==Ph4ntOm Security Team==

Issue 0x03, Phile #0x02 of 0x07

```
|=-----------------------------------------------------------------------=|
|=---------------------=[       专访wordexp       ]=---------------------=|
|=-----------------------------------------------------------------------=|
|=-----------------------------------------------------------------------=|
|=-------------------=[           By wordexp           ]=-----------------=|
|=-----------------------------------------------------------------------=|
```

[目录]

一、据您所知现在都还有哪些严重的0DAY没被公开？

主流应用软件方面目前微软公司的IE6/7和PPT 2003 SP3前一阵子就有EXP在外面跑了，adobe公司的FLASH产品中也有一个能被利用的漏洞，拿到的人应该也不少，往后的一到三个月内就应该出补丁或是有相关新闻，当然以我们目前的视界能看到的只有很少很少的一部分，其实国外的很多安全机构比如：idefense和zdi可以确定还有不少没被公开的漏洞，只是这些漏洞可能并不是我们想象的那么通用，成功率也许有限。

其实经常听到朋友问这个问题，说白了就是个消息的打听，这个一方面要提高自己的敏感度，注意随时关注国内外相关网站的新闻，比如NORTON和MCAFEE的网站经常有一些抓到的0DAY样本的新闻，有时还有一些细节。还有就是消息的共享，你提前能知道消息并告知朋友，以后也许人家也会这样对你。

二、挖漏洞有什么窍门？可以具体谈谈fuzzer怎样构造样本么？

这个问题太为难我了，很多人比我更有资格回答这个问题。我只是斗胆胡说几句。

要说窍门，应该是不同软件的洞窍门还不一样，然后还要看挖洞的目的，如果是为了出名在bugtraq之类的邮件列表上能多露几次脸，那么可以尽可能的找那种用户少关注少的软件特别是WEB脚本程序的洞。如果是公司有任务必须往CVE、MS上报多少条漏洞那么可以找大公司的二三线产品的漏洞或是有一定用户数但版本很久不更新的软件的洞，而且这些洞是不一定要可利用的。如果是为了混zdi、idefense那么可以把fuzz到的POC只要看起来有可能被利用的都提交上去，也可以找默认情况下不支持的功能的洞，多少可以骗点钱。

如果是要挖卖得出去也能利用的洞，比较通用的一些窍门我能想到的是：

1. 找大众软件生僻功能，生僻协议/文件格式的洞
2. 找几乎没有文档化的功能的洞
3. 新版本软件为了向下兼容所支持的老协议/老文件格式的洞
4. 新版本软件增加的新功能/新格式
5. 不容易fuzz到的洞，比如数据是加密/压缩/编码过的，或是有验效的
6. 某软件某功能刚出了漏洞，马上测试其它同类软件同类功能是否有类似漏洞
7. 多分析老漏洞，善于总结前人挖漏洞的经验技巧，很多不同的洞其实都有相类似的发掘方式和思路

第二个问题，我以文件型漏洞举例子，fuzz样本的构造，首先是按照上述几个窍门来生成原始模版，这样相对可以弄出一些人家不太容易fuzz到的数据格式结构，当然在生成原始模版的前期功课也是很花时间的。做好样本后就是写具体的fuzz程序， 如果对文件格式比较熟，那么可以节约很多的时间，我比较喜欢的一个办法是fuzz某一些功能的洞，那么就先看格式，把数据在文件中的位置先手工定位，然后小粒度的测试，要注意的是可能与某功能相关联的数据比较杂乱数据很可能并不是连续存放的。一般1-4KB的数据要不了多少时间就可以手工测试完毕。另外具体测试时，数据替换的长度（一次替换几个字节），替换的内容也是非常重要的。为此我们将提供一个PPT 2003 sp3的“0day”poc，在这个“0day”中数据替换的步进就必须为1字节，而且值也必须为一个固定的值才能触发出错。最后要注意的就是错误的捕捉，有些洞是打开就退出进程，有些是打开要停顿一定时间才退出进程，有些是CPU 100%程序挂起，有些是关闭时触发，甚至有些是文档打开后进行某种操作才会触发，当然还有一种情况进程不退出，也没有提示，也不出错，象这种情况一般依靠进程/窗口/CPU来检测错误的fuzz就失效了。

三、请问您对0day市场有什么看法？

很复杂的一个圈子，搞技术的不搞技术的啥人都有，不过目前看来很多都是为了各种利益混这个圈子。简单说就是：

池塘不大但人杂，水深。

四、您建立wordexp这个blog的初衷是什么？为什么叫wordexp，而不是pdfexp或者是exclexp？

初衷就是团队成员工作之余发发劳骚，聊聊八卦的地方，希望大家别见怪。另外这个名字是因为我们几个人搞客户端的漏洞都比较多，所以随便就取了这么个名字。

五、请问0day是咋来的？

最初当然是某个人找出来的。从这个0DAY的发掘者到最终的用户中间可能会只有一层关系，也可能会有N层关系，也许直到这个0DAY被补上，使用者也不知道洞是谁挖到的。下面举几个例子吧：

情况一：A挖到一个0day，但对黑产没有了解或接触，或者也不想靠这个赚钱，或者觉得漏洞不值钱，或者压根以为漏洞不能够被利用，那么A有可能把这个漏洞公开给类似PST的网站，网站上的代码通常是POC或是只有一部分细节。这时黑产中的漏洞研究者B，很快会看到这个消息，并且分析POC然后写出EXP。随后B再联系具体的使用者C或是自己使用。最终或是因为这个0DAY的POC被公开，也或许因为EXP被杀毒软件公司抓到样本等等，软件厂商推出补丁。在这个过程中B可能是一个人也可能是很多水平各不相同的人，所以公开的0DAY的EXP有时是千差万别，有的好用，有的很差。

