# 开源与安全的纠结

## ——开源系统的安全问题笔记

■ 文/安天实验室首席技术架构师 江海客

20世纪90年代中期开始，开源是否导致安全成为一个公众性的话题。在关于Linux和Windows优劣的旷日持久的争论中，很重要的一条是Linux是否比Windows安全。宣扬Linux安全论的大多是Linux的开发者和爱好者。但在这一点上，有统计支持的分析过程并不多见，也缺少在安全机制、漏洞分析、漏洞挖掘方法等角度的对比分析。Linux安全论群体最核心的信心支撑来自于Linux的开源性。

当这种信仰驱动的信心逐渐传染至一般的信息工作者和网管阶层的时候，我们诧异地发现，这种比对已经不只存在于开源操作系统和主流商用操作系统之间了，在WEB平台、数据库系统等领域，这种信心被放大为一个声音——“开源导致安全”，并且这种声音正在被用于游说各国政府……

现在是我们正本清“源”的时候了。

### 一个古老的“伪”命题

“Windows由1000个人开发，Linux由全世界人一起开发，你说哪个更安全！”

笔者第一次遭遇Linuxer，就听到了这句话。

这次讨论发生于90年代末期，笔者提出如下观点：“由于Win9x系统是基于桌面设计的，而Linux是基于服务器应用设计的，显然Linux开放的服务要多于Win9x，所以从远程攻击获取权限的角度，攻击Linux会比攻击Win9x容易。因此从安全性的角度，Linux显然不适合于作为终端系统”。

笔者很具象的论述，遭到了“开放导致安全”的哲学式反击。

此时，还没有发生文摘一中的争论。

●文摘一（引自http://en.wikipedia.org/wiki/Linus'_Law）

开源活动的领军人物Eric Raymond在开源的纲领性文献《大教堂和市集》中说：Given enough eyeballs, all bugs are shallow. 即“足够多的审视可以让所有的bug无所遁形”。这句话特别有名，被称之为Linus定律。

微软工程师Shawn Hernan对此的反驳是，上述定律基于两个假设：1、开源软件比闭源软件有更多的代码审计（Code Review）；2、代码审计可以让软件更安全。但在现实场景中，这两个假设并非都成立。

对于这场争论，我们希望补充的是，虽然漏洞是可被攻击者利用的一种特殊的bug，但漏洞挖掘的方法和一般程序纠错的方法存在一定的专业性差异。

而大多数设计人员和开发者实际上并没有受过对抗漏洞挖掘、错误注入或者格式模糊（fuzzing）的专门训练。他们所编写的稳定、可靠、高效的代码，多数未必是安全的代码。从这个角度，笔者还有一个（缺少数据证实的）观点：大部分重大漏洞的发现，来自于少数的安全分析人员，而不是庞大的开发者团体自身。

源码审计是诸多安全分析方法中最直观的一种，从这个意义上来看，开源系统的安全初衷是为安全分析者提供更多的分析资源。但除了能获得源码，建立安全的开源系统还有三个要素缺一不可：能在有限的具有安全分析能力的人之中，能获得足够多的参与者；参与者愿意把分析结果反馈给开发者以改善系统；项目开发和维护人员积极地接受问题并修补。

开源安全论的明显软肋在开源形成一种潮流的本身时已经出现， Linux、Firefox等少数系统或软件被聚焦从而使他取得了感觉不错的改进体验，但这种经验具有普适价值么？让我们来看文摘二的观点。

●文摘二(引自http://www.zdnet.co.uk/news/application-development/2006/10/26/red-hats-cox-warns-on-open-source-security-39284341/)

Linux内核主要维护人员之一Alan Cox在2006年伦敦的LinuxWorld大会上发言称：

“许多开放源代码项目远谈不上安全。许多资金都被用来破坏开放源代码系统的安全。媒体上经常

有这样的字眼：开放源代码软件更安全、更可靠，缺陷也更少。这是一种危险的观点。”

“许多分析都只关注知名度很高的项目。对SourceForge上150个项目的分析显示，它们在安全方面的表现不如Linux。只有部分项目的‘高质量’是名符其实的。”

