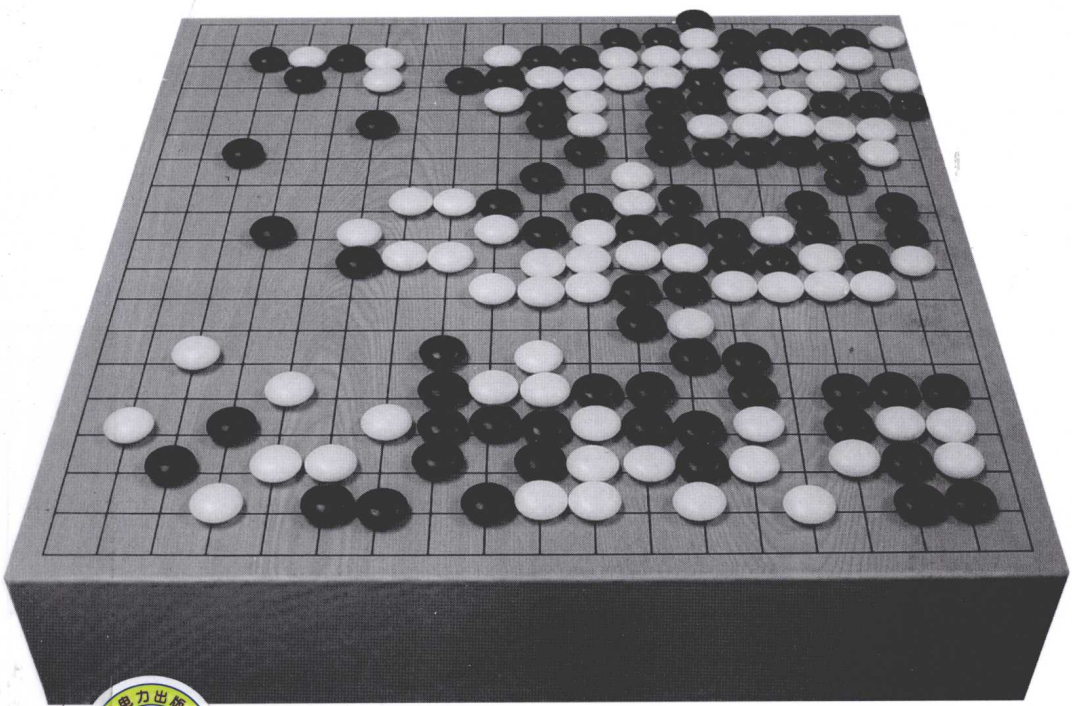
BINARY HACKS™

黑客秘笈100选



高林 哲、鵜飼 文敏、佐藤 祐介、
浜地 慎一郎、首藤 一幸 著
蒋斌 杨超 译

O'REILLY®
中国电力出版社

BINARY HACKS™

黑客秘笈100选

高林 哲、鵜飼 文敏、佐藤 祐介、
浜地 慎一郎、首藤 一幸 著
蒋斌 杨超 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目（CIP）数据

BINARY HACKS™：黑客秘笈100选 /（日）高林哲等著；蒋斌，杨超译.
北京：中国电力出版社，2009
书名原文：BINARY HACKS™
ISBN 978-7-5083-8793-2

I. B… II. ①高… ②蒋… ③杨… III. 计算机网络－安全技术
IV. TP393.08

中国版本图书馆CIP数据核字（2009）第065928号

北京市版权局著作权合同登记
图字：01-2009-2300号

书　　名/ BINARY HACKS
书　　号/ ISBN 978-7-5083-8793-2
责任编辑/ 胡顺增
封面设计/ O'Reilly Japan，张健
出版发行/ 中国电力出版社（www.infopower.com.cn）
地　　址/ 北京三里河路6号（邮政编码100044）
经　　销/ 全国新华书店
印　　刷/ 汇鑫印务有限公司
开　　本/ 787毫米×1092毫米　18开本　21.25印张　379千字
版　　次/ 2010年1月第1版　2010年1月第1次印刷
印　　数/ 0001—3000册
定　　价/ 39.00元（册）

