1. **反弹shell的拦截程序seesaw测试总结**

**原理:**

基于Netlink Socket的进程监控，主动检测进程打开FD，发现并处理反弹Shell行为。检查进程启动打开的FD，如果为Shell进程，打开了Socket而未使用/dev/tty、/dev/pts/n、/dev/ptmx等终端，则确认为反弹Shell。

**代码下载：<https://codeload.github.com/zhanghaoyil/seesaw/zip/master>**

**测试过程：**

被攻击机：172.21.70.44

攻击机：localhost-117.121.33.224

1. 使用开源代码seesaw对反弹shell进行检测和拦截。程序安装过程中可能会碰到一些坑。需要耐心调试。难度不大，但考验细节。如果是python2.6，过程中间需要安装setuptools。
2. 先看看如果没有使用seesaw拦截的攻击场景。

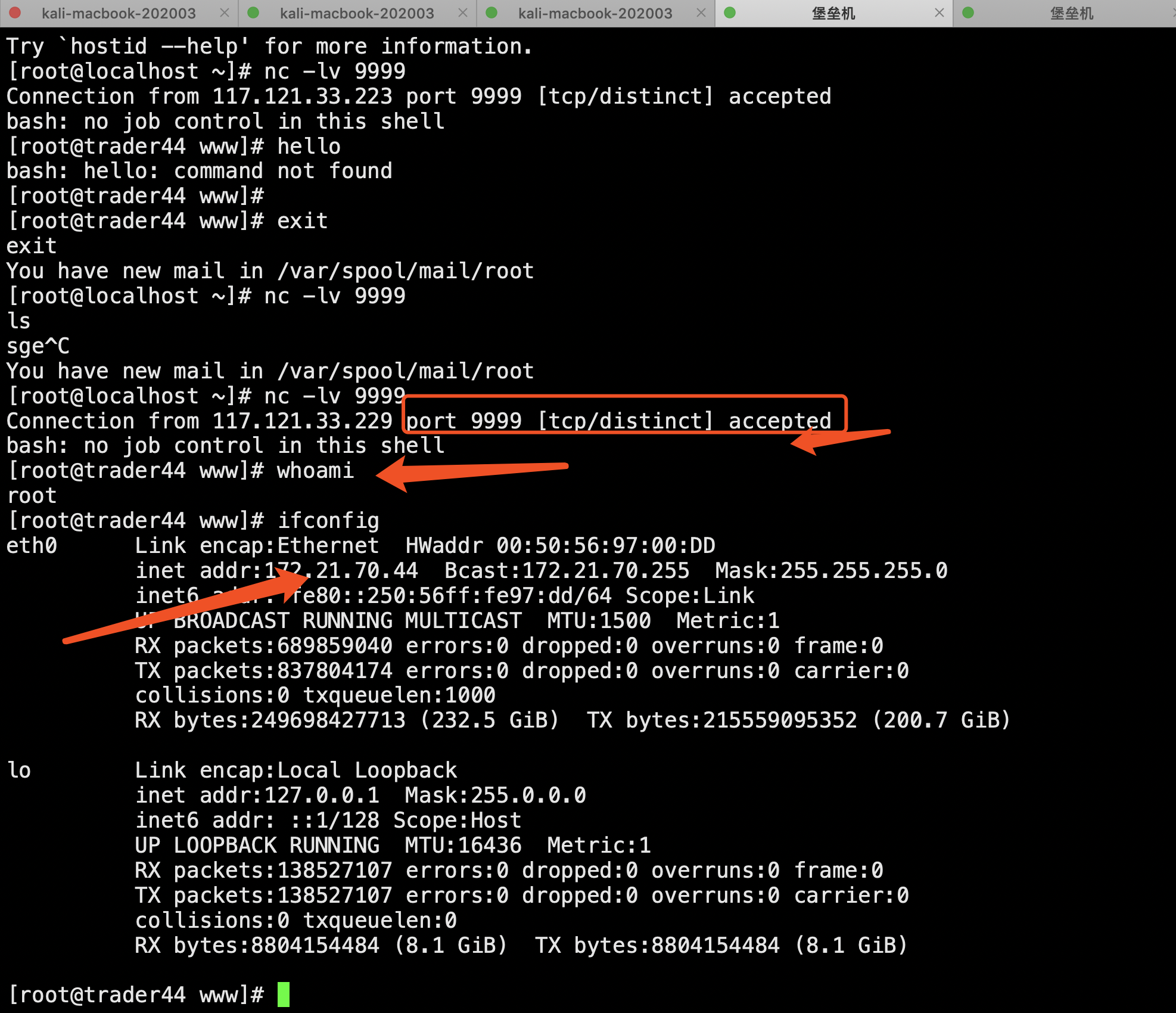


图1.1：攻击机，成功通过远程执行反弹shell连接上被攻击机

如果是使用了seesaw拦截反弹shell后的场景如下：



图1.2：被攻击机执行反弹shell命令

攻击机做端口监听，反弹连接建立后，迅速被切断。

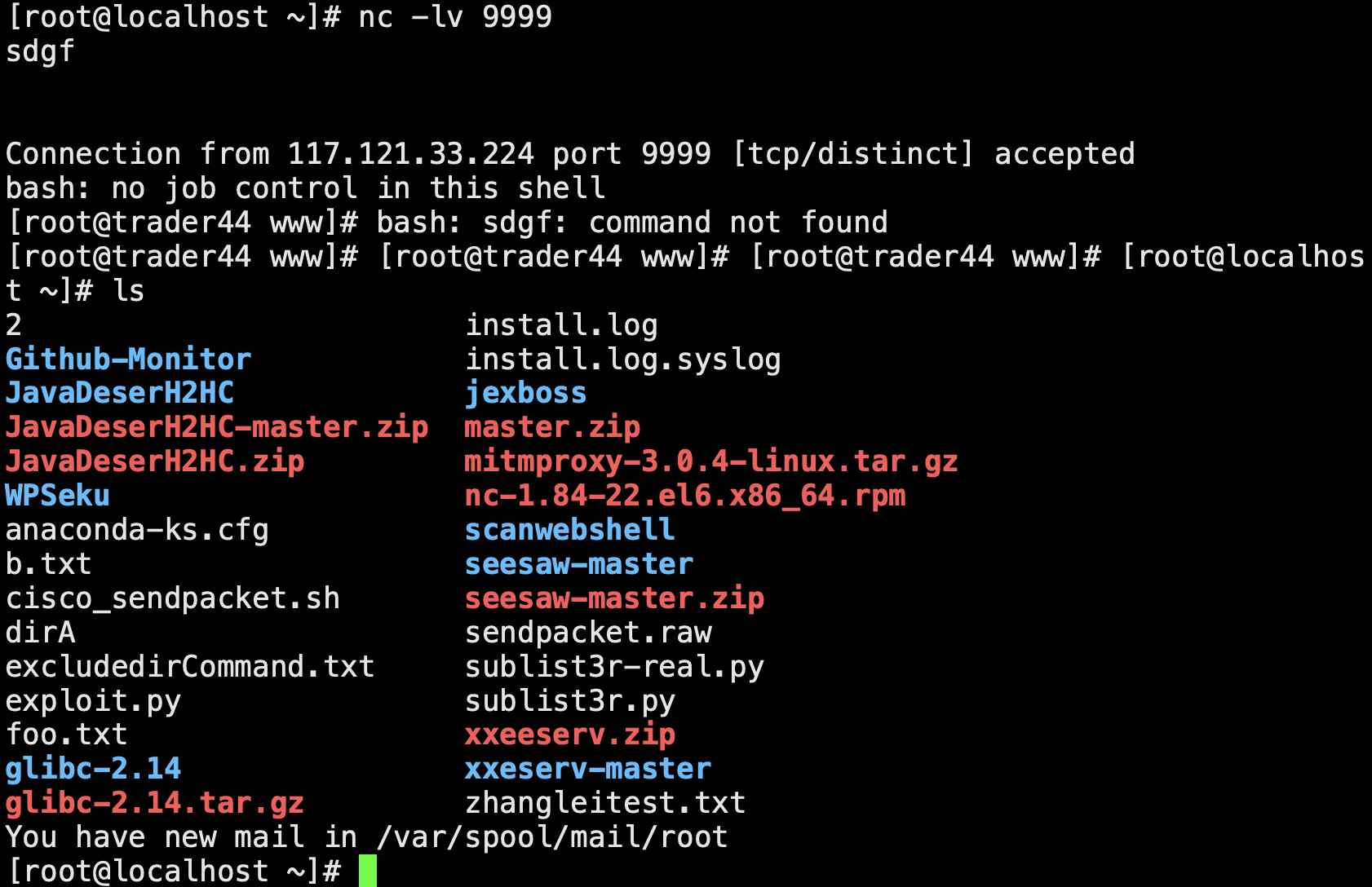


图2：攻击机连接被攻击机后，迅速被切断连接

