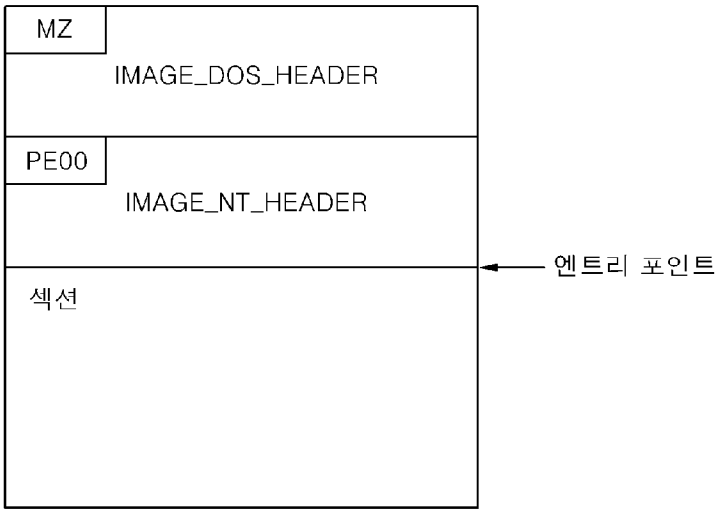
[Fig. 2]


114
120
114-1
프로세스 생성모듈
114-2
코드 삽입모듈
114-3
복원모듈
114-4
권한 변경모듈
메모리

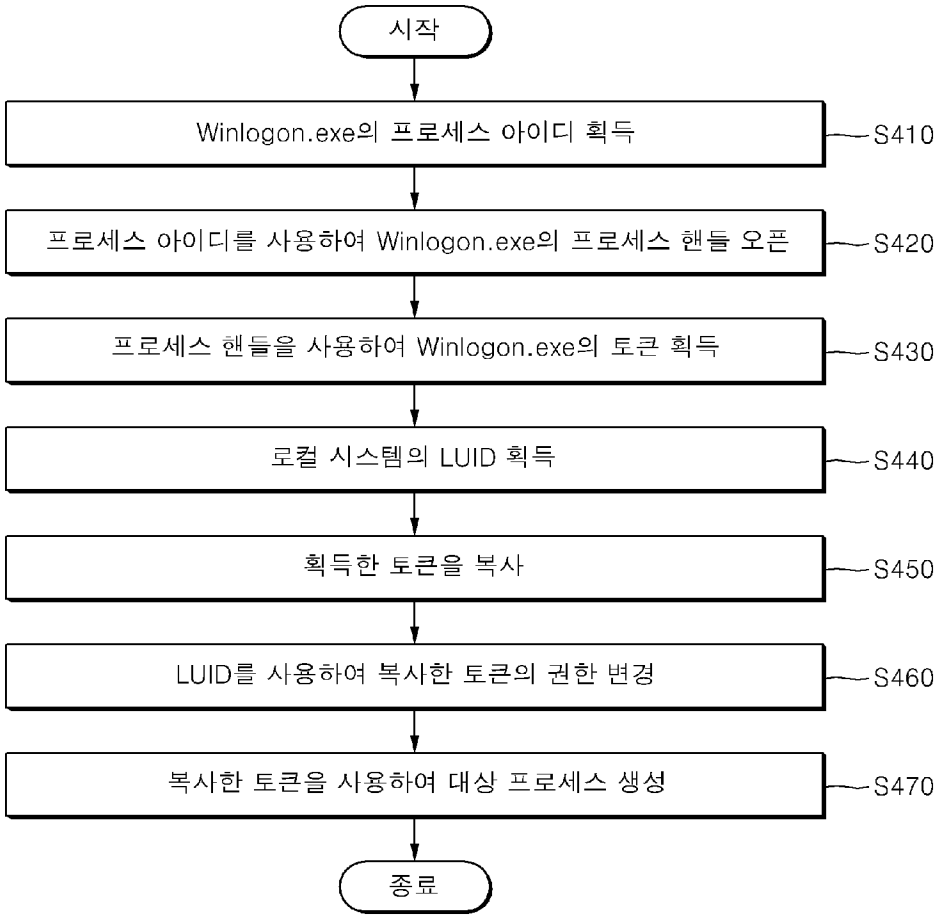
[Fig. 3]