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构O'Reilly Media, Inc. 授权中国电力出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的*The Whole Internet User's Guide & Catalog*（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到 WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商 —— 每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体 —— 他们本身是相关领域的技术专家、咨询专家，而现在则编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

目录

本书寄语 1

编写说明 3

前 言 7

第 1 章 介绍 13

1. Binary Hack 入门 13
2. Binary Hack 用语的基础知识 15
3. 用 File 查询文件的类型 22
4. 用 od 转储二进制文件 24

第 2 章 目标文件 Hack 30

5. ELF 入门 30
6. 静态链接库和共享库 40
7. 通过 Idd 查阅共享库的依赖关系 44
8. 用 readelf 表示 ELF 文件的信息 47
9. 用 objdump 来转储目标文件 49
10. 用 objdump 反汇编目标文件 53
11. 用 objcopy 嵌入可执行文件的数据 57

12. 用 nm 检索包含在目标文件里的符号 58
13. 用 strings 从二进制文件中提取字符串 64
14. 用 C++filt 对 C++ 的符号进行转储 66
15. 用 addr2line 从地址中获取文件名和行号 66
16. 用 strip 删除目标文件中的符号 68
17. 用 ar 操作静态链接库 69
18. 在链接 C 程序和 C++ 程序时要注意的问题 70
19. 注意链接时的标识符冲突 76
20. 建立 GNU/Linux 的共享库，为什么要用 PIC 编译？ 82
21. 用 statifier 对动态链接的可执行文件进行模拟静态链接 85

第 3 章 GNU 编程 Hack 88

22. GCC 的 GNU 扩展入门 88
23. 在 GCC 上使用内联汇编（inline assembler） 93
24. 活用在 GCC 的 built in 函数上的最优化 97
25. 不使用 glibc 写 Hello World 100
26. 使用 TLS（Thread-Local Storage） 104
27. 根据系统不同用 glibc 来更换加载库 106
28. 由链接后的库来变换程序的运行 109
29. 控制对外公开库的符号 111
30. 在对外公开库的符号上利用版本来控制动作 114
31. 在 main()的前面调用函数 121
32. GCC 根据生成的代码来生成运行时的代码 124
33. 允许 / 禁止运行放置在 stack 里的代码 126
34. 运行放置在 heap 上的代码 128
35. 建成 PIE（位置独立运行形式） 129
36. 用 C++ 书写同步方法（synchronized method） 132
37. 用 C++ 生成 singleton 136
38. 理解 g++ 的异常处理（throw 篇） 141
39. 理解 g++ 的异常处理（SjLj 篇） 142
40. 理解 g++ 的异常处理（DWARF2 篇） 149

41. 理解 g++ 异常处理的成本 153

第 4 章 安全编程 Hack 156

42. GCC 安全编写入门 156
43. 用 -ftrapv 检测整数溢出 160
44. 用 Mudflap 检测出缓冲区溢出 163
45. 用 -D_FORTIFY_SOURCE 检测缓冲区溢出 166
46. 用 -fstack-protector 保护堆栈 170
47. 将进行位遮蔽的常量无符号化 173
48. 注意避免移位过大 175
49. 注意 64 位环境中 0 和 NULL 的不同之处 176
50. POSIX 的线程安全函数 179
51. 安全编写信号处理的方法 182
52. 用 sigwait 将异步信号进行同步处理 187
53. 用 sigsafe 将信号处理安全化 190
54. 用 Valgrind 检测出内存泄漏 198
55. 使用 Valgrind 检测出错误的内存访问 200
56. 用 Helgrind 检测出多线程程序的 bug 204
57. 用 fakeroot 在相似的 root 权限中运行进程 207

第 5 章 运行时 Hack 211

58. 程序转变成 main() 211
59. 怎样调用系统调用 220
60. 用 LD_PRELOAD 更换共享库 223
61. 用 LD_PRELOAD 来 lap 既存的函数 225
62. 用 dlopen 进行运行时的动态链接 228
63. 用 C 表示回溯 232
64. 检测运行中进程的路径名 237
65. 检测正在加载的共享库 240
66. 掌握 process 和动态库 map memory 246
67. 用 libbfd 取得符号的一览表 250

