from:5up3rh3i'blog

浅谈 web 漏洞挖掘—特殊变量 fuzz?

目前对 web 应用程序漏洞的挖掘方法有：
1． 纯粹的黑盒[如：AWVS、APPScan 等]
2． 纯粹的白盒[如：grep 等]
一些其他的方法如 coolq 大牛基于 php 内核编译器[这个思路早些跟 boy 说过，不过马来人比我还懒，谁叫按不懂 c 呢?]。

在《web 代码安全边缘性问题》里，提到了一些代码“非主流？”一些漏洞模式，里面提到的查找方法主要是基于 grep“静态”分析代码。这里我给大家介绍下我的“动静结合”的一些其他的思路。

一、 实现思路
基于 http 代理把构造的变量提交个目标。“动静结合”首先通过 grep 的静态分析出“可能”有问题的变量[静]，然后构造这些变量通过我们的 fuzz 工具，也就是代理服务器提交给目标，如果通过浏览器[ie、ff 等]触发。

二、 特殊变量
1． Header 变量
Header 变量的问题在《web 代码安全边缘性问题》提到过，很容易被忽视，那么我们基于 http 代理我们很容易植入我们的变量如：
$headers{"X-Forwarded-For"} = "Test31425926’";
a.对于基于 sql 的程序很容易造成 sql 注射如：
Sablog-X Ver 1.1 getip() Vulnerability
http://superhei.blogbus.com/logs/2006/09/3436056.html

b.对于文本数据库很容易造成 phpcode 注射如：
Bmb
D:\>Findstr /S /I /N /dbig_smile:\phproot\bmb2007\bmb\ "Test31425926" *.php
D:\phproot\bmb2007\bmb\:
datafile\guest.php:2:<?php //|娓稿?|1163859032|Test31425926?|f|0|Firefox 1.5.0.8|Windo
ws XP|t||
容易出现问题的 header 变量：X-Forwarded-For、User-Agent、CLIENT-IP … 还有 Referer 对限制来路的突破

2． 全局变量
a. zend_hash_del_key_or_index_vulnerability ：
http://www.hardened-php.net/hphp/ze…nerability.html
b. php5-globals-vulnerability：http://www.ush.it/2006/01/25/php5-g…-vulnerability/
c. 变量/数组变量未初始化

在 fuzz 脚本里我们可以通过 cookie 提交。

3． Upfile 漏洞 fuzz

常见模式：

1． 直接上传[没有任何过滤]

2． 被动过滤[过滤不完全：php3 php4 asa cer …]

3． Content-Type

4． Null 截断

5． Apache 文件解析特性

6． PHP RFC1867 Vul

7． 后缀大小写规则

4． 二次攻击的 fuzz

二次攻击的主要类型是变量通过 UPDATA 或者 INSERT 提交给数据库[包括文本数据库]，然后在提取造成二次攻击，如果我们通过 fuzz 脚本自动提交构造的变量，再通过浏览器正常的访问来探测触发。