这是来自开源阵营自身的观点，这是一种资源摊薄的困境。这种摊薄也证明安全问题从一定意义上说是一种成本代价问题。

笔者始终的观点是安全来自于如何有效地聚合善意而专业的关注，并以此驱动改善。开源与否，是在如何聚合这种专注方面的不同组织方法。在这方面，尽管开源系统号称具有所谓“全世界”的参与的可能性，但由于非开源系统先行找到了盈利模式（销售），往往就具备了承担更大安全成本的能力。一个典型的例子就是微软专门而庞大的应急处理体系。

### 代码开放与算法公开

开源安全论者，最具有类比支撑价值的观点是关于加密算法公开与代码开放的类比。他们认为既然“公开的算法一定比保密的算法安全”，那么开源系统一定比封闭系统安全。他们声称找到的重要的支持者来自安全阵营，那就是Bruce Schneier（著名密码学家、安全研究者，著有《应用密码学》等）。

●文摘三（引自http://www.schneier.com/crypto-gram-9909.html#OpenSourceandSecurity）

Bruce Schneier：

Public security is always more secure than proprietary security. It's true for cryptographic algorithms, security protocols, and security source code. For us, open source isn't just a business model; it's smart engineering practice.

公开安全总是比私有安全要好，对加密算法、安全协议、安全源码来说都是如此。对我们而言，开源不仅仅是一种商业模式，还是一种更好的工程实践。

作为安全学徒，对于宗师的观点是无法不谨慎思考的。我们认为Bruce Schneier的观点并非泛指各种系统，而是仅就“加密算法、安全协议、安全源码”这一狭窄的领域，这一领域的特点是：

- 这些算法本身的逻辑安全性有数学可证明性
- 经过反复的数学分析淘汰而剩下的主流算法是有限的
- 实现主流算法所需的代码规模比较小
- 算法的标准化实现被广泛复用和考察

后三点可以回归到前一节的观点，安全性来自于足够专业而善意的审视和快速的修补。

其实从密码学的历史上说，最初并不存在所谓“公开的算法一定比保密的算法安全”的观点，否则盟军就没有必要花巨大的代价去获取“恩尼格玛”（Enigma）了。与其说是公开导致了安全，不如说是在反复的密码攻防实战中，密码学家逐渐认识到：承载算法运行过程的载体是无法保证不被敌方获取，“算法的安全性，不应该依赖于算法本身的保密”。因此，数字化时代到来后，算法设计被赋予了这样一个前提——安全的算法应依赖于密钥保密、而不是算法保密。这样，只要实现对作为静态数据的密钥的安全保管即可，而不需要用鲜血和生命去保护那个笨拙的黑匣子。

同时，不能把安全算法领域普适到广泛开发领域的重要原因，是因为算法本身具有一定的数学可证明性。例如PKI体制整体上是基于NP问题的；RSA算法的安全性尽管一直没能证明等价于大整数分解，但至少是基于大整数分解的。另一方面，从操作系统到应用软件的安全性，尽管存在若干安全开发原则，但不可能存在理论性证明。

开源主义者曾把美国政府选定标准算法的过程作为开放导致安全的案例——分组加密的标准算法AES、目前的HASH标准 SHA-1，以及正在遴选的SHA-3。这些确实来源于全球性的公开征集和选拔过程，但我们也需要看到的是：

它们并不是通过开放某一个算法并不断修正达到目的，而是组织征集了多个备选对象，在全球密码学家和研究人员的分析、破解和评估中不断淘汰。

它们的安全性并非来自于开放本身，而在于通过开放的方法，组织一个选秀过程。与其说是开放性保证安全，还不如说遴选过程无法在算法封闭的前提下实现。

这个过程中，对淘汰起到关键作用的是全球大大小小的密码学术团队的几乎全员参与，包括中国的密码学家们也参与其中。基本上来看，其对数学精英的聚合效应甚至超出了当年的哥德巴赫猜想等数论难题。这种全世界数学家给美国打工的组织形式，基本上只能是一个不具备类比性的个案。

### 传统对决与对称性博弈

Linux和Windows哪个更好？这种比较被粉丝群逼成一场超过10年的马拉松讨论，从Win9x一直比较到Windows 7。

笔者最初以为这是一场关公战秦琼式的比较，因为当时Windows只是一个桌面系统。用服务器系统和桌面系统比较稳定性，就像用桌面系统和服务器系统比较易用性一样没有价值。笔者曾提醒争论者说，能想象一下如果让Linux也支持即插即用、自动加载大量驱动，并能持续向前兼容大量前台应用，那么Linux的稳定性表现一定还会比Windows好么?