시작
대상 프로세스의 프로세스 핸들을 이용하여 메모리로부터 대상 프로세스의 실행 코드 획득
S210
실행 코드의 엔트리 포인트 탐색
S220
대상 프로세스의 가상 메모리 영역 할당
S230
지역 변수에 Load Library를 호출하는 Load DLL Code 기록
S240
엔트리 포인트의 접근 권한 변경
S250
현재의 엔트리 포인트 저장
S260
할당된 메모리 영역에 Load DLL Code 저장
S270
엔트리 포인트에 Load DLL Code로 이동하는 점프 코드 삽입
S280
엔트리 포인트의 접근 권한 복구
S290
종료

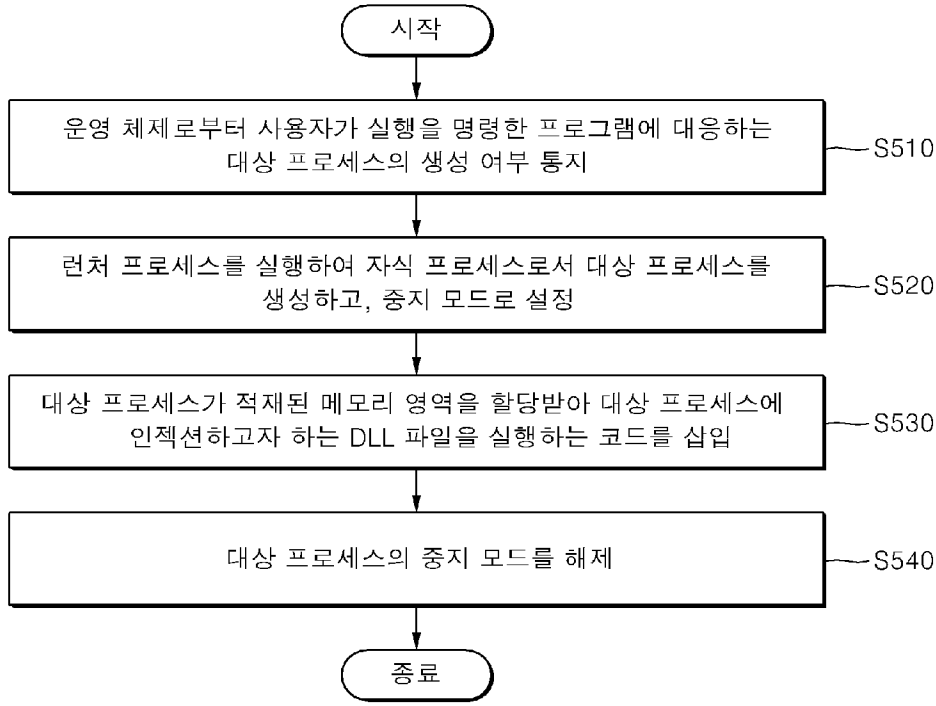
[Fig. 4]


MZ
IMAGE_DOS_HEADER
PE00
IMAGE_NT_HEADER
엔트리 포인트
섹션

[Fig. 5]


시작
Winlogon.exe의 프로세스 아이디 획득
S410
프로세스 아이디를 사용하여 Winlogon.exe의 프로세스 핸들 오픈
S420
프로세스 핸들을 사용하여 Winlogon.exe의 토큰 획득
S430
로컬 시스템의 LUID 획득
S440
획득한 토큰을 복사
S450
LUID를 사용하여 복사한 토큰의 권한 변경
S460
복사한 토큰을 사용하여 대상 프로세스 생성
S470
종료

[Fig. 6]


시작
운영 체제로부터 사용자가 실행을 명령한 프로그램에 대응하는 대상 프로세스의 생성 여부 통지
S510
런처 프로세스를 실행하여 자식 프로세스로서 대상 프로세스를 생성하고, 중지 모드로 설정
S520
대상 프로세스가 적재된 메모리 영역을 할당받아 대상 프로세스에 인젝션하고자 하는 DLL 파일을 실행하는 코드를 삽입
S530
대상 프로세스의 중지 모드를 해제
S540
종료