68. 运行 C++ 语言时进行 demangle 254
69. 用 ffcall 动态决定签名，读出函数 257
70. 用 libdwarf 取得调试信息 261
71. 通过 dumper 简化 dump 结构体的数据 265
72. 自行加载目标文件 268
73. 通过 libunwind 控制 call chain 275
74. 用 GNU lightning Portable 生成运行编码 278
75. 获得 stack 的地址 281
76. 用 sigaltstack 处理 stack overflow 285
77. hook 面向函数的 enter/exit 294
78. 从 signal handler 中改写程序的 context 297
79. 取得程序计数器的值 299
80. 通过自动改写来改变程序的操作 300
81. 使用 SIGSEGV 来确认地址的有效性 303
82. 用 strace 来跟踪系统调用 305
83. 用 ltrace 来跟踪进程调用共享库的函数 307
84. 用 Jockey 来记录、再生 Linux 的程序运行 309
85. 用 prelink 将程序启动高速化 310
86. 通过 livepatch 在运行中的进程上发布补丁 313

第 6 章 profile 调试器 Hack 321

87. 使用 gprof 检索 profile 321
88. 使用 sysprof 搜索系统 profile 324
89. 使用 oprofile 获取详细的系统 profile 326
90. 使用 GDB 操作运行进程 330
91. 使用硬件调试的功能 332
92. C 程序中 break point 的设定可以用断点这个说法 336

第 7 章 其他的 Hack 338

93. Boehm GC 的结构 338
94. 请注意处理器的存储器顺序 343

95. 对 Portable Coroutine Library（PCL）进行
轻量的并行处理 348
96. 计算 CPU 的 clock 数 351
97. 浮点数的 bit 列表现 354
98. x86 的浮点数运算命令的特殊性 356
99. 用结果无限大和 NaN 化运算来生成信号 360
100. 文献介绍 363

本书寄语

在2005年年末，以本书的作者高林先生等为中心举办了“Binary 2.0会议”。听到这个消息的时候，我不由笑了。表面上看这是一个用流行的Web2.0和轻量语言格式叙述低级别的技术、推崇不协调的策划，但是它的定位确实很有意义。

随着计算机高性能化的发展，编程的环境也大大改变了。想要编程首先必须要有编辑器和编译器的说法已经过时了。现在的程序员从一开始就拥有能够提供大量帮助的非常有效的开发工具。一旦有了新方案，就能方便地运用轻量级脚本语言、已有的库以及网络相关的各种服务轻松进行组合，使用非常方便。

若能熟练掌握最先进的工具，并能把巧妙的程序设计方法付诸实施的话，那实在是件很美妙的事情。然而，仅仅是查询一下编译器里跳出的数字信息，或是更新一下运行中的程序，确实没有太多意思。像让存储器内充满数据，从而耗尽机器周期这样的高超技术，尽管曾有人跃跃欲试，但若要真的付诸实施，则不是件容易的事情。

那么，为何当今社会中二进制如此重要？

工具的力量，或者说是抽象的力量。通过特定切入点对现实加以简化，并忽略与本质无关的复杂程序，只把精力集中到需要思考的问题上。但是，抽象化中必然存在其成立的前提。当系统正常工作的时候，前提的存在往往不会引人注意，但当系统的使用达到极限，前提便不复存在。抽象的界限会出现紊乱。在抽象的高级“世界”，不管构建如何强大的逻辑，根基松动则逻辑

也会随之土崩瓦解。如果对抽象界限所面对的世界缺乏起码的了解，要想重新构建抽象世界则完全不可能。

只使用高级工具的程序员，是在抽象化的世界里游刃有余的人。在抽象化世界可以做很多有趣的事情，但如果是光靠兴趣的话就并非那样容易了。如果对抽象化的范围没有把握好，就无法察觉自身世界的限制，对于专业编程人员而言这是致命的。因为不仅不能解决问题，还不能彻底理解抽象世界构建者的设想。例如，函数指令、回车等习惯用语都被作为真理接受，若不知道这个原理，想要熟练使用计算机就会很困难。如果你想拥有程序员的实力，那就好好利用本书吧。