而NT架构走上历史舞台后，这种比较开始出现了一定的对位性。安全开始成为一个焦点辩题，在Linux游说者看来，如果不能证实Linux比Windows更安全，就失去了开源优势的理论基础。但真正落实到攻防演练的时候，在很长时间的结论是：作为服务器系统，两者的默认配置都高度不安全；强化配置出来的系统，都相对比较安全。很难比较哪个的默认配置更不安全，或者哪个的强化配置更安全。

在架构性安全机制上，我们确实可以看到开源系统的活跃。如2000年前后，Linux支持了DEP（数据执行保护），而微软则是到了2004年的Windows XP SP2才支持；而ASLR（地址空间布局随机化）的概念2001年出现后，率先尝鲜的是OpenBSD，其后是Linux，最后才是Windows。

是封闭性导致了这种进展缓慢么？微软十年磨一剑的传统方式肯定对此有影响。但同时，在使用新的安全机制时，其巨大的软件库资源的向前兼容性就变成了一个巨大的历史包袱，而这种兼容本身也是商业系统的企业责任之一。我们看到微软今天谨慎的DEP默认设置，我们也可能看到其易用和兼容高于安全的出发点——尽管在Vista上微软出现了以安全还是易用为主导的反复，但到了Windows 7，又回归了最初的选择。

而这个阶段也是Linux开始向桌面扩展的时候，在扩展更灵活便利的驱动支持过程中，也开始引进硬件关联的漏洞。比如在USB方面，就出现了比Windows更多的缺陷。

但在这个长期比较中，关于下列问题，一直有不同的信息源论述不同的结论：

- 漏洞数量及致命漏洞数量
- 平均修补漏洞的时间

让我们看看三组数据，见文摘四。

文摘四

US-CERT：

2005年的5198个漏洞通告中，812个来自Windows，2328个来自Unix/Linux（包括Mac OS X及其他基于Unix/Linux的产品）。

US-CERT：

2005年某一时间段的统计，微软的Windows出现了250次安全漏洞，其中有39个安全漏洞的危险程度达到了40分或者40分以上。而Red Hat Linux只有46次安全漏洞，其中只有3个安全漏洞的危险程度在40分以上。（注意只是对一个Linux发行版的统计。）

赛门铁克：

2006年下半年，在微软Windows上共发现了39处漏洞，其中12个属高优先级或严重级。微软修复这些漏洞的平均时间仅为21天。而Linux操作系统在2006年下半年共曝露了208处漏洞，修复这些漏洞平均耗时58天。

相关的对比也发生在IE和Firefox之间。在2005年，一份资料说明，Firefox比IE有更多的漏洞，但其修补的更快。这份资料本身来自于开源社区，但微软全面的应急体制成型后，我们反而越来越难看到有关的比较了。

笔者一直认为，安全性是可以评级但无法排序的。如果非要找到一个庸俗的比较安全性的方法，那就是所谓的黑客大赛。以浏览器为例，在Pwn2Own黑客大赛上连续两年被率先攻破的既不是IE也不是Firefox，而是苹果的Safari，显然这是一个非开源的系统，也是安全性最没有得到重视的系统。

如果非要在攻防的角度来诠释开放性的损益的话，那么基本的哲学应基于不对称性的争夺，即入侵的过程是对信息了解的天平逐步向攻方倾斜的过程。

从这个意义上说，似乎一个差异出现了。非开源的系统的不对称性是三层的：厂商和用户之间存在着系统信息的不对称性（系统不对称），而用户和攻击者之间存在着配置的不对称性（配置不对称）。而开源系统的不对称性是两层的，即系统对于厂商、用户和攻击者都是开放的，其对称性只有配置的不对称性。

显然，在非开源的情况下，攻击者对于系统级漏洞的挖掘，主要是二进制级的方法，而在开源系统中则变为了源码级别的。那么有趣的问题出现了，开源成了改善了用户的安全认知和降低了攻击者的难度的双刃剑。

