深入分析Linux操作系统深度安全加固（5） PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/144/2021_2022__E6_B7_B1_E5_85_A5_E5_88_86_E6_c103_144450.htm 8. 自行扫描 普通的安全加固基本上是做完了，现在让我们来对自己做的系统做一个风险评估，推荐使用 nessus latest version [homepage:http://www.nessus.org](既然从头到尾用的都是开源的东西，这里也继续节约成本，呵呵)也许你觉得自己的系统没有问题了，但有时 nessus 还是能报告出一些问题，比如一个第三方的 webmail 有某些安全缺陷，如果没有问题最好，有问题我们再回去修补。 9. 高级技巧 以上的措施已经足以让大多数入侵者望而却步，接下来的部分给那些对安全极度敏感的偏执狂 缓冲区溢出对策中有: stackgurad，stackshield，formatguard，heapguard，pointguard 等编译技术，但他们需要重新编译源码，不仅麻烦而且会使系统性能有所下降.所以这里打算用防止缓冲区溢出的内核补丁。 比较熟知的是 PaX 内核补丁，它主要通过数据区 [heap/bss/stack] 不可执行代码来防御直接覆盖返回地址后跳转到数据区执行 shellcode 的一些exploitPaX的站点好像访问不了，但用google可以找到很多对应较新内核的PaX下载http://home.hetnet.nl/~ottolander/pax/pax.html。 甜祥夔衄腔祛堤击，但却可以挡住市面上相当数量的 exploit，现在那些关于如何绕过补丁的高级 exploit 技巧已经很不神秘，但是书写那样的攻击程序通常要满足一定的条件，即使那样的程序被写出来，函数，文件指针被成功覆盖，可能在这个系统上还是无法把那“ 溢出成果 ”传递给攻击者--仍然没有办法得

到 shell 或是建立一个连接。 lids Linux 上的入侵检测和防护系统，内核补丁，通过一个比 root 更大的 ring0 权限来提供增强的访问控制，甚至连 root 都不能改变，已有现成资料，不在此讨论。站点：http://www.lids.org lids 和缓冲区溢出补丁可能不兼容，欢迎知道真相的朋友告诉我。 10. 日志策略 主要就是创建对入侵相关的重要日志的硬拷贝，不至于应急响应的时候连最后的黑匣子都没有可以把他们重定向到打印机，管理员邮件，独立的日志服务器及其热备份 11. Snort 入侵检测系统 对入侵响应和安全日志要求较高的系统有此必要；对于一般的系统而言，如果管理员根本不会去看一大堆日志，那么它白白占用系统资源就如同鸡肋一样 12. 最后的建议 关心 bugtraq 上的漏洞列表； 订阅厂商的安全公告； 勤打补丁； 站在攻击者的角度去思考如何防御。 小结 对攻击的思考：假设有一个技术高超的入侵者，拥有自行挖掘系统底层漏洞的能力，他发现了 apache 的一个漏洞，并书写了 remote exploit，这个漏洞暂时还没有出现在 bugtraq 上，处于“ 未知 ”状态，如果入侵者试图攻击我们的系统，他必须能挖掘一个 apache 并且是 root 级的远程溢出： 在 shellcode 中植入代码杀死 httpd 进程，并且把 sh 绑定在 80 端口。 在 80 端口复用。 让 shellcode 执行 iptables -F OUTPUT/INPUT，前提是他猜到有这么回事以上均需要溢出后是 root 权限，并且是能绕过 PaX 的高级 exploit，另外 apach e杀掉后会自动重启如果想攻击 sshd，因为 iptables 将丢弃所有来自外网访问 sshd 的包，所以即使有远程溢出 (当然别忘了 PaX)，此路不通其他的方法，如果脚本攻击可以获得允许远程登录 ssh 用户的明文口令，或是利用脚本缺陷直接添加系统账号，这不仅需要系统 root

权限，而且 /etc/passwd 已经被 chattr 过，满足以上条件，并且攻破 server2，就有希望得到 shell但提升权限的机会不大!普通脚本攻击在此无效，当然如果该系统并不运行 CGI 的话，此路更是不通诚然入侵者很可能在 http 上破坏你的脚本，不过第三方的 web 安全加固暂不在本文讨论之列以上条件对大多数入侵者足够苛刻，可以说几乎不可能实现。 但是我们为此也牺牲了不少，并且这些措施依赖一定的环境而实现安全性和易用性，需要读者站在自己的角度寻找他们的平衡点。