表面上的介绍就到此为止。其实，本书的真正魅力并不在此。

孩提时代，在大件垃圾场看到被扔掉的电视机，也有过打开满是灰尘的电视机外壳往里面看的经历吧。看到像血管一样缠绕着的各种颜色的接线、隐藏在玻璃管中的奇怪的零部件，就像偷看了什么奇特的东西一样的兴奋。即使到了已经成年的现在，在Web上的关于个人电脑系列的新闻网站上看到最新的个人电脑分析文章，就不由自主地联想到了真实的线路板照片。

如果你和我以及本书的编者们一样拥有相同兴趣的话，在当前这种万物具备的开发环境中，就应该有种不满足的感觉。总是想着如何打开黑箱，然后把里面看个究竟。如果时间允许的话，甚至想自己组装出个什么东西来。

开发效率低也好，土里土气不怎么出色也罢，总之在抽象化的世界里我们总能够发现它令人惊叹的魅力。你选择了与专业编程相关的东西，那么你现在就站在这个原点上了。充满挑战的21世纪也好，五彩缤纷的大千世界也好，何不带着好奇心在这个看似低水平的世界搜寻一番看看呢？那么这本书就是你最好的参考选择。

Enjoy Hacking.

川合史朗

编写说明

关于编者

高林 哲　Satoru Takabayashi

软件工程师。1997年开发了全文检索系统Namazu，此后一直从事大量免费软件的开发。2005年负责IPA“未知软件创造事业”中“源码搜索引擎”的开发，被称为超级创造师。工学博士，兴趣是“破坏一切所知道的”。*http://0xcc.net/*。

鵜飼 文敏 Humitoshi Ukai

Debian 项目的官方成员、Debian JP 项目前负责人、日本 Linux 协会前会长、The Free Software Initiative of Japan 副理事长、2004 和 2005 年度“未知软件创造事业”项目经理。自从攻读研究生期间研究用 PC98 架构操作 386BSD 和 Linux 以来，就一直沉浸在免费操作系统的世界里。作为 Debian JP 项目的创建人员，一直以 Debian 为活动中心，并进行着 *Debian.or.jp* 以及 *linux.or.jp* 等的运营管理。

佐藤 祐介 Yusuke Sato

软件工程师，早稻田大学理工部毕业后，从事软件开发，现在在某制造企业进行信息家电类的安全系统脆弱性检查。日本 SELinux 成员会（LIDS-JP）、JSSM 安全 OS 研究会、Linux 国际财团安全部成员。

浜地 慎一郎 Shinichiro Hamaji

喜欢将技术运用于不同寻常的领域，爱好广泛也有过很多尝试，但最终兴趣转移到了免费软件的批量生产上来。量子信息研究院研究生。

首藤 一幸 Kazuyuki Shudo

以“要做出别人做不出的东西”为格言的工程师，开发了Java线程转移系统、Just-In-Time编译器、覆盖构造工具系列等的软件。Utagoe株式会社董事、技术总监，信息科学博士。研究领域涉及广域分散处理、程序语言处理系统和信息安全等方面。

关于贡献

本书收录了许多Binary Hacker寄出的大量Hack。在此介绍一下他们的简况。

後藤 正德 Masanori Goto

在计算机领域中，从事Debian GNU C Library和Linux内核等开放源代码软件的开发工程的活动。Debian项目正式开发者、YLUG（Yokohama Linux Users Group）发起人。现在在制造研究所里参与数据存储器、PC Cluser等的研发。

中村 実 Minoru Nakamura

在命令集和ABI的夹缝中生存着的现实中的Binary Hacker。开发了MIPS、SPARC、Alpha用的编译器，x86、SPARC、Itanium2用的Java VM的优化调试器。