文摘五（引自http://news.csdn.net/n/20080109/112583.html）

Coverity公司是著名的源码审计产品提供商，每年对多个开源软件进行安全审计并发布报告。

2008年的分析结果被CSDN报道，称“平均每

一千行开源代码就有一个安全漏洞”（报告原文为we find that a typical static analysis defect density is approximately 1 per 1000 lines of code），并引用了大量数据。

2009年的报告称，代码函数越短小，发现的代码漏洞越少。换句话说，如果一个开发人员编写是由几行代码组成的程序“action”，代码行越少，就会发现越少的错误。2009年报告发现，开源软件的总体完整性、质量，与安全性都有提升，近三年来的瑕疵比例降低了16%。

2010年，Coverity测试的Android内核发现了359个软件缺陷，其中88个高危漏洞。这个内核版本可能被用在流行的手机和其他基于Android的设备。Coverity Scan在开源项目中发现的缺陷将近半数属于高风险。

### 新安全层次

我们讲到这里，问题并未完成，因为上述讨论都依托于三个前提出现。

第一，软件的发布者是善意的。

第二，软件的使用者是无辜的。

第三，攻击者不是发布者或者合法使用者中的一员。

传统开源主义者一直期望世界是一个君子世界，是一个泾渭分明的世界，他们确实有所努力。但事实却远为残酷。

让我们看一个非常纠结的例子。

文摘六

2011年3月3日，Google官方Android市场上出现58款应用软件被植入名为DroidDream的木马，攻击了至少5万台手机。该木马释放提权工具，利用系统漏洞获得root权限，回传用户隐私信息，并具有下载其他恶意代码的能力。Google随后删除了这些软件，并向受攻击的手机远程推送安全工具以清除该木马。恶意代码如此大规模通过官方市场传播，是前所未有的。而后不到一周，这一安全工具又被植入BgServ木马，被攻击者上传到第三方网站分发。

这是恶意攻击在典型的开放式操作系统+半开放的商务体系环境下的一次综合爆发。在传统PC环境下，多数软件是以编译型二进制结果存在的。因此，尽管恶意软件也在向植入、破解、汉化等灰色环节渗透，从而试图与工具软件捆绑，但其成本是不低的。其中最重要的制约因素，就包括二进制PE程序的签名机制相对完善。但类似APK这种过度开放的文件格式，加上其中使用的解释型中间代码，以及极为松散的签名机制，导致一个软件太易于被改头换面为一个“新”版的工具，并被植入木马。

这个问题在开源环境下是泛存在的，比如我们如何验证某一Wireshark、libpcap是原厂版本，而不是植入了后门？在PC环境下，我们可以靠数字签名，但开源体系呢？相关机制虽然也存在，但并不是默认的、普适的方法。

当然，在软件发布者的善意本身遭到怀疑的情况下，开源系统也多了一种新的优势，那就是心理安全感。这种感觉应该是封闭系统依靠用户协议所无法达到的。

但从开源系统的角度，也面临着如何在巨大规模的代码中证明并不存在某些恶意的构造的问题。此外，又如何证明二进制的版本与源码是一致的呢？

目前的自动化代码审计机制基本是建立在代码发布者没有恶意基础上建立的bug发现机制，代码发布者只要存在这种恶意，其就会有意规避相关的检测方法，并以测试验证其是否达到了目的。

### 政府角色

“自由操作系统和CPU是中国IT从业人员永远的痛。这种痛决定了，坚持者需要绷带来包扎伤口，来暗示自己没有受伤，这条绷带就是Linux”。在认为从头做起已经没有可能的情况下，开源组织开始了一场积极的面向中国政府的Linux游说中。而对立面则显然是产业巨头微软。

这个问题绝不是中国特色的，从欧盟到日本、从印度到韩国，微软霸权都是某种挥之不去的阴影。在这个层面，代码是否公开似乎已经比代码是否安全更为重要，他们关心的不是产品安全品质，而是产品中是否潜藏着恶意与阴谋。

文摘六和文摘七或许背后也是美国产业界在政府层面的对决。

文摘七（引自原文地址：http://blog.sina.com.cn/s/blog_4b8a02690100gip6.html）

中国开源软件推进联盟主席陆首群老师在博客中提到：

一份美国开源人士为奥巴马政府起草的“白皮书”中，指出：

“在过去十年里，开源软件在美国军队和情报部门得到广泛应用，这主要取决于它的安全优势”

“最近美国国家安全局（NSA）证实：发现开放源代码在网络安全方面优于私有代码”

