

THE FLORIDA STATE UNIVERSITY  
COLLEGE OF ARTS AND SCIENCES

**EXTENDING THE CORONER'S TOOLKIT VIA  
AGGREGATE DATABASE**

By

JaRay Jasper

A Master Project submitted to the  
Department of Computer Science  
In partial fulfillment of the  
Degree of Master of Science

Degree Awarded:  
Spring Semester, 2004

The members of the Committee approve the master project of student 's  
name defended on date defended.

---

Dr. Lois Hawkes  
Major Professor

---

Dr. Mike Burmester  
Committee Member

---

Jeffrey Bauer  
Committee Member

## DEDICATION

I dedicate this to God and my Family. I'm calling out names because I can. I did this for me, but without the support of Moms, Pops, Big Sis, Baby Bro and my two favorite type little nieces I would never have come to college much less any University in the first place. Also, for raising me in a fashion so when I did come here I didn't freak out and act a fool like most kids seem to do when they first get to school.

Other members of my family include, my homeboy since the 3<sup>rd</sup> grade Steve Lewis, my homegirl Ana Clemente and the homies Donald Baker and Vince Bruington. You people have given me nothing but love since we were all youngsters hanging out in California and are still here for me.

I also dedicate this to the Rodriguez Family, especially Shandy and Millie. Ever since I've been stuck out here in Florida, you have always treated me like family and looked out for me and fed me.

This goes out to Karen Cahill too. You made sure I didn't get into any trouble when I moved out there to Texas and had to finish high school and have been a true to me ever since.

To the Maintenance Team led by Jackie, to Sam, to Khandys and to the FGAMP program because you all helped me to get through grad school.

I like to say thanks and much love to all of you for your parts in making me who I am and helping me survive for this long.

Last, but never the least I dedicate this to Elisa Jonte. If I had never met you I'm pretty sure I would have never made it out of undergrad. I don't think you know how many times I was ready to call it quits, but seeing and talking with you all the time is what helped me to walk on through. You've played the biggest part in making me a man since I left my home eight years ago. You will always have my love and respect. Thank you.

## **ACKNOWLEDGEMENTS**

I'd like to acknowledge Dr. Lois Hawkes. She was my major professor and kept on me, ensuring me I would finish this project, this semester! Also, I'd like to acknowledge Dr. Mike Burmester and Jeff Bauer as my committee members and for providing me the skills needed to execute this project as I am a newbie to computer related security and Unix based systems. Finally, I'd like to acknowledge the Department of Defense Information Assurance Scholarship Program for providing me with funding this past year.



# Table of Contents

Table of Figures .....	i
Table of Tables .....	iii
Abstract .....	iv
1.0 Introduction .....	1
1.1 Plan of Investigation .....	2
1.2 TCT Background .....	3
1.2.1 Grave-Robber .....	3
1.2.2 The C Tools .....	4
1.2.3 Unrm and Lazarus .....	5
1.2.4 Mactime .....	5
1.2.5 TCT Run Times and Results.....	5
1.3 Project Purpose .....	6
1.4 The Architecture .....	7
1.5 Section Overview .....	10
2.0 Attack Scenario .....	10
2.0.1 Scenario Explained .....	11
2.0.2 Project Assumptions .....	11
2.1 GRcompare Script.....	12
2.1.1 Demonstration of GRcompare .....	12
2.2 DELread Script.....	19
2.3 MACread script.....	23
2.4 UNRMread script.....	26
2.5 Simple-MySQL .....	34
2.5.1 Running Simple-MySQL .....	35
2.5.2 Creating a Database .....	37
2.5.3 Creating the Database Tables .....	38
2.5.4 Showing Tables .....	39
2.5.5 Showing Columns .....	40
2.5.6 Inserting a Record .....	41
2.5.7 Showing Records.....	43
2.5.8 Exiting Simple-MySQL .....	44
3.0 Project Scripts' Run Times & Results.....	44
4.0 Conclusion .....	46
REFERENCES .....	48
APPENDIX A - Full procedure .....	49
APPENDIX B - Examining DELread's results .....	54
APPENDIX C - Examining MACread's results .....	69

APPENDIX D - Examining UNRMread's results .....	76
APPENDIX E - MySQLcc reinforcement.....	80
APPENDIX F - Project Scripts .....	84

## Table of Figures

Figure 1: Overall data flow.....	8
Figure 2: Data filtration steps.....	9
Figure 3: Database data .....	10
Figure 4: Initial image of root's home directory.....	13
Figure 5: Grave-robbert results .....	13
Figure 6: Running GRcompare.....	14
Figure 7: GRcompare directory .....	15
Figure 8: GRcompare results .....	16
Figure 9: Missing.txt format .....	17
Figure 10: Missing .shosts file .....	18
Figure 11: Example ssh-knownhosts file .....	18
Figure 12: Running DELread.....	20
Figure 13: DELread results.....	21
Figure 14: Example _keyword.txt file.....	22
Figure 15: Example inode.txt file .....	22
Figure 16: Running MACread.....	24
Figure 17: Results from MACread .....	25
Figure 19: Example keyword.txt files created by MACread .....	26
Figure 20: Running UNRMread .....	28
Figure 21: Results from running UNRMread on single unrm file .....	28
Figure 22: Results from running UNRMread on lazarus files .....	29
Figure 23: example result from keyword 'bob' on single file .....	30
Figure 24: example result from keyword 'bob' on lazarus files .....	30
Figure 25: Example result from keyword 'Feb' on single file .....	31
Figure 26: Example result from keyword 'Feb' on lazarus files .....	31
Figure 27: Example result from keyword 'user=' on single file .....	32
Figure 28: Example result from keyword 'user=' on lazarus files .....	32
Figure 29: Example result from keyword 'nc-1.10' on single file .....	33
Figure 30: Example result from keyword 'nc-1.10' on lazarus files .....	33
Figure 31: User kills ssh after being compromised.....	34
Figure 32: Running Simple-MySQL.....	36
Figure 33: Creating a database .....	37
Figure 34: Creating the Tables .....	38
Figure 35: Showing tables .....	39
Figure 36: Column descriptions of a table .....	40
Figure 37: Example record file (form) .....	41
Figure 38: Inserting a record .....	42
Figure 39: Showing records.....	43

Figure 40: Exiting Simple-MySQL .....	44
Figure 41: DEL-keyword.txt file used for DELread .....	54
Figure 42: COMMAND= keyword.txt file.....	55
Figure 43: Hostbased keyword.txt file .....	56
Figure 44: Hostbased keyword.txt file part 2 .....	57
Figure 45: Inode.txt file for the hostbased keyword.....	57
Figure 46: Password keyword.txt file part 1.....	58
Figure 47: Password keyword.txt file part 2.....	59
Figure 48: rhost= keyword.txt file .....	59
Figure 49: uid= keyword.txt file part 1 .....	60
Figure 50: uid= keyword.txt file part 2 .....	61
Figure 52: RH-DELETED contents part 2.....	63
Figure 53: RH-DELETED contents part 3.....	64
Figure 54: RH-DELETED contents part 4.....	65
Figure 55: RH-DELETED contents part 5.....	66
Figure 56: RH-DELETED contents part 6.....	67
Figure 57: RH-DELETED contents part 7.....	68
Figure 58: MAC-keyword.txt file used with MACread .....	69
Figure 59: Bash keyword results .....	69
Figure 60: ..c keyword results.....	70
Figure 61: Mail keyword results.....	71
Figure 62: m.c keyword results part 1 .....	72
Figure 63: m.c keyword results part 2 .....	73
Figure 64: m.c keyword results part 3 .....	74
Figure 65: sbin keyword results.....	75
Figure 66: DEL-keyword.txt used with UNRMread .....	76
Figure 67: example result from keyword 'bob' on lazarus files part 277	
Figure 68: example result from keyword 'bob' on lazarus files part 377	
Figure 69: Example result from keyword 'COMMAND=' on lazarus files .....	78
Figure 70: Example result from keyword 'Feb' on single unrm file ...	78
Figure 71: Example result from keyword 'Feb' on lazarus files .....	79
Figure 72: Setup before Simple-MySQL is executed .....	80
Figure 73: Creating a database – MySQLcc view .....	81
Figure 74: Showing tables – MySQLcc view .....	81
Figure 75: Column descriptions of a table – MySQLcc view .....	82
Figure 76: Inserting a record – MySQL view .....	83
Figure 77: Viewing records – MySQL view.....	83

## Table of Tables

Table 1: dd image sizes and creation times .....	5
Table 2: Data size returned by tool.....	6
Table 3: TCT run times.....	6
Table 4: Run time for project scripts.....	44
Table 5: File reduction analysis .....	45
Table 6: Data size reduction analysis.....	46

## **Abstract**

The main focus of this project was to provide tools to assist home users and small company system administrators with an inexpensive way to do basic analysis and storage of computer forensics data gathered via The Coroner's Toolkit. Not in any way did this project attempt to provide methods of data gathering and/or analysis aimed towards gathering 'legal' evidence that could be presented in a court of law. Most home users and small companies don't have the time or finances to raise lawsuit wars on anybody, but do need to find out what has been done on their network.

Four of the five shell scripts produced from this project serve as the main information filtering tools. They are GRcompare, DELread, MACread and UNRMread and are written to specifically filter through the raw data collected by The Coroner's Toolkit. The fifth script Simple-MySQL provides a simple interface that connects to a MySQL database and allows for the creation of databases, tables and other functions.

All project scripts are written to utilize the Bourne shell, are modifiable and can be run on any Unix based system (this statement assumes the user has the basic shell commands in their path i.e. ls, grep, diff...).

## 1.0 Introduction

In today's technological society, it is not unusual for home based networks, small office networks much less huge corporation and university networks to be attacked by someone trying to gain access or control of their system. Sure Intrusion Detection Systems (IDS) and firewalls can detect/prevent many attempts of system or network infiltration, but once one's system or network is compromised there is not much an IDS or firewall can do. So what can be done?

Unlike the big corporations or universities, most home users and small office "mom and pop" companies do not have the financial clout to hire a professional or computer forensic specialist to analyze their computers once compromised much less take legal action against the culprit who compromised them. However, there are open source and free forensics toolkits that can be utilized by the home user and the system administrator of these small companies for the purpose of collecting any possible evidence left by the attacker, computer by computer. Major benefits of using one of these toolkits are: becoming more aware of your network/system and being able to prevent future attempts of similar compromises.

One such toolkit is The Coroner's Toolkit (TCT). "TCT is a collection of programs by Dan Farmer and Wietse Venema for a post-mortem analysis of a UNIX system after break-in."<sup>(1)</sup> The tools that come in TCT are used for the purpose of collecting data not for doing any analysis on the data once gathered. Any analytical methods or techniques used on the data provided by TCT are strictly up to the person(s) doing the analysis. Therefore, the methods used during the process of this project may or may not be used by someone else, but are offered as a starting point for information filtering.

There are four sets of tools that are packaged in TCT:<sup>(2)</sup>

- **grave-robber**
- **the C tools (pcat, ils, icat, file)**
- **unrm and lazarus**
- **mactime**

The *grave-robber* tool captures various types of data quickly via order of volatility (this order is given in the *Section 1.2.1 Grave-Robber*) and creates MD5 hashes of the evidence to preserve its integrity. The optimum way to run *grave-robber* is to collect the volatile data on a live system, shut down the system, image the drive, and then use *grave-robber*'s -f option against a copy of the file systems. However, due to time constraints on this project *grave-robber* will only be run on an image of the compromised drives as discussed in *Section 1.1 Plan of Investigation*.

The C tools set of tools record and analyze processes and inode data. They are run by *grave-robber* for most circumstances, but can be run manually and individually if there is a need to. Running some of these tools (i.e. *pcat*)

requires *grave-robber* to be run on a live system, while other tools (i.e. *ils* and *icat*) can be run on the drive image.

The *unrm* and *lazarus* set of tools recover and analyze the unallocated disk blocks on a file system. *Unrm* collects information in unallocated portions of the file system. *Lazarus* analyzes raw data from *unrm* and attempts to classify what type of data it contains.

The *mactime* tool helps create a chronological timeline of when files have been Modified, Accessed, or Created (MAC) for each inode, along with their associated filenames.

It is the purpose of this project to create an aggregate database used for the storage of critical data gathered by TCT from multiple compromised computers. Therefore, this paper will present a method for collecting data via TCT, filtering critical data via shell scripts (we have written and provide for public use) and long-term storage of that data, which can be applied by the home user or system administrator.

## 1.1 Plan of Investigation

In this project, the two compromised computers' images were first stored on the Intermediate Storage Box (ISB). This was accomplished by first, using the unix 'dd' command to take a bit by bit image of each computer's hard-drive. The images were directly exported from the computers to the ISB via a USB connection. TCT's tools were then run on each of the individual images. The data was arranged using the default scheme TCT uses to organize data. Various analytical methods were used to compare each computer's data to a baseline computer configuration as a first step towards erroneous information filtering. Data that could not be directly compared to the baseline went through a pseudo-manual information filtering process using our shell scripts to further separate non-useful data from critical data whose results could further be used to reduce the gross amount of critical data to be analyzed and sent to the Database Box (DBB).

Once the critical data was filtered from each computer's set of data, it was exported from the ISB to the DBB via a USB connection. The critical data was then stored into a single database. The data was further organized into a series of incident specific tables (i.e. backdoors, altered files, etc.).

The independent variables for this project were the analysis techniques used to filter data, the construction methodology of the database and the export method used to transfer data between storage locations.

The dependent variables for this project were the critical data stored in the DBB, the amount of data that can be stored in the ISB and DBB and the number of compromised computers that can be analyzed at one time.



## 1.2 TCT Background

The Coroner's Toolkit is free for use by the public and runs on Unix based systems. It is a collection of tools written in the C and perl programming languages. These tools are used primarily for the purpose of data gathering not for doing data analysis. As the authors Dan Farmer and Wietse Venema state when describing the tools, "If there was a theme it'd be an effort towards the reconstruction of the past - determining as much as possible what happened with a static snapshot of a system."<sup>(3)</sup>

There are four sets of tools that are package in TCT:<sup>(2)</sup>

- **grave-robber**
- **the C tools (pcat, ils, icat, file)**
- **unrm and lazarus**
- **mactime**

Each tool will be discussed separately in the following sections. The discussions will consist of a description of each tool, what sort of data each tool collects and how that tool organizes the data it collects (if available). It will not discuss how each tool is run or the options that could be run with them.

### 1.2.1 Grave-Robber

The grave-robber tool is the main tool in the TCT. However, it is not really a tool itself. It's more tool-caller so to speak. It can run the set of C tools and given a command line option it runs the mactime tool amongst the other tools it runs by default. Its main function is to collect as much information on the system as it can. It collects data by Order of Volatility, in order: <sup>(4)</sup>

- **Memory**
- **Unallocated file system**
- **netstat, route, arp, etc.**
- **ps/ lsof, capture all process data**
- **stat & MD5 on all files, strings on directories**
- **Config, log, interesting files (cron, at, etc.)**

Grave-robber generates two log files, coroner.log and error.log. All the commands executed by grave-robber and when they were executed are stored in the coroner log. All the output going to stderr is stored in the error log.

All captured data is saved in the data subdirectory. To be precise it is actually kept in a subdirectory of the data subdirectory, the name of the subdirectory corresponding to the name of the system and the time that the grave-robber tool has been run. The grave-robber also creates a symbolic link -

"data/hostname" - to the actual data dir. In the hostname subdirectory there are several files and subdirectories of interest. The follow list identifies the specific files and subdirectories grave-robber will create: <sup>(5)</sup>

- **command\_out**
- **strings\_log**
- **body**
- **body.S**
- **coroner.log**
- **error.log**
- **deleted\_files**
- **pcat**
- **conf\_vault**

The description of each can also be found in reference five. Finally, *grave-robber* creates MD5 signatures for all of the data it collects.

### 1.2.2 The C Tools

Not in any order, the first C tool is *pcat*. *Pcat* copies process memory from a live system. It hooks onto the process specified with `process_id` and copies the contents of its memory to standard output. By default, *pcat* skips over holes in the process address space. Consequently, absolute location information is lost. <sup>(8)</sup>

The second C tool is *ils*. *ils* lists inode information. *ils* opens the named device and lists inode information. By default, *ils* lists only the inodes of removed files. <sup>(9)</sup> *Icat* is C tool number three and copies files by inode number. *Icat* opens the named device and copies the files with the specified inode numbers to standard output. <sup>(10)</sup> Rounding out the C tools is the *file* tool.

*File* classifies files into various types. It tests each argument in an attempt to classify it. There are three sets of tests, performed in this order:

First, are the file system tests. The file system tests are based on examining the return from a `stat` system call. The program checks to see if the file is empty, or if it's some sort of special file. Second, are the magic number tests. The magic number tests are used to check for files with data in particular fixed formats. These files have a 'magic number' stored in a particular place near the beginning of the file that tells the operating system that the file is a binary executable, and which of several types it is. Finally, are the language tests. The language tests look for particular strings that can appear anywhere in the first few blocks of a file. <sup>(11)</sup>

### 1.2.3 Unrm and Lazarus

The tool set of *unrm* and *lazarus* are used for “bringing the dead to life” so to speak. *Unrm* is a tool that collects all the unallocated space in a file system. It basically gathers information on anything that has had space allocated to it and then had that space freed some time in the past. Also, *unrm* will return the same amount of data as the file system had unallocated space. Therefore, if a file system has 10 gigabytes worth of unallocated space, *unrm* will return 10 gigabytes worth of data. The data collected by *unrm* is then used by *lazarus* to try and restore some sensibility or recognition of the data.

*Lazarus* attempts to analyze the data it receives from *unrm*. It can also be run on other sources of data such as system memory and swap. *Lazarus* takes the following steps when analyzing data:

First, it reads in a chunk of data. Then it tries to determine if the data is text or binary. If the data is text, it checks it against a set of regular expressions in an attempt to determine what it is. If the data is binary, the file command is run over the chunk of data. Now, if the block of data is successfully recognized it is marked as a data or binary block and if the block immediately recognized before it has the same mark the two blocks are concatenated. If the block is not recognized it is only concatenated to the previous block if that block was recognized. Finally, *lazarus* outputs the data blocks to a directory called “blocks” and it creates a corresponding map to the blocks.

### 1.2.4 Mactime

The *mactime* tool is used to search through a file system or database in order to display every file’s modified, accessed and created times (MAC times). The tool itself does not create any directories or files to save its output, so it is usually run on a database created by the grave-robber tool. Given a command line option *grave-robber* can be made to collect all the MAC times for the file system. It then creates and stores this data into a database. The *mactime* tool can then be used to query the database for a specific time period and output any file whose MAC times fit the search criteria.

### 1.2.5 TCT Run Times and Results

dd Image Sizes		
Operating System	Image Size (GB)	Creation Time
RedHat	22	2-3 days
Mandrake	3.7	10 hours

**Table 1: dd image sizes and creation times**

The image sizes are the same for both the baseline and compromised images. TCT was run on these images and as shown in the following tables the results and run times are dependent on the size of image.

TCT Results		
	RedHat	Mandrake
<b>unrm</b>	19 GB	2.5 GB
<b>ils &amp; icat</b>	107 MB	0 (got corrupted)
<b>mactime</b>	21 MB	16 MB

**Table 2: Data size returned by tool**

The size of *grave-robber*'s result is not displayed because it is dependent on how *grave-robber* is configured. The mactime results are actually the size of the databases created by *grave-robber*.

TCT Run Times		
	RedHat (days)	Mandrake (days)
<b>Grave-robber</b>	1.5	5 (hrs)
<b>ils &amp; icat</b>	1.5	----
<b>unrm + lazarus</b>	3 + Stopped after 3 days	.33 + 3 = 3.33

**Table 3: TCT run times**

The *mactime* run times are not presented because the mactime database was actually created during the execution of *grave-robber* as stated above. Due to time constraints *lazarus* was not run on the *unrm* file for the RedHat box.

### 1.3 Project Purpose

Other forensic toolkits besides TCT are available for doing data collection and analysis. One such toolkit is the Sleuth Kit<sup>(7)</sup>, which is also free and can be used for gathering forensics data on Unix based systems. The Sleuth Kit was based somewhat on TCT using some of its code and design. Another tool, Encase made available by Guidance SoftWare<sup>(6)</sup> supports Windows operating systems as well as Unix based. However, with a price tag around \$2000 this tool can be too expensive for the general public. There are many more forensics tools and toolkits available, but for purposes of originality and cost, this project utilizes TCT.

After surfing the web and reading various documents, articles and presentations on the steps to follow when attempting to do computer forensics, it

was found that not one really mentioned any methods for information filtering or long-term storage of data for future and/or further analysis or for simple reference. Also, these documents did not make any reference to how someone who owns more than one computer could set up a system that would allow him or her to not only do forensics analysis on more than one computer, but also be able to trace and store evidence from the multiple computers in an organized fashion. It was felt that providing home users, possibly system administrators and other users who are not professional or computer forensic specialists with a system that would allow for both long-term storage and cross comparability options would be very useful.

This project studied the feasibility of a collecting and storing data from a distributed network (computers with different operating systems) gathered by The Coroner's Toolkit into a single aggregate database for future analysis or reference. The Coroner's Toolkit was used to gather data from the 'dd' images of two compromised computers. A 250GB external hard drive was used as the Intermediate Storage Box in order to store the 'dd' images of the compromised computers along with an image of the baseline configuration of a the computers before compromise. Another 250GB external hard drive was used as the Database Box to hold a database for which all critical data exported from Intermediate Storage Box would be stored.

This project produced four shell scripts *GRcompare*, *DELread*, *MACread* and *UNRMread* that served as the main information filtering tools. This project also produced the *Simple-MySQL* shell script that functioned as an interactive menu that allowed for database and table creation, record insertion and other functions. These scripts are discussed in *Section 2 - Attack Scenario*. The first four shell scripts were written to filter the information collected by one of the TCT tools and can be modified by the user if desired. The last shell script was written to organize and place the information returned from the other four scripts into a MySQL database and can also be modified by the user.

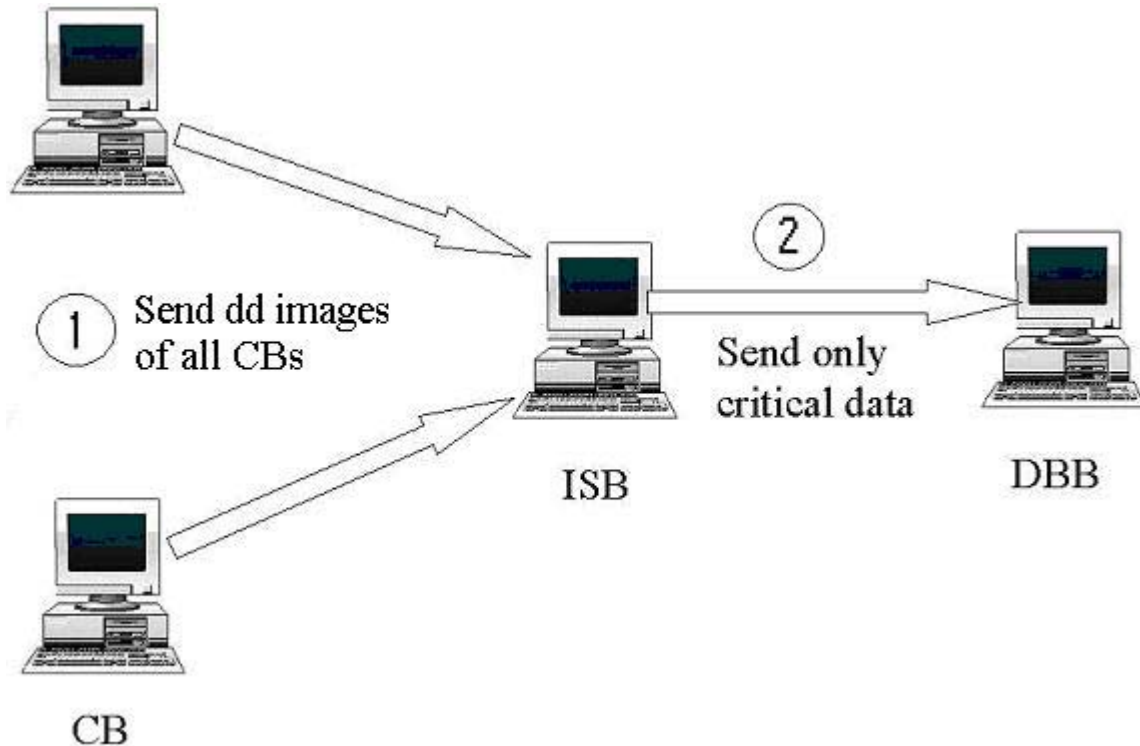
This project had four objectives:

- **To collect data from multiple compromised computers and store it directly on the ISB.**
- **To apply information filtering techniques to determine what critical data needed to be exported to the DBB.**
- **To store the critical data from each computer into a single database for possible future analysis or reference.**
- **To determine that evidence of the compromise could be traced to the DBB.**

## 1.4 The Architecture

The minimum requirements for this project are listed as:

- 2 computers or dd images of 2 computers before and after being compromised
- 1 Intermediate Storage Box (ISB) used to store the data gathered from TCT
- 1 Main Database Box (DBB) used to store only the critical data extracted from the main ISB
- The Coroner's Toolkit



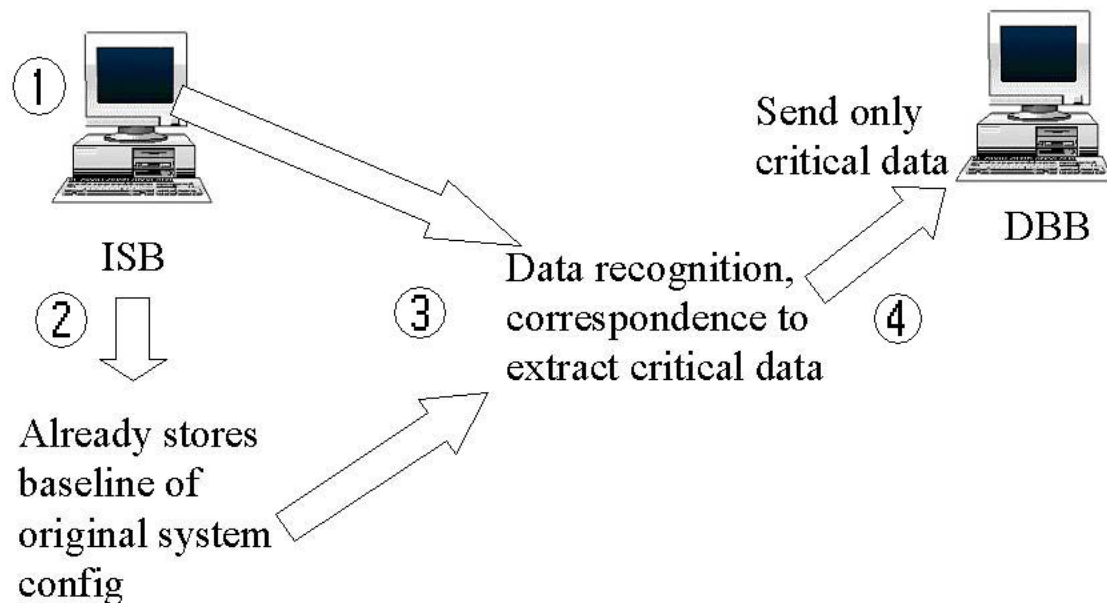
**Figure 1: Overall data flow**

The overall data flow figure above depicts the path the data gathered from the compromised computers (also called compromised boxes, hence CB) will take in order to be stored in the database located in the DBB.

Where:

- Step 1 Represents the use of the Unix dd command to image the hard-drives or individual file systems from each of the CBs where these images will be sent directly to the ISB via a USB link and analyzed.
- Step 2 Shows that only the critical data filtered out of the total data collected from the CBs will be sent to the DBB for storage in the database, also via a USB link.

Figure 2 below shows a general idea of the steps taken to filter out the critical data from the masses. Data recognition and correspondence would first be done in the ISB in order to find any differences between the results of the Toolkit and the baseline.



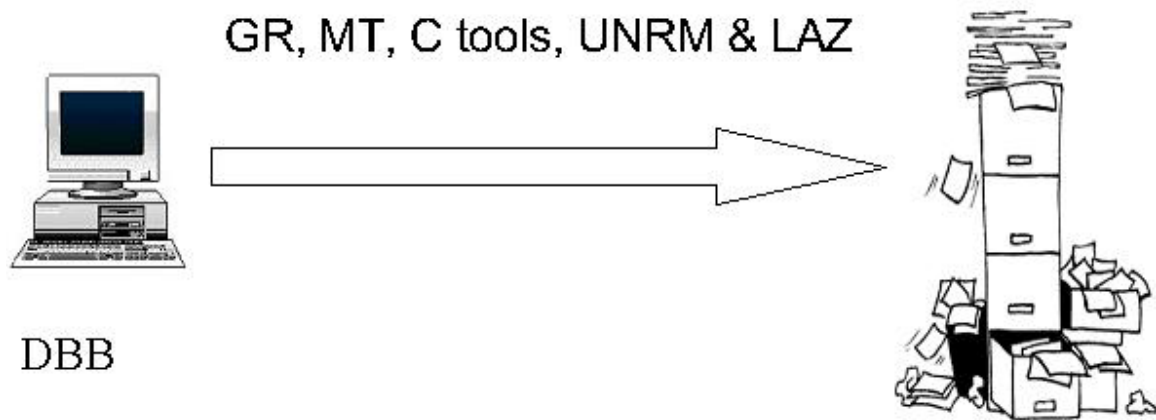
**Figure 2: Data filtration steps**

Where:

- Step 1 Represents all the data being stored in the ISB after TCT has been run on the dd images.
- Step 2 Indicates that a baseline of the original system configuration is also stored on the ISB. The baseline was gathered from running TCT (except for the urnm and lazarus tool) on a computer “out of the box”.
- Step 3 Implies that information filtering techniques will be applied inside the ISB. The techniques used to filter the critical data can be as simple as using the Unix command ‘diff’ or as complicated as doing manual filtering.
- Step 4 Finally presents the critical data collected from the filtration process being sent from the ISB to the DBB for storage in the database.

Instead of using multiple databases for storing the data gathered by each tool individually, it was felt that using one database to store all the data would further simplify the entire process for the user. By using multiple tables, the retrieving performance of the database should not be hampered too much. Also, for data that inherently cannot be placed in a database, the database will hold the

location value for that data instead. Figure 3 shows the general type of data that will be stored in the database.



**Figure 3: Database data**

## 1.5 Section Overview

This paper consists of five sections. The first section has just been presented and consisted of an introduction, plan of investigation, some background information, project purpose and the general architecture this project will utilize. *Section 2 – Attack Scenario*, will present the attack scenario that lead to the network compromise, project assumptions and a discussion of each script specifically written to aid in the information filtering process or database setup. *Section 3 – Project Scripts' Run Times & Results*, will display tables that hold information pertaining to the time it took to run the Project Scripts, the sizes of the initial raw data collected by TCT and the size of data produced by the Project Scripts for analysis. *Section 4 – Conclusion*, will deliver measurable results from the Project Scripts, closing remarks with a brief restatement of the results of the project. *Section 5 – Appendix*, will hold all everything from the Procedure to the Project Scripts intermediate results to example keyword files.

## 2.0 Attack Scenario

In order to test and demonstrate the shell scripts written for purposes of filtering the information collected by TCT an attack scenario was implanted. Due to time and policy constraints placed on this project an actual hack into the network was not performed. Instead ideas from some attacks described in some web-based documents were performed in a simplified manner. The scenario with assumptions is discussed below. Also, the most readily available operating systems for this project were RedHat 9 Linux and Mandrake 9.2 Linux and therefore were the only ones tested on.