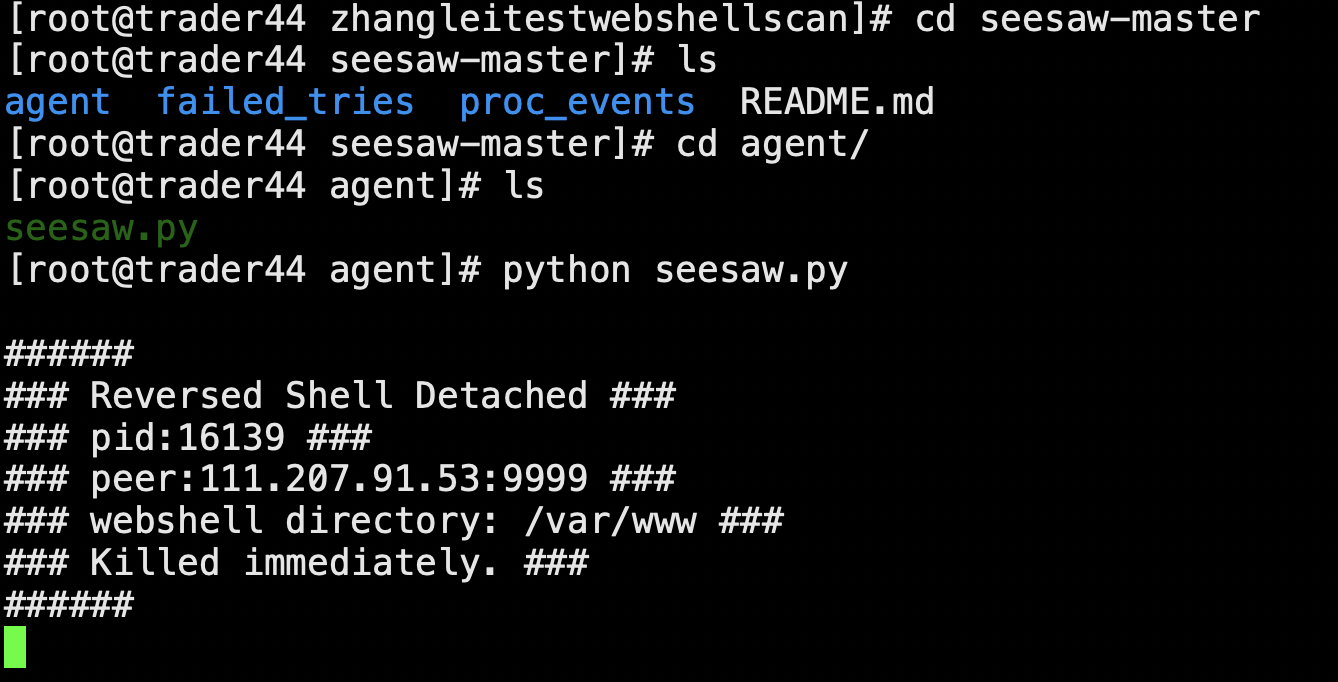


图3：被攻击机成功拦截反弹shell

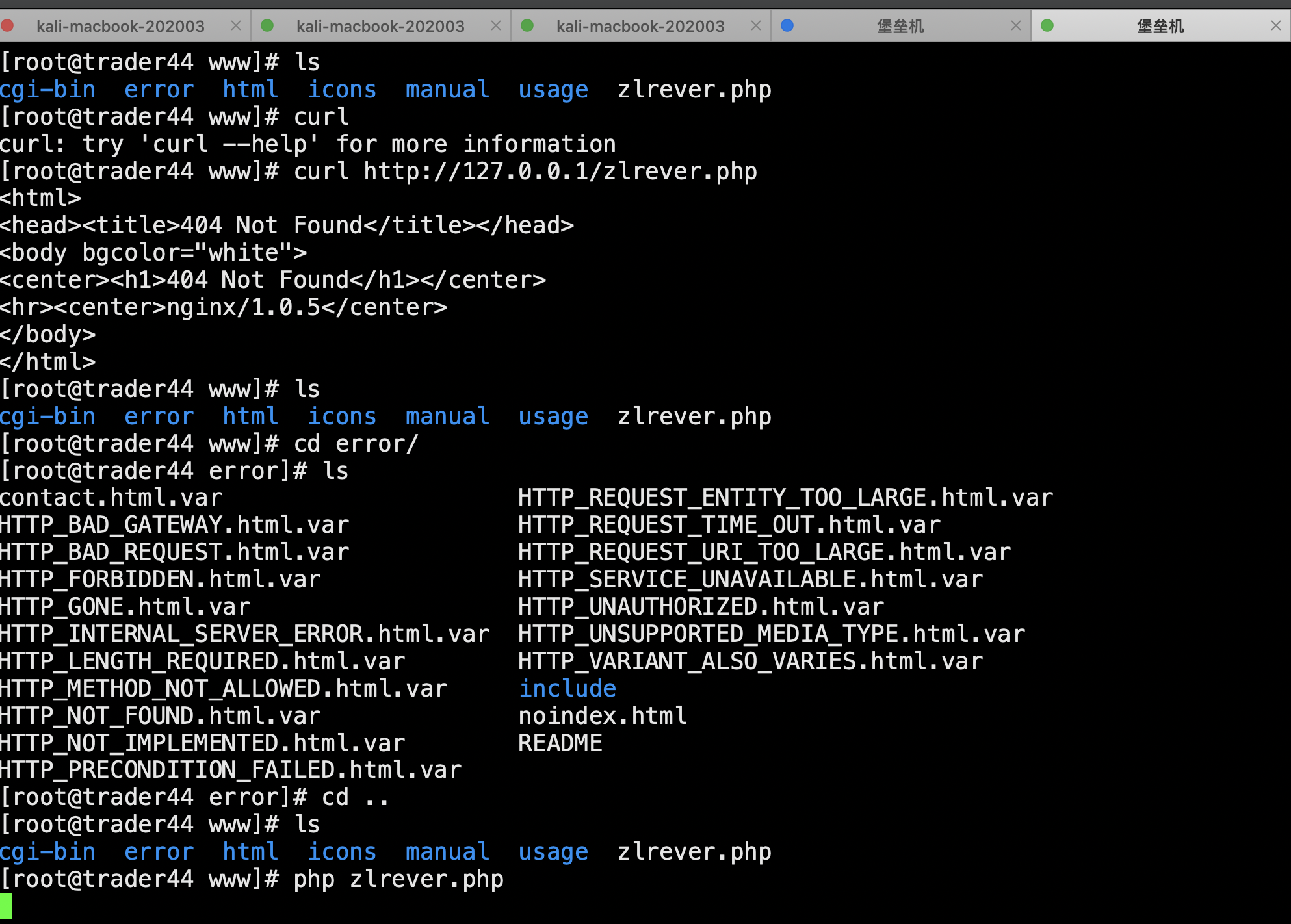


图4：被攻击机执行反弹连接脚本zlrever.php

通过如下的shell脚本调用第二个perl程序，实现对本机的webshell恶意程序的检测和告警。扫描结果会自动输出到一个日志文件。

|  |
| --- |
| #!/bin/bash  :<<!  by zhangleird 20200317  usage: ./dirc.sh $dir1 $dir2  !  cd /root  touch dirA  cd /usr/local/www  dir1=$1  dir2=$2  ls -1| grep -v ${dir1}|grep -v ${dir2}> /root/dirA  scanDir=$(cat /root/dirA)  ls\_date=`date +%Y%m%d`  cd /root/  touch log${ls\_date}  for subdir in $scanDir;do  # echo $subdir  /usr/bin/perl /root/ScanWebShellNew.pl /usr/local/www/$subdir 1 >>log${ls\_date}  echo nextecho.zhang  done  # cat log+${ls\_date} |