情况二：A是黑产中的一员，挖到一个0DAY并卖给X，或是接受使用者X的定制并找到0DAY，X偷偷的使用0DAY，这样的情况一般0DAY的生存期会比较长一些，因为这才算是真正的私洞，知道的人不多。但是在X的使用中，EXP可能被别的黑产从业者Y抓到样本，然后Y把样本提供给技术员T分析并重新写出EXP，而成果Y和T分享。这时T可能再次把EXP卖给其它的黑产使用者W，同样X或Y在使用一段时间后也可能交换或者再出手给其它的买家，而且这个过程是可以无限次重复的，当然时间越久知道的人越多，0DAY就越掉价。

情况三：白帽子A挖到一个0day，并提交给软件厂商B。假设A是个真真正正的白帽子，也假设这个0day的确也只被A发现了，但0day到了厂商B那儿，最终会找公司内负责安全的部门对漏洞进行研究，假设这个公司内部的研究者是C，C有可能在圈子中也有其它从事黑产的朋友或是自己本身就偷偷的在参与黑产，那么在金钱或是感情的作用下C完全有可能违背道德，写出EXP出售或是使用。象这种情况，0day可能刚出来没几天就被补上了，或者圈子里的很多人根本就没机会见到0day，知道有这么个东东的时候早就被补上了。

情况四：软件厂商B在代码审计或软件测试过程中发现了漏洞，自己在新版本中偷偷的补上了漏洞，但并没有在相对老的版本中打补丁，也没有公开任何细节和公告。 研究人员A通过补丁比较，直接定位出老版本中的漏洞位置，然后动态调试找到触发方式，并写出EXP，由于很多情况下老版本的软件反而用户更多，所以这样的0day还是有一定的价值。

情况五：研究员A挖到了一个Nday而这个Nday以前在圈内并不为人所知，或是研究员A研究出某Nday的新利用方式，比如说可以和某某软件结合看起来和以前的EXP完全不一样，或是成功率有很大的提高。研究员A以较低的价格出售给中间人B。B拿到EXP后，发现圈内还没人有，成功率各方面也还不错，于是号称0day到处叫卖。如果买家发现问题，B就装傻说自己也上当了。现在象B这样的人其实也是不少的， 因为很多最终用户对技术并不是特别懂，而且有些Nday测试起来也不是那么简单，如果刚好EXP效果不错，可能就忽悠过去了。那么我们能看到的情况就是，江湖传言又出了个0day，或是某某手上有0day，但等呀等就是见不到东东，最终传言不了了之或是被人家揭发出来。

所以要搞到0day可以自己挖，可以补丁比较，也可以分析已公开信息快速写出EXP，可以买，也可以换，不要命也可以偷抢骗，技术手段非技术手段都是可能的。也正是因为上述情况的多样性，所以经常有不怎么搞技术的人，手上也有些0day。

六、请问在安全圈子谁是您的偶像？

我的偶象是那种啥技术不懂，还能发大财的，不PF不行。

七、请问溢出这面红旗还能打多久？

仅仅是溢出这块，我们团队里面意见也大不相同，另外几个成员还是比较乐观的，如果比较全面的分析这个问题，首先要看站在什么人的角度来看这个问题，是黑产工作者是安全公司还是软件生产商。假设以黑产工作者的角度来看，那么我是非常非常悲观的，因为溢出漏洞从技术角度上说：有一个通用性和成功率的问题，直观的说就是有一个效果的问题，再深一点说就是经济成本的问题。往后走溢出漏洞单从个数上说还是会有很多的。但是现在从编译器和OS（/GS、/SafeSEH、/DYNAMICBASE、DEP、PEB随机等等）到CPU（NX），软件公司和硬件厂商已经越来越关注安全问题，几十年来溢出漏洞最关键的命脉无非是数据能够被当做代码来执行，以前这一点基本上不被软硬件厂商所重视，这几年来人家开始重视了，开始从体系上解决这个问题，那么这个命脉也将因为各种防范检测技术的运用被卡得越来越死，另外现在很多软件也有自动升级功能了。

往后走是个什么样的情况，我想应该是上面提到的各种技术随着新型CPU和OS的占有率越来越高，被越来越多的应用。一个溢出漏洞的成功率将会大大下降，再加上主流软件公司的产品也越来越安全，以后那种一个漏洞打天下的局面将会越来越少(现在黑产工作者的网马都是漏洞合集了，无非就是提高成功率)，具体的情况也许就是现在我有一个IE的0DAY，100个人看也许能中10-20个，以后可能手上能用的就变成某个第三方控件的0DAY，100个人看网页就能中1-2个吧。当你使用溢出漏洞的时间，人力，金钱成本和产出完全不成正比的时候，也基本上算溢出这面红旗倒下的时候。

估计也就三四年以后，具体指标就是上面提到的各种检测技术的普及率，至少往后的发展不会是车到山前必有路。如果把挖溢出漏洞当成一个产业，也就是个夕阳产业。

八、您对我们杂志以及我们小组的发展有什么建议？

不走商业路线是正确的，反正你们那群人也不差钱，就不定期搞搞科普工作吧，为普及中国安全事业做点贡献，同时也可以锻炼你们各方面的能力，继续努力!

-EOF-