### 2.0.1 Scenario Explained

Attacker gains access to the RedHat 9 Linux box by some unknown method. The Redhat box is the only box on the local network that has an internet connection. The Redhat box's specifications are as follows:

- Network name little.box.com
- 22 Gigabyte hard drive
- Kernel 2.4.20-8
- 2.66 GHz processor
- Pentium 4
- 1 Gigabyte physical memory
- eth0 address 192.xxx.xxx.xxx connected to internet
- eth1 address 192.168.123.3 connected to local Mandrake Linux box

Attacker compromises Mandrake 9.2 Linux box via RedHat box local network connection. Specifications for the Mandrake box are:

- Network name boss.box.com
- 3.7 Gigabyte hard drive
- Kernel 2.4.22-10mdk
- 400 MHz processor
- Pentium 2
- 128 Megabytes physical memory
- eth0 address 192.168.123.2 connected to local RedHat Linux box

Attacker presence was determined due to the user opening a SSH session on the Mandrake box where information was displayed informing the user that the last login time for that user was just over a minute prior to this session. At this point the user killed the SSH daemon and disconnected the local network from the internet and the investigation began.

### 2.0.2 Project Assumptions

- User has some ideas of how the network was compromised due to the SSH session that lead to the acknowledgement of a compromise
- User has taken baseline image of all computers on the network (not required for all information filtering scripts, but helpful)
- Due to sudden discovery the attacker was unable to touch the mactimes for any or all of the files and/or directories back to original times and dates
- TCT is used as data collection method as the filtering scripts were written in general to read the format of its output.

- Attacker has not written over all data to make evidence gathering impossible

## 2.1 GRcompare Script

The intent of the *GRcompare* script is to recursively search and compare two directories that hold the results from running the *grave-robber* tool. In essence, due to the way the script is written a comparison can be made on any two directories. For instance, if you have the before and after compromised images, after mounting both images as loopbacks this script can be run on the filesystems contained within images themselves. This is useful as it is possible to walk through the entire system finding new files, new hidden files and files that have been deleted or renamed. So what is the point of running the *grave-robber* tool?

If the above option is implemented there is a possibility that a plethora of information will be returned. This result can be caused by comparing two different partitions (i.e. Image1/etc and Image2/opt) or if an authorized user has modified many of the files after the base image was taken, this could basically defeat the purpose of the script. *Grave-robber* will return only the information it is configured to return by the user, which can be a lot or a little. Regardless when the *GRcompare* script is run on the returned data it keeps the user from having to look at every single piece of information gathered.

It searches through the main directories for files with the same name and recursively searches any 'same name' subdirectories and so on. Any file or directory that appears in one directory, but not the other will have its name placed in a 'missing.txt' file as seen in Figure 9. Therefore, if this file is empty both main directories contain all files and subdirectories with the same name. However, this does not mean that the contents of file A in directory 1 are the same as file A in directory 2.

When *GRcompare* comes across two files with the same name it does a comparison with the 'cmp' command. If the comparison fails a file ending with 'diff\_file.txt' will be made using the actual files name as the prefix that contains the complete contents of both files and where they differ as captured with a 'diff' command as seen in Figure 8.

### 2.1.1 Demonstration of GRcompare

Figure 4 below is of the directory where the result of running *GRcompare* will be stored. This location can be manually changed inside the script if the user wants to store results somewhere else.

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# ls
anaconda-ks.cfg      linux-2.4.23          PROJECT_PICS-BAK
CDD                  linux-2.4.23.tar      RH
ddimages.txt         MAN                   RH-DELETED
Desktop              MANDRAKE_UNRMread_file_dump  RH-GRdata
exchanges            MAN-GRdata            RH-img
forensic.ppt         MAN-img               test
hacked               MANWWW                tmp
images.tar           mysqlstuff             upgrade.log
imgtest              notes-McIver-UsingMySQLInt.pdf  upgrade.log.syslog
install.log          original              xtra
install.log.syslog   PROJECT

[root@dhcp122115 root]#
```

Figure 4: Initial image of root's home directory.

In the figure above there is a directory named "RH-GRdata". This directory contains the results from running *grave-robber* on the base and compromised images of the RedHat box as seen in Figure 5.

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

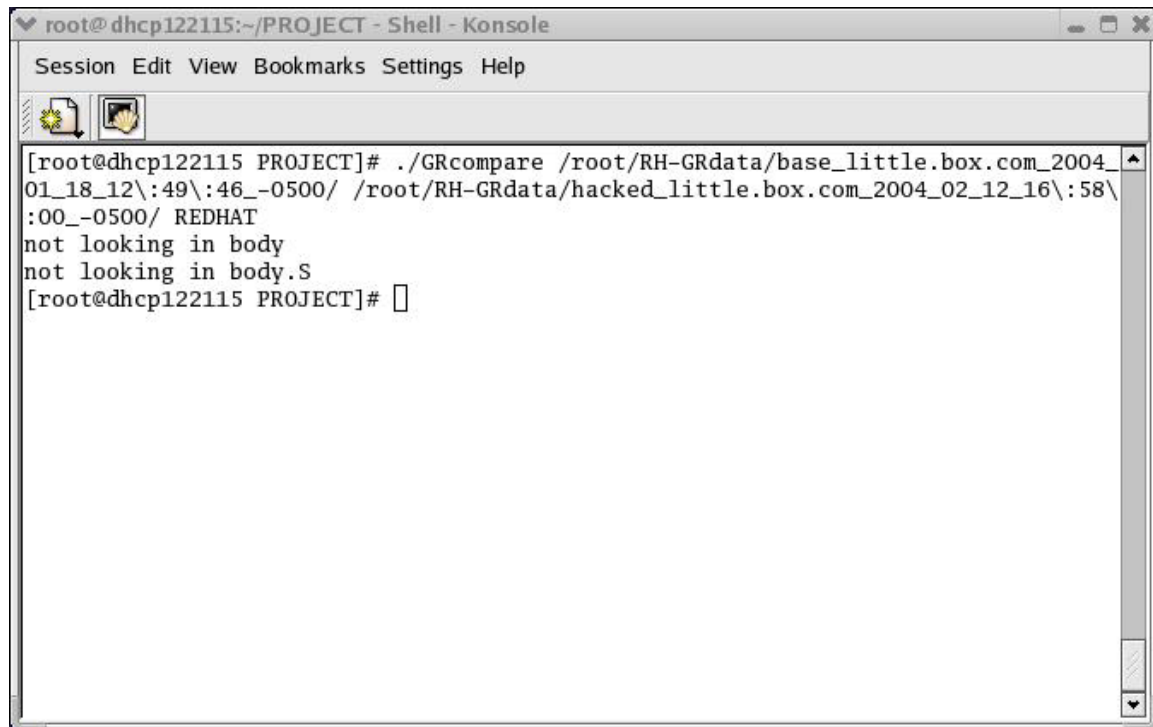
[root@dhcp122115 root]# ls RH-GRdata/
base_little.box.com
base_little.box.com_2004_01_18_12:49:46_-0500
hacked_little.box.com
hacked_little.box.com_2004_02_12_16:58:00_-0500
[root@dhcp122115 root]# ls RH-GRdata/base_little.box.com
body body.S command_out conf_vault MD5_all MD5_all.md5 trust user_vault
[root@dhcp122115 root]# ls RH-GRdata/hacked_little.box.com
body body.S command_out conf_vault MD5_all MD5_all.md5 trust user_vault
[root@dhcp122115 root]#
```

Figure 5: Grave-robber results

The titles "base" and "hacked" were appended to the front of "little.box.com" as a way to be able to identify which run is which. The *grave-*

*robber* tool was run with the same options on both images and therefore created storage locations with the same name for each image.

*GRcompare* is run on the command line and requires as arguments: two directories and the name of the operating system the image was taken from. This is shown in Figure 6 below.



```
root@dhcp122115:~/PROJECT - Shell - Konsole
Session Edit View Bookmarks Settings Help

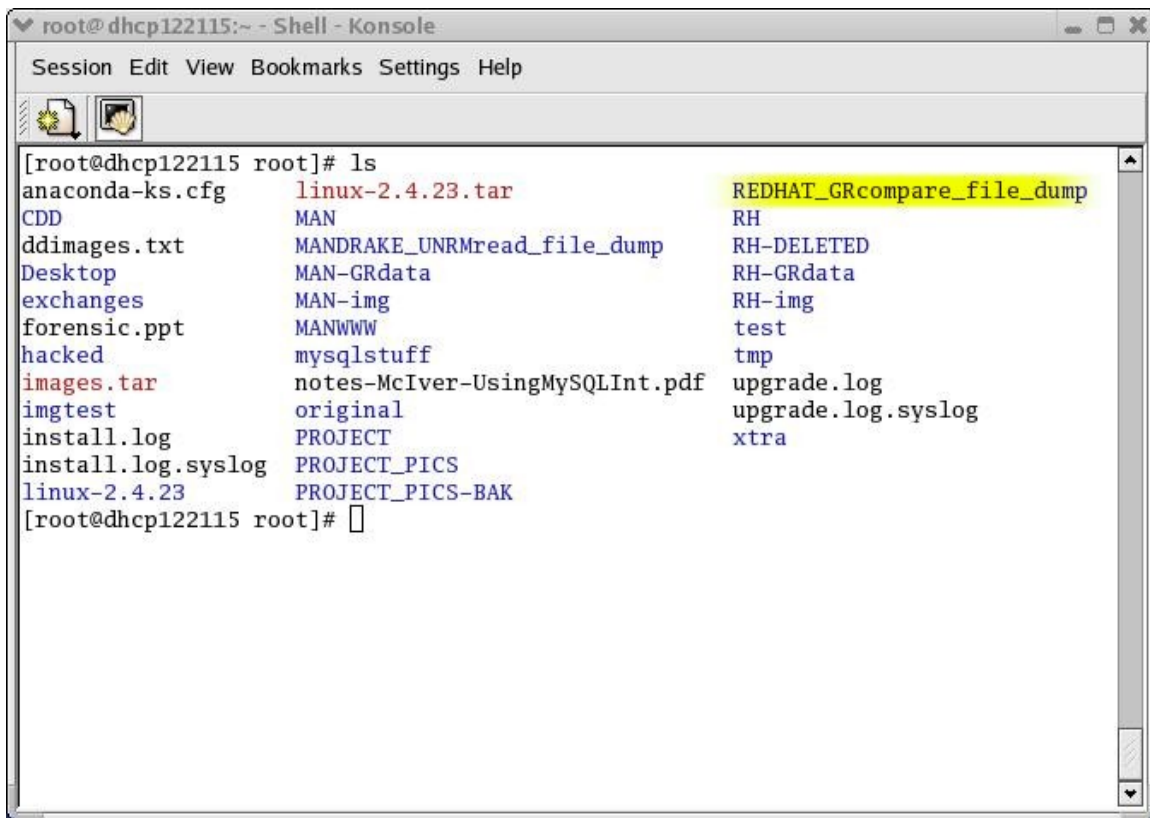
[root@dhcp122115 PROJECT]# ./GRcompare /root/RH-GRdata/base_little.box.com_2004_01_18_12\:49\:46_-0500/ /root/RH-GRdata/hacked_little.box.com_2004_02_12_16\:58\:00_-0500/ REDHAT
not looking in body
not looking in body.S
[root@dhcp122115 PROJECT]#
```

**Figure 6: Running GRcompare**

As seen in the figure above, the third argument “REDHAT” is the name of the operating system the data being compared belongs to. In actuality this third argument could be anything and is not case sensitive. It is merely used as a way to identify the results made from this particular run and therefore any text type here will be accepted. Also, the output “not looking in body” and “not looking in body.S” appear. As seen in Figure 6 these are files created by *grave-robber*, but they are ignored by *GRcompare*.

The reason the aforementioned files are not looked at is they are actually Mactime databases. As stated previously in this document *grave-robber* can be used to collect and store the mactimes from the images. These files contain any newly created, accessed or modified file’s and/or executable’s name and therefore would contain dramatically different results. Recognizing this, a different approach is used to examine their contents and will be discussed in *Section 2.3 MACread script*.

After executing *GRcompare* a new directory is created. This is shown in Figure 7.

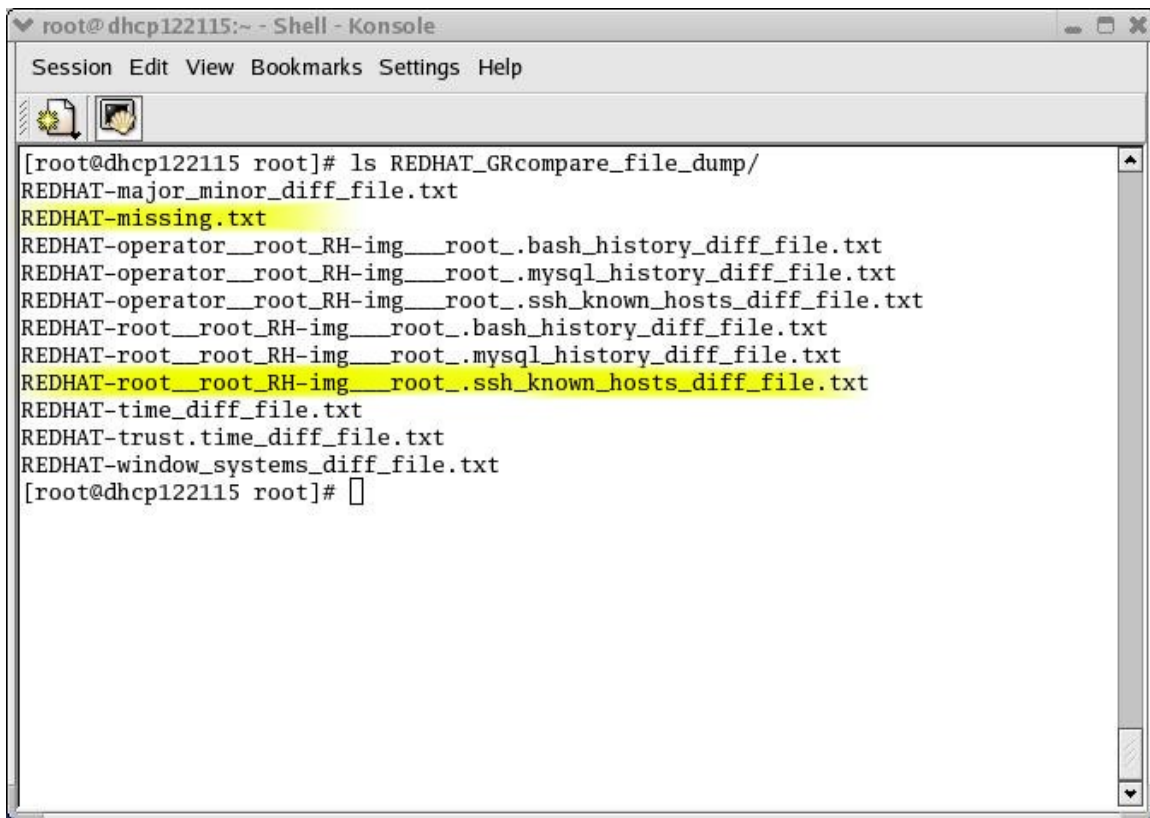


```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# ls
anaconda-ks.cfg      linux-2.4.23.tar      REDHAT_GRcompare_file_dump
CDD                  MAN                   RH
ddimages.txt         MANDRAKE_UNRMread_file_dump  RH-DELETED
Desktop              MAN-GRdata            RH-GRdata
exchanges            MAN-img               RH-img
forensic.ppt         MANWWW                test
hacked               mysqlstuff             tmp
images.tar           notes-McIver-UsingMySQLInt.pdf  upgrade.log
imgtest              original               upgrade.log.syslog
install.log          PROJECT               xtra
install.log.syslog   PROJECT_PICS
linux-2.4.23         PROJECT_PICS-BAK
[root@dhcp122115 root]#
```

**Figure 7: GRcompare directory**

The new directory is named “REDHAT\_GRcompare\_file\_dump”. “REDHAT” is taken from the third argument of the command line execution of the script and the “GRcompare” tells the user that this directory contains the results from the *GRcompare* script. Viewing the contents of the new directory shows various files, all of which contain the discrepancies found between the two images as displayed in Figure 8.



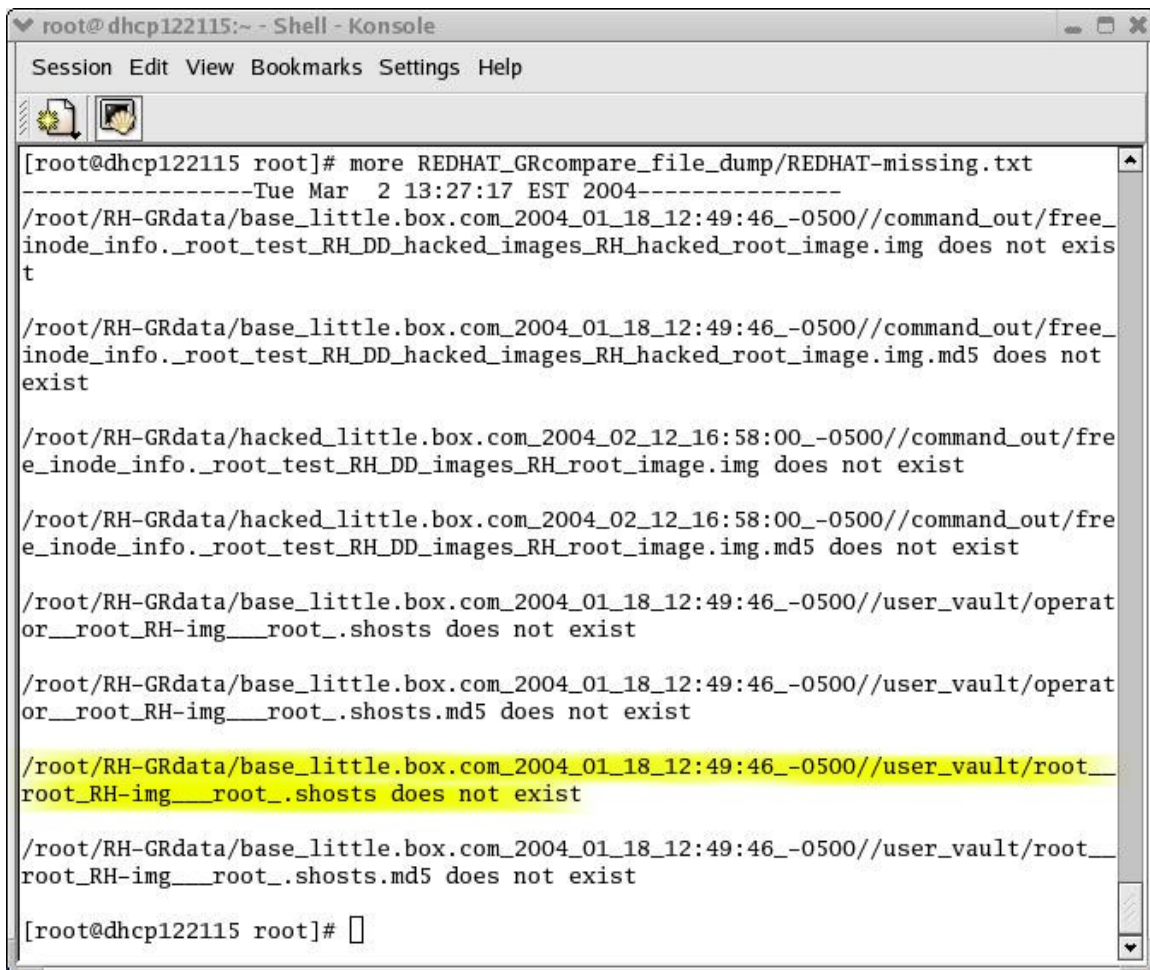
```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# ls REDHAT_GRcompare_file_dump/
REDHAT-major_minor_diff_file.txt
REDHAT-missing.txt
REDHAT-operator__root_RH-img__root_.bash_history_diff_file.txt
REDHAT-operator__root_RH-img__root_.mysql_history_diff_file.txt
REDHAT-operator__root_RH-img__root_.ssh_known_hosts_diff_file.txt
REDHAT-root__root_RH-img__root_.bash_history_diff_file.txt
REDHAT-root__root_RH-img__root_.mysql_history_diff_file.txt
REDHAT-root__root_RH-img__root_.ssh_known_hosts_diff_file.txt
REDHAT-time_diff_file.txt
REDHAT-trust.time_diff_file.txt
REDHAT-window_systems_diff_file.txt
[root@dhcp122115 root]#
```

**Figure 8: GRcompare results**

The main files of interest here are the “REDHAT-missing.txt” and the “REDHAT-root\_\_root\_RH-img\_\_root\_.ssh\_known\_hosts\_diff\_file.txt”. The other files may contain evidence of the compromise and is up to the user to further investigate. Also, depending on how the user configures the *grave-robber* tool before running it, more or less files may be collected which contain different evidence.

Either way the missing.txt file should be examined first as it contains missing file information. This means someone (probably the attacker) may have added new files or removed previously existing ones. The format of the missing.txt file is displayed in Figure 9.



```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_GRcompare_file_dump/REDHAT-missing.txt
-----Tue Mar  2 13:27:17 EST 2004-----
/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//command_out/free_
inode_info._root_test_RH_DD_hacked_images_RH_hacked_root_image.img does not exist

/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//command_out/free_
inode_info._root_test_RH_DD_hacked_images_RH_hacked_root_image.img.md5 does not
exist

/root/RH-GRdata/hacked_little.box.com_2004_02_12_16:58:00_-0500//command_out/fre
e_inode_info._root_test_RH_DD_images_RH_root_image.img does not exist

/root/RH-GRdata/hacked_little.box.com_2004_02_12_16:58:00_-0500//command_out/fre
e_inode_info._root_test_RH_DD_images_RH_root_image.img.md5 does not exist

/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//user_vault/operat
or__root_RH-img__root_.shosts does not exist

/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//user_vault/operat
or__root_RH-img__root_.shosts.md5 does not exist

/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//user_vault/root__
root_RH-img__root_.shosts does not exist

/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//user_vault/root__
root_RH-img__root_.shosts.md5 does not exist

[root@dhcp122115 root]#
```

**Figure 9: Missing.txt format**

The missing.txt file contains the date the information was gathered on. This allows for the *GRcompare* script to be executed numerous times as any new entries will simply be appended to the bottom of this file using the date a separator.

The file is read as: First everything before the '/' list the directory entered as one of the command line arguments. Everything following the '/' list the path to the file that does not exist in the directory. In other words, taking the second to last entry in the file for example. It is reporting that the file '/user\_vault/root\_\_root\_RH-img\_\_root\_.shosts' does not exist in the directory '/root/RH/Grdata/base\_little.box.com'. This means this particular file only exists in a subdirectory of hacked\_little.box.com. This also happens to be a very important piece of information because it means this 'hidden' file did not originally exist when the user set up the initial network. The contents of this can be seen in Figure 10.



```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# cat RH-GRdata/hacked_little.box.com/user_vault/root__root_RH-img__root_.shosts
Mon Feb 16 11:33:20 EST 2004
shady.user.org root

[root@dhcp122115 root]#
```

**Figure 10: Missing .shosts file**

In this file we see an entry for “shady.user.org”. This entry allows any member from shady.user.org to open an SSH session with root access to the local user’s RedHat box. Not only does the attacker have root access to the user’s RedHat box, but also the attacker does not need to use a password to open an ssh session. This provides the user with key information that can be used in some of the scripts discussed in the following sections to help further filter out critical information.

Looking back at Figure 10 this new information gives the user some insight to what the “REDHAT-root\_\_root\_RH-img\_\_root\_.ssh\_known\_hosts\_diff\_file.txt” may contain. This information is presented in Figure 11.

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_GRcompare_file_dump/REDHAT-root__root_RH-img__root_.ssh_known_hosts_diff_file.txt
-----Tue Mar  2 13:27:17 EST 2004-----
/root/RH-GRdata/base_little.box.com_2004_01_18_12:49:46_-0500//user_vault/root__root_RH-img__root_.ssh_known_hosts ---different from--- /root/RH-GRdata/hacked_little.box.com_2004_02_12_16:58:00_-0500//user_vault/root__root_RH-img__root_.ssh_known_hosts
Sun Jan 18 13:55:42 EST 2004
diablo.cs.fsu.edu,128.186.121.40 ssh-rsa AAAAB3NzaC1yc2EAAAAB | Thu Feb 12 18:35:25 EST 2004
garnet.acns.fsu.edu,146.201.2.25 ssh-dss AAAAB3NzaC1kc3MAAACB | diablo.cs.fsu.edu,128.186.121.40 ssh-rsa AAAAB3NzaC1yc2EAAAAB
> 192.168.123.2 ssh-rsa AAAAB3NzaC1yc2EAAAABIAAAIEA3XKD4Ie6+AF | garnet.acns.fsu.edu,146.201.2.25 ssh-dss AAAAB3NzaC1kc3MAAACB
> shady.user.org ssh-rsa AAAAB3NzaC1yc2EAAAABIAAAIEA3XKD4Ie6+A |
> 192.168.123.3 ssh-rsa AAAAB3NzaC1yc2EAAAABIAAAIEAw43mZ1yt673 |

[root@dhcp122115 root]#
```

**Figure 11: Example ssh-knownhosts file**

The format of the diff\_file.txt files is shown in the figure above. Once again the date is the first entry in the file, followed by a statement that can be simplified to “directory1/fileA -- is different from -- directory2/fileA”. The two files appear side by side and the ‘>’ character is used to point out where the two files differ. Collecting the information in this manner saves the user from have to view



each file separately and having to collect and rename multiple files when moving them for permanent storage in the DBB.

## 2.2 DELread Script

The *DELread* script is used to search the output created from the *ils* and *icat* tools. The deleted files are recovered and written to the “RH-DELETED” directory using the following code provided by Thomas Roessler:<sup>(2)</sup>

```
# ./ils -rf ext2fs /image/dev_hda1.img | awk -F '|' '($2=="f") {print $1}'  
| while read i; do /usr/local/tct-1.07/bin/icat /image/dev_hda1.img $i  
> /tmp/deleted/$i; done
```

This code is entered on the command line and the path to ‘icat’ is dependent on where the user installs The Coroner’s Toolkit. Also, the ‘/tmp/deleted/’ directory has to be created before this code is executed and can be called whatever the user wants to call it. As stated previously, for this project deleted files were recovered and stored to in a directory named “RH-DELETED”.

The recovered files can be in any format such as: English text, ASCII text, data, ELF, pictures, gz, etc. and are giving numbers as file names (see Appendix B). Recognizing this and meeting the requirement to filter extraneous information, *DELread* is written to only read the ‘text’ files, data files and ELF files. Applying the search to only these file formats combined with the use of keywords decreased the total amount of information the user had to look through from 3360 files to 210 files (the user doesn’t actually have to look through 210 files either, only the keyword files created from the 210 files that contained matches) and from 102MB to 3MB (actually less than 3Mb because much of the information is redundant based on the keywords used) for this project. The user can also choose whether or not to view the other information.

*DELread* basically is a keyword search script. It requires as arguments: the directory that contains the recovered files, a keyword file named “DEL-keyword.txt” and the name of the operating system as shown in Figure 12. The DEL-keyword.txt file used for *DELread* can be found in Appendix B as Figure 41. As with *GRcompare* a new directory is created taking the name of the operating system and the script as identifiers. The DEL-keyword file is the where the user places the keywords used to extract information from the recovered files.

```
root@dhcp122115:~/PROJECT - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 PROJECT]# ./DELread /root/RH-DELETED/ DEL-keyword.txt REDHAT
[root@dhcp122115 PROJECT]# ls ..
anaconda-ks.cfg      linux-2.4.23.tar      REDHAT_DELread_file_dump
CDD                  MAN                   REDHAT_GRcompare_file_dump
ddimages.txt         MANDRAKE_UNRMread_file_dump  RH
Desktop              MAN-GRdata            RH-DELETED
exchanges            MAN-img               RH-GRdata
forensic.ppt         MANWWW               RH-img
hacked               mysqlstuff            test
images.tar           notes-McIver-UsingMySQLInt.pdf tmp
imgtest              original              upgrade.log
install.log          PROJECT               upgrade.log.syslog
install.log.syslog   PROJECT_PICS          xtra
linux-2.4.23         PROJECT_PICS-BAK
[root@dhcp122115 PROJECT]#
```

**Figure 12: Running DELread**

The '@' character is used as the comment character in the DEL-keyword file. The user can add new entries to this file with the restriction that only one keyword be placed on each line. The user can delete entries from the file or as just stated place the '@' character immediately in front of a keyword (i.e. @ssh) so when *DELread* comes across this entry it will be ignored and not used as a search keyword. Due to the way the script is written the user can repeatedly search the recovered files for new information based on information gathered from previous searches.

The contents of the "REDHAT\_DELread\_file\_dump" directory are listed in Figure 13 below.

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# ls REDHAT_DELread_file_dump/
REDHAT-1007645 REDHAT-1910787 REDHAT-2945045 REDHAT-3782674 REDHAT-5087258 REDHAT-686082
REDHAT-1015815 REDHAT-1918979 REDHAT-2951176 REDHAT-3794948 REDHAT-5107732 REDHAT-714755
REDHAT-1017871 REDHAT-1925157 REDHAT-2965514 REDHAT-3799058 REDHAT-5117985 REDHAT-778255
REDHAT-1064966 REDHAT-1927179 REDHAT-3026955 REDHAT-3803139 REDHAT-5126164 REDHAT-819220
REDHAT-1067014 REDHAT-1974278 REDHAT-303130 REDHAT-3852309 REDHAT-5177392 REDHAT-839695
REDHAT-1112073 REDHAT-2021378 REDHAT-305163 REDHAT-3872798 REDHAT-5183490 REDHAT-854021
REDHAT-1130517 REDHAT-2127878 REDHAT-3063821 REDHAT-3889155 REDHAT-5255187 REDHAT-864282
REDHAT-1185811 REDHAT-213002 REDHAT-3080198 REDHAT-3895300 REDHAT-5255188 REDHAT-894988
REDHAT-1269765 REDHAT-2134035 REDHAT-3082250 REDHAT-4030496 REDHAT-5275655 REDHAT-938003
REDHAT-1273876 REDHAT-2148414 REDHAT-3090441 REDHAT-4030498 REDHAT-530435 REDHAT-991246
REDHAT-1282233 REDHAT-2162697 REDHAT-3102732 REDHAT-4030500 REDHAT-5359620 REDHAT-COMMAND=_inode_list.txt
REDHAT-1318914 REDHAT-2168857 REDHAT-315401 REDHAT-4030502 REDHAT-5367814 REDHAT-COMMAND=_keyword.txt
REDHAT-1417246 REDHAT-2191374 REDHAT-3155977 REDHAT-4081677 REDHAT-5378061 REDHAT-exportfs_inode_list.txt
REDHAT-1441819 REDHAT-2195468 REDHAT-3260450 REDHAT-4087811 REDHAT-546819 REDHAT-exportfs_keyword.txt
REDHAT-1441820 REDHAT-225314 REDHAT-3260485 REDHAT-4163586 REDHAT-5470211 REDHAT-Feb_inode_list.txt
REDHAT-1441822 REDHAT-2265191 REDHAT-3266571 REDHAT-4188171 REDHAT-5570592 REDHAT-Feb_keyword.txt
REDHAT-1441830 REDHAT-2269195 REDHAT-3280906 REDHAT-4251651 REDHAT-5570594 REDHAT-filetypes.txt
REDHAT-1441833 REDHAT-2357320 REDHAT-3283038 REDHAT-4292709 REDHAT-5570595 REDHAT-hostbased_inode_list.txt
REDHAT-1441835 REDHAT-2363459 REDHAT-3293206 REDHAT-4294663 REDHAT-5570596 REDHAT-hostbased_keyword.txt
REDHAT-1441840 REDHAT-2377751 REDHAT-3309580 REDHAT-4392989 REDHAT-5570599 REDHAT-Logfiles_inode_list.txt
REDHAT-1441843 REDHAT-2416644 REDHAT-3311620 REDHAT-4438046 REDHAT-5570601 REDHAT-Logfiles_keyword.txt
REDHAT-1441846 REDHAT-245765 REDHAT-3371012 REDHAT-4454405 REDHAT-5570603 REDHAT-pam_unix_inode_list.txt
REDHAT-1441852 REDHAT-2459695 REDHAT-3399708 REDHAT-4597786 REDHAT-5570604 REDHAT-pam_unix_keyword.txt
REDHAT-1527827 REDHAT-2461739 REDHAT-3524610 REDHAT-464899 REDHAT-5570607 REDHAT-password_inode_list.txt
REDHAT-1564831 REDHAT-2465794 REDHAT-3596304 REDHAT-466952 REDHAT-5570609 REDHAT-password_keyword.txt
REDHAT-1564832 REDHAT-2469893 REDHAT-362517 REDHAT-4702224 REDHAT-5570610 REDHAT-rhost=_inode_list.txt
REDHAT-1564838 REDHAT-2471969 REDHAT-3643431 REDHAT-4757533 REDHAT-5570611 REDHAT-rhost=_keyword.txt
REDHAT-1660931 REDHAT-247822 REDHAT-3647505 REDHAT-4767757 REDHAT-5570617 REDHAT-rpm_inode_list.txt
REDHAT-1720338 REDHAT-2484229 REDHAT-3647507 REDHAT-4802577 REDHAT-5570619 REDHAT-rpm_keyword.txt
REDHAT-1728526 REDHAT-2486293 REDHAT-3653657 REDHAT-485380 REDHAT-5570620 REDHAT-ssh_inode_list.txt
REDHAT-1728538 REDHAT-2568195 REDHAT-3686405 REDHAT-4862014 REDHAT-5570621 REDHAT-ssh_keyword.txt
REDHAT-1787992 REDHAT-2580485 REDHAT-36895 REDHAT-4876290 REDHAT-5570622 REDHAT-uid=_inode_list.txt
REDHAT-1816578 REDHAT-2627601 REDHAT-3690514 REDHAT-4884482 REDHAT-5570623 REDHAT-uid=_keyword.txt
REDHAT-1824778 REDHAT-2674691 REDHAT-3719176 REDHAT-4888581 REDHAT-5570632 REDHAT-user=_inode_list.txt
REDHAT-1826836 REDHAT-26772 REDHAT-3737612 REDHAT-4900866 REDHAT-5589107 REDHAT-USER=_inode_list.txt
REDHAT-1851406 REDHAT-280582 REDHAT-3776546 REDHAT-4907032 REDHAT-559107 REDHAT-user=_keyword.txt
REDHAT-1855503 REDHAT-2930715 REDHAT-3776549 REDHAT-499722 REDHAT-5617700 REDHAT-USER=_keyword.txt
REDHAT-1859591 REDHAT-2932767 REDHAT-3778628 REDHAT-5011463 REDHAT-575496
REDHAT-1863699 REDHAT-2934802 REDHAT-3778631 REDHAT-5054466 REDHAT-598048
REDHAT-1884163 REDHAT-2936841 REDHAT-3778632 REDHAT-5072925 REDHAT-606215
[root@dhcp122115 root]#
```

Figure 13: DELread results

As *DELread* walks through the RH-DELETED directory whose complete contents can be viewed in Appendix B, it takes a keyword from DEL-read.txt and if and searches a file for matches. If the keyword is not found within the file then the next keyword in the DEL-read file is used to check for matches in the file. If none of the keywords happen to be found in the file then no information for that file is written to the new directory and the next file is searched for all the keywords and so on. This feature automatically filters out any file that does not contain information the user is looking for.