|  |
| --- |
| #!/usr/bin/perl  #Scan WebShell for LAKE2  #Desc: A small tools that find webshell with perl, it can check ASP/PHP/JSP/ASP.Net script, enjoy hacking :-)  #Author: lakehu[TSRC]  #Date: 2013-10-30  #Version: 1.1.1  use File::Find;  #php webshell str  @php\_code\_array = (  '\beval(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bassert(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bsystem(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bpassthru(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bexec(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bpcntl\_exec(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bshell\_exec(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bpopen(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bproc\_open(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bpreg\_replace(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bcreate\_function(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\bob\_start(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '\barray\_map(\s|\/\\*.\*?\\*\/)\*\(\s\*.\*?\s\*\)',  '`.\*?`',  '(phpspy|4ngel|wofeiwo|c99shell|webshell|php\_nst|reDuh)',  '\$[\w-\_\'\\[\\]{}\.\$\\*/|]+(\s|\/\\*.\*?\\*\/)\*\(.\*?\)'  );  @asp\_code\_array = (  ''  );  #asp.net webshell str  @aspx\_code\_array = (  ''  );  #jsp webshell str  @jsp\_code\_array = (  ''  );  if(@ARGV!=2){  print "\n";  print "\* Simple Scan WebShell by lake2 [ TSRC ] \n";  print "\* know it then hack it !\n";  print "\* Usage: ScanWebShell.pl <Path> <Type> \n";  print "\* Type: 1 - PHP\n";  print " 2 - ASP\n";  print " 3 - ASP.Net\n";  print " 4 - JSP\n";  print "\* TSRC Website: http:\\\\security.tencent.com\n";  exit;  }  my $postfix;  $postfix = '';  my @str\_code;  if($ARGV[1]==1)  {  $postfix = '\.php$';push(@str\_code, @php\_code\_array);  }  elsif($ARGV[1]==2)  {  print "NO PUBLIC! Do you used ASPSecurity ?\n";exit;  }  elsif($ARGV[1]==3)  {  print "NO PUBLIC!\n";exit;  }  elsif($ARGV[1]==4)  {  print "NO PUBLIC!\n";exit;  }  else  {  print "ERROR: unkown type !\n";exit;  }  #old Perl is not Switch -\_-!! FucK !!!!  #switch($ARGV[1]){  # case 0 { print "get out!\n";exit; }  # case 1 { $postfix = '\.php$';push(@str\_code, @php\_code\_array); }  # case 2 { $postfix = '\.(asp|cdx|cer)$';push(@str\_code, @asp\_code\_array); }  # case 3 { $postfix = '\.aspx$';push(@str\_code, @aspx\_code\_array); }  # case 4 { $postfix = '\.jsp$';push(@str\_code, @jsp\_code\_array); }  # else { print "ERROR: unkown type !\n";exit; }  #}  print "start scanning ..... \n-----------------\n";  $scan\_path = $ARGV[0];  if(substr($scan\_path, length($scan\_path)-1, 1) ne "/"){$scan\_path.="/";}  find(\&wanted, $scan\_path);  print "----------------\ndone !\n ";  sub wanted {  if (-f $File::Find::name) {  if ($File::Find::name=~/$postfix/i) {  checkfile($File::Find::name);  }  }  }  sub checkfile{  my($filepath) = @\_;  my($content);  $content = openfile($filepath);  if($content ne ""){  foreach $item (@str\_code){  if($content =~ /$item/is){ # fix bug : ig -> is, \s will contain \r\n  print $filepath." -> ".$&."\n";  }  }  }  }  sub openfile{  my($filepath) = @\_;  my(@string);  unless (open (MYFILE, $filepath)) {  print ("[-]ERROR: open file $filepath fail !\n");  return "";  }  @string= <MYFILE>;  close(MYFILE);  return join("", @string);  } |