“与私有软件相比开源软件更加安全”

“有证据显示，从操作系统到中间件，到数据库，到浏览器以及Java等，比之私有软件产品，开源软件产品在其生命周期里面临更少的安全问题”

其主要依据是：

（1）在开源代码中不留“后门”（不隐藏秘密）

（2）开源软件比私有软件的“漏洞（Vulnerability）”少

（3）“开源软件产品的开放性意味着安全秘密不在代码中，它必须在代码外进行管理”

文摘八（引自http://www.fortify-china.com/web/fd/c11694/w10059423.asp）

Fortify公司对11个开源软件包进行了安全评估（Tomcat、Derby、Geronimo、Hibernate、Hipergate、Jboss、Jonas、OFBiz、OpenCMS、Resin、Struts），结果是：

1、一共从中找出了22826个跨站点脚本和15612个SQL注入式(injection)安全漏洞

2、当Fortify试图把这些问题提交给开源社区时，“在三分之二的情况下，你不会得到任何回应，他们没有提供任何电话号码，那么你应该向谁询问安全信息？这很难讲。”

因此，针对上文提到的“开源人士建议奥巴马政府使用开源软件以加强安全性”，Fortify提出反对意见，建议政府谨慎对待开源产品：http://www.iteye.com/news/5856

而美国政府对于Checkpoint收购Sourcefire的否决，似乎也证明，在所有者改变的情况下，美国政府也不相信仅靠源码开放就可以保证系统中不出现猫腻。

作为职业安全工作者，笔者希望提醒有关人士，开源不是照单全收的理由，只要存在主观的故意，有太多技巧可以在开放的代码中埋下潜藏的杀机，比如在一个复杂的条件组合下才能触发的漏洞。这样的隐患根本无法从代码逻辑角度去判定，而且即使被发现，也不能证明是被故意埋下的。以下的文摘或许值得注意。

文摘八（引自http://tech.ccidnet.com/art/1101/20101220/2274721_1.html；http://lwn.net/Articles/419865/）

2010年12月14日，号称史上最安全操作系统的OpenBSD，被其创始人希欧·德若特自己爆料称，他们所提供的OpenBSD网络数据安全加密协议可能早在10年前就为美国联邦调查局（FBI）预留了“后门”。

当然，相关事务未得到具体的证实，就像微软的NSA密钥事件一样，都没有最终的确定。

**总结**

本文的目的，并不是试图加入关于开源与封闭孰是孰非的争论，也不是试图对Windows和Linux哪个更好的无厘头争论发表观点。笔者认同开源是一个革命性的方法，笔者认同开源的践行者们的信仰和努力，并对此保持敬意。

安全过去是一个技术问题，而在地下经济发达的今天，又混杂了浓重的社会学色彩。但无论如何，它觉不是一个“宗教”或者“情感”问题。因好感而信任，只适用于人与人，但不适用于评价代码、软件和系统。

对于希望改善安全性的软件开发者。笔者需要提醒他们：安全来自更多善意而专业的关注，并以这种关注驱动改善。开放是聚合这种关注的方法之一，但在巨大的地下经济产业链条存在的前提下，对于很多并非“知名度很高”的系统来说，开放使遭遇偶然性风险的概率大大增加，花朵可以在露天种植，但访问它的不一定只有蜜蜂，也可能有害虫。

正在为选择开源或非开源产品而纠结的用户也是值得同情的，这是一个可以为任何需求找到产品的时代，这也是任何产品都无法证实自己可靠性的时代。用户少了很多DIY的乐趣，也多了很多辨识的成本。但落实到用户层面的安全问题是一个个体问题，除非用户有精力、并且期望对产品做出个性化的功能修改，否则是否满足功能、成本、维护代价等等，才是选择的首先要素。同时，除了根本不可用的系统，否则任何软件和产品都不是安全的，安全来自于对系统内嵌安全机制的有效发挥，以及辅助使用第三方的安全产品。而其最终的安全，往往不取决于你选择了什么的系统，而是你如何管理和配置它。

笔者最后的结论是：尽管开源与安全这两个话题存在着太多的关联和纠结，但开源既非安全的充分条件，也非必要条件；甚至既非安全的朋友，当然也非安全的敌人，而是一种令安全工作者爱恨纠结的存在。