When a keyword is matched in a file, the file is copied to the new directory and the operating system's name is appended to the front of the file number (i.e. REDHAT-1007645). Also, two files are created using the operating system's name and the matched keyword as file identifiers. The files made for the matched keyword end with extensions '\_keyword.txt' and '\_inode.txt' whose formats appear in Figures 14 and 15. In addition, if the same keyword is found in multiple files, the new files are copied over and the matching lines are appended to the already existing keyword's keyword.txt file while the new files number is added to the keyword's inode.txt file.

```

Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-USER\=_keyword.txt
-----entries found in file /root/RH-DELETED//1564832-----
Feb  4 15:29:09 little sudo:      root : TTY=pts/2 ; PWD=/etc/ssh ; USER=root ; COMMAND=/usr/sbin/useradd
Feb  4 15:29:19 little sudo:      root : TTY=pts/2 ; PWD=/etc/ssh ; USER=root ; COMMAND=/usr/sbin/useradd sdsd
Feb  4 17:25:15 little sudo:      rpg  : command not allowed ; TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=useradd gp
Feb  4 17:35:24 little sudo:      rpg  : TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=list
Feb  4 17:35:44 little sudo:      rpg  : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=list
Feb  4 17:39:22 little sudo:      rpg  : command not allowed ; TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=useradd gp
Feb  4 17:53:14 little sudo:      rpg  : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/useradd -u 0 -d /root
bossy
Feb  4 17:59:20 little sudo:      rpg  : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/useradd -u 0 go
Feb  4 18:00:08 little sudo:      rpg  : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/useradd -d /root bossy
Feb  4 18:00:30 little sudo:      rpg  : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/usermod -u 0 bossy
Feb  4 18:06:32 little sudo:      rpg  : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb  4 18:34:57 little sudo:      rpg  : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb  4 18:36:31 little sudo:      rpg  : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi /etc/profile
Feb  4 18:37:22 little sudo:      rpg  : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi .bash_profile
Feb  4 18:38:53 little sudo:      rpg  : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb  4 18:39:34 little sudo:      rpg  : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi /etc/profile
-----entries found in file /root/RH-DELETED//2459695-----
USER=root
-----entries found in file /root/RH-DELETED//5570611-----
203) Added "USER=%s" to logging functions.
[root@dhcp122115 root]#

```

**Figure 14: Example \_keyword.txt file**

This example file shows that the keyword being search for was “USER=” and it was matched in three separate files. The matching lines of all files were copied to a central file for the user’s benefit. Also, this example figure just happens to present new evidence of the compromise and presents possible new keywords such as *sudo*, *bossy* and *rpg* that can be used in future searches as the owner of this network did not originally have such users or use the sudo command.

Accompanying the keyword.txt file is the inode.txt file that lists the files in which the keywords found matches.

```

Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-USER\=_inode_list.txt
1564832
2459695
5570611
[root@dhcp122115 root]#

```

**Figure 15: Example inode.txt file**

The inode.txt example file above simply list the files that the contain entries in the keyword.txt file. This file is important because the keyword.txt file can become very large depending on the keyword being searched for, resulting in difficulties

for the user in identifying which of the recovered files contain information on a particular keyword. Also, this listing enables the user to have the option of viewing the entire contents of a matched file giving the user some insight on what the file contains instead of blindly searching all the copied over files.

## 2.3 MACread script

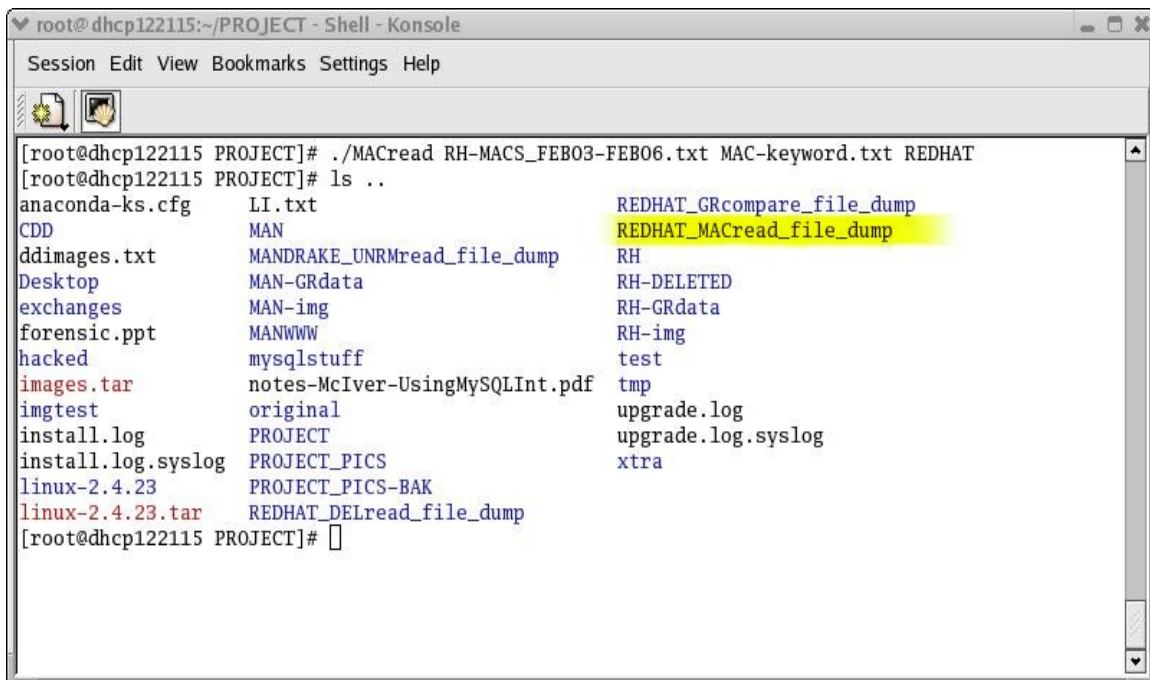
The *MACread* script is written to search through a single file created by redirecting the output from the *mactime* tool. For the purposes of this project the command line argument for creating this file was:

```
# ./mactime -b <location of the 'body' file created by Grave-  
Robber> [start date]-[end date] >> < filename.txt>
```

The 'body' file created by the *grave-robber* tool is a database file that uses the '|' character as field separators which makes this file hard to read. Also, this file contains any modified, accessed or created file time dating back to when the system was first made operational. The body file for the RedHat box in this project was 21Mb in size. By using the *mactime* tool and specifying specific start and stop dates, a file of 64Kb in size was created and examined. Using *MACread* on this file along with another keyword search file, the total amount of information made available to the user was 16Kb.

The *MACread* script takes as arguments: the name of the file the user wants to search, the "MAC-keyword.txt" file which contains the keywords used to search the file and the name of the operating system. The MAC-keyword.txt file can be viewed in Appendix C as Figure 58. As the previous scripts do, *MACread* also creates its own directory for result storage using the operating system and script name arguments as identifiers. Both of these features can be seen in the following Figure 16.



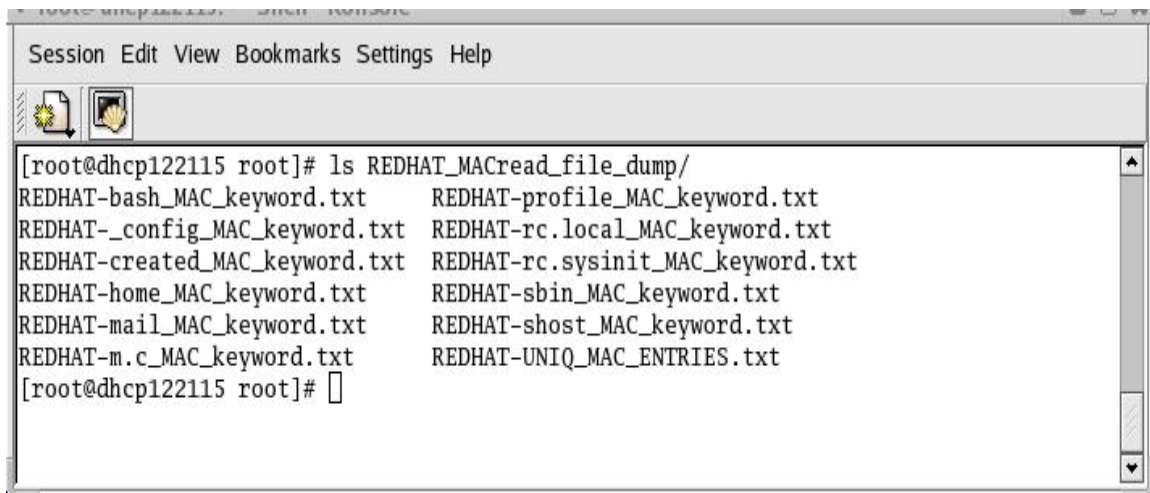


```
root@dhcp122115:~/PROJECT - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 PROJECT]# ./MACread RH-MACS_FEB03-FEB06.txt MAC-keyword.txt REDHAT
[root@dhcp122115 PROJECT]# ls ..
anaconda-ks.cfg      LI.txt                REDHAT_GRcompare_file_dump
CDD                  MAN                   REDHAT_MACread_file_dump
ddimages.txt          MANDRAKE_UNRMread_file_dump  RH
Desktop               MAN-GRdata            RH-DELETED
exchanges             MAN-img               RH-GRdata
forensic.ppt          MANWWW                RH-img
hacked                mysqlstuff            test
images.tar            notes-McIver-UsingMySQLInt.pdf tmp
imgtest               original              upgrade.log
install.log           PROJECT               upgrade.log.syslog
install.log.syslog    PROJECT_PICS          xtra
linux-2.4.23          PROJECT_PICS-BAK
linux-2.4.23.tar       REDHAT_DELread_file_dump
[root@dhcp122115 PROJECT]#
```

**Figure 16: Running MACread**

The MAC-keyword.txt file follows the same guidelines as the DEL-keyword.txt file. The reason the DEL-keyword.txt file is not allowed to be used as the keyword file for *MACread* is because the format of the file to be searched is different from the files *DELread* searches through. Keywords such as 'm.c' or '..c' are used to find occurrences of files that have been 'modified and created' or just 'modified or created'. That does not mean that many of the keywords used for *DELread* cannot be used for *MACread*. For instance, the user could place 'ssh' and 'Feb' in either keyword file, but would not place USER=, COMMAND= or hostbased keyword in the MAC-keyword file as these words more then likely will not have occurrences in the file containing mactimes. Therefore, the use of two uniquely named files saves the user from having to continuously comment out, delete and read all the keywords that would not be suitable for searches in one instance, but valid for another. With this pointed out, the contents stored in the *MACread* file dump directory can be seen in Figure 17.

A screenshot of a terminal window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and two icons (a folder and a document). The terminal shows a root prompt at a DHCP IP address. The user runs 'ls REDHAT\_MACread\_file\_dump/' and the output lists ten files in two columns: REDHAT-bash\_MAC\_keyword.txt, REDHAT-\_config\_MAC\_keyword.txt, REDHAT-created\_MAC\_keyword.txt, REDHAT-home\_MAC\_keyword.txt, REDHAT-mail\_MAC\_keyword.txt, REDHAT-m.c\_MAC\_keyword.txt, REDHAT-profile\_MAC\_keyword.txt, REDHAT-rc.local\_MAC\_keyword.txt, REDHAT-rc.sysinit\_MAC\_keyword.txt, REDHAT-sbin\_MAC\_keyword.txt, REDHAT-shost\_MAC\_keyword.txt, and REDHAT-UNIQ\_MAC\_ENTRIES.txt.

```
[root@dhcp122115 root]# ls REDHAT_MACread_file_dump/
REDHAT-bash_MAC_keyword.txt      REDHAT-profile_MAC_keyword.txt
REDHAT-_config_MAC_keyword.txt   REDHAT-rc.local_MAC_keyword.txt
REDHAT-created_MAC_keyword.txt   REDHAT-rc.sysinit_MAC_keyword.txt
REDHAT-home_MAC_keyword.txt      REDHAT-sbin_MAC_keyword.txt
REDHAT-mail_MAC_keyword.txt      REDHAT-shost_MAC_keyword.txt
REDHAT-m.c_MAC_keyword.txt       REDHAT-UNIQ_MAC_ENTRIES.txt
[root@dhcp122115 root]#
```

**Figure 17: Results from MACread**

Many of the keyword.txt files created contain redundant information, this is unavoidable because the information gathered depends on the keywords used. Therefore a file named “REDHAT-UNIQ\_MAC\_ENTRIES.txt” is created to hold all the unique information collected from all the keyword.txt files created by *MACread*. This gives the user the option to search a given keyword’s file for clues or search all the results at once for evidence. It’s important to understand that these results can only tell the user that something has been modified, accessed or created. The results will not show how something was modified, what was being done while being accessed or the contents of the creation. The format of the keyword file is displayed in Figure 19.

```

[Session Edit View Bookmarks Settings Help]

[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-_config_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 03 04 14:12:19    2473 m.c -rw----- root/bossy root    /root/RH-img/etc/ssh/sshd_config~
Feb 03 04 14:28:42    2473 .a. -rw----- root/bossy root    /root/RH-img/etc/ssh/sshd_config~
Feb 03 04 14:31:28    2471 m.c -rw----- root/bossy root    /root/RH-img/etc/ssh/sshd_config
Feb 05 04 17:51:47    1167 .a. -rw-r--r-- root/bossy root    /root/RH-img/etc/ssh/ssh_config
[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-profile_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
                191 .a. -rw-r--r-- 501    bossy    /root/RH-img/.bash_profile
                191 m.c -rw-r--r-- bob     bob     /root/RH-img/root/.bash_profile
Feb 04 04 18:39:59    841 m.c -rw-r--r-- root/bossy root    /root/RH-img/etc/profile
                191 m.c -rw-r--r-- 501    bossy    /root/RH-img/.bash_profile
                191 mac -rw-r--r-- bob     bob     /root/RH-img/home/bob/.bash_profile
                191 mac -rw-r--r-- rpg     rpg     /root/RH-img/var/rpg/.bash_profile
                191 .a. -rw-r--r-- root/bossy root    /root/RH-img/etc/skel/.bash_profile
[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-shost_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 03 04 14:24:36     19 m.c -rw-r--r-- root/bossy root    /root/RH-img/root/.shosts~
Feb 03 04 14:56:23     19 .a. -rw-r--r-- root/bossy root    /root/RH-img/root/.shosts~
[root@dhcp122115 root]# 

```

**Figure 19: Example keyword.txt files created by MACread**

The above figure actually presents information from three different keyword.txt files created by *MACread*. The first of these example files inform the user that the `sshd_config` file has not only been modified, accessed and created, but its ownership has been shared between user *root* and user *bossy*. The second file informs the user that new users are present in *bossy*, *bob* and *rpg* and also possible modifications have been made to the `/etc/profile` file and root's `.bash_profile` file. The third file shows the backups left from the creation of a `.shosts` file in root's home directory. Further findings made by *MACread* can be found in Appendix C.

## 2.4 UNRMread script

The *UNRMread* script is used to search either the single file created as output from the *unrm* tool or however many files are created when the *lazarus* tool is run on the output from *unrm*. The *unrm* tool is the most time-consuming tool in The Coroner's Toolkit. As explained earlier, the size of *unrm*'s output is



dependent on the amount of unallocated space left on the user's hard drive, which can easily result in gigabytes of data being returned. Given this fact, it can take *lazarus* hours, days or weeks to completely run on *unrm*'s output. If *lazarus* can be run on the data in a reasonable time frame as decided by the user it creates a directory where it stores its recovered files.

Understanding this *UNRMread* is written to be able to accept as arguments: either a directory or a single file to search through the "DEL-keyword.txt" file and the name of the operating system. The DEL-keyword.txt file can be found in Appendix D as Figure 66. The reason the DEL-keyword.txt file can be used with *UNRMread* as well as *DELread* is because the format of the output created by *unrm* is simply one huge data file that contains a variety of formats including text. Running *lazarus* on *unrm*'s output clears up much of the garbage and creates a directory similar to that of the "RH-DELETED". If *UNRMread* is run on the *unrm* file, the resulting keyword files will contain much garbage caused by the matching of contents in executable, sound and archive files. This is unavoidable if the user cannot afford the time it takes to run *lazarus* on the *unrm* data file. Either way all the files can be forced read as text files and searched for any keywords.

In this project the *unrm* result was a 2.5 Gigabyte (GB) data file for the Mandrake box. Running *UNRMread* on this file decreased the total amount of information the user would have to search through to 38 Megabytes (MB). *Lazarus* was also run on the *unrm* data file and it created a total of 95364 individual files that also totaled 2.5 GB of data. Because *lazarus* already sorts and classifies the information it produces, *UNRMread* is able to filter out the majority of files that cause garbage to appear in the files the user searches through. Running *UNRMread* on the *lazarus* files decreased the amount of files considered to contain valuable information to 2847 and the total amount of data the user would have to search through to 8.5 MB.

Again, *UNRMread* also creates its own directory for result storage using the operating system and script name arguments as identifiers. These features can be seen in the following Figure 20. The following discussion of *UNRMread* will not discuss the file format of its results because the format is the same as for *DELread*. The discussion will focus more on showing that whichever method *UNRMread* is run under the same results will be found. The DEL-keyword search file this time included a few more keywords based on the results found from running the previous scripts on the RedHat box. This file can be seen in Appendix D.

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# ./UNRMread /root/MAN/MAN_hacked_unrm/MAN_hacked_root_unrm.results DEL-keyword.txt MANDRAKE-unrmfile

[root@dhcp122115 root]# ./UNRMread /root/MAN/MAN_hacked_blocks/ DEL-keyword.txt MANDRAKE-BLOCK

[root@dhcp122115 root]# ls
anaconda-ks.cfg      install.log.syslog      MAN-img                RH
BOP_GRcompare_file_dump  jjj                    MANWWW                RH-DELETED
CDD                  key                    mysqlstuff             RH-DEL-ls.txt
ddimages.txt         linux-2.4.23           notes-McIver-UsingMySQLInt.pdf  RH-GRdata
Desktop              linux-2.4.23.tar       original               RH-img
ELF-TEST             lll                    PROJECT                test
exchanges            MAN                    PROJECT_PICS           tmp
FLEtest             manbloxx              PROJECT_PICS-BAK       upgrade.log
forensic.ppt         MANDDY                REDHAT2_DELread_file_dump  upgrade.log.syslog
hacked               MANDDY2               REDHAT3_DELread_file_dump  xtra
images.tar           MANDRAKE-BLOCK_UNRMread_file_dump  REDHAT_DELread_file_dump
imgtest             MANDRAKE-unrmfile_UNRMread_file_dump  REDHAT_GRcompare_file_dump
install.log          MAN-GRdata             REDHAT_MACread_file_dump

[root@dhcp122115 root]#

```

**Figure 20: Running UNRMread**

Figures 21 and 22 show the contents stored in the file dump directories for both execution methods of *UNRMread*.

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

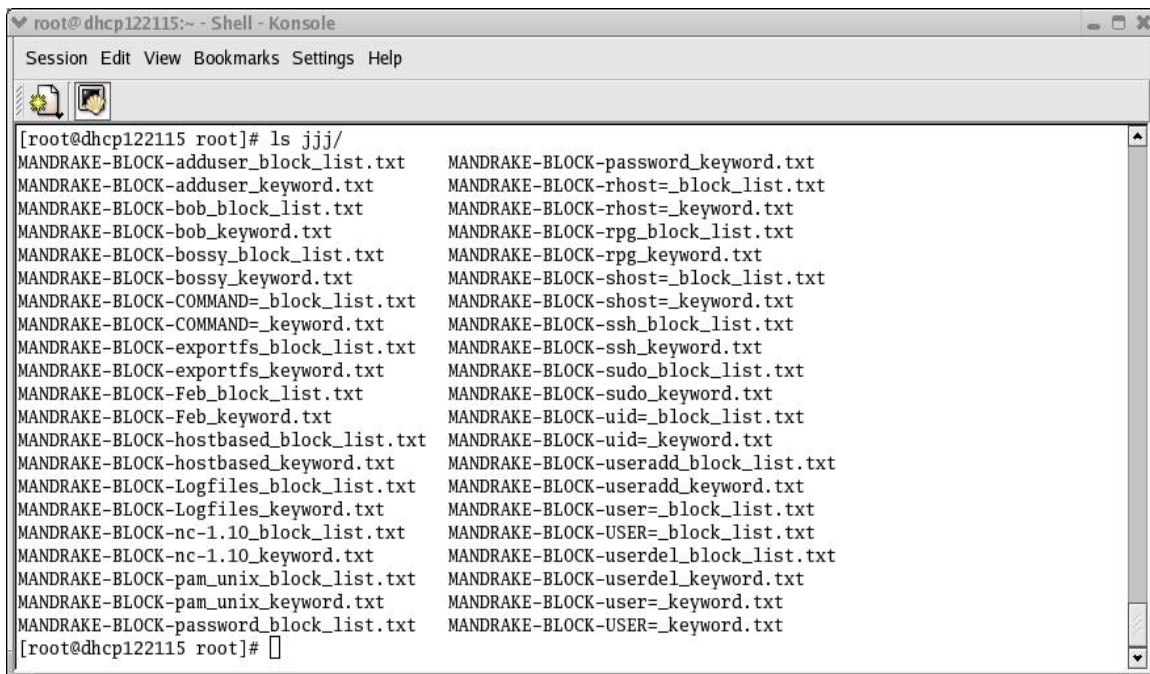
[root@dhcp122115 root]# ls MANDRAKE-unrmfile_UNRMread_file_dump/
MANDRAKE-unrmfile-adduser_keyword.txt  MANDRAKE-unrmfile-rhost=_keyword.txt
MANDRAKE-unrmfile-bob_keyword.txt      MANDRAKE-unrmfile-rpg_keyword.txt
MANDRAKE-unrmfile-bossy_keyword.txt    MANDRAKE-unrmfile-shost=_keyword.txt
MANDRAKE-unrmfile-COMMAND=_keyword.txt MANDRAKE-unrmfile-ssh_keyword.txt
MANDRAKE-unrmfile-exportfs_keyword.txt MANDRAKE-unrmfile-sudo_keyword.txt
MANDRAKE-unrmfile-Feb_keyword.txt       MANDRAKE-unrmfile-uid=_keyword.txt
MANDRAKE-unrmfile-hostbased_keyword.txt MANDRAKE-unrmfile-useradd_keyword.txt
MANDRAKE-unrmfile-Logfiles_keyword.txt  MANDRAKE-unrmfile-userdel_keyword.txt
MANDRAKE-unrmfile-nc-1.10_keyword.txt   MANDRAKE-unrmfile-user=_keyword.txt
MANDRAKE-unrmfile-pam_unix_keyword.txt  MANDRAKE-unrmfile-USER=_keyword.txt
MANDRAKE-unrmfile-password_keyword.txt

[root@dhcp122115 root]#

```

**Figure 21: Results from running UNRMread on single unrm file**

The only results from executing *UNRMread* on entire *unrm* data file are the keyword files created from being matched somewhere in the data file. No other information is needed since all the results come from the same file and being that the file will be gigabytes in size, copying the original file to another location serves no purpose.



```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# ls jjj/
MANDRAKE-BLOCK-adduser_block_list.txt  MANDRAKE-BLOCK-password_keyword.txt
MANDRAKE-BLOCK-adduser_keyword.txt      MANDRAKE-BLOCK-rhost=_block_list.txt
MANDRAKE-BLOCK-bob_block_list.txt       MANDRAKE-BLOCK-rhost=_keyword.txt
MANDRAKE-BLOCK-bob_keyword.txt          MANDRAKE-BLOCK-rpg_block_list.txt
MANDRAKE-BLOCK-bossy_block_list.txt      MANDRAKE-BLOCK-rpg_keyword.txt
MANDRAKE-BLOCK-bossy_keyword.txt         MANDRAKE-BLOCK-shost=_block_list.txt
MANDRAKE-BLOCK-COMMAND=_block_list.txt  MANDRAKE-BLOCK-shost=_keyword.txt
MANDRAKE-BLOCK-COMMAND=_keyword.txt     MANDRAKE-BLOCK-ssh_block_list.txt
MANDRAKE-BLOCK-exportfs_block_list.txt   MANDRAKE-BLOCK-ssh_keyword.txt
MANDRAKE-BLOCK-exportfs_keyword.txt      MANDRAKE-BLOCK-sudo_block_list.txt
MANDRAKE-BLOCK-Feb_block_list.txt        MANDRAKE-BLOCK-sudo_keyword.txt
MANDRAKE-BLOCK-Feb_keyword.txt           MANDRAKE-BLOCK-uid=_block_list.txt
MANDRAKE-BLOCK-hostbased_block_list.txt  MANDRAKE-BLOCK-uid=_keyword.txt
MANDRAKE-BLOCK-hostbased_keyword.txt     MANDRAKE-BLOCK-useradd_block_list.txt
MANDRAKE-BLOCK-Logfiles_block_list.txt   MANDRAKE-BLOCK-useradd_keyword.txt
MANDRAKE-BLOCK-Logfiles_keyword.txt      MANDRAKE-BLOCK-user=_block_list.txt
MANDRAKE-BLOCK-nc-1.10_block_list.txt    MANDRAKE-BLOCK-USER=_block_list.txt
MANDRAKE-BLOCK-nc-1.10_keyword.txt       MANDRAKE-BLOCK-userdel_block_list.txt
MANDRAKE-BLOCK-pam_unix_block_list.txt   MANDRAKE-BLOCK-userdel_keyword.txt
MANDRAKE-BLOCK-pam_unix_keyword.txt      MANDRAKE-BLOCK-user=_keyword.txt
MANDRAKE-BLOCK-password_block_list.txt  MANDRAKE-BLOCK-USER=_keyword.txt
[root@dhcp122115 root]#
```

**Figure 22: Results from running UNRMread on lazarus files**

Unlike running *UNRMread* on a single file as done above, if the option to run it on the results of *lazarus* is chosen the contents of the directory will look like that of the *DELread* file dump directory (Since many more files are copied over to the *UNRMread* file dump directory the figure above only contains the keyword and block list files created by *unrm*). Even though it may look like the results from the *lazarus* run contains much more information, the actual size of the keyword files for this run totals 8.5 MB as compared to the 38 MB created from keyword files from the single *unrm* file. This means that approximately 30 MB of garbage is contained in the keyword files from the single *unrm* file. Even so, the same information can be found all the examples below will show.

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

--
Feb  2 19:47:15 boss groupadd[9825]: new group: name=kop, gid=503
Feb  2 19:49:36 boss usermod[9860]: change user `bob' GID from `502' to `503'
Feb  3 13:29:02 boss usermod[15530]: change user `bob' GID from `503' to `0'
Feb  3 14:57:04 boss useradd[15891]: new group: name=gog, gid=503
--
Feb  3 14:57:55 boss userdel[15894]: remove group `gog'
Feb  3 14:58:26 boss userdel[15897]: delete user `bob'
Feb  3 14:58:49 boss groupdel[15900]: remove group `bob'
Feb  3 14:58:52 boss useradd[15901]: new group: name=bob, gid=502
Feb  3 14:58:52 boss useradd[15901]: new user: name=bob, uid=502, gid=502, home=
/home/bob, shell=/bin/bash
Feb  3 14:59:44 boss useradd[15946]: new group: name=gog, gid=503
--
Feb  3 18:46:51 boss userdel[16250]: remove group `gp'
Feb  3 18:46:54 boss userdel[16251]: delete user `bob'
Feb  3 18:46:54 boss userdel[16251]: remove group `bob'
Feb  7 07:57:42 boss webmin[1628]: Webmin starting

```

Figure 23: example result from keyword ‘bob’ on single file

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb  2 19:37:46 boss su(pam_unix)[9669]: session closed for user bob
Feb  2 19:42:21 boss su(pam_unix)[9732]: session opened for user bob by (uid=0)
Feb  2 19:43:03 boss su(pam_unix)[9732]: session closed for user bob
Feb  2 19:43:21 boss su(pam_unix)[9779]: session opened for user bob by (uid=0)
Feb  2 19:43:47 boss su(pam_unix)[9779]: session closed for user bob
Feb  2 19:47:15 boss groupadd[9825]: new group: name=kop, gid=503
Feb  2 19:49:36 boss usermod[9860]: change user `bob' GID from `502' to `503'
Feb  2 19:50:07 boss su(pam_unix)[9863]: session opened for user bob by (uid=0)
Feb  2 19:50:28 boss su(pam_unix)[9863]: session closed for user bob
Feb  2 19:51:22 boss su(pam_unix)[9910]: session opened for user bob by (uid=0)
Feb  2 19:51:54 boss su(pam_unix)[9910]: session closed for user bob
Feb  2 19:52:12 boss su(pam_unix)[9956]: session opened for user bob by (uid=0)
Feb  2 19:52:26 boss su(pam_unix)[9956]: session closed for user bob
Feb  2 19:53:16 boss su(pam_unix)[10000]: session opened for user bob by (uid=0)
Feb  2 19:53:24 boss su(pam_unix)[10042]: session opened for user root by (uid=502)
Feb  2 19:53:28 boss su(pam_unix)[10042]: session closed for user root
Feb  2 19:53:30 boss su(pam_unix)[10000]: session closed for user bob
Feb  3 05:01:02 boss msec: changed mode of /var/log/security/sgid.today from 644 to 640
--
Feb  3 13:25:11 boss su(pam_unix)[15314]: session closed for user bin
--More--(1%)

```

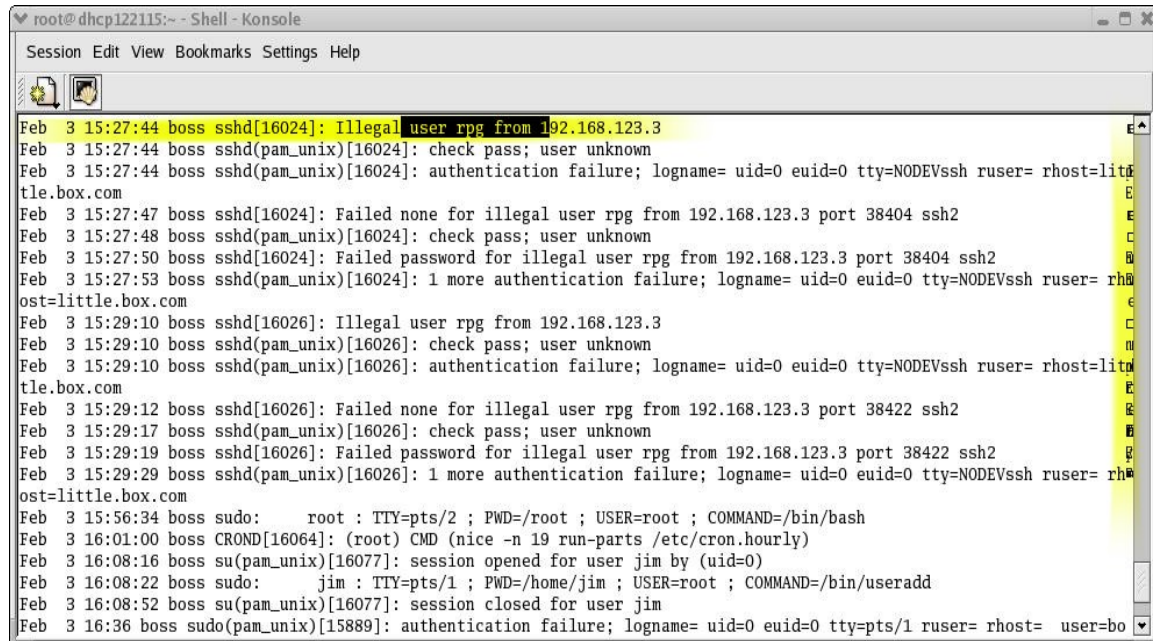
Figure 24: example result from keyword ‘bob’ on lazarus files

The above figures came from the keyword *bob*. Figure 23 shows new groups named *kop* and *gog* being created and also shows that *bob*’s group identification (GID) was temporarily changed to that of the root group before the user was removed.

Figure 24 taken from the *lazarus* file run also reports the new group *kop* being created. As for the other group *gog* and the changing of *bob*’s GID, these results can be found in Appendix D in Figures 67 and 68. This information was

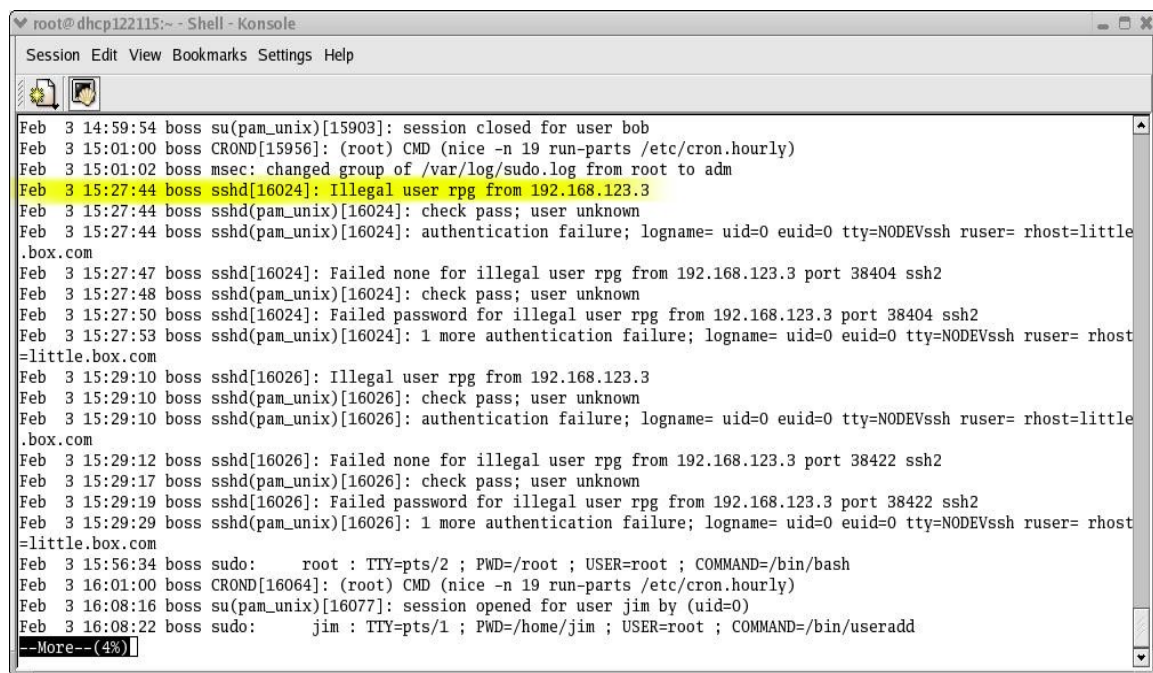


actually stored in different files and was not actually stored next to each other as it appears in the figure given for the single file results.



```
Feb 3 15:27:44 boss sshd[16024]: Illegal user rpg from 192.168.123.3
Feb 3 15:27:44 boss sshd(pam_unix)[16024]: check pass; user unknown
Feb 3 15:27:44 boss sshd(pam_unix)[16024]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:27:47 boss sshd[16024]: Failed none for illegal user rpg from 192.168.123.3 port 38404 ssh2
Feb 3 15:27:48 boss sshd(pam_unix)[16024]: check pass; user unknown
Feb 3 15:27:50 boss sshd[16024]: Failed password for illegal user rpg from 192.168.123.3 port 38404 ssh2
Feb 3 15:27:53 boss sshd(pam_unix)[16024]: 1 more authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:29:10 boss sshd[16026]: Illegal user rpg from 192.168.123.3
Feb 3 15:29:10 boss sshd(pam_unix)[16026]: check pass; user unknown
Feb 3 15:29:10 boss sshd(pam_unix)[16026]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:29:12 boss sshd[16026]: Failed none for illegal user rpg from 192.168.123.3 port 38422 ssh2
Feb 3 15:29:17 boss sshd(pam_unix)[16026]: check pass; user unknown
Feb 3 15:29:19 boss sshd[16026]: Failed password for illegal user rpg from 192.168.123.3 port 38422 ssh2
Feb 3 15:29:29 boss sshd(pam_unix)[16026]: 1 more authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:56:34 boss sudo: root : TTY=pts/2 ; PWD=/root ; USER=root ; COMMAND=/bin/bash
Feb 3 16:01:00 boss CROND[16064]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
Feb 3 16:08:16 boss su(pam_unix)[16077]: session opened for user jim by (uid=0)
Feb 3 16:08:22 boss sudo: jim : TTY=pts/1 ; PWD=/home/jim ; USER=root ; COMMAND=/bin/useradd
Feb 3 16:08:52 boss su(pam_unix)[16077]: session closed for user jim
Feb 3 16:36 boss sudo(pam_unix)[15889]: authentication failure; logname= uid=0 euid=0 tty=pts/1 ruser= rhost= user=bo
```

Figure 25: Example result from keyword 'Feb' on single file



```
Feb 3 14:59:54 boss su(pam_unix)[15903]: session closed for user bob
Feb 3 15:01:00 boss CROND[15956]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
Feb 3 15:01:02 boss msec: changed group of /var/log/sudo.log from root to adm
Feb 3 15:27:44 boss sshd[16024]: Illegal user rpg from 192.168.123.3
Feb 3 15:27:44 boss sshd(pam_unix)[16024]: check pass; user unknown
Feb 3 15:27:44 boss sshd(pam_unix)[16024]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:27:47 boss sshd[16024]: Failed none for illegal user rpg from 192.168.123.3 port 38404 ssh2
Feb 3 15:27:48 boss sshd(pam_unix)[16024]: check pass; user unknown
Feb 3 15:27:50 boss sshd[16024]: Failed password for illegal user rpg from 192.168.123.3 port 38404 ssh2
Feb 3 15:27:53 boss sshd(pam_unix)[16024]: 1 more authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:29:10 boss sshd[16026]: Illegal user rpg from 192.168.123.3
Feb 3 15:29:10 boss sshd(pam_unix)[16026]: check pass; user unknown
Feb 3 15:29:10 boss sshd(pam_unix)[16026]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:29:12 boss sshd[16026]: Failed none for illegal user rpg from 192.168.123.3 port 38422 ssh2
Feb 3 15:29:17 boss sshd(pam_unix)[16026]: check pass; user unknown
Feb 3 15:29:19 boss sshd[16026]: Failed password for illegal user rpg from 192.168.123.3 port 38422 ssh2
Feb 3 15:29:29 boss sshd(pam_unix)[16026]: 1 more authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com
Feb 3 15:56:34 boss sudo: root : TTY=pts/2 ; PWD=/root ; USER=root ; COMMAND=/bin/bash
Feb 3 16:01:00 boss CROND[16064]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
Feb 3 16:08:16 boss su(pam_unix)[16077]: session opened for user jim by (uid=0)
Feb 3 16:08:22 boss sudo: jim : TTY=pts/1 ; PWD=/home/jim ; USER=root ; COMMAND=/bin/useradd
--More--(4%)
```

Figure 26: Example result from keyword 'Feb' on lazarus files

Some of the garbage in Figure 25 can be seen on the right border of the terminal window. However, this figure shows attempts to log into SSH on the Mandrake box from user *rpg* from 'little.box.com', which happens to be the

RedHat box. This same information is found in Figure 26 from the lazarus run and there is no garbage for the user to read through.

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb  4 17:35:43 boss sshd[21929]: Accepted password for root from 192.168.123.3 port 57419 ssh2
--
Feb  4 17:49:28 boss rpc.mountd: authenticated mount request from little.box.com:693 for / (/)
Feb  4 17:51:53 boss sshd(pam_unix)[22127]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruse
r= rhost=little.box.com user=root
Feb  4 17:51:58 boss sshd[22127]: Accepted password for root from 192.168.123.3 port 32854 ssh2
--
Feb  4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb  4 17:53:29 boss sshd(pam_unix)[22171]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruse
r= rhost=little.box.com user=root
Feb  4 17:53:34 boss sshd[22171]: Accepted password for root from 192.168.123.3 port 32874 ssh2
--
Feb  4 17:53:54 boss sshd(pam_unix)[22171]: session closed for user root
Feb  4 17:54:24 boss sshd(pam_unix)[22215]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruse
r= rhost=little.box.com user=root
Feb  4 17:54:58 boss sshd[22215]: Accepted password for root from 192.168.123.3 port 32886 ssh2
--
[ --with-gnats-service=SERVICE-NAME ]
[ --with-gnats-user=USERNAME ]
[ --with-gnats-port=PORT-NUMBER ]
--

```

**Figure 27: Example result from keyword ‘user=’ on single file**

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb  4 17:35:43 boss sshd[21929]: Accepted password for root from 192.168.123.3 port 57419 ssh2
--
Feb  4 17:49:28 boss rpc.mountd: authenticated mount request from little.box.com:693 for / (/)
Feb  4 17:51:53 boss sshd(pam_unix)[22127]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb  4 17:51:58 boss sshd[22127]: Accepted password for root from 192.168.123.3 port 32854 ssh2
--
Feb  4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb  4 17:53:29 boss sshd(pam_unix)[22171]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb  4 17:53:34 boss sshd[22171]: Accepted password for root from 192.168.123.3 port 32874 ssh2
--
Feb  4 17:53:54 boss sshd(pam_unix)[22171]: session closed for user root
Feb  4 17:54:24 boss sshd(pam_unix)[22215]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb  4 17:54:58 boss sshd[22215]: Accepted password for root from 192.168.123.3 port 32886 ssh2
-----entries found in file /root/MAN/MAN_hacked_blocks//110389.s.txt-----
(if (not (string= "" sql-user))
  (setq params (append (list (concat "--user=" sql-user)) params)))
(if (not (null sql-mysql-options))
  (if (not (null sql-mysql-options))
    [ --with-gnats-service=SERVICE-NAME ]
    [ --with-gnats-user=USERNAME ]
    [ --with-gnats-port=PORT-NUMBER ]
  ))
)
-----entries found in file /root/MAN/MAN_hacked_blocks//1190575.t.txt-----
[ --with-gnats-service=SERVICE-NAME ]
[ --with-gnats-user=USERNAME ]
[ --with-gnats-port=PORT-NUMBER ]
--
'--with-gnats-user=USERNAME'

```

**Figure 28: Example result from keyword ‘user=’ on lazarus files**

These figures not only report a successful nfs mount request from little.box.com to mount the root ‘/’ partition of the Mandrake box, they go further to show the netcat rpm ‘nc-1.10-18’ being installed. Also, notice in Figure 27 all the information displayed appears to be clumped together. However, Figure 28 actually shows the sections starting with “(if (not.... ))” and “[ --with....]” are found in different files.

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more MANDRAKE-unrmfile_UNRMread_file_dump/MANDRAKE-unrmfile-nc-1.10_keyword.txt
-----entries found in file /root/MAN/MAN_hacked_unrm/MAN_hacked_root_unrm.results-----
myspell-hyph-en-1.0.2-0.20020727.1mdk.noarch.rpm
nc-1.10-18.i386.rpm
ncurses-5.3-1.20030215.3mdk.i586.rpm
---
myspell-hyph-en-1.0.2-0.20020727.1mdk.noarch.rpm
nc-1.10-18.i386.rpm
ncurses-5.3-1.20030215.3mdk.i586.rpm
---
Feb  4 17:49:28 boss rpc.mountd: authenticated mount request from little.box.com:693 for / (/)
Feb  4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb  4 17:56:33 boss sshd: stop succeeded
---
Feb  4 17:51:58 boss sshd(pam_unix)[22127]: session opened for user root by (uid=0)
Feb  4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb  4 17:53:29 boss sshd(pam_unix)[22171]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruse
r= rhost=little.box.com user=root
---
  □Z□□BB□□□  U.\□!□.$ {p0□□.wcN^□□□Q`#^LLK~□#yX
                                DU`~□mNkJ□"O;□□Gm
--More--(10%)
```

Figure 29: Example result from keyword ‘nc-1.10’ on single file

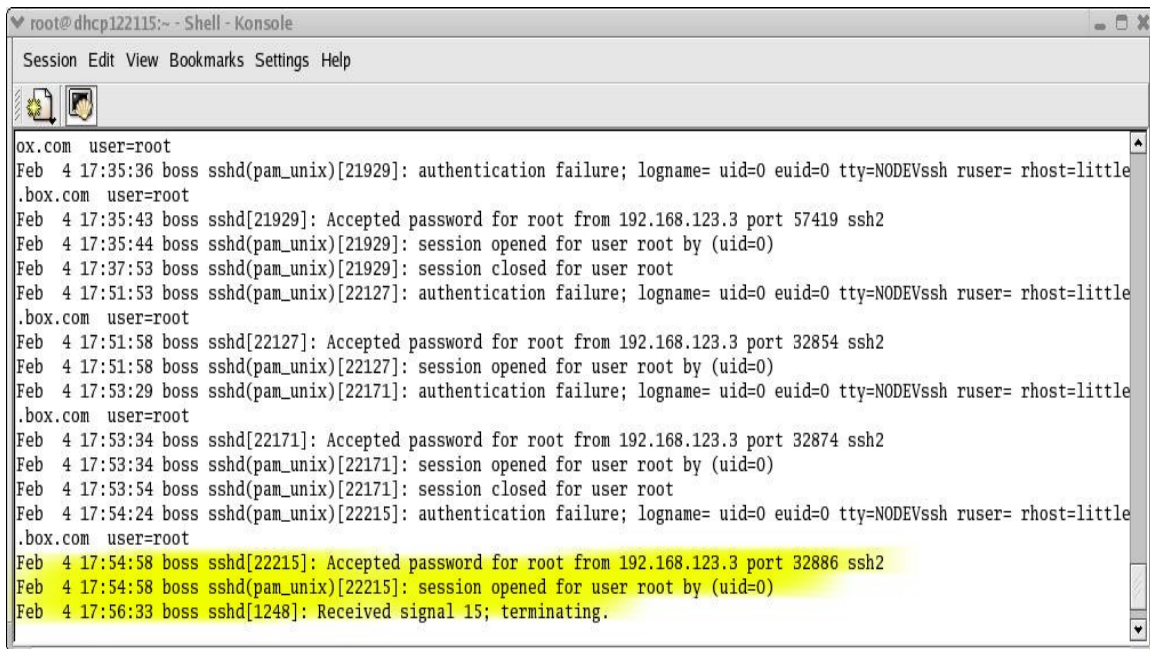
```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more MANDRAKE-BLOCK_UNRMread_file_dump/MANDRAKE-BLOCK-nc-1.10_keyword.txt
-----entries found in file /root/MAN/MAN_hacked_blocks//1093339.1.txt-----
myspell-hyph-en-1.0.2-0.20020727.1mdk.noarch.rpm
nc-1.10-18.i386.rpm
ncurses-5.3-1.20030215.3mdk.i586.rpm
-----entries found in file /root/MAN/MAN_hacked_blocks//1093339.1.txt-----
Feb  4 17:49:28 boss rpc.mountd: authenticated mount request from little.box.com:693 for / (/)
Feb  4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb  4 17:56:33 boss sshd: stop succeeded
---
Feb  4 17:51:58 boss sshd(pam_unix)[22127]: session opened for user root by (uid=0)
Feb  4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb  4 17:53:29 boss sshd(pam_unix)[22171]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser=
.box.com user=root
--More--(8%)
```

Figure 30: Example result from keyword ‘nc-1.10’ on lazarus files

Just like previous two figures, Figures 29 and 30 show the successful nfs mount request from little.box.com to mount the root ‘/’ partition of the Mandrake box and the successful installation of netcat on Mandrake. However, these two figures also show the point at which the user killed all SSH sessions feeling the network had been compromised. The figure below actually displays the infiltrator’s and user’s interaction with ssh.





```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

ox.com user=root
Feb 4 17:35:36 boss sshd(pam_unix)[21929]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb 4 17:35:43 boss sshd[21929]: Accepted password for root from 192.168.123.3 port 57419 ssh2
Feb 4 17:35:44 boss sshd(pam_unix)[21929]: session opened for user root by (uid=0)
Feb 4 17:37:53 boss sshd(pam_unix)[21929]: session closed for user root
Feb 4 17:51:53 boss sshd(pam_unix)[22127]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb 4 17:51:58 boss sshd[22127]: Accepted password for root from 192.168.123.3 port 32854 ssh2
Feb 4 17:51:58 boss sshd(pam_unix)[22127]: session opened for user root by (uid=0)
Feb 4 17:53:29 boss sshd(pam_unix)[22171]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb 4 17:53:34 boss sshd[22171]: Accepted password for root from 192.168.123.3 port 32874 ssh2
Feb 4 17:53:34 boss sshd(pam_unix)[22171]: session opened for user root by (uid=0)
Feb 4 17:53:54 boss sshd(pam_unix)[22171]: session closed for user root
Feb 4 17:54:24 boss sshd(pam_unix)[22215]: authentication failure; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little
.box.com user=root
Feb 4 17:54:58 boss sshd[22215]: Accepted password for root from 192.168.123.3 port 32886 ssh2
Feb 4 17:54:58 boss sshd(pam_unix)[22215]: session opened for user root by (uid=0)
Feb 4 17:56:33 boss sshd[1248]: Received signal 15; terminating.
```

**Figure 31: User kills ssh after being compromised**

The highlighted section in the above figure shows the user logging into SSH. Upon logging into ssh the message informing the user of the last time SSH was log into with the root account was display, which happened to be approximately one minute before the user actually logged in. At this point the user kills the ssh daemon and begins the forensics analysis.

## 2.5 Simple-MySQL

The *Simple-MySQL* script is an interactive menu that allows the users automatically to connect to their MySQL database and create a database to store the evidence of the compromised network in an organized manner. The keyword files, block files and diff files created by the previous scripts are very specific and due to the naming structure the information contained in each file is self-evident. However, if there is no method by which the user can quickly look up information pertaining to a particular part of the compromise (i.e. which users were added?) the user will continually have to open and search through each file to find the information. Also, some of the evidence may not be found because it did not match the exact keyword being searched (i.e. opencat was found in the mactime database in the file REDHAT-created\_MAC\_keyword.txt file where the search keyword was ‘.c’ see APPENDIX C Figure 60).

Recognizing this, the decision was made to place the evidence into a database. A database allows for fast retrieval and allows the user to store evidence such as illegally added new users. Thereby saving the time it would take to search through all the files the next time the user can’t remember where exactly that information is available.



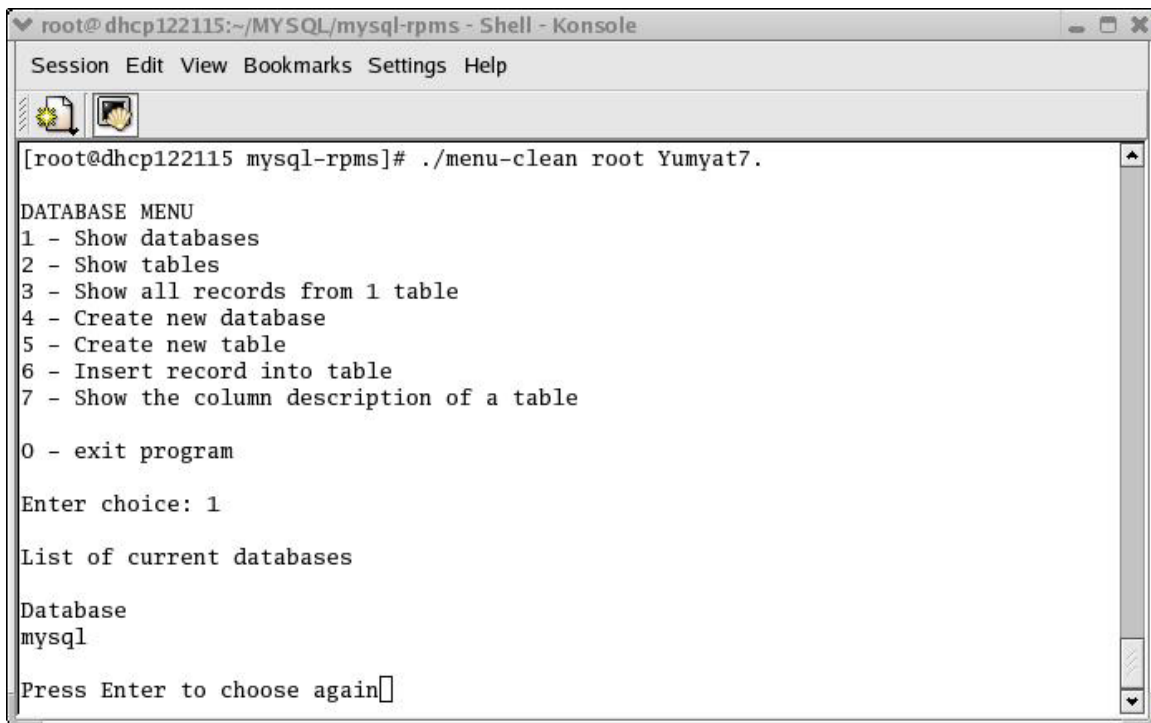
It provides the user with the options to show databases, records of a table, create a database, create four prespecified tables, to insert records into the four tables via a text file (form) and show the column descriptions for each table. It requires as arguments, the name and password of the user who is allowed to access the MySQL databases. The options to delete/update records, tables and databases were intentionally excluded. With this sort of interface a user could accidentally delete valued information, therefore it was decided that the user should perform these actions via direct interaction with the database.

These actions can be easily performed with another tool made available by MySQL AB known as 'MySQL Control Center' (MySQLcc). *MySQLcc* is a Graphical User Interface (GUI) that allows the user to perform all the administrative actions on a MySQL database. So, why go through the trouble of writing a script that only performs some predefined actions and only creates predefined tables? For one, many, many people have trouble getting the precompiled version of *MySQLcc* to work. I spent over three hours alone trying to get it to run without it continuously given segmentation faults and closing whenever I entered the password to login. Also, the source version takes a while to compile and all its dependencies need to be installed first. Because the MySQL database was installed on an external hard drive some of the dependencies (i.e. Qt) would not compile correctly, which did not allow for the complete installation of the source *MySQLcc*. Finally, the time constraint only allowed for so much time to be spent on these issues.

Since command line is always the way to go in these instances a script that saves the user much time in database setup seemed natural. Eventually, I did get *MySQLcc* running therefore the explanation and walkthrough of *Simple-MySQL* will only deliver screen captures from the script, while screen captures from *MySQLcc* will be offered in Appendix E as a visual reinforcement for the script.

### **2.5.1 Running Simple-MySQL**

*Simple-MySQL* is run from the command line and requires the username and password of the user allowed to connect to the database. The password entered on the command line is the password created specifically for that user to access MySQL. It is not the password that user uses to log onto the computer!! Once executed the initial menu is displayed as shown in Figure 32.



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 mysql-rpms]# ./menu-clean root Yumyat7.

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table

0 - exit program

Enter choice: 1

List of current databases

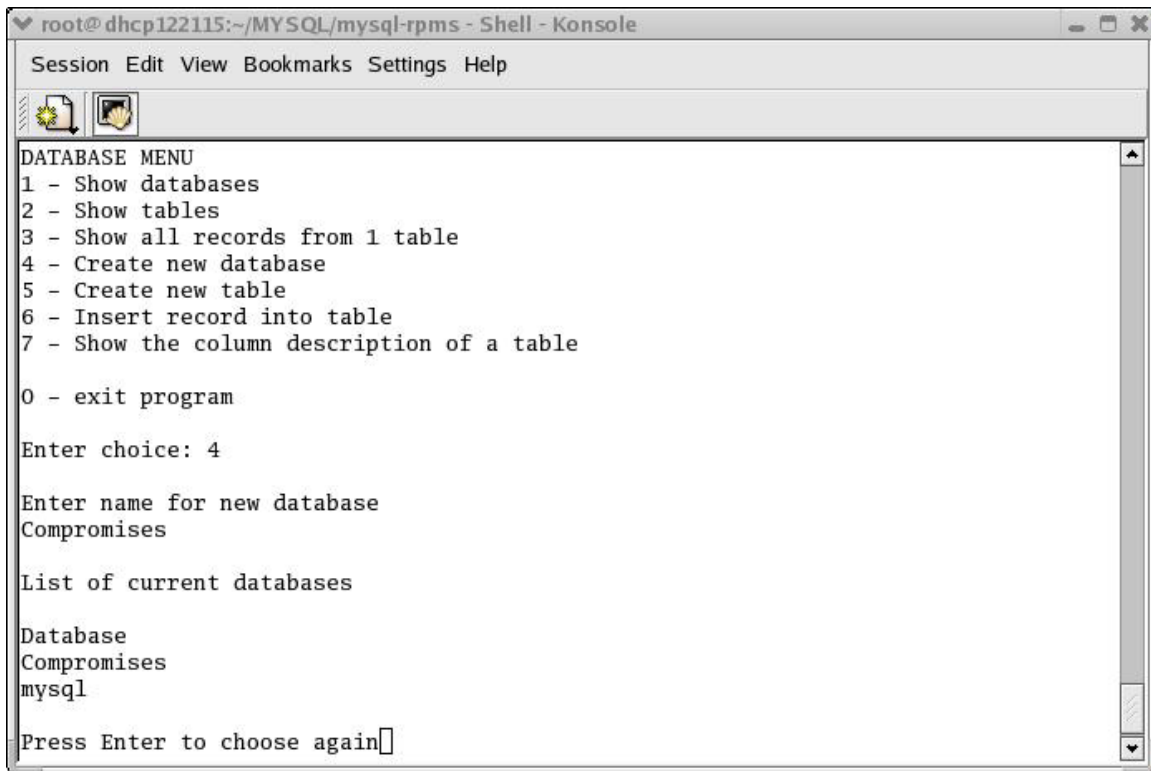
Database
mysql

Press Enter to choose again
```

**Figure 32: Running Simple-MySQL**

Along with the 'DATABASE MENU' the above figure shows the first option of the menu being executed. The option to show all current databases is chosen and it shows a database named 'mysql' already exists. This database was actually created during the installation of the MySQL database. With this option completed the user presses the enter key and is allowed to chose another option.

## 2.5.2 Creating a Database



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table

0 - exit program

Enter choice: 4

Enter name for new database
Compromises

List of current databases

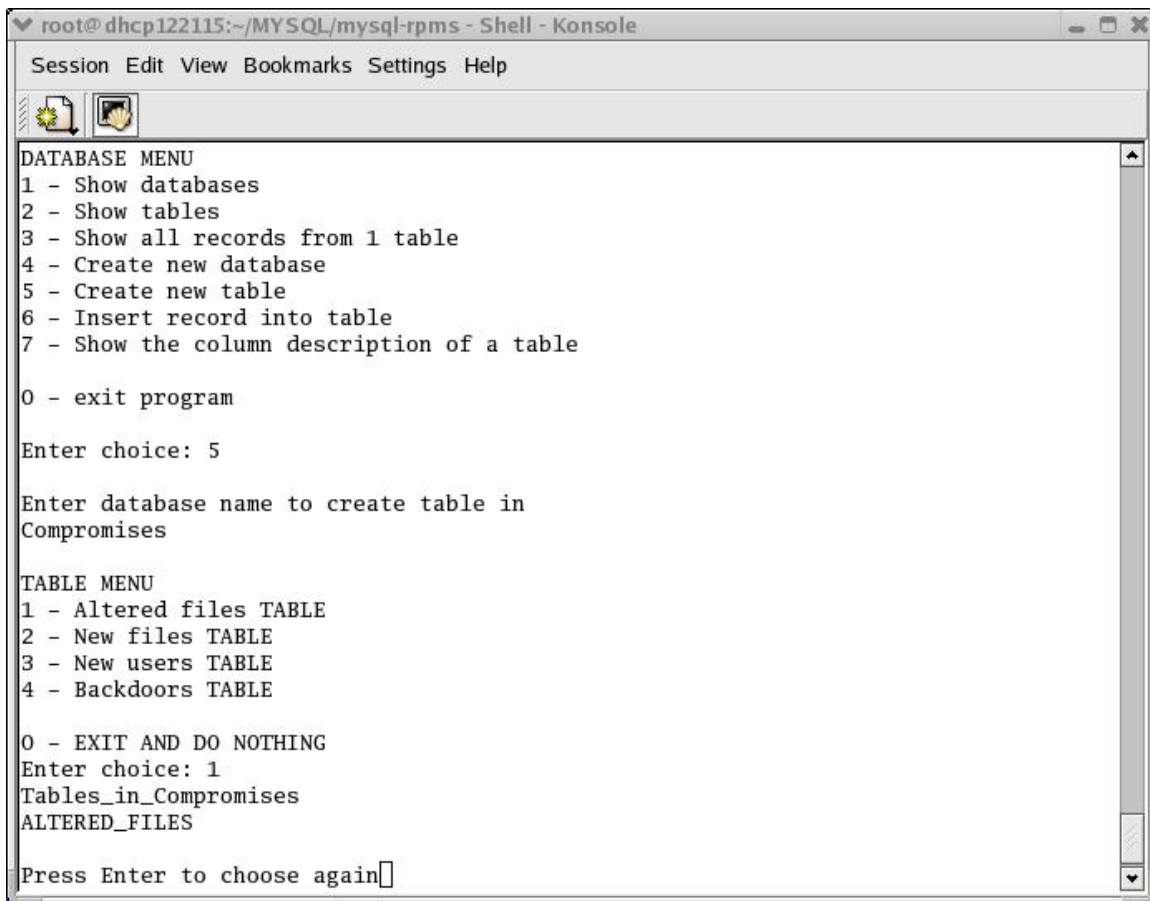
Database
Compromises
mysql

Press Enter to choose again
```

**Figure 33: Creating a database**

To create a database the user simply selects option four and presses enter. A prompt pops up asking the user to enter the name for the new database. After entering 'Compromises' as the database's name and pressing enter the updated list of current databases is displayed and the user presses enter to continue on.

### 2.5.3 Creating the Database Tables



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table

0 - exit program

Enter choice: 5

Enter database name to create table in
Compromises

TABLE MENU
1 - Altered files TABLE
2 - New files TABLE
3 - New users TABLE
4 - Backdoors TABLE

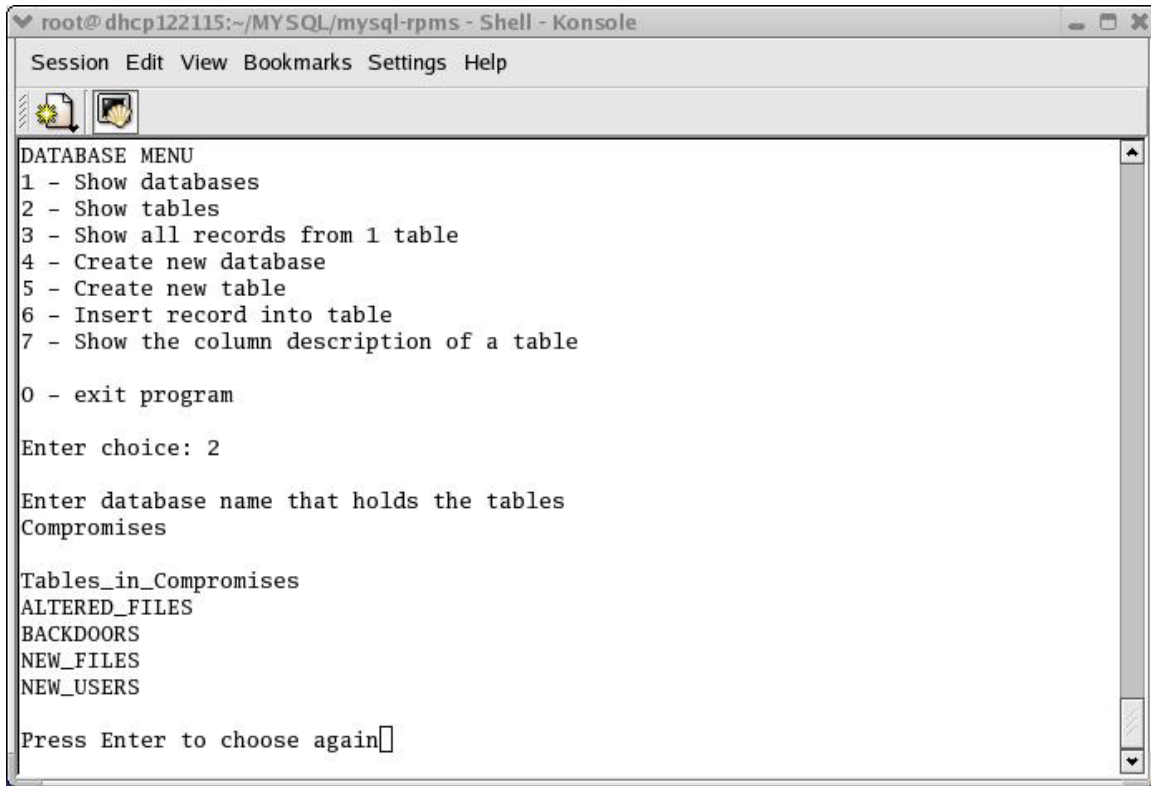
0 - EXIT AND DO NOTHING
Enter choice: 1
Tables_in_Compromises
ALTERED_FILES

Press Enter to choose again
```

**Figure 34: Creating the Tables**

Option five at the database menu is chosen to create a new table. The user is prompted for the database in which the table will be created. This simple feature is set just in case the user decides to use multiple databases to store evidence. After the database name is entered the user presses enter and the 'TABLE MENU' is displayed. The table menu list the four predefined tables created as minimum storage options. The user can modify the script to add more table structures and will simply have to add those options to the table menu.

## 2.5.4 Showing Tables



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table

0 - exit program

Enter choice: 2

Enter database name that holds the tables
Compromises

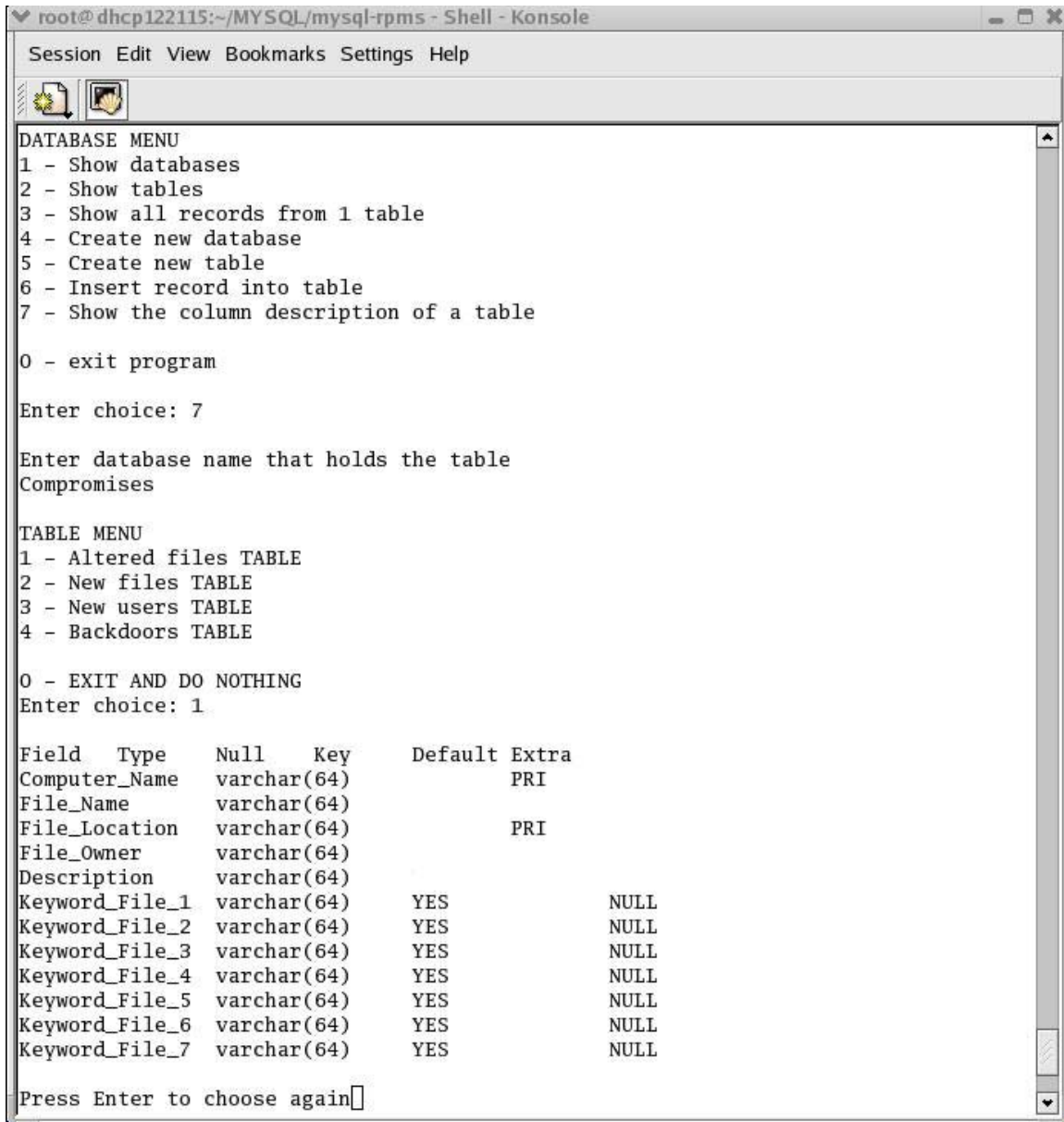
Tables_in_Compromises
ALTERED_FILES
BACKDOORS
NEW_FILES
NEW_USERS

Press Enter to choose again
```

**Figure 35: Showing tables**

After the user has created all the tables deemed necessary, by selecting the second option as shown in Figure 35 the user can see at any time what tables are in what database. Notice here the user is also prompted for the name of the database to display the tables for.

## 2.5.5 Showing Columns



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table

0 - exit program

Enter choice: 7

Enter database name that holds the table
Compromises

TABLE MENU
1 - Altered files TABLE
2 - New files TABLE
3 - New users TABLE
4 - Backdoors TABLE

0 - EXIT AND DO NOTHING
Enter choice: 1

Field Type Null Key Default Extra
Computer_Name varchar(64) PRI
File_Name varchar(64)
File_Location varchar(64) PRI
File_Owner varchar(64)
Description varchar(64)
Keyword_File_1 varchar(64) YES NULL
Keyword_File_2 varchar(64) YES NULL
Keyword_File_3 varchar(64) YES NULL
Keyword_File_4 varchar(64) YES NULL
Keyword_File_5 varchar(64) YES NULL
Keyword_File_6 varchar(64) YES NULL
Keyword_File_7 varchar(64) YES NULL

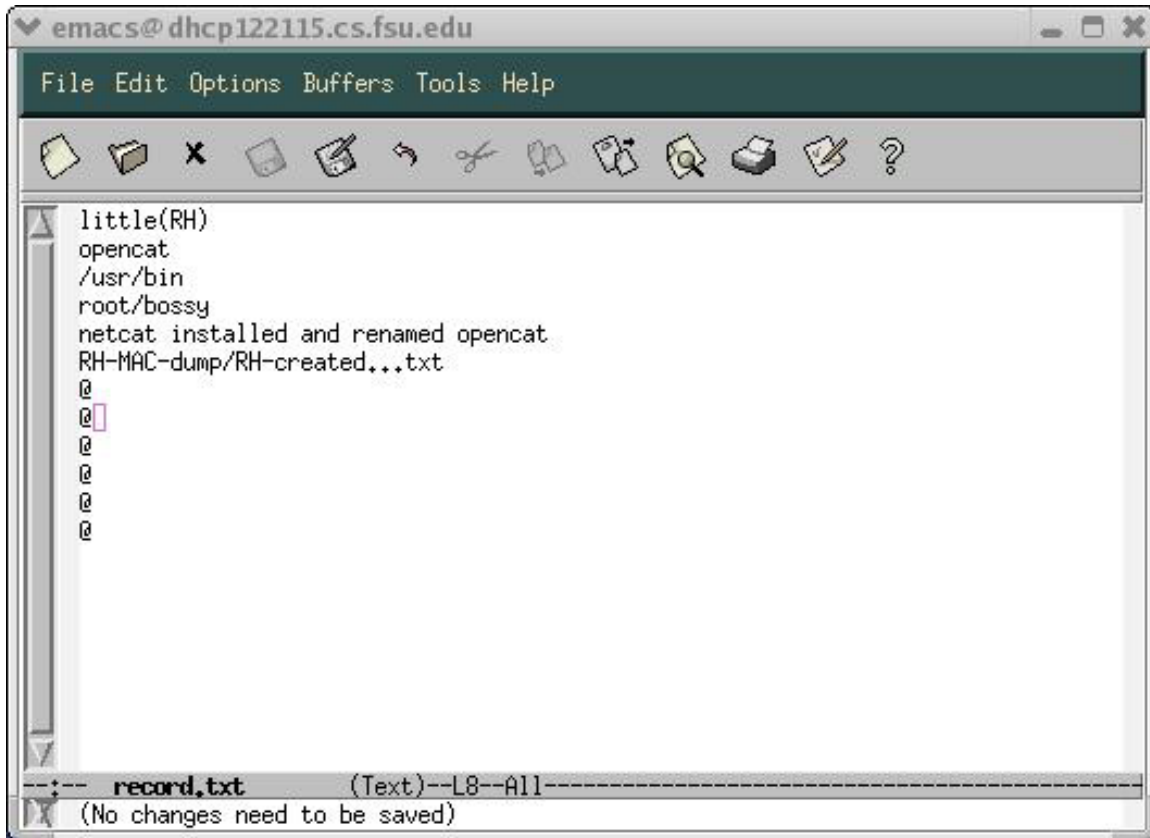
Press Enter to choose again
```

**Figure 36: Column descriptions of a table**

By selecting option seven, the user can obtain a column description of any table that has been created. This feature is more of a precautionary measure. Just in case the user doesn't look at script to see how the tables are constructed and doesn't look at the example record file as displayed Figure 36 in the following section, the user will at least know what kind of information can be entered into each table.

## 2.5.6 Inserting a Record

Entering records into any of the tables requires the user to make a form. This form can take the form of a simple text file with field contents being placed on different lines.



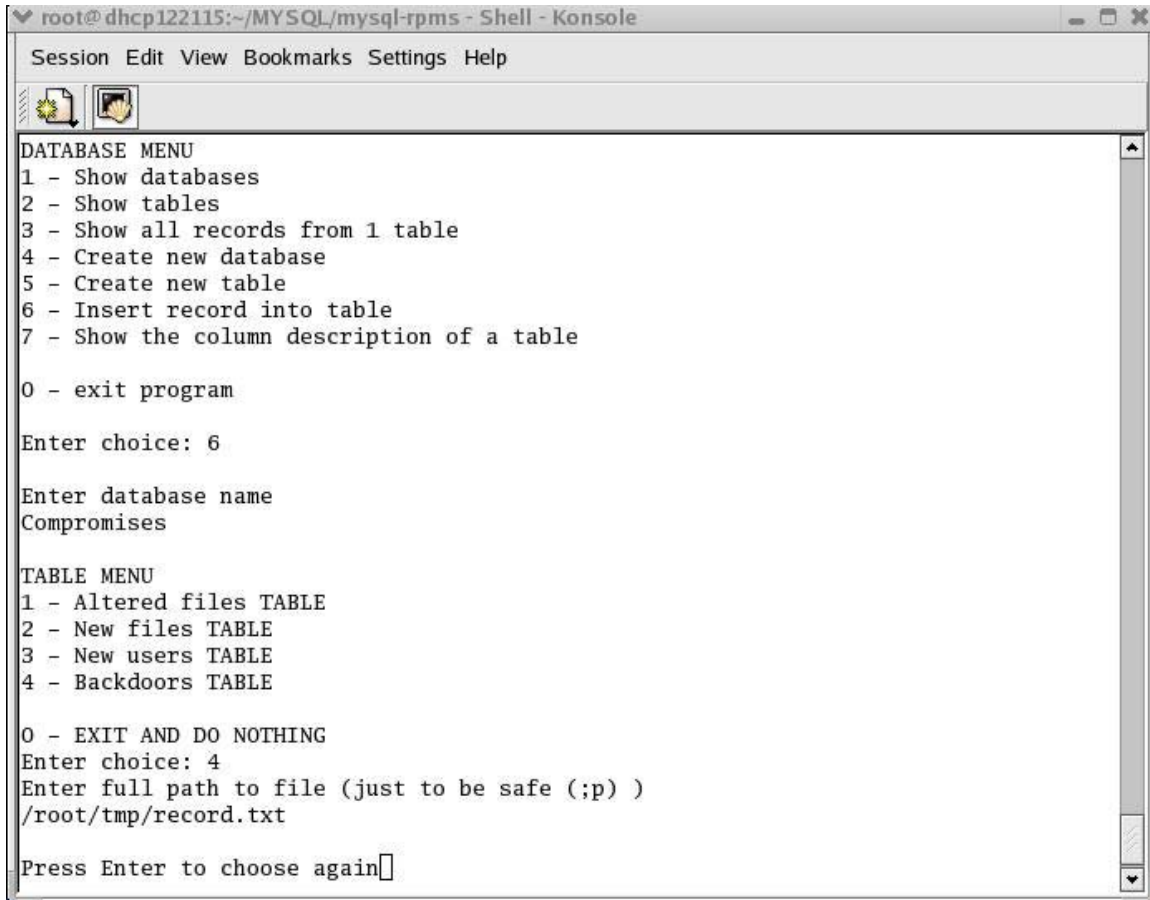
**Figure 37: Example record file (form)**

What makes the use of a file created in a text editor such as 'emacs' easy for the user to utilize is its speediness and simplicity. Users can enter in field contents line by line just as fast as they would in a GUI, but without the hassle of dealing with the error pop-ups from entering the wrong data format. If the wrong data format is entered into the form then the contents won't be inserted into the table. The user simply goes back to the form, which can be left open through the duration of the *Simple-MySQL* session, makes the changes, resaves the file and steps through the process to insert a record again.

The file can also be reused, thereby saving the user from have to create a new file. This also allows the user to use the same file path and do a simple copy and paste when ask for the record's location.

The '@' character is used as a null character and placeholder for the form. It can only be used for the last seven lines of file as these lines correlate with the 'Keyword\_File\_#' fields of each table. The keyword file fields are the only fields allowed to contain 'NULL' values and the '@' character must be present for the

lines where the user does not want information entered into the keyword file fields.



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table

0 - exit program

Enter choice: 6

Enter database name
Compromises

TABLE MENU
1 - Altered files TABLE
2 - New files TABLE
3 - New users TABLE
4 - Backdoors TABLE

0 - EXIT AND DO NOTHING
Enter choice: 4
Enter full path to file (just to be safe (;p) )
/root/tmp/record.txt

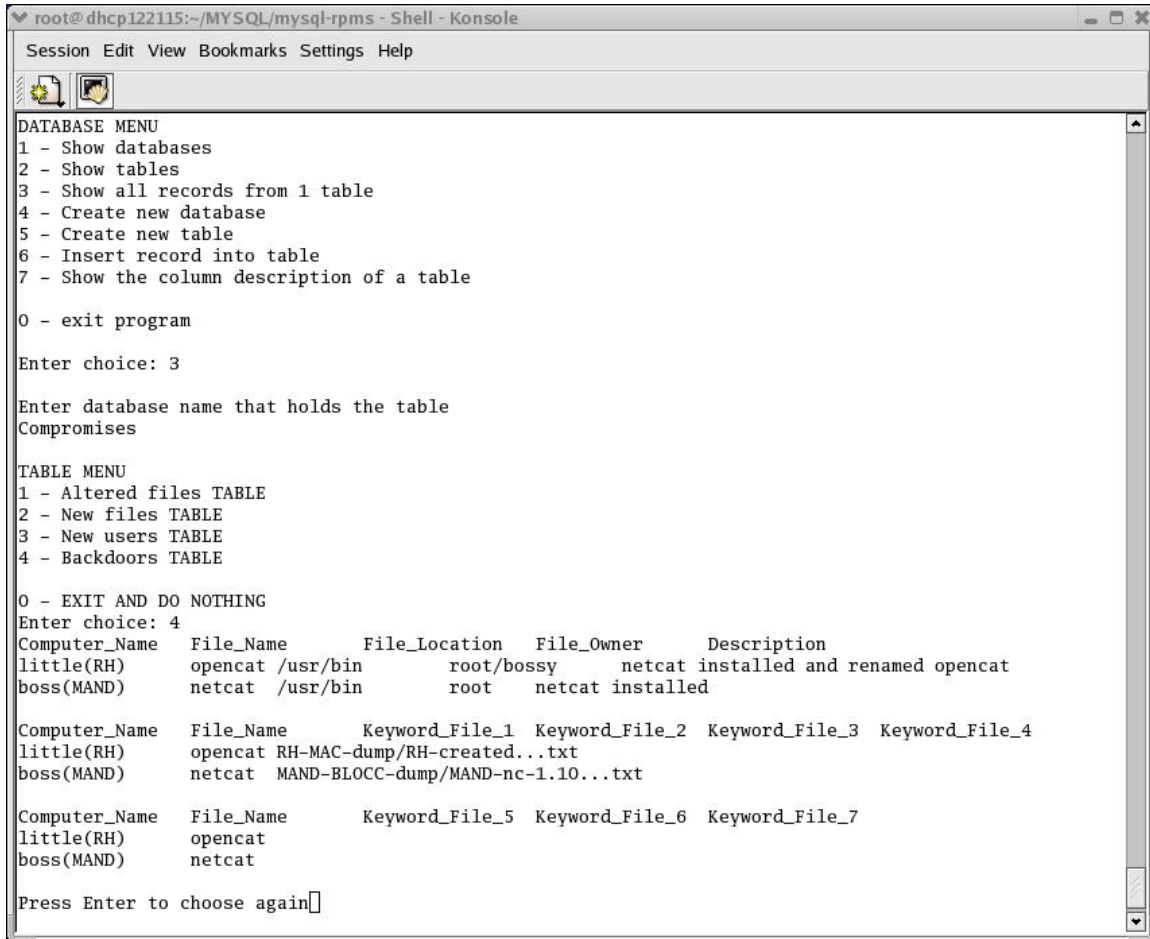
Press Enter to choose again
```

**Figure 38: Inserting a record**

Option six allows the user to enter a record into a table. After entering the name of the database and choosing a table from the menu, the user is prompted to enter the full path of the file that contains the information to be entered into the table. The file can be named whatever the user wants to name it and if the option to reuse the same file is taken then the user can copy the path and just paste it every time the choose to insert a record. If the 'Press Enter to choose again' line appears then the record was entered successfully.



## 2.5.7 Showing Records



```
root@dhcp122115:~/MYSQL/mysql-rpms - Shell - Konsole
Session Edit View Bookmarks Settings Help

DATABASE MENU
1 - Show databases
2 - Show tables
3 - Show all records from 1 table
4 - Create new database
5 - Create new table
6 - Insert record into table
7 - Show the column description of a table
0 - exit program

Enter choice: 3

Enter database name that holds the table
Compromises

TABLE MENU
1 - Altered files TABLE
2 - New files TABLE
3 - New users TABLE
4 - Backdoors TABLE
0 - EXIT AND DO NOTHING
Enter choice: 4

Computer_Name  File_Name      File_Location  File_Owner    Description
little(RH)     opencat /usr/bin      root/bossy     netcat installed and renamed opencat
boss(MAND)     netcat  /usr/bin      root           netcat installed

Computer_Name  File_Name      Keyword_File_1 Keyword_File_2  Keyword_File_3  Keyword_File_4
little(RH)     opencat RH-MAC-dump/RH-created...txt
boss(MAND)     netcat  MAND-BLOCC-dump/MAND-nc-1.10...txt

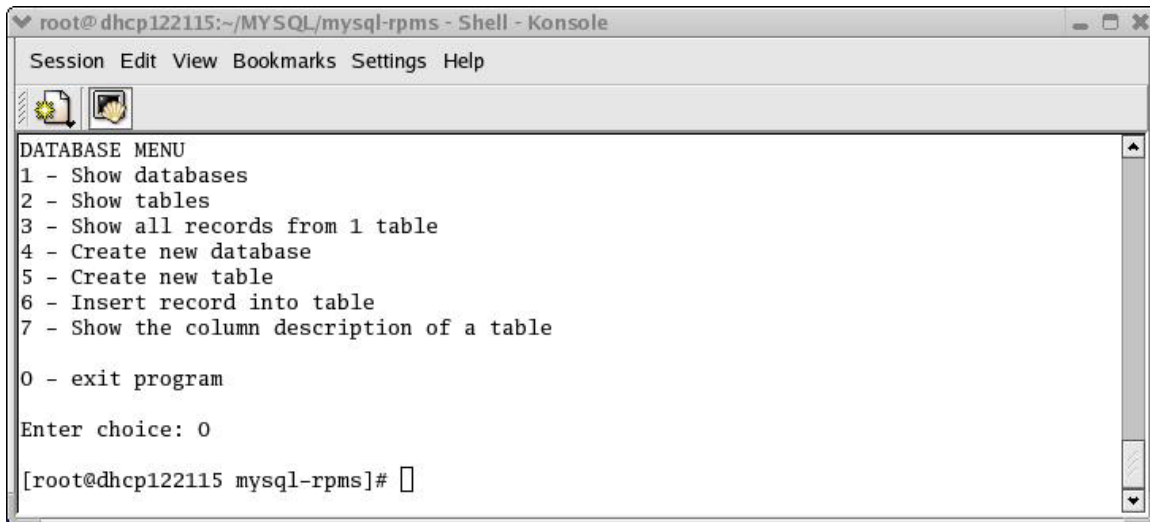
Computer_Name  File_Name      Keyword_File_5 Keyword_File_6  Keyword_File_7
little(RH)     opencat
boss(MAND)     netcat

Press Enter to choose again
```

**Figure 39: Showing records**

When the user wants to view the records contained in any table then option three is chosen. As seen in the figure above two records are currently in the BACKDOORS table (one of them being the contents found in the record file in Figure 37). Due to the way MySQL displays its information, I had to split up its output into three printouts. I chose to use the Computer\_Name and File\_Name fields as constants so the user could still tell what information belongs to what record entry. Users should also be aware that even though each field can contain up to 64 characters, when the information is displayed the column titles tend not be located directly above its contents. Therefore users should try to use as few characters as necessary to still allow them to understand their entries.

## 2.5.8 Exiting Simple-MySQL



**Figure 40: Exiting Simple-MySQL**

To exit the *Simple-MySQL* script the user simply needs to choose the '0' option and hit enter. The user is then returned to the command line. Also any errors the MySQL database comes across when the user attempts one of the options will be displayed on the screen for the user.

## 3.0 Project Scripts' Run Times & Results

Project Scripts' Run Times		
	RedHat (mins)	Mandrake (days)
GRcompare	3	3 (min)
DELread	10	Data was corrupted
MACread	3	----
UNRMread (unrm.results)	----	1.5
UNRMread (block files)	----	1

**Table 4: Run time for project scripts**

Due to time constraints *UNRMread* was not run on the RedHat data. To be more specific it was stopped after three days of running. *Lazarus* was stopped from running on the results of RedHat's *unrm* after three days also. All times are dependent on the amount of data presented in Table 5: below.

File Reduction				
	Number of files before filtering with scripts (A)	Number of files after filtering with scripts (B)	Number of files created by scripts for user analysis (C)	Percent decrease in number of files (D)
GRcompare	>2	>2	2	--
DELread (RH)	3359	210	13	99.61
MACread (RH)	1	--	10	--
UNRMread (MAN unrm single file)	1	--	10	--
UNRMread (MAN lazarus blocks)	95364	2847	10	99.99

**Table 5: File reduction analysis**

Column A in the table above refers to the total amount of files created by TCT. Since the inode files were corrupted for the Mandrake (MAN) box *DELread* was not executed on this result. Column B shows the number of original files that contained matches to the keywords. Column C delivers the actual number of files the user has to analyze for evidence. Column D reports the percent decrease in number of files the user has to analyze.

Where

- $D = 100 - 100(C/A)$

Data Reduction				
	Amount of total data before script filtration (MB) (A)	Amount of total data after script filtration (MB) (B)	Amount of actual data for user analysis (MB) (C)	Percent decrease in data size (D)
GRcompare	--	--	--	--
DELread (RH)	102	29	3.2	96.86
MACread (RH)	64KB	16KB	16KB	75.00
UNRMread (MAN single file)	2500	38	38	98.48
UNRMread (MAN lazarus blocks)	2500	138	8.5	99.66

## Table 6: Data size reduction analysis

The data reduction table holds the same description as that for the file reduction table.

## 4.0 Conclusion

Anyone who attempts to do computer forensics analysis can expect to spend a considerable amount of time just gathering the raw data needed for analysis. Even after the data has been collected, if the persons attempting to analyze the data do not have a method for which to do analysis they can waste many hours, days or even weeks trying to come up with a way to find the information they are searching for. Also, if they just plan to create random files containing the information they searched for (which can get very unorganized, very fast) a substantial loss of valuable time can occur due to the researching of all those files to find some specific incident. This project had four objectives to assist this process.

The first objective was *to collect data from multiple compromised computers and store it directly on the Intermediate Storage Box*. This objective was successfully completed via the USB connections. The dd images were transferred to the ISB without any need to store the images temporarily on the computers that were compromised.

The second objective was *to apply information filtering techniques to determine what critical data needed to be exported to the DBB*. This objective was also achieved completely. Deciding to write scripts that could take to output from The Coroner's Toolkit and thoroughly search it for user defined keyword proved extremely critical for filtering out the critical data.

- *DELread* reduced the amount of important files from 3359 to 13 (99.61% reduction) and the amount of data for analysis from 102 to 3.2 Megabytes (96.86% reduction)
- *UNRMread* when used on *lazarus* output reduced the amount of important files from 95364 to 10 (99.99% reduction) and the amount of data for analysis from 2500 to 8.5 Megabytes (99.66% reduction)
- *UNRMread* when used on *unrm* output reduced the amount of data for analysis from 2500 to 38 Megabytes (98.48% reduction)
- *MACread* reduced the amount of data for analysis from 64 to 16 Kilobytes (75% reduction)

The third objective was *to store the critical data from each computer into a single database for possible future analysis or reference*. Again, successfully completed. *Simple-MySQL* allowed for the creation of a database and tables for the storage of user specified evidence.

The ability to store user specific evidence satisfies the forth objective of *tracing evidence of the compromises to the DBB*. Thus bringing a successful end to this project.

## REFERENCES

- 1) <http://www.porcupine.org/forensics/tct.html>
- 2) <http://www.securityfocus.com/infocus/1503>
- 3) TCT README.FIRST
- 4) <http://www.fish.com/forensics/freezing.pdf>
- 5) Italics from graver-robber.README
- 6) <http://www.guidancesoftware.com/products/EnCaseForensic/index.shtm>
- 7) <http://www.sleuthkit.org/sleuthkit/download.php>
- 8) pcat man page
- 9) lls man page
- 10) lcat man page
- 11) file man page

## APPENDIX A - Full procedure

This appendix provides the full procedure used to set up and execute this project (except for the parts demonstrated in the paper i.e. Simple-MySQL)

ISB – Intermediate Storage Box

DBB – Database Box

RH – RedHat

MAN – Mandrake

TCT – The Coroner's Toolkit

OS – operating system

All the instructions containing the 'mount and umount' commands assume the user is in the parent directory of the directory being mounted and unmounted.

### Setting up the external hard drives ISB & DBB

1. Connect the USB jacks into the USB ports on the Linux box
2. Connections were recognized as /dev/sda and /dev/sdb on my box
3. Enter 'fdisk /dev/sda' on the command line
4. At the prompt enter 'm' to view the list of options
5. At the prompt enter 'n' to add a new partition
6. At the prompt enter 'p' to make a primary partition
7. At the 'First cylinder' prompt press enter to accept the default start point
8. At the 'Last cylinder' prompt enter '+155000M' (155GB)
9. Back at the main prompt enter 't' to change the system partition system id
10. At the 'partition number' prompt enter '1'
11. At the 'Hex code' prompt enter '83' for a general 'ext2' Linux id
12. Repeat steps 5-11 for however many partitions are to be created.  
Remember to change the cylinder sizes, partition numbers, and system id numbers to fit the sizes and operating systems are being worked with.
13. Once satisfied with all the partitions to be created enter 'w' at the main prompt to write the table to the disk and exit fdisk.
14. On the command line enter 'mkfs.ext2 /dev/sda#' for each # partition (or whatever make filesystem type fits the partitions created by the user)

### Creating mount and storage directories

- On the Linux box make the mount directories
1. On the command line enter 'mkdir RH'
  2. On the command line enter 'mkdir RH-img'
  3. On the command line enter 'mkdir MAN'
  4. On the command line enter 'mkdir MAN-img'
  5. On the command line enter 'mkdir MYSQL'

Users can name these directories anything and make them wherever they choose, but at a minimum they will need five directories to mount everything

(these five are only if two computers are being analyzed, increase as the number of computers increases).

6. Enter 'mount /dev/sda1 /RH/' (if this is your Redhat partition) on the command line
7. Enter 'mkdir RH\_DD\_images' on the command line
8. Enter 'mkdir RH\_DD\_hacked\_images' on the command line
9. Enter 'mkdir RH\_base\_deleted' on the command line
10. Enter 'mkdir RH\_hacked\_deleted' on the command line
11. Enter 'mkdir RH\_base\_unrm' on the command line
12. Enter 'mkdir RH\_hacked\_unrm' on the command line

The RH\_base\_deleted and mkdir RH\_base\_unrm are not really necessary unless you want to gather this information just to have.

13. Enter 'mount /dev/sda2 /MAN/' (if this is your MANDRAKE partition) on the command line
14. Repeat steps 7-12 replacing 'RH' with 'MAN'
15. Enter 'umount RH/' on the command line
16. Enter 'umount MAN/' on the command line

### Taking dd images

1. Connect the ISB to the RH box via the USB connector
2. On the command line enter 'mount /dev/sda1 RH/'
3. On the command line enter 'dd if=/dev/hda5 of=/root/RH/RH\_DD\_images/RH\_root\_image.img' (Replace the '5' in hda5 with the number of the device the root partition is assigned to)
4. On the command line enter 'umount RH/'
5. Disconnect the USB connector from the RH box
6. Repeat steps 1-5 for the MAN box

If the 'fdisk' steps were performed on the RH box the user may have to enter the fdisk program after connecting the ISB to the MAN box and write the table to the MAN system by entering 'w' and pressing enter.

### Installing TCT

Downloaded TCT from the official site at  
<http://www.porcupine.org/forensics/tct.html>

1. Connect the ISB to the RH box via the USB connector
2. On the command line enter 'mount /dev/sda1 RH/'
3. Download and store the 'tct-1.14.tar.gz' in the RH directory
4. Enter 'gunzip RH/tct-1.14.tar.gz' on the command line
5. Enter 'tar -xvf RH/tct-1.14.tar' on the command line



6. Enter 'cd RH/tct-1.14' on the command line
7. Enter 'make' on the command line
8. Repeat steps 1-7 for the MAN box

When TCT is installed it stores the absolute paths to its scripts. This means if the user uses a different directory to mount the ISB to, the programs won't work. Therefore it is just easier to install TCT on all partitions.

### Running TCT

The steps used to run TCT for this project are the same as found in reference \_\_\_\_\_. Since this project did not run TCT on a live system I will extract only the instructions used to run it on a dead system (a.k.a. an imaged drive).

1. # mount /dev/sda1 RH/ (if not already mounted)
2. # mount -o ro,loop,nodev,noexec,nosuid,noatime  
RH/RH\_DD\_hacked\_images/ RH\_hacked\_root\_image.img RH-img/
3. # cd /RH/tct-1.14/bin
4. # ./grave-robber -c /root/RH-img -o LINUX2 -MivVt (only use LINUX2 if working on a Linux box other OS types are available read the man pages)
5. # ./ils -rf ext2fs  
/root/RH/RH\_DD\_hacked\_images/RH\_hacked\_root\_image.img | awk -F '|' '{(\$2=="f") {print \$1}} |while read i; do /root/RH/tct-1.14/bin/icat /root/RH/RH\_DD\_hacked\_images/RH\_hacked\_root\_image.img \$i > /root/RH/RH\_hacked\_deleted/\$i; done
6. # ./unrm /root/RH/RH\_DD\_hacked\_images/RH\_hacked\_root\_image.img  
> /root/RH/RH\_hacked\_unrm/RH\_hacked\_root\_unrm.results
7. # ./lazarus -h /root/RH/RH\_hacked\_unrm/RH\_hacked\_root\_unrm.results
8. # ./mactime -b /<path to the>/<mactime database>/body 2/03/2004-2/06/2004 >> RH-MACS\_FEB03-FEB06.txt

These steps are performed for both the baseline and compromised images from the RH and MAN boxes with the exceptions of steps 5-7. These steps do not have to be performed on the baseline images unless the user just wants this information for some reason. Also, since the mactime database was created when grave-robber was run in our case. The step 8 shows how to create a single file with specific start and stop dates to be used by the MACread script.

After running grave-robber on the baseline and compromised images the words 'base' and 'hacked' were appended to the otherwise both runs would create directories with the same name but different time stamps.

For Example:

Running grave-robber on the baseline image then on the compromised image would create-

localhost.localdomain...12\40  
localhost.localdomain...16\40

So to make this less confusing add base and hack to the directory names-

base\_localhost.localdomain...12\40  
hacked\_localhost.localdomain...16\40

Now each can be identified easily.

### Running the Project Scripts

Change to whatever directory is holding the scripts

1. # ./GRcompare /<path to>/base\_localhost.localdomain...12\40 /<path to>/hacked\_localhost.localdomain...16\40 RH
2. # ./DELread /root/RH/RH\_hacked\_deleted/ DEL-keyword.txt RH
3. # ./MACread /<path to the file>/<created in step 8 of Running TCT>/ RH-MACS\_FEB03-FEB06.txt MAC-keyword.txt RH

UNRMread can be run in two different manners depending on the data available. If the user only has the unrm.results file created in step 6 of Running TCT then this command is issued

4. # ./UNRMread /root/MAN/MAN\_hacked\_unrm/ \MAN\_hacked\_unrm/MAN\_hacked\_root\_unrm.results DEL-keyword.txt MAN-unrmfile

If the user was able to run and complete step 7 of Running TCT then a directory containing a mass number of text file will be created by lazarus and the user can run UNRMread in this fashion

4. # ./UNRMread /root/<path to directory created by lazarus>/ DEL-keyword.txt MAN-block

### Installing MySQL

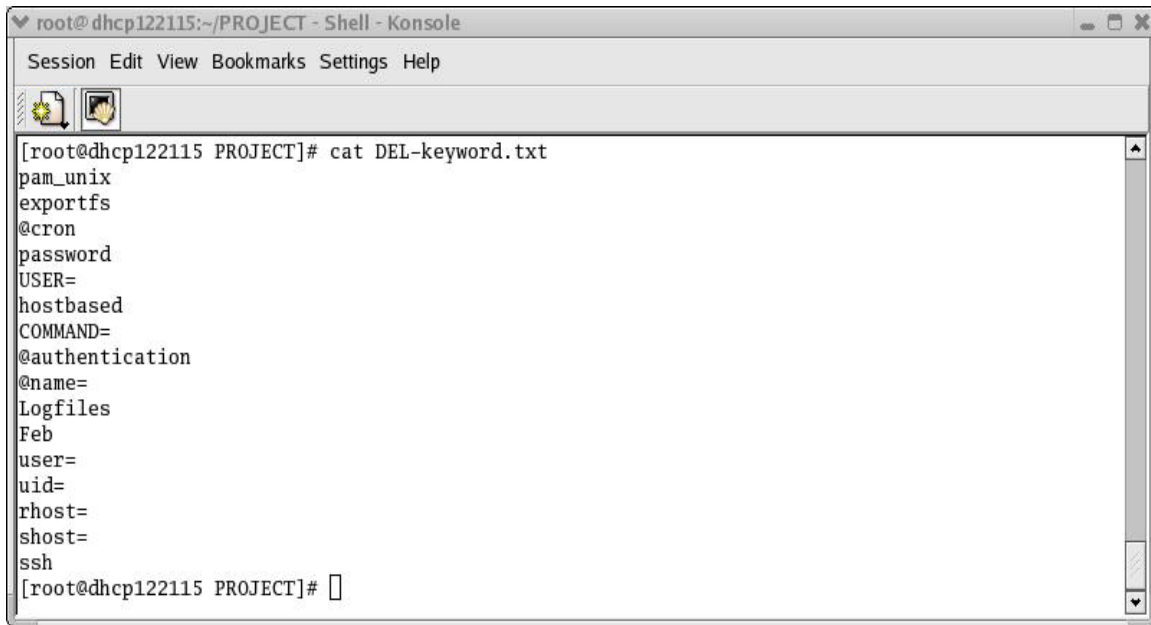
Download made from [www.mysql.com](http://www.mysql.com)

1. Connect the DBB to the Linux box via the USB connector
2. On the command line enter 'mount /dev/sda1 MYSQL/'
3. Download and store the 'mysql-standard-4.0.18-pc-linux-i686.tar.gz' in the MYSQL directory

4. Download and store the 'mysqlcc-0.9.3-linux-glib23.tar.gz' in '/usr/local/'
5. Enter 'gunzip MYSQL/mysql-standard-4.0.18-pc-linux-i686.tar.gz' on the command line
6. Enter 'tar -xvf MYSQL/mysql-standard-4.0.18-pc-linux-i686.tar' on the command line
7. Enter 'cd MYSQL' on the command line
8. Enter 'ln -s mysql-standard-4.0.18-pc-linux-i686 mysql' on the command line
9. Enter 'cd mysql/' on the command line
10. Follow the installation directions found in the 'INSTALL-BINARY' file
11. Enter 'cd /usr/local' on the command line
12. Enter 'gunzip mysqlcc-0.9.3-linux-glib23.tar.gz' on the command line
13. Enter 'tar -xvf mysqlcc-0.9.3-linux-glib23.tar' on the command line
14. Enter 'ln -s mysqlcc-0.9.3-linux-glib23.tar mysqlcc' on the command line
15. Enter './mysqlcc/mysqlcc' on the command line to start MySQLcc

## APPENDIX B - Examining DELread's results

This appendix provides more evidence of the compromise gathered from the keyword files created by *DELread*. It also shows the original amount of data returned by the *ils* and *icat* tools run on the RedHat box that *DELread* had to filter through.

A screenshot of a terminal window titled "root@dhcp122115:~/PROJECT - Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". Below the menu bar is a toolbar with icons for a file, a shell, and a search. The terminal content shows the command "[root@dhcp122115 PROJECT]# cat DEL-keyword.txt" followed by the output of the file: "pam\_unix", "exportfs", "@cron", "password", "USER=", "hostbased", "COMMAND=", "@authentication", "@name=", "Logfiles", "Feb", "user=", "uid=", "rhost=", "shost=", "ssh". The prompt "[root@dhcp122115 PROJECT]#" is shown at the bottom of the terminal.

```
root@dhcp122115:~/PROJECT - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 PROJECT]# cat DEL-keyword.txt
pam_unix
exportfs
@cron
password
USER=
hostbased
COMMAND=
@authentication
@name=
Logfiles
Feb
user=
uid=
rhost=
shost=
ssh
[root@dhcp122115 PROJECT]#
```

Figure 41: DEL-keyword.txt file used for DELread

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-COMMAND=_keyword.txt
-----entries found in file /root/RH-DELETED//1564832-----
Feb  4 15:27:22 little sshd[6266]: Accepted password for rpg from 192.168.123.3 port 38428 ssh2
Feb  4 15:29:09 little sudo:      root : TTY=pts/2 ; PWD=/etc/ssh ; USER=root ; COMMAND=/usr/sbin/useradd
Feb  4 15:29:19 little sudo:      root : TTY=pts/2 ; PWD=/etc/ssh ; USER=root ; COMMAND=/usr/sbin/useradd sdsd
Feb  4 15:29:19 little useradd[6310]: new group: name=sdsd, gid=501
--
Feb  4 16:08:22 little useradd[6580]: new user: name=rpg, uid=501, gid=501, home=/var/rpg, shell=/bin/bash
Feb  4 17:25:15 little sudo:      rpg : command not allowed ; TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=useradd gp
Feb  4 17:35:24 little sudo:      rpg : TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=list
Feb  4 17:35:44 little sudo:      root : TTY=pts/6 ; PWD=/root ; USER=root ; COMMAND=list
Feb  4 17:39:22 little sudo:      rpg : command not allowed ; TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=useradd gp
Feb  4 17:53:14 little sudo:      rpg : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/useradd -u 0 -d /root
bossy
Feb  4 17:59:20 little sudo:      rpg : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/useradd -u 0 go
Feb  4 18:00:08 little sudo:      rpg : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/useradd -d /root bossy
Feb  4 18:00:08 little useradd[7190]: new group: name=bossy, gid=502
Feb  4 18:00:08 little useradd[7190]: new user: name=bossy, uid=502, gid=502, home=/root, shell=/bin/bash
Feb  4 18:00:30 little sudo:      rpg : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/usermod -u 0 bossy
Feb  4 18:05:52 little sshd[7205]: Accepted password for rpg from 192.168.123.3 port 40428 ssh2
Feb  4 18:06:32 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb  4 18:11:25 little sshd[7303]: Accepted password for rpg from 192.168.123.3 port 40502 ssh2
--
Feb  4 18:34:49 little sshd[7549]: Accepted password for rpg from 192.168.123.3 port 40796 ssh2
Feb  4 18:34:57 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb  4 18:36:31 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi /etc/profile
Feb  4 18:37:22 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi .bash_profile
Feb  4 18:38:42 little sshd[7626]: Accepted password for rpg from 192.168.123.3 port 40844 ssh2
Feb  4 18:38:53 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb  4 18:39:34 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi /etc/profile
Feb  5 11:40:26 little sshd[8754]: Accepted hostbased for root from 192.168.123.2 port 1068 ssh2
[root@dhcp122115 root]#
```

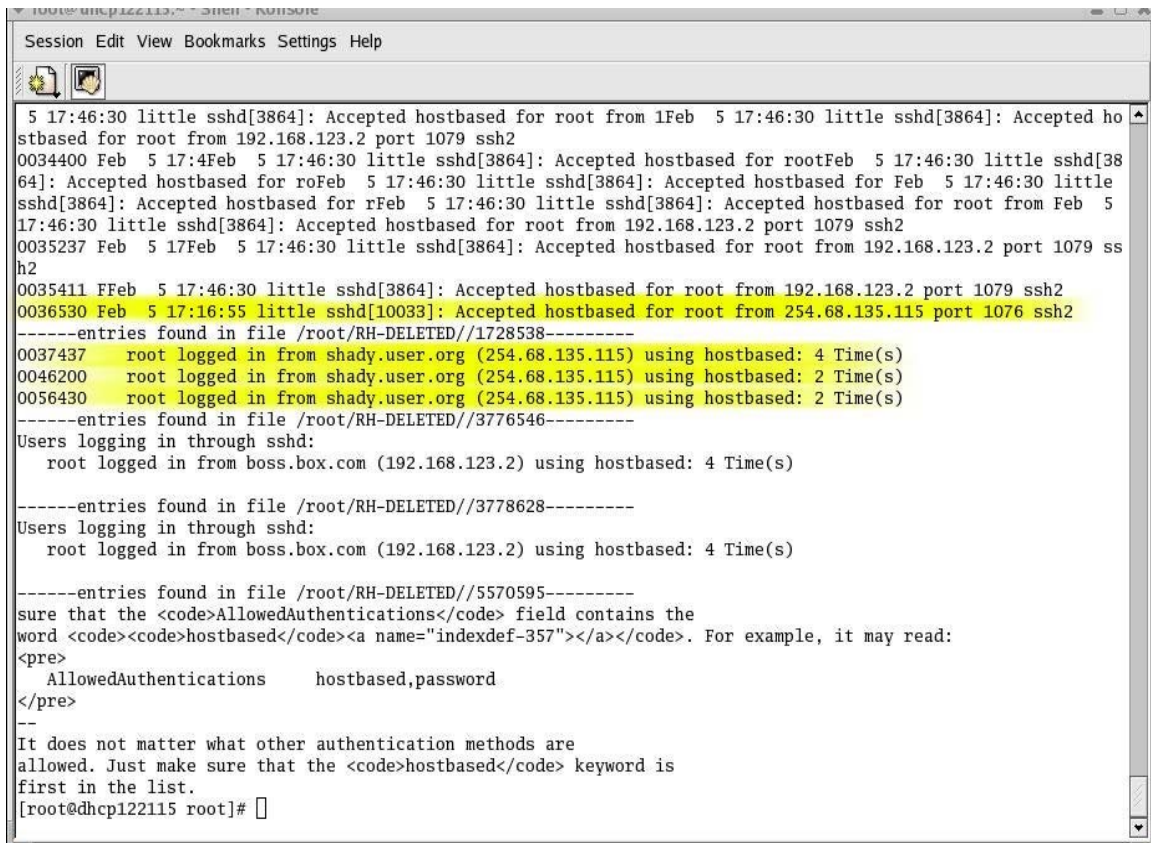
**Figure 42: COMMAND= keyword.txt file**

Shows new users being added: rpg, bossy  
Shows ssh and sudo interactions from user rpg  
Shows possible editing of /etc/profile file  
Shows hostbased authentication used for ssh

```
Session Edit View Bookmarks Settings Help
[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-hostbased_keyword.txt
-----entries found in file /root/RH-DELETED//1564832-----
Feb 3 14:32:14 little sshd[4460]: Server listening on 0.0.0.0 port 22.
Feb 3 14:32:24 little sshd[4464]: Accepted hostbased for root from 192.168.123.2 port 1042 ssh2
Feb 3 14:56:55 little sshd[4460]: Received signal 15; terminating.
--
Feb 3 14:57:11 little sshd[4564]: Server listening on 0.0.0.0 port 22.
Feb 3 14:57:19 little sshd[4568]: Accepted hostbased for root from 192.168.123.2 port 1046 ssh2
Feb 3 15:06:53 little adduser[4649]: new group: name=bob, gid=500
Feb 3 15:06:53 little adduser[4649]: new user: name=bob, uid=500, gid=500, home=/home/bob, shell=/bin/bash
Feb 4 15:01:45 little sshd[6127]: Accepted hostbased for root from 192.168.123.2 port 1065 ssh2
Feb 4 15:07:41 little useradd[6173]: new group: name=rpg, gid=501
--
Feb 4 15:29:34 little userdel[6311]: remove group `sdsd'
Feb 4 16:06:35 little sshd[6530]: Accepted hostbased for root from 192.168.123.2 port 1066 ssh2
Feb 4 16:07:01 little userdel[6574]: delete user `rpg'
--
Feb 4 18:39:34 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi /etc/profile
Feb 5 11:40:26 little sshd[8754]: Accepted hostbased for root from 192.168.123.2 port 1068 ssh2
Feb 5 16:48:44 little sshd[8754]: fatal: login_init_entry: Cannot find user "root"
--
Feb 5 17:16:48 little sshd[10029]: Server listening on 0.0.0.0 port 22.
Feb 5 17:16:55 little sshd[10033]: Accepted hostbased for root from 192.168.123.2 port 1076 ssh2
Feb 5 17:23:21 little groupdel[10101]: remove group `rpg'
--
Feb 5 17:28:51 little useradd[10115]: new user: name=rpg, uid=78, gid=78, home=/var/rpg, shell=/bin/bash
Feb 5 17:32:07 little sshd[10123]: Accepted hostbased for root from 192.168.123.2 port 1077 ssh2
Feb 5 17:39:58 little sshd[10029]: Received signal 15; terminating.
Feb 5 17:41:53 little sshd[2843]: Server listening on 0.0.0.0 port 22.
Feb 5 17:46:30 little sshd[3864]: Accepted hostbased for root from 192.168.123.2 port 1079 ssh2
Feb 6 11:56:00 little sshd[2843]: Received signal 15; terminating.
-----entries found in file /root/RH-DELETED//1564838-----
--More--(38%)
```

**Figure 43: Hostbased keyword.txt file**

Shows hostbased authentication used with ssh  
Shows new user being added: bob



```
Session Edit View Bookmarks Settings Help

5 17:46:30 little sshd[3864]: Accepted hostbased for root from 1Feb 5 17:46:30 little sshd[3864]: Accepted ho
stbased for root from 192.168.123.2 port 1079 ssh2
0034400 Feb 5 17:4Feb 5 17:46:30 little sshd[3864]: Accepted hostbased for rootFeb 5 17:46:30 little sshd[38
64]: Accepted hostbased for roFeb 5 17:46:30 little sshd[3864]: Accepted hostbased for Feb 5 17:46:30 little
sshd[3864]: Accepted hostbased for rFeb 5 17:46:30 little sshd[3864]: Accepted hostbased for root from Feb 5
17:46:30 little sshd[3864]: Accepted hostbased for root from 192.168.123.2 port 1079 ssh2
0035237 Feb 5 17Feb 5 17:46:30 little sshd[3864]: Accepted hostbased for root from 192.168.123.2 port 1079 ss
h2
0035411 FFeb 5 17:46:30 little sshd[3864]: Accepted hostbased for root from 192.168.123.2 port 1079 ssh2
0036530 Feb 5 17:16:55 little sshd[10033]: Accepted hostbased for root from 254.68.135.115 port 1076 ssh2
-----entries found in file /root/RH-DELETED//1728538-----
0037437 root logged in from shady.user.org (254.68.135.115) using hostbased: 4 Time(s)
0046200 root logged in from shady.user.org (254.68.135.115) using hostbased: 2 Time(s)
0056430 root logged in from shady.user.org (254.68.135.115) using hostbased: 2 Time(s)
-----entries found in file /root/RH-DELETED//3776546-----
Users logging in through sshd:
root logged in from boss.box.com (192.168.123.2) using hostbased: 4 Time(s)

-----entries found in file /root/RH-DELETED//3778628-----
Users logging in through sshd:
root logged in from boss.box.com (192.168.123.2) using hostbased: 4 Time(s)

-----entries found in file /root/RH-DELETED//5570595-----
sure that the <code>AllowedAuthentications</code> field contains the
word <code><code>hostbased</code><a name="indexdef-357"></a></code>. For example, it may read:


```

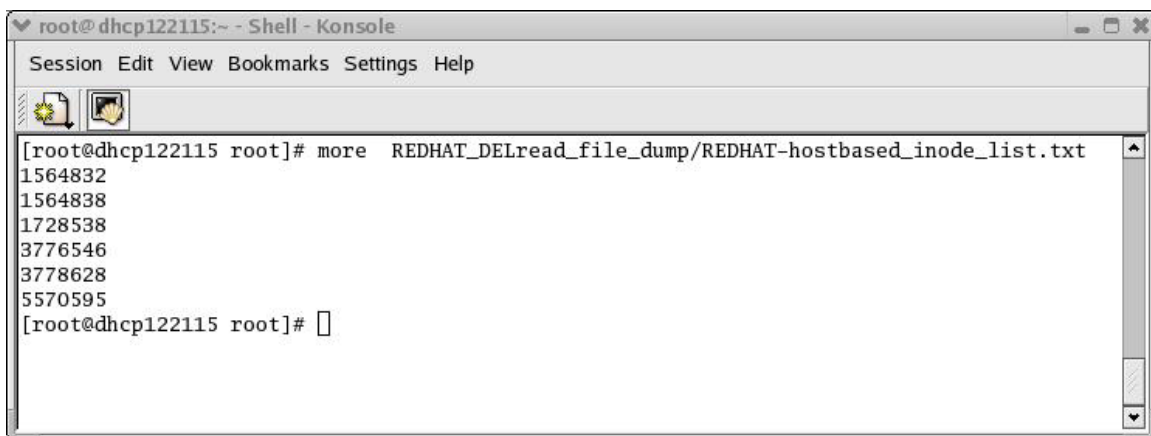
AllowedAuthentications    hostbased,password

```


It does not matter what other authentication methods are
allowed. Just make sure that the <code>hostbased</code> keyword is
first in the list.
[root@dhcp122115 root]#
```

**Figure 44: Hostbased keyword.txt file part 2**

Shows hostbased authentication on ssh from IP 254.64.135.115  
Shows root access from shady.user.org



```
root@dhcp122115:~ - Shell - Konsole

Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-hostbased_inode_list.txt
1564832
1564838
1728538
3776546
3778628
5570595
[root@dhcp122115 root]#
```

**Figure 45: Inode.txt file for the hostbased keyword**

Shows which copied inode files contain information regarding the 'hostbased' keyword.



```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb 3 13:40:11 little sshd[4323]: Failed password for root from 192.168.123.2 port 1033 ssh2
Feb 3 13:42:00 little sshd[4323]: fatal: Timeout before authentication for 192.168.123.2
---
Feb 4 15:23:08 little useradd[6257]: new user: name=rpg, uid=78, gid=78, home=/var/rpg, shell=/bin/bash
Feb 4 15:27:22 little sshd[6266]: Accepted password for rpg from 192.168.123.3 port 38428 ssh2
Feb 4 15:29:09 little sudo:      root : TTY=pts/2 ; PWD=/etc/ssh ; USER=root ; COMMAND=/usr/sbin/useradd
---
Feb 4 18:00:30 little sudo:      rpg : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/usermod -u 0 bossy
Feb 4 18:05:52 little sshd[7205]: Accepted password for rpg from 192.168.123.3 port 40428 ssh2
Feb 4 18:06:32 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
Feb 4 18:11:25 little sshd[7303]: Accepted password for rpg from 192.168.123.3 port 40502 ssh2
Feb 4 18:14:29 little userdel[7352]: delete user `rpg'
---
Feb 4 18:14:49 little useradd[7353]: new user: name=rpg, uid=78, gid=78, home=/var/rpg, shell=/bin/bash
Feb 4 18:20:52 little sshd[7382]: Failed password for rpg from 192.168.123.3 port 40619 ssh2
Feb 4 18:21:06 little last message repeated 2 times
Feb 4 18:21:25 little sshd[7385]: Failed password for rpg from 192.168.123.3 port 40626 ssh2
Feb 4 18:24:10 little sshd[7393]: Accepted password for rpg from 192.168.123.3 port 40662 ssh2
Feb 4 18:27:41 little sshd[7428]: Accepted password for rpg from 192.168.123.3 port 40707 ssh2
Feb 4 18:28:09 little sshd[7463]: Accepted password for rpg from 192.168.123.3 port 40714 ssh2
Feb 4 18:29:24 little userdel[7473]: delete user `rpg'
---
Feb 4 18:30:29 little useradd[7479]: new user: name=rpg, uid=78, gid=78, home=/var/rpg, shell=/bin/bash
Feb 4 18:30:51 little sshd[7482]: Failed password for rpg from 192.168.123.3 port 40747 ssh2
Feb 4 18:30:53 little sshd[7482]: Accepted password for rpg from 192.168.123.3 port 40747 ssh2
Feb 4 18:34:49 little sshd[7549]: Accepted password for rpg from 192.168.123.3 port 40796 ssh2
Feb 4 18:34:57 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
---
Feb 4 18:37:22 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi .bash_profile
Feb 4 18:38:42 little sshd[7626]: Accepted password for rpg from 192.168.123.3 port 40844 ssh2
Feb 4 18:38:53 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
-----entries found in file /root/RH-DELETED/1564838-----
0024655 Feb 3 13:40:11 little sshd[4323]: Failed password for root from 254.68.135.115 port 1033 ssh2
0025673 Feb 3 13:18:26 little sshd[4155]: Accepted password for root from 254.68.135.115 port 1032 ssh2
--More--(0%)
```

**Figure 46: Password keyword.txt file part 1**

Shows user rpg being added and logging into local ssh.  
Shows foreign IP address logging into ssh



```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb  4 18:21:25 little sshd[7385]: Failed password for rpg from 192.168.123.3 port 40626 ssh2
Feb  4 18:24:10 little sshd[7393]: Accepted password for rpg from 192.168.123.3 port 40662 ssh2
Feb  4 18:27:41 little sshd[7428]: Accepted password for rpg from 192.168.123.3 port 40707 ssh2
Feb  4 18:28:09 little sshd[7463]: Accepted password for rpg from 192.168.123.3 port 40714 ssh2
Feb  4 18:29:24 little userdel[7473]: delete user `rpg'
---
Feb  4 18:30:29 little useradd[7479]: new user: name=rpg, uid=78, gid=78, home=/var/rpg, shell=/bin/bash
Feb  4 18:30:51 little sshd[7482]: Failed password for rpg from 192.168.123.3 port 40747 ssh2
Feb  4 18:30:53 little sshd[7482]: Accepted password for rpg from 192.168.123.3 port 40747 ssh2
Feb  4 18:34:49 little sshd[7549]: Accepted password for rpg from 192.168.123.3 port 40796 ssh2
Feb  4 18:34:57 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
---
Feb  4 18:37:22 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/bin/vi .bash_profile
Feb  4 18:38:42 little sshd[7626]: Accepted password for rpg from 192.168.123.3 port 40844 ssh2
Feb  4 18:38:53 little sudo:      rpg : TTY=pts/3 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/visudo
-----entries found in file /root/RH-DELETED//1564838-----
0024655 Feb  3 13:40:11 little sshd[4323]: Failed password for root from 254.68.135.115 port 1033 ssh2
0025673 Feb  3 13:18:26 little sshd[4155]: Accepted password for root from 254.68.135.115 port 1032 ssh2
0026034 Feb  3 12:34:29 little sshd[4071]: Accepted password for root from 254.68.135.115 port 1031 ssh2
0026175 Feb  3 12:33:08 little sshd[4069]: Failed password for root from 254.68.135.115 port 1030 ssh2
0026334 Feb  3 12:33:02 little sshd[4069]: Failed password for root from 254.68.135.115 port 1030 ssh2
-----entries found in file /root/RH-DELETED//1720338-----
pcretest      (1) - a program for testing Perl-compatible regular expressions
chage          (1) - change user password expiry information
gpasswd        (1) - administer the /etc/group file
-----entries found in file /root/RH-DELETED//1728538-----
0046322      root logged in from shady.user.org (254.68.135.115) using password: 2 Time(s)
0046503      root/password from 254.68.135.115: 3 Time(s)
0056311      rpg logged in from little.box.com (192.168.123.3) using password: 9 Time(s)
0056612      rpg/password from 254.68.135.115
-----entries found in file /root/RH-DELETED//1859591-----

s in the Workspace Switcher. Properties a _sound when new mail arrives into run dialog POP3-server on sn this command before we
check the maillic_html) references for all your panels Buttonw date in tooltip boxows from a _ll workspacese expander arrow of
a window --More--(1%)

```

**Figure 47: Password keyword.txt file part 2**

Shows root user from shady.user.org logging into local box  
Shows user rpg logging into local box from foreign IP

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-rhost\=_keyword.txt
-----entries found in file /root/RH-DELETED//1728526-----
Unknown Entries:
authentication failure; logname= uid=0 euid=0 tty=:0 ruser= rhost= user=root: 1 Time(s)

-----entries found in file /root/RH-DELETED//1728538-----
0025636      authentication failure; logname= uid=0 euid=0 tty=:0 ruser= rhost= user=root: 1 Time(s)
0042300      2 more authentication failures; logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=little.box.com user=rpg: 1 Ti
me(s)
0042525      authentication failure; logname=root uid=0 euid=0 tty=pts/2 ruser= rhost= user=rpg: 5 Time(s)
0042672      authentication failure; logname=rpg uid=0 euid=0 tty=pts/7 ruser= rhost= user=rpg: 1 Time(s)
0043036      authentication failure; logname=root uid=0 euid=0 tty=pts/6 ruser= rhost= user=rpg: 7 Time(s)
0043203      authentication failure; logname=root uid=0 euid=0 tty=pts/5 ruser= rhost= user=rpg: 4 Time(s)
[root@dhcp122115 root]#

```

**Figure 48: rhost= keyword.txt file**

Shows failed attempts by new user rpg and root to log into ssh

```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_DELread_file_dump/REDHAT-uid=_keyword.txt
-----entries found in file /root/RH-DELETED//1564832-----
Feb  3 13:18:58 little adduser[4201]: new group: name=bob, gid=500
Feb  3 13:18:58 little adduser[4201]: new user: name=bob, uid=500, gid=500, home=/home/bob, shell=/bin/bash
Feb  3 13:20:10 little userdel[4203]: delete user `bob'
--
Feb  3 15:06:53 little adduser[4649]: new group: name=bob, gid=500
Feb  3 15:06:53 little adduser[4649]: new user: name=bob, uid=500, gid=500, home=/home/bob, shell=/bin/bash
Feb  4 15:01:45 little sshd[6127]: Accepted hostbased for root from 192.168.123.2 port 1065 ssh2
Feb  4 15:07:41 little useradd[6173]: new group: name=rpg, gid=501
Feb  4 15:07:41 little useradd[6173]: new user: name=rpg, uid=501, gid=501, home=/, shell=/bin/bash
Feb  4 15:08:37 little userdel[6175]: delete user `rpg'
--
Feb  4 15:11:04 little useradd[6193]: new group: name=rpg, gid=78
Feb  4 15:11:04 little useradd[6193]: new user: name=rpg, uid=78, gid=78, home=/, shell=/bin/bash
Feb  4 15:19:49 little userdel[6251]: delete user `rpg'
--
Feb  4 15:23:08 little useradd[6257]: new group: name=rpg, gid=78
Feb  4 15:23:08 little useradd[6257]: new user: name=rpg, uid=78, gid=78, home=/var/rpg, shell=/bin/bash
Feb  4 15:27:22 little sshd[6266]: Accepted password for rpg from 192.168.123.3 port 38428 ssh2
--
Feb  4 15:29:19 little useradd[6310]: new group: name=sdsd, gid=501
Feb  4 15:29:19 little useradd[6310]: new user: name=sdsd, uid=501, gid=501, home=/home/sdsd, shell=/bin/bash
Feb  4 15:29:34 little userdel[6311]: delete user `sdsd'
--
Feb  4 16:08:22 little useradd[6580]: new group: name=rpg, gid=501
Feb  4 16:08:22 little useradd[6580]: new user: name=rpg, uid=501, gid=501, home=/var/rpg, shell=/bin/bash
Feb  4 17:25:15 little sudo:      rpg : command not allowed ; TTY=pts/5 ; PWD=/var/rpg ; USER=root ; COMMAND=useradd gp
--
Feb  4 18:00:08 little useradd[7190]: new group: name=bossy, gid=502
Feb  4 18:00:08 little useradd[7190]: new user: name=bossy, uid=502, gid=502, home=/root, shell=/bin/bash
Feb  4 18:00:30 little sudo:      rpg : TTY=pts/6 ; PWD=/var/rpg ; USER=root ; COMMAND=/usr/sbin/usermod -u 0 bossy
--
--More-- (20%)
```

**Figure 49: uid= keyword.txt file part 1**

Shows new users being added: bob, rpg, bossy



root@little: ~ - Shell - Konsole <4>

Session Edit View Bookmarks Settings Help

[root@little root]# ls DELETED/

1001498	1241202	1669134	2117646	2650166	3094539	3582028	38928	4200464	4599845	5115934	5570593	647204
1003532	1241203	1673220	2117648	2652163	3100681	358407	38929	4202599	4599846	5115935	5570594	647207
100362	1243147	1677579	2117649	2654215	3102732	3584081	38930	4204563	4599847	5115936	5570595	647208
1005576	1247433	167951	2117650	2656265	3104787	3586100	38931	4206597	4599848	5115937	5570596	647209
1007645	1249307	1682785	2117651	2660355	3108884	3588221	38932	4208665	4603938	5115938	5570597	649240
1009672	124951	1682786	2117652	2662403	3110932	3590180	3893252	4208666	4605981	5115939	5570598	651302
1011731	1251353	1683660	2117653	266243	3112980	3594526	38933	4208667	4605982	5115940	5570599	653322
1015815	1253387	1687641	2117654	2664452	311383	3596304	38934	4208668	4605983	5117979	5570600	655374
1017871	1255431	1690021	2117655	2666522	3115028	3596307	38935	4208669	4605984	5117985	5570601	655375
1019928	1257500	1691661	2117656	2670594	3121190	3596308	38936	4208670	4605985	512022	5570602	655376
1021956	1259549	1693791	2117657	2672645	3123236	3596310	38937	4212754	4608010	5122054	5570603	655377
1024017	1261579	1695766	2117658	2674691	3125260	3596311	38938	4214818	460803	5124158	5570604	655378
10241	1263627	1695767	2119689	26757	3129363	3596312	38939	4216874	4610052	5126164	5570605	655379
102410	1267723	1699866	2121756	2676758	3131429	3596313	38940	4218882	4612271	5128216	5570606	655380
10242	1269765	170220	2123790	26771	313386	3596316	38941	421900	4614149	5132291	5570607	655381
10243	127003	1703942	2125828	26772	3135502	3596317	38942	4220932	4618266	5134342	5570608	655382
10245	1271827	1706052	2127878	2678786	3137581	3596318	38943	4225037	4622341	514091	5570609	655383
10246	1273876	1710091	2129934	2680888	3139610	3596319	38944	4227087	4626478	5142543	5570610	655384
10247	1277980	1710092	213002	268316	3143729	3598512	38945	4227088	4628525	5148691	5570611	655385
10248	1280213	1712173	2131995	2684937	3145734	3600429	38946	4227089	4630558	5152781	5570612	655386
10249	1282233	1716235	2134035	2689026	3147914	3602449	38947	4227090	4634632	5154818	5570613	655387
10250	1284104	1718293	2138123	2691086	3149844	360477	38948	4227091	4636682	5156933	5570614	65540
10251	1288216	1720337	2140183	2695175	315401	3606542	38949	4227092	4640807	5158960	5570615	657413
10252	1290262	1720338	2142248	2701362	315404	3608579	38950	4227093	4642842	5160992	5570616	659479
10253	129094	1720339	2146306	270381	315405	3612726	38951	4227094	4644871	516122	5570617	663570
10254	1292345	1720340	2148414	270384	3155977	3614743	38952	4227095	4646929	5163015	5570618	665612
10255	1294345	1720341	215051	270385	3162125	3616803	38953	4227096	4648973	5165065	5570619	667656
10256	1296475	1720342	2150551	270386	3168314	3618818	3895300	4227097	464899	5169155	5570620	669710
10257	1298508	1720343	2152676	270387	3172382	3620984	38954	4227098	4651028	5171244	5570621	673815
10258	1300589	1720344	2154503	2707463	3174410	3622925	38955	4229435	4653067	5173252	5570622	675852
10259	1302534	1720345	2160672	2709509	317444	362517	38956	4235291	4655126	5175323	5570623	679945
10260	1304701	1720346	2162697	2711568	317445	3627058	38957	4239369	4655127	5177392	5570624	684048
1026069	1306660	172041	2168857	2713629	317446	3629073	38958	4241433	4655128	5179517	5570625	686082
10261	1308750	1722391	2170896	2715653	317447	3631130	38959	4241434	4657156	5181459	5570626	688135
10262	1310725	1724424	217096	2717700	317448	3637261	38960	4241435	4663306	518179	5570627	690187
10263	131087	1726483	2172936	2719751	317449	3643431	38961	4241436	4665373	5183490	5570628	692233
10264	131091	1728526	2174983	2721882	317450	3645462	3898057	4241437	4667436	5185594	5570629	694287
10265	131093	1728538	2177089	272425	3176452	364563	3899416	4241438	4669454	5187611	5570630	696348

Figure 51: RH-DELETED contents part 1




root@little:~ - Shell - Konsole <4>													
Session Edit View Bookmarks Settings Help													
													
10266	131094	1730573	2179092	2725891	3178506	3647505	3901462	4241439	466952	5189637	5570631	700457	
10267	131095	1732790	2187272	2727939	3180563	3647506	3903494	4243464	4671536	5191691	5570632	702479	
10268	131096	1734662	2191374	2734083	3182734	3647507	3905907	4245560	4673570	5193797	5570633	704521	
10269	131097	1736712	2193478	2736147	3184646	3649544	3907597	4247564	4675611	5195805	5570634	706577	
10270	131098	1738765	2195468	2738192	3186696	3651592	3911689	4249606	4679694	5197839	5570635	708691	
10271	131099	1740807	2197561	2740227	3188759	3653646	3913735	4251651	4681743	5199909	5570636	710708	
10272	131100	1744097	2203656	2742275	3190803	3653655	3915779	4253709	4685827	5201945	5570637	710711	
10273	131101	1744911	2205709	2744329	3192918	3653656	3921924	4255764	4688046	520212	5570638	712717	
10274	131102	1744912	2207775	274460	3194952	3653657	3924021	4257825	4689925	5204004	5570639	714755	
10275	131103	1744913	2209802	2746371	3196991	3653658	3924022	4259999	4696068	5206053	5570640	716806	
10276	131104	1744914	2211875	2750476	3199037	3657758	3924023	426062	4698118	5208103	5570641	716808	
10277	131105	1744915	221373	2752515	3203079	3657759	3924024	4262249	4702224	5210150	5570642	71683	
10278	131106	1744916	2213895	2754567	3205505	3657760	3924025	4269344	4704269	5212163	5570643	720915	
10279	131107	1744917	2215938	2756611	3207303	3657761	3924026	4270136	4710420	5216262	5570644	724999	
10280	131108	1744918	2217995	2758660	3209220	3657762	3924027	4272141	471237	5220355	5570645	725002	
10281	131109	1744919	2220045	2760710	3211275	3657763	3924028	4276226	4712462	5222489	5570646	725003	
1028102	1312792	1744920	2224147	2762807	3215368	3657764	3924029	4278293	4714500	5224614	5570647	725004	
10282	1314824	1744923	2230287	276486	321582	3657765	3924030	4280336	4716554	5226583	5570648	725005	
10283	1316985	1744924	2232382	2766850	3217434	3657766	3924031	4280339	4718597	5229061	5570649	725023	
10284	1318914	1744925	223240	2768899	3219465	3657767	3924032	4280340	4720754	5236752	5570650	727046	
10285	1321016	1744926	2234391	2770948	3221510	3657768	3924033	4280341	4722722	5238831	5570651	729154	
10286	1323045	1744927	2236451	2772995	3223578	3657769	3924034	4280342	4726824	5249044	5572637	731143	
10287	1325183	1744928	2244614	2775066	3225663	3659784	3924035	4280343	473099	5249045	5574715	733197	
10290	1327128	1744929	2246659	2777138	3227698	3663875	3924036	4280344	4732937	5249046	5576756	737288	
10291	1329211	1744930	2248707	2779139	3229722	3665940	3924037	4280345	4734989	5249047	5578803	737291	
10292	1331272	1744931	2250754	2781187	3231780	3667980	3924038	4280346	4737028	5249048	5578804	737292	
10293	1333264	1744932	2252807	2783235	3233931	3670040	3924039	428043	4739092	5249049	5578805	737293	
10294	1335384	1749012	225314	2785283	323589	3672072	3924040	4282413	4741126	5251148	5578806	737294	
10297	1337417	1751060	2254858	278537	3235903	3674132	3924041	4282415	4743180	5255187	5578807	737295	
10298	1339412	1753121	2256906	2787366	3239940	3676194	3924042	4284436	4745250	5255188	5580948	737296	
10299	1341543	1755142	2258950	2789390	3244069	3678223	3924043	4286475	4749332	5255189	5582896	737297	
10300	1343494	1757223	2263093	2791443	3250207	3682399	3924044	4288721	4751387	5255190	5586964	737298	
10301	1345651	1759240	2265191	2793475	3254276	3684385	3924045	4292709	475143	5255191	5589107	737299	
1030165	1347635	1761327	2269195	2795535	325671	3686405	3924046	4294663	4753413	5255192	559107	737300	
10304	1349724	176135	2273304	2799623	3258381	368654	3924047	4296736	4757533	5257222	5591098	737301	
10305	135185	1763411	227333	2801667	3260450	36876	3924048	4298764	4759570	5259311	5599329	737302	
10306	1353818	1765463	227336	2803715	3260485	36878	3924049	4300812	4761613	5261368	5601300	737303	
10307	1357920	1767437	227337	280582	3262477	36879	3924050	4304983	4763653	5263374	5603379	737304	
10308	1362001	1769518	227338	2805863	3264517	36880	3924051	4309016	4765716	526339	5603382	737305	

Figure 52: RH-DELETED contents part 2

root@little:~ - Shell - Konsole <4>												
Session Edit View Bookmarks Settings Help												
10309	1364233	1771528	227345	2807811	3266571	36881	3924052	4311053	4767757	5265421	5603383	737306
10310	1366115	1773678	227346	2811919	3270708	36882	3924053	4313095	4769826	5267544	5605518	737307
10311	1368539	17745	2275337	2818051	3272772	36883	3924054	4315157	4771847	5275655	5607438	737308
10312	1370181	1777725	2277379	2822147	3274758	36884	3924055	4317208	477195	5279761	5609481	737309
10313	137224	1779738	2287666	282640	3274760	3688475	3924056	4319252	4773898	5281844	561178	73731
10314	1374326	1779739	2293775	282642	3274761	36885	3930116	432179	4775940	5283920	5615648	737310
10315	1376303	1779740	229435	2828291	3274762	36886	393281	4323502	4777997	528393	5617700	739338
10316	1378408	1779741	2295818	2830339	3274764	36887	3936286	4325385	4780034	5285929	5621791	739339
10317	1380377	1779742	2297868	2832387	3274765	36888	3938318	4327434	4782084	5287950	5625951	739340
10318	1382477	1779743	2299937	2834435	3276812	36889	3940364	4329514	4784387	5289999	5628000	739341
10319	1384504	1779744	2301970	2836483	327698	36890	3942465	4331550	4788266	5292058	5629963	739342
10320	1386579	1779745	2304070	2840609	327701	36891	3944697	4333607	4790292	5294082	5632102	739343
10321	1388573	1779746	2306102	2842627	327702	36892	3948815	4335640	479245	5296132	5636326	739344
10322	1390642	1779747	2308159	2844673	327703	36893	3952695	4337670	4794402	5298181	5642397	739345
1032207	1392722	1781778	2310366	2846723	327704	36894	395273	4339721	4796449	5300227	5644303	739346
1032208	139278	178181	2314300	284749	327705	36895	3954728	4341773	4798479	5302352	5648489	741399
1032209	1396811	1785917	231477	2848813	327706	36896	3956803	434288	4802577	5304324	5648492	741402
1032210	1400865	1787992	2316301	2850819	327707	36897	3958804	4343820	4804613	530435	5648493	741403
1032211	1402960	1792029	2318399	2852877	327712	36898	3964951	4345862	4806672	5306524	5648494	745488
1034245	1404994	1792106	2320405	2854915	327713	36899	3971091	4347918	4808709	5308461	5648495	749582
1036342	1406998	1794076	2326546	2859011	327714	36900	3973136	4352087	4810769	5310468	5648496	751619
1038480	1409057	1794113	2328585	2861059	327715	36901	3973139	4354072	4812824	5312540	5648497	753746
1042458	1411126	1796124	2330677	2865186	327716	36902	3973140	4356105	4814877	5314584	5648498	755722
1044491	141328	1796185	2332684	2867205	327717	36903	3973141	4358210	4818952	5320735	5648499	757786
104455	1415196	1798178	2336786	286757	327718	36904	3973142	4360243	4821018	5322758	5652489	75784
1046594	1417246	1798246	2338818	286760	32784	36905	3973143	4362351	4823067	5324819	565281	763909
1052682	1421340	1800220	2342966	286764	3278852	3690514	3973144	436270	4825119	532488	5654588	770053
1056787	1421341	1800346	2347143	286765	3278854	36906	3973145	4364314	4827146	53260	5656585	772156
1061099	1421342	1802274	2349077	286766	3280906	36907	3973146	4366393	4829189	5326855	5660751	772159
1064966	1421343	180247	2353164	286767	3283038	36908	3973147	4368400	4833283	5328929	5662727	772160
106503	1421344	1802947	2355377	286768	3293206	36909	397323	4370445	483334	5330970	5664783	774176
1067014	1421345	1804317	235569	286769	3295260	36910	397324	4372495	4835610	5333002	5666828	778255
1071177	1421346	1804349	2357320	286770	3297305	36911	397325	4374543	4835613	5335119	5668904	77890
1073159	1421347	1806344	2359301	286771	329732	36912	397326	4376585	4835614	5337099	5670920	780310
1075211	1423402	1810452	2361377	286772	3301401	36913	397327	4378636	4835615	5343236	5672990	788495
1077271	1425492	1812527	2363459	2869286	3309580	36914	397328	4380684	4839460	5345301	567300	792582
1077274	1431647	1816578	2365457	28703	3311620	36915	397329	4382732	4841497	534540	5677063	796676
1077275	1433636	1818696	2367507	28704	3315767	36918	397330	438289	4841498	5347333	5677068	798732
1077277	1433637	1822725	2369571	2871301	3317792	36925	397331	4384810	4841499	5349379	5677069	79875

Figure 53: RH-DELETED contents part 3

root@little:~ - Shell - Konsole <4>													
Session Edit View Bookmarks Settings Help													
1077277	1433637	1822725	2369571	2871301	3317792	36925	397331	4384810	4841499	5349379	5677069	79875	
1077278	1433638	182295	2371612	2877444	331905	3692551	397332	4386846	4841500	5351430	5679116	801176	
1077279	1433639	1824778	2375740	2879491	3319815	36927	397333	4388885	4841501	5355541	5685283	805272	
1077280	1433640	1826836	237599	2881539	3323914	36929	397334	4390939	4841502	5357599	5687476	806922	
1077281	1433641	1828871	2377751	2883623	3325960	36930	397335	4392983	4841503	5359620	5695730	811021	
1077282	143377	1830951	2379797	288785	3328002	36937	397336	4392984	4841504	5365825	569595	811023	
1077283	1435698	1832996	2384128	288788	3330112	36938	397337	4392985	4841505	5365827	5697603	811024	
1077284	1437778	1835038	2387968	288789	3338301	36939	397338	4392986	4841506	5365828	5701643	813089	
1077285	1441819	1837082	2392087	2889731	333858	3694595	3975189	4392987	4841507	536614	5703712	813092	
1077286	1441820	1843240	2394335	2891779	3340315	3696675	3977223	4392988	485380	5367814	5705739	813093	
1077287	1441821	18443	2396191	2891783	3342352	3698698	3983385	4392989	4855845	5369902	5707819	815116	
1079324	1441822	1845284	239626	2891784	3346444	3700743	3985423	4395023	4857870	5369904	5709826	815119	
1081357	1441830	1847321	239631	2893868	3348537	3702811	3991673	4397062	4862009	5369905	5711888	815120	
1081358	1441833	1849349	239632	2895875	3352581	3702814	3993620	4399116	4862012	5371907	571415	819214	
1081373	1441835	1851406	239633	2897932	3356675	3702815	399421	4401167	4862013	5371908	5720131	819220	
1081374	1441837	1853500	2398243	2902018	335876	3702816	3995682	4403238	4862014	5371918	5724248	81923	
1081375	1441839	1855503	2400268	2908199	3360778	3704855	3997740	440399	4862015	5373965	5728266	821258	
1081376	1441840	1859591	2402307	2908200	3362836	3706900	3999781	440402	4862016	5376027	5730316	821260	
1081377	1441843	1861653	2404389	2908201	3364888	370699	4001798	440403	4862017	5378061	5732387	823329	
1081378	1441846	1863699	2408459	2908202	3366996	3710985	4005892	440404	4862018	5378063	5734439	825355	
1081379	1441849	186386	2410505	2908203	3369001	3719176	4005893	440405	4862019	5380146	573450	827397	
1081381	1441850	1865739	2412569	2908204	3371012	3721233	4007949	440406	4862020	5382181	573453	827400	
1081382	1441851	1867808	2414599	290863	3373099	3723279	4009990	4405309	4862021	5384219	573454	827409	
1081383	1441852	1869829	2416644	290864	3375120	3725335	4012075	4407459	4864076	538744	573455	827410	
1083443	1441853	1880072	241690	290865	3377165	3729468	4014107	4409362	4866071	5388306	573456	827411	
1085480	14432	1882117	2418711	290866	3379273	373025	401433	4413474	4870159	5390340	573457	827412	
108553	1443845	1882118	2420756	290867	3385367	3731495	401438	4415501	4874318	5392477	573458	827413	
1087569	1445911	1882119	2422798	290868	3387396	3737612	401439	4417583	487475	5392480	573459	827414	
1089561	1450050	1882120	2424835	290869	3389457	3739662	401440	4417585	4876290	5392481	57346	827415	
1091605	1452107	1882121	2424878	290870	3391493	3741699	401441	4417586	4878421	5392482	573460	827416	
1093637	145414	1882122	2424879	290871	3393557	3743747	401442	4417587	4880404	5392483	573461	829445	
1095754	1456179	1882123	2424880	290872	3395634	3745817	401443	4417588	4882481	5392484	573462	833572	
1099782	1458192	1882124	2433138	290873	3397638	3747852	401444	4417589	4884482	5392485	573463	837637	
1101865	1460235	1882125	2435080	290874	3399708	374804	4016148	4417590	4886535	5398551	573464	839695	
1103902	1462301	1882126	2437128	2912259	3401768	374807	4018206	4417591	4888581	5398552	573465	83975	
1105927	1466422	1882127	2439176	2914307	3403792	374808	4020258	4417592	4890641	5398553	573466	849937	
110602	1468447	1882128	2441223	2916369	3405832	3749908	4022278	4417593	489480	5398554	573467	849938	
1107974	1470586	1882129	2443392	2918403	3409923	3751942	4024324	4419609	4896774	5398555	573468	849943	
1112073	1474604	1882130	2449408	2920451	3411971	3753993	4026421	4423686	4898856	5398556	573469	849944	

Figure 54: RH-DELETED contents part 4




root@little:~ - Shell - Konsole <4>															
Session Edit View Bookmarks Settings Help															
															
1114223	147510	1882131	2449435	2922530	3416106	3756039	4028422	4425748	4900866	5398558	57347	849945			
1116207	1476649	1882132	2451464	2924617	3418277	3758114	4030496	4429856	4902926	5398559	5736545	849946			
1118282	1480754	1882133	2453633	2926598	342020	3760152	4030497	4431892	4904975	5398560	5738513	849947			
1122315	1482834	1884163	2455606	2928645	3422246	3762223	4030498	4433924	4907032	5398561	5740553	849948			
1124356	1486887	188424	245765	292868	3424272	3764261	4030499	4438046	4909109	5398562	5746693	849949			
112643	1488951	1886224	2459695	2930715	3426357	3766303	4030500	4442115	4911160	5398563	5748741	849950			
1126432	1490978	1888313	24601	2932767	3432516	376842	4030501	4444178	49166	5398564	5750797	849951			
1128454	1495106	1890359	24603	2934802	3434510	3770407	4030502	444465	4921393	5400602	5754890	849952			
1130517	149530	1892391	24604	2936841	3436586	3772429	4032541	4446211	4925682	540687	575496	851981			
1132619	1497135	1894456	24605	2938885	3438614	3774484	4034573	4448269	4935693	5408800	5756933	854021			
1134603	1499187	1900552	2461739	2943006	3440648	3776546	403477	4450340	4937845	5408802	5759013	856124			
1136690	1501244	1902610	2463751	2945045	344096	3776549	4036638	4454405	4939788	5411008	577585	858143			
1138694	1505307	1904653	2465794	2947131	3442704	3778628	4038664	4456501	4941848	5413795	579609	86020			
1138695	1507383	190475	2467882	2949123	3442705	3778631	4040751	4460551	4943989	542781	583755	860204			
1138708	1509418	1910787	2469893	294936	3442706	3778632	4044811	4462599	4945941	5435472	585759	862234			
1138709	1515552	1912860	2471969	294939	3442707	3780649	4046861	4464664	4950033	544781	587927	864282			
1138710	151570	1916939	2478181	294940	3442708	3782674	4048925	446513	4952072	5468161	589873	866350			
1138711	1521682	1918979	247822	2951176	3444762	3786757	4050982	4466700	4954142	5468162	591900	866353			
1138712	1523776	1921030	2484229	2953221	3446808	3788833	4053022	4468808	4956170	5468163	59395	866354			
1138713	1525776	1925157	2486293	2955457	3452930	378900	4055053	4470804	495627	5468164	593955	868408			
1140761	1527827	192651	2488337	2959409	3457179	3794948	405512	4472845	4958211	5468165	595998	868411			
1143372	1529892	1927179	2490427	2961414	3459074	3797035	4057090	4474914	4960260	5468166	598048	868412			
1144837	1531911	1931281	2492434	2963490	3461122	3799058	4059149	4478983	4962309	5468167	600079	868413			
1146901	1533965	1933347	2494505	2965514	346123	3801101	4061213	4481035	4964356	5468168	602150	868414			
114739	1536015	1935430	2496531	2967557	3463170	3803139	4063271	4483087	4966465	546819	604182	868415			
114742	153607	1935431	249938	2969625	3465218	3807245	4065310	4485170	4970500	5470209	606215	868416			
114743	1538058	1939479	249941	297008	3467311	3809293	4067341	448525	4976643	5470210	610321	868417			
1150986	1542148	1945625	249942	2973730	3469314	380938	4071440	4487172	497668	5470211	612380	868418			
1150987	1544196	1945626	2502769	2977877	3473410	3815427	4073492	4489229	4978699	5470212	61443	868419			
1153027	1550359	194571	2504735	2979875	3475458	3819540	4075532	4491275	4980741	5472259	614431	868420			
1155080	1552399	1947655	2506787	2983941	3477506	3821602	407563	4493322	4982815	5474316	6146	868421			
1155083	1554509	1951762	2508817	2986031	3479578	3823624	4077586	4495578	4984839	5476388	6147	868422			
1155084	1556496	1953820	2510901	2988047	3481602	3829770	4081677	4497422	4987077	5480474	6148	870407			
1157129	155655	1955926	2514947	299106	348180	382989	4083736	4501524	4987080	5482622	6149	872460			
1161222	1558788	1957924	2519043	2992192	3483659	3831813	4085794	4503769	4987081	5482624	6150	874587			
1163279	1560610	1959957	2521094	2994227	34851	3833873	4087811	4505634	4988965	5482625	6153	876550			
1163282	1564831	1961990	2523139	3000352	3485698	3833874	4089881	450571	4988968	5486607	6154	880666			
1163283	1564832	1964070	2525198	3004462	34858	3833875	4093994	4507651	4988969	5488642	6155	88067			
1165354	1564838	1966098	2527235	3006521	34861	3833876	4093997	4509735	4991027	548868	6156	884747			

Figure 55: RH-DELETED contents part 5



root@little:~ - Shell - Konsole <4>												
Session Edit View Bookmarks Settings Help												
116747	1566743	196619	2529284	3008589	3487746	3833877	4093998	4511776	499722	5490697	6157	886831
1167600	1568770	1968148	2531331	3010566	3489794	3833879	4093999	4513795	4999172	5492754	6158	890886
1169439	1570906	1970207	2533437	301096	3491842	3833880	4094000	4515867	5001230	5492756	6159	892939
1171469	1575086	1972244	2535427	3012613	3493890	3833881	4094001	4517997	5003271	5496841	6160	894988
1173671	1576976	1974278	2537529	3018758	3495938	3833882	4094002	4519953	5005319	5500939	6161	894991
1175563	157743	1976823	2539523	3020828	3497986	3833883	4094003	4521987	5007374	5503056	6162	897031
1179659	1579475	1982470	253966	3022853	3500034	3833884	4094004	4524082	5009415	5505099	6163	901131
1181706	1581103	1984520	253967	3024912	3502082	3833885	4094005	4526083	5011463	5507079	6164	90115
1181707	1583231	198667	253968	3026955	350212	3833886	4096012	452615	5013510	5509167	616468	907269
1181708	1585174	1990710	253969	3028996	3504130	3833887	409611	4528141	5015589	550927	6165	911366
1181709	1587259	1992736	253970	303130	3506185	3833888	40967	4530179	5017639	5511175	6166	913419
1181710	1589337	1994766	253971	3035144	3508226	3833889	4098093	4532247	501773	5515270	6167	915462
1181711	1591381	1998874	253972	3039237	3510274	3840008	4100109	4536340	5021730	5521471	6168	917510
1181712	1595408	20	253973	3041322	3512347	3842052	4102146	4538384	5023761	5527663	6169	919558
1181713	1597890	2002969	253974	3043333	3514371	3844111	4104194	4540440	5023762	5527665	6170	92164
1181714	159791	2005068	253975	3047428	3516418	3846157	4106242	4542479	5023763	5527666	6171	923749
1181715	1599739	2011142	253976	3049560	3518535	3848202	4108290	4544527	5023764	5527667	6172	927750
1181716	1601675	2013211	2541574	3051542	3520514	3850268	4110354	454665	5023765	5527668	6173	929798
1181717	1603637	2017288	2543619	305163	3522607	385060	4112404	4547167	5023766	5527669	6174	929801
1183776	1607751	2019353	2545706	305166	352337	3852309	4114469	4548626	5023767	5527670	6175	929802
1185811	1609946	2021378	2547715	305167	352340	3854340	4116522	4548627	5023768	5527671	6176	931845
1187847	1613851	2027565	2551811	3053574	352341	3856408	411660	4548628	5023771	5527672	6177	938003
1187850	1616046	202765	2553899	3063821	352342	3856409	411663	4548629	5023772	5527673	6178	940037
118792	161798	2029592	2555907	3065874	352343	3856410	411664	4550669	5023773	5527674	6179	94214
1191947	1619981	2029593	2557962	3067993	352344	3856411	411665	4552706	5023774	5527675	6180	944134
1193993	1624107	2031622	2560010	3069962	352345	3856413	411666	4554775	5023775	5529601	6181	946187
1196086	1626119	2033670	256011	3072018	352346	3856414	411667	4556816	5027844	5529638	6182	950284
1198194	1632308	2035718	2562051	307241	352347	3856415	411668	4560948	5027850	5529640	6183	952335
1200139	1634321	2037766	2564099	307244	352348	3856416	411669	4562948	5029900	5529641	6184	958618
1202182	1636551	2039814	2566154	307245	352349	3856417	411670	4564994	5032088	553017	6185	960799
1204238	1638430	2041862	2568195	307246	352350	3856418	411671	456725	5034015	553018	618560	96258
1206298	163853	2043910	2570250	307247	352351	3856419	411672	4569111	5036038	553019	6186	970763
1208328	1640480	2045958	2572319	307248	352352	3856420	411673	4571168	503822	553020	6187	972804
120850	1642528	2048006	2574346	307249	352353	3860490	411674	4575235	5040149	553021	6188	974886
1212425	1642529	204855	2576400	307250	3524610	3862538	411675	4577295	5044228	553022	6189	974887
1214582	1642530	2052102	2580485	307251	3526694	3866629	4118532	4579340	5046287	553023	6190	978946
1214585	1642531	2056198	2582869	307252	3528706	3870726	4120637	4581402	5054466	5537835	6191	978949
1214586	1642532	2058244	2584596	307253	3530754	387110	4122670	4583719	5056570	5537837	6192	978957
1218571	1642533	2060295	2586640	307254	3532952	3872798	4124690	4585501	5058595	5537838	6193	978960

Figure 56: RH-DELETED contents part 6

root@little: ~ - Shell - Konsole <4>

Session Edit View Bookmarks Settings Help

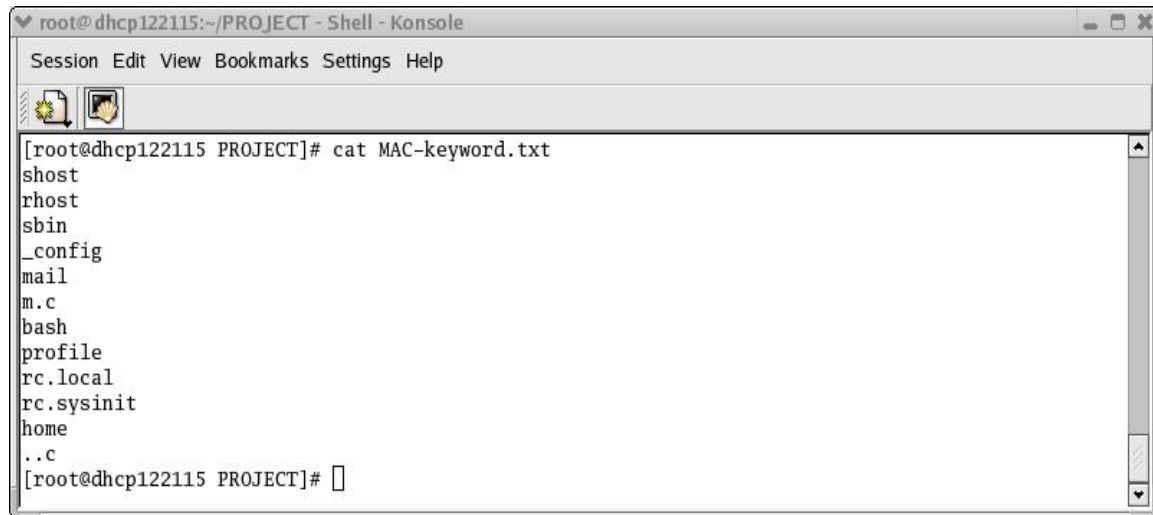
1200139	1634321	2037766	2564099	307244	352348	3856416	411669	4562948	5029900	5529641	6184	958618
1202182	1636551	2039814	2566154	307245	352349	3856417	411670	4564994	5032088	553017	6185	960799
1204238	1638430	2041862	2568195	307246	352350	3856418	411671	456725	5034015	553018	618560	96258
1206298	163853	2043910	2570250	307247	352351	3856419	411672	4569111	5036038	553019	6186	970763
1208328	1640480	2045958	2572319	307248	352352	3856420	411673	4571168	503822	553020	6187	972804
120850	1642528	2048006	2574346	307249	352353	3860490	411674	4575235	5040149	553021	6188	974886
1212425	1642529	204855	2576400	307250	3524610	3862538	411675	4577295	5044228	553022	6189	974887
1214582	1642530	2052102	2580485	307251	3526694	3866629	4118532	4579340	5046287	553023	6190	978946
1214585	1642531	2056198	2582869	307252	3528706	3870726	4120637	4581402	5054466	5537835	6191	978949
1214586	1642532	2058244	2584596	307253	3530754	387110	4122670	4583719	5056570	5537837	6192	978957
1218571	1642533	2060295	2586640	307254	3532952	3872798	4124690	4585501	5058595	5537838	6193	978960
1222733	1642534	2062344	2588682	307255	3534885	3874830	4128802	4585502	505869	5537839	6194	978961
1224715	1642535	2064561	2592797	307256	3534886	3876874	4132872	4587525	5060652	5539845	6195	978962
1226766	1642536	2066467	2594826	307257	3534887	3878925	413707	458768	5064711	5543943	6196	978963
1226768	1642537	2068547	2600970	307258	3534888	3883011	4139041	4597775	5066788	5546034	6197	978965
1226769	1644621	2070648	260099	307259	3534889	3885069	4141069	4597778	5068815	5546036	6198	978966
1226770	1646603	2072667	2605063	307260	3536898	3887120	4147235	4597779	5070935	5546037	6199	978967
1226775	1652740	2078805	2607111	307261	3538946	3889155	4149696	4597780	5072925	5546038	6200	978968
1226776	1654804	2080814	2609163	307262	3540994	3891203	415775	4597781	5074998	5548041	6201	978970
1226777	1656835	2089008	2611222	30747	3543042	389138	4163586	4597782	507915	5550086	6202	981007
1226778	1658886	208958	2613258	30749	354379	389139	4167700	4597783	5083138	5550088	6203	983057
1226779	165897	2091113	2615307	3076134	3545090	38914	4169730	4597784	5085261	5550089	6204	98306
1226780	165899	2095182	2619397	3078205	3547160	389140	4171783	4597785	5085262	5550090	6205	985097
1226781	165901	2099226	262150	3080198	3549210	389141	4173958	4597786	5087258	555124	620578	987177
1226783	165902	2101294	2623547	3082250	3551235	389142	4175878	4597789	5089302	5554185	6206	989188
1226784	165903	2103425	2625561	3084308	3553282	38915	4177950	4597790	5091346	5556279	624673	991246
1226786	165904	2105656	2627601	3084309	3555379	38916	417800	4597791	5093388	5560327	626727	993299
1226787	165905	2107479	2629654	3084310	3557418	38917	4179987	4597792	5095449	5564486	628795	995332
1226788	165906	2109556	2631683	3084311	3559427	38918	4182028	4597793	509973	5568550	630819	997388
1228811	165907	2111574	2633731	3084312	3563522	38919	4184071	4597794	5101579	5570585	632836	999463
122892	165908	2113563	2637826	3084313	356449	38920	4186121	4599837	5103623	5570586	634908	filetyp
1230881	165909	2115600	2639886	3084314	3565570	38921	4188171	4599838	5107732	5570587	634911	
1232907	165910	2117640	2641927	3084315	3567618	38922	4192265	4599839	5109769	5570588	634912	
1234997	165911	2117641	264239	3084316	3569666	38923	4194337	4599840	5111845	5570589	63492	
1237003	165912	2117642	264242	3088433	3573762	38924	4196364	4599841	5113859	557059	636957	
1239051	1660931	2117643	264243	3090441	3575810	38925	419902	4599842	5115931	5570590	641052	
1241198	1663014	2117644	2643989	3092487	3577931	38926	4200460	4599843	5115932	5570591	643076	
1241201	1667373	2117645	2646023	309281	3580421	38927	4200463	4599844	5115933	5570592	645169	

[root@little root]#

Figure 57: RH-DELETED contents part 7

## APPENDIX C - Examining MACread's results

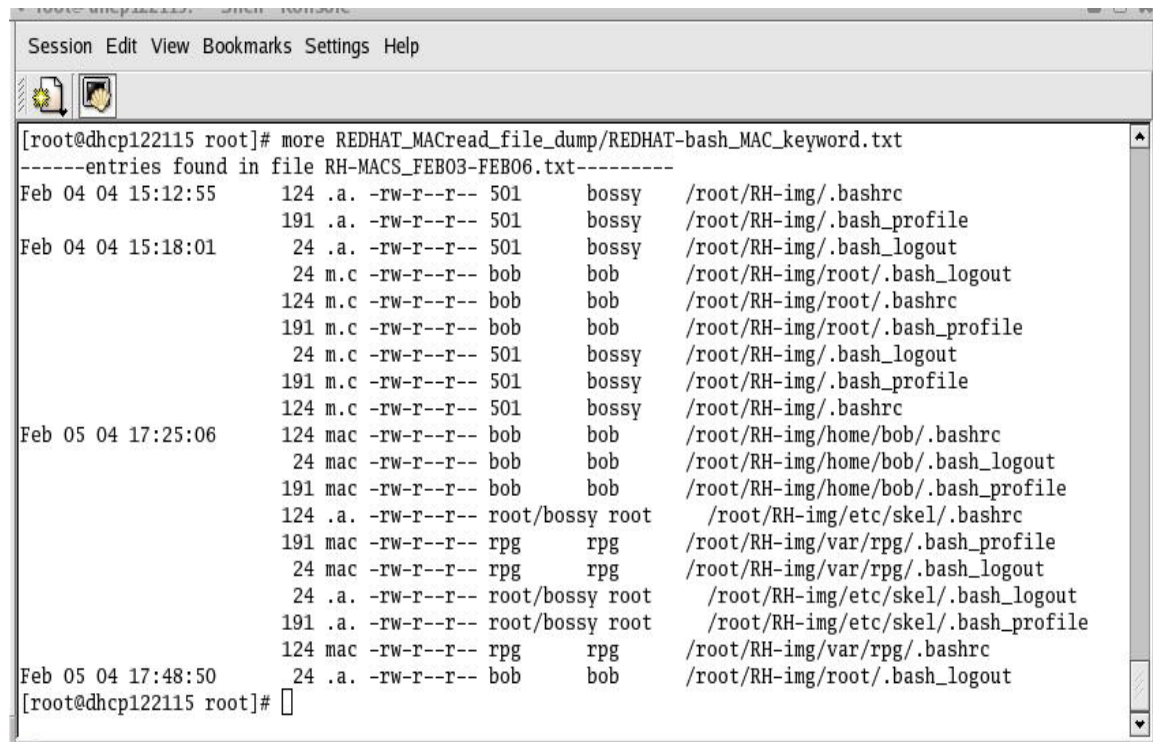
This appendix provides more evidence of the compromise gathered from the keyword files created by *MACread*.



```
root@dhcp122115:~/PROJECT - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 PROJECT]# cat MAC-keyword.txt
shost
rhost
sbin
_config
mail
m.c
bash
profile
rc.local
rc.sysinit
home
..c
[root@dhcp122115 PROJECT]#
```

Figure 58: MAC-keyword.txt file used with MACread

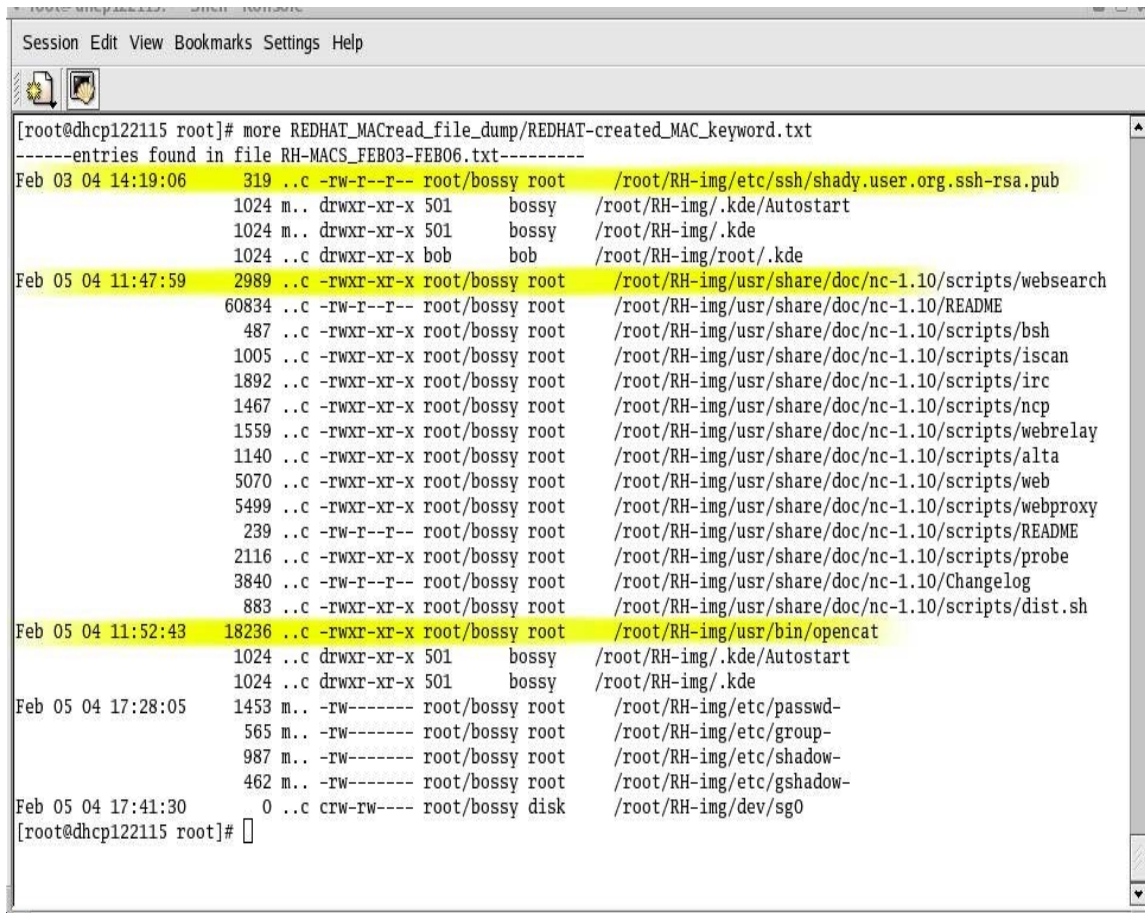


```
root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-bash_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 04 04 15:12:55      124 .a. -rw-r--r-- 501      bossy    /root/RH-img/.bashrc
                      191 .a. -rw-r--r-- 501      bossy    /root/RH-img/.bash_profile
Feb 04 04 15:18:01      24 .a. -rw-r--r-- 501      bossy    /root/RH-img/.bash_logout
                      24 m.c -rw-r--r-- bob      bob      /root/RH-img/root/.bash_logout
                      124 m.c -rw-r--r-- bob      bob      /root/RH-img/root/.bashrc
                      191 m.c -rw-r--r-- bob      bob      /root/RH-img/root/.bash_profile
                      24 m.c -rw-r--r-- 501      bossy    /root/RH-img/.bash_logout
                      191 m.c -rw-r--r-- 501      bossy    /root/RH-img/.bash_profile
                      124 m.c -rw-r--r-- 501      bossy    /root/RH-img/.bashrc
Feb 05 04 17:25:06      124 mac -rw-r--r-- bob      bob      /root/RH-img/home/bob/.bashrc
                      24 mac -rw-r--r-- bob      bob      /root/RH-img/home/bob/.bash_logout
                      191 mac -rw-r--r-- bob      bob      /root/RH-img/home/bob/.bash_profile
                      124 .a. -rw-r--r-- root/bossy root    /root/RH-img/etc/skel/.bashrc
                      191 mac -rw-r--r-- rpg      rpg      /root/RH-img/var/rpg/.bash_profile
                      24 mac -rw-r--r-- rpg      rpg      /root/RH-img/var/rpg/.bash_logout
                      24 .a. -rw-r--r-- root/bossy root    /root/RH-img/etc/skel/.bash_logout
                      191 .a. -rw-r--r-- root/bossy root    /root/RH-img/etc/skel/.bash_profile
                      124 mac -rw-r--r-- rpg      rpg      /root/RH-img/var/rpg/.bashrc
Feb 05 04 17:48:50      24 .a. -rw-r--r-- bob      bob      /root/RH-img/root/.bash_logout
[root@dhcp122115 root]#
```

Figure 59: Bash keyword results



Shows creations of .bash\_profile and .bashrc files for new users rpg, bob, and bossy.



```

[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-created_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 03 04 14:19:06    319 ..c -rw-r--r-- root/bossy root    /root/RH-img/etc/ssh/shady.user.org.ssh-rsa.pub
                    1024 m.. drwxr-xr-x 501    bossy    /root/RH-img/.kde/Autostart
                    1024 m.. drwxr-xr-x 501    bossy    /root/RH-img/.kde
                    1024 ..c drwxr-xr-x bob     bob      /root/RH-img/root/.kde
Feb 05 04 11:47:59   2989 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/websearch
                    60834 ..c -rw-r--r-- root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/README
                    487  ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/bsh
                    1005 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/iscan
                    1892 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/irc
                    1467 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/ncp
                    1559 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/webrelay
                    1140 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/alta
                    5070 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/web
                    5499 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/webproxy
                    239  ..c -rw-r--r-- root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/README
                    2116 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/probe
                    3840 ..c -rw-r--r-- root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/Changelog
                    883  ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/share/doc/nc-1.10/scripts/dist.sh
Feb 05 04 11:52:43   18236 ..c -rwxr-xr-x root/bossy root    /root/RH-img/usr/bin/opencat
                    1024 ..c drwxr-xr-x 501    bossy    /root/RH-img/.kde/Autostart
                    1024 ..c drwxr-xr-x 501    bossy    /root/RH-img/.kde
Feb 05 04 17:28:05   1453 m.. -rw----- root/bossy root    /root/RH-img/etc/passwd-
                    565 m.. -rw----- root/bossy root    /root/RH-img/etc/group-
                    987 m.. -rw----- root/bossy root    /root/RH-img/etc/shadow-
                    462 m.. -rw----- root/bossy root    /root/RH-img/etc/gshadow-
Feb 05 04 17:41:30    0  ..c crw-rw---- root/bossy disk    /root/RH-img/dev/sg0
[root@dhcp122115 root]#

```

**Figure 60: ..c keyword results**

- Shows creation of public key for ssh  
shady.user.org.ssh-rsa.pub
- Shows creation of documents for netcat  
/usr/share/doc/nc-1.10
- Shows creation of executable file netcat renamed  
/usr/bin/opencat

```
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-mail_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 03 04 13:18:58      0 mac -rw-rw---- 500      mail      /root/RH-img/var/spool/mail/bob
                      0 mac -rw-rw---- 501      mail      /root/RH-img/var/spool/mail/rpg
Feb 04 04 15:29:19      0 mac -rw-rw---- 501      mail      /root/RH-img/var/spool/mail/sdsd
                      0 mac -rw-rw---- bob      mail      /root/RH-img/var/spool/mail/bossy
14 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/Mail -> ../../bin/mail
27 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/mailq -> /etc/alternatives/mta-mailq
23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/purgestat -> ../../usr/sbin/sendmail
23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/etc/alternatives/mta-mailq -> /usr/bin/mailq.sendmail
dmail
23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/etc/alternatives/mta-rmail -> /usr/bin/rmail.sendmail
dmail
23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/mailq.sendmail -> ../../usr/sbin/sendmail
1
27 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/rmail -> /etc/alternatives/mta-rmail
23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/hoststat -> ../../usr/sbin/sendmail
1365 .a. -rw-r--r-- root/bossy root    /root/RH-img/root/.mailcap
291 .a. -rw-r--r-- root/bossy root    /root/RH-img/etc/mailcap

[root@dhcp122115 root]#
```

**Figure 61: Mail keyword results**

Shows new mail accounts being created for users:  
bob, rpg, sdsd, bossy

```

[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-m.c_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 03 04 13:18:58      0 mac -rw-rw---- 500   mail   /root/RH-img/var/spool/mail/bob
Feb 03 04 14:12:19    2473 m.c -rw----- root/bossy root   /root/RH-img/etc/ssh/sshd_config~
Feb 03 04 14:12:49     44 m.c -rw-r--r-- root/bossy root   /root/RH-img/root/.emacs.d/auto-save-list/.save-4384-
little.box.com~
Feb 03 04 14:24:36     19 m.c -rw-r--r-- root/bossy root   /root/RH-img/root/.shosts~
Feb 03 04 14:29:58     44 mac -rw-r--r-- root/bossy root   /root/RH-img/root/.emacs.d/auto-save-list/.save-4446-
little.box.com~
Feb 03 04 14:31:28    2471 m.c -rw----- root/bossy root   /root/RH-img/etc/ssh/sshd_config
1024 m.c drwxr-xr-x 500   500   /root/RH-img/usr/local/mysqlcc-0.9.3-linux-glibc23
152 mac -rwxr-xr-x root/bossy root   /root/RH-img/usr/local/mysqlcc-0.9.3-linux-glibc23/pa
ctice_auto_sql_open
      0 mac -rw-rw---- 501   mail   /root/RH-img/var/spool/mail/rpg
Feb 04 04 15:17:37    1024 m.c drwx----- root/bossy root   /root/RH-img/var/run/sudo
Feb 04 04 15:29:19      0 mac -rw-rw---- 501   mail   /root/RH-img/var/spool/mail/sdsd
Feb 04 04 18:00:08     381 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.kde/Autostart/.directory
847 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.emacs
1686 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.kde/Autostart/Autorun.desktop
120 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.gtkrc
24 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.bash_logout
      0 mac -rw-rw---- bob    mail   /root/RH-img/var/spool/mail/bossy
1024 m.c drwx----- bob    bob    /root/RH-img/root/.kde/Autostart
124 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.bashrc
191 m.c -rw-r--r-- bob    bob    /root/RH-img/root/.bash_profile
Feb 04 04 18:05:32     607 m.c -r--r----- root/bossy root   /root/RH-img/etc/sudoers
Feb 04 04 18:29:12    1046 m.c -rw----- root/bossy root   /root/RH-img/etc/#shadow#
Feb 04 04 18:30:29      5 m.c -rw----- root/bossy root   /root/RH-img/etc/passwd.lock~
Feb 04 04 18:39:59     841 m.c -rw-r--r-- root/bossy root   /root/RH-img/etc/profile
1024 m.c drwxr-xr-x root/bossy root   /root/RH-img/usr/share/doc/nc-1.10
24576 m.c -rw-r--r-- rpm    rpm    /root/RH-img/var/lib/rpm/Name
1024 m.c drwxr-xr-x root/bossy root   /root/RH-img/usr/share/doc/nc-1.10/scripts
31744 m.c drwxr-xr-x root/bossy root   /root/RH-img/usr/share/man/man1
9216 m.c drwxr-xr-x root/bossy root   /root/RH-img/usr/share/doc
22605824 m.c -rw-r--r-- rpm    rpm    /root/RH-img/var/lib/rpm/Packages
Feb 05 04 11:48:00   65536 mac -rw-r--r-- rpm    rpm    /root/RH-img/var/lib/rpm/Provideversion
--More-- (31%)

```

**Figure 62: m.c keyword results part 1**

Shows mail accounts being made for users:

bob, rpg ,bossy

Shows .shosts~ and /etc/sshd\_config files modification and creation

Shows rpm usage

Shows modification and creation of /etc/sudoers file

Session Edit View Bookmarks Settings Help						
Feb 05 04 11:48:00	65536	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Provideversion
	53248	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Shalheader
	114688	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Requireversion
	200704	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Requirename
	626688	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Dirnames
	12288	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Group
	45056	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Sigmd5
	311296	m.c	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Providename
	5300224	m.c	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Basenames
	5279744	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Filemd5s
	16384	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Installtid
	12288	mac	-rw-r--r--	rpm	rpm	/root/RH-img/var/lib/rpm/Triggername
Feb 05 04 11:59:28	36864	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/usr/bin
Feb 05 04 12:00:29	706	m.c	-rw-r--r--	root/bossy	root	/root/RH-img/usr/lib/mozilla-1.2.1/searchplugins/googl
e.src						
	1024	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/usr/lib/mozilla-1.2.1/searchplugins
	52028	m.c	-rw-----	root/bossy	root	/root/RH-img/tmp/lecture_1_17.pdf
Feb 05 04 15:12:25	52879	m.c	-rw-----	root/bossy	root	/root/RH-img/tmp/zhang.pdf
Feb 05 04 15:39:26	230246	m.c	-rw-----	root/bossy	root	/root/RH-img/tmp/lab7_lecture.pdf
ads.rdf	14764	m.c	-rw-r--r--	root/bossy	root	/root/RH-img/root/.mozilla/default/zjpr9pmu.slt/downlo
Feb 05 04 16:34:42	0	m.c	-rw-r--r--	root/bossy	root	/root/RH-img/etc/passwd.bak
Feb 05 04 17:03:50	1024	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/root/.emacs.d/auto-save-list
Feb 05 04 17:06:46	1024	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/etc/rc.d/rc2.d
	13	mac	lrwxrwxrwx	root/bossy	root	/root/RH-img/etc/rc.d/rc2.d/K20nfs -> ../init.d/nfs
Feb 05 04 17:10:24	13	m.c	lrwxrwxrwx	root/bossy	root	/root/RH-img/etc/rc.d/rc5.d/S60nfs -> ../init.d/nfs
	1024	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/etc/rc.d/rc3.d
	13	mac	lrwxrwxrwx	root/bossy	root	/root/RH-img/etc/rc.d/rc3.d/S60nfs -> ../init.d/nfs
	1024	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/etc/rc.d/rc5.d
	1024	m.c	drwxr-xr-x	root/bossy	root	/root/RH-img/etc/rc.d/rc4.d
	13	mac	lrwxrwxrwx	root/bossy	root	/root/RH-img/etc/rc.d/rc4.d/S60nfs -> ../init.d/nfs
Feb 05 04 17:18:28	67668	m.c	-rwxr-xr-x	root/bossy	root	/root/RH-img/bin/ls
Feb 05 04 17:24:27	381	m.c	-rw-r--r--	501	bossy	/root/RH-img/.kde/Autostart/.directory
	24	m.c	-rw-r--r--	501	bossy	/root/RH-img/.bash_logout
	847	m.c	-rw-r--r--	501	bossy	/root/RH-img/.emacs
	1686	m.c	-rw-r--r--	501	bossy	/root/RH-img/.kde/Autostart/Autorun.desktop

**Figure 63: m.c keyword results part 2**

Shows modification and creation in the /usr/bin directory :  
Directory where netcat was stored



```

root@dhcp122115: ~ - ssh - Konsole
Session Edit View Bookmarks Settings Help

Feb 05 04 17:25:06 124 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.bashrc
381 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.kde/Autostart/.directory
1024 m.c drwxr-xr-x bob bob /root/RH-img/home/bob/.kde/Autostart
24 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.bash_logout
120 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.gtkrc
1024 m.c drwx----- bob bob /root/RH-img/home/bob
1024 m.c drwxr-xr-x bob bob /root/RH-img/home/bob/.kde
1024 m.c drwxr-xr-x root/bossy root /root/RH-img/home
1686 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.kde/Autostart/Autorun.desktop
191 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.bash_profile
847 mac -rw-r--r-- bob bob /root/RH-img/home/bob/.emacs
Feb 05 04 17:28:51 381 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.kde/Autostart/.directory
1024 m.c drwxr-xr-x rpg rpg /root/RH-img/var/rpg/.kde/Autostart
6 mac -rw----- root/bossy root /root/RH-img/etc/shadow.lock
6 mac -rw----- root/bossy root /root/RH-img/etc/group.lock
1485 m.c -rw-r--r-- root/bossy root /root/RH-img/etc/passwd
6 mac -rw----- root/bossy root /root/RH-img/etc/gshadow.lock
191 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.bash_profile
6 mac -rw----- root/bossy root /root/RH-img/etc/passwd.lock
24 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.bash_logout
1024 m.c drwxr-xr-x root/bossy root /root/RH-img/var
1024 m.c drwx----- rpg rpg /root/RH-img/var/rpg
124 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.bashrc
120 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.gtkrc
847 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.emacs
1024 m.c drwxr-xr-x rpg rpg /root/RH-img/var/rpg/.kde
575 m.c -rw-r--r-- root/bossy root /root/RH-img/etc/group
1686 mac -rw-r--r-- rpg rpg /root/RH-img/var/rpg/.kde/Autostart/Autorun.desktop
470 mac -r----- root/bossy root /root/RH-img/etc/gshadow
Feb 05 04 17:29:08 1045 m.c -r----- root/bossy root /root/RH-img/etc/shadow
Feb 05 04 17:37:06 1024 mac drwxr-xr-x root/bossy root /root/RH-img/root/tmp
1024 m.c drwx----- root/bossy root /root/RH-img/var/run/sudo/rpg
Feb 05 04 17:41:31 67332 m.c -rw-r--r-- root/bossy root /root/RH-img/var/log/ksyms.1
862 m.c -rw-r--r-- root/bossy root /root/RH-img/var/log/gdm/:0.log.1
Feb 05 04 17:42:41 0 mac -rwxr-xr-x root/bossy root /root/RH-img/tmp/orbit-root/linc-cce-0-21826b1cc359d
Feb 05 04 17:42:44 0 mac -rwxr-xr-x root/bossy root /root/RH-img/tmp/orbit-root/linc-cd1-0-59364d8d1bb97
[root@dhcp122115 root]#

```

**Figure 64: m.c keyword results part 3**

Shows modification and creation in /etc/passwd file and /etc/group  
 Shows modification and creation in /var/run/sudo/rpg directory



```
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more REDHAT_MACread_file_dump/REDHAT-sbin_MAC_keyword.txt
-----entries found in file RH-MACS_FEB03-FEB06.txt-----
Feb 03 04 15:06:53      7 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/sbin/adduser -> useradd
Feb 04 04 18:00:30    55960 .a. -rwxr-xr-x root/bossy root    /root/RH-img/usr/sbin/usermod
      11 .a. lrwxrwxrwx root/bossy root    /root/RH-img/etc/rmt -> ../sbin/rmt
Feb 04 04 18:38:53    55768 .a. -rwxr-xr-x root/bossy root    /root/RH-img/usr/sbin/visudo
      22 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/makemap -> ../../usr/sbin/makemap
      14 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/disable -> ../sbin/accept
      23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/purgestat -> ../../usr/sbin/sendmail
ail
      14 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/rdistd -> ../sbin/rdistd
      23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/mailq.sendmail -> ../../usr/sbin/sendmail
sendmail
      14 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/enable -> ../sbin/accept
      23 .a. lrwxrwxrwx root/bossy root    /root/RH-img/usr/bin/hoststat -> ../../usr/sbin/sendmail
il
Feb 05 04 17:02:04    9896 .a. -rwxr-xr-x root/bossy root    /root/RH-img/usr/sbin/showmount
      33108 .a. -rwxr-xr-x root/bossy root    /root/RH-img/sbin/chkconfig
Feb 05 04 17:16:48    1819 .a. -rwxr-xr-x root/bossy root    /root/RH-img/sbin/service
      36056 .a. -rwxr-xr-x root/bossy root    /root/RH-img/usr/sbin/userdel
Feb 05 04 17:28:12    20280 .a. -rwxr-xr-x root/bossy root    /root/RH-img/usr/sbin/groupdel
      56088 .a. -rwxr-xr-x root/bossy root    /root/RH-img/usr/sbin/useradd
[root@dhcp122115 root]#
```

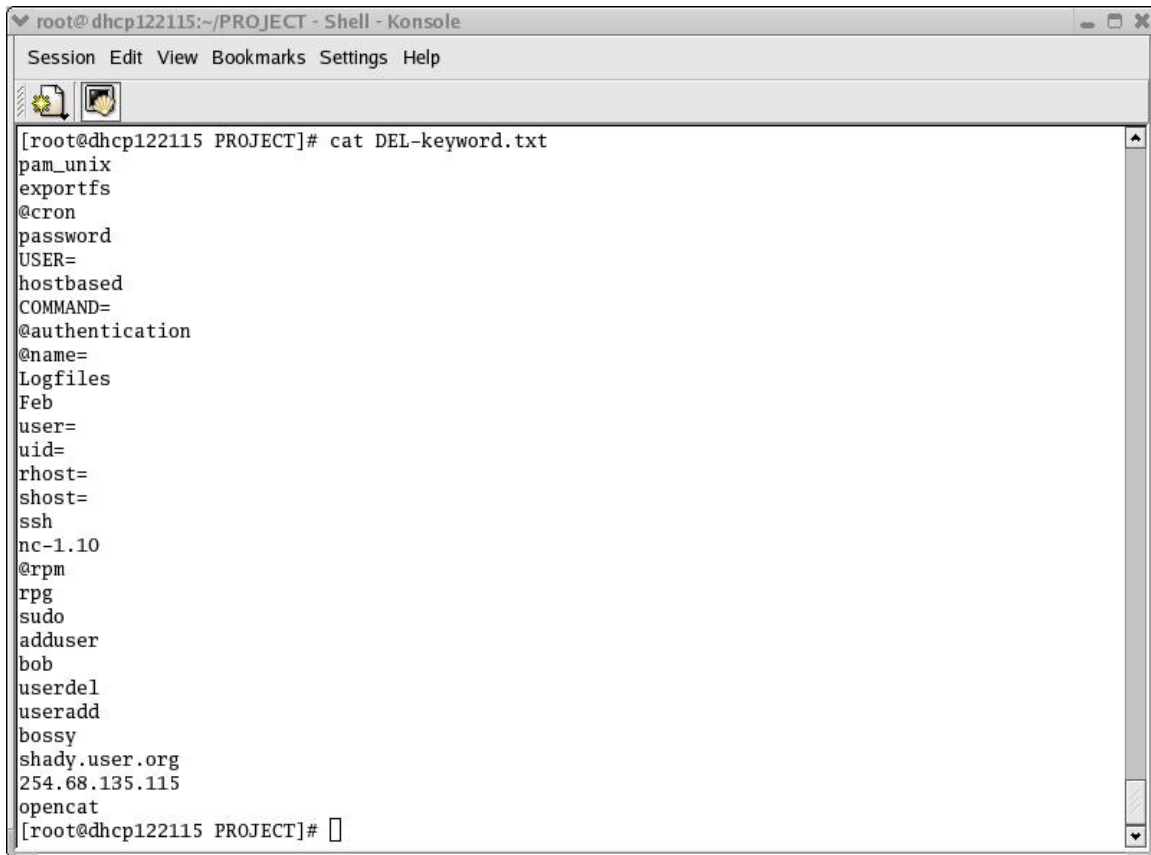
**Figure 65: sbin keyword results**

Shows following commands /usr/sbin being accessed

- adduser
- usermod
- visudo
- showmount
- checkconfig
- service
- userdel
- groupdel
- useradd

## APPENDIX D - Examining UNRMread's results