中村 孝史 Takashi Nakamura

参与本书编写的重要成员，在控制设备方面具备专业Binary Hacker的知识。因为PC的OS会将硬件部分隐藏起来，有些不方便，但是如果没有了OS会更加麻烦。

田中 哲 Akira Tanaka

经常使用Ruby的人，可能Conservative GC就是引他走向Binary Hacker的道路。

八重樫 剛史 Takeshi Yaegashi

每天都是新的开始。

野首 貴嗣 Takatsugu Nokubi

将文件指令库一体化的 Perl 模块、做着 File::MMagic 的维护工作。继续着在没有 Namazu、Kakasi 等库的情况下将内容强行库化的工作。

前言

本书的书名是底层的编程技术。所谓底层的意思是指接近于原始状态的电脑。

软件世界是随着抽象化的重叠而进步的。汇编程序是对机器语言的抽象，C语言则是对汇编程序的抽象。因此，在C语言的基础上还存在着对C进行封装的各种脚本语言。抽象化将底层的复杂部分隐藏起来，以高效率、高安全性的方法为开发者提供编程手段。

但是，如果把底层的技术完全忽略也是不能进行编程工作的。想要彻底追求性能、想尽可能地提高可信赖度、想要解决偶尔发生的“像谜一样的错误”等，在这些时候，底层的基础知识是必不可少的。这是因为抽象化并不能保证万无一失。

例如，一旦Ruby和Perl的脚本由于段异常而异常终止的问题发生之后，就有必要在C的标准环境下探求原因，有的时候为了解决特殊的问题，譬如像“转换运行中的机器语言的代码”这样的技巧就很重要。如果不了解底层的基础知识，这些问题就不能解决。

本书的目的是为了介绍能在这些场合使用的“Binary Hack”技能。Binary Hack的名称是由0或1也就是在编程中处于最底层的二进制而来的。本书将Binary Hack定义为“应用了底层软件技术的编程技能”，因为这是基本工具的使用方法，所以也涵盖了一些应用了OS安全编程以及处理器功能的高水平技巧。

一直以来，由于这样的技能没能被好好地总结和归纳，出现了“知道的人则知道，不知道的人则不知道”的情况。因此，本书尝试着将这些技能总结从而达到谁都可以使用的效果。本书的编写以在实践中确实有用的Hack为中心，但也包含了一些实用效果不是很大却相当有趣的Hack。希望能通过本书，让读者们在掌握实际可用技能的同时也能体会到底层技术的乐趣。

本书的能力与限制

本书中编录了在Binary Hack中必不可少的基本工具的使用方法，还有能够处理像GCC的扩展功能、OS的系统级调用以及内联汇编程序（Inline Assembler）等高技术水平的问题。本书的目标平台为UNIX，特别是以GNU/Linux为重点。虽然不能处理Win32API的Windows程序中存在的Binary Hack的问题，但是在采用了Cygwin的GNU基本的开发环境中，本书的大多数Hack都能适用。

必备知识和参考文献

本书的读者要能在UNIX的命令行上能够进行基本的操作。关于UNIX可以参考《UNIX环境编程》（Brian W.Kernighan、Rob Pike著）等书籍。

另外，本书虽介绍了C和C++编程语言的使用规则，但是没有涉及C程序设计语言如何操作这些编程语言的基本知识。关于C和C++可以分别参考《C程序设计语言》（B.W.Kernighan、D.M.Rich著）、《C++程序设计语言》（Bjarne Stroustrup著）等书籍。在[Hack #100]中介绍了很多参考文献。一部分的Hack中使用了汇编语言，由于在本书中有详细的解释说明，所以即便没有汇编语言方面的知识也能读懂。

本书的构成

第一章 介绍

围绕着Binary Hack的初步知识，解释并说明了在本书中使用的各类技术用语，还介绍了Binary Hack的最基本工具。

第二章 目标文件Hack

目的在于加深对作为可执行文件以及共有库的目标文件的理解。首先，对GNU/Linux等系统中使用的ELF进行了相关说明，其次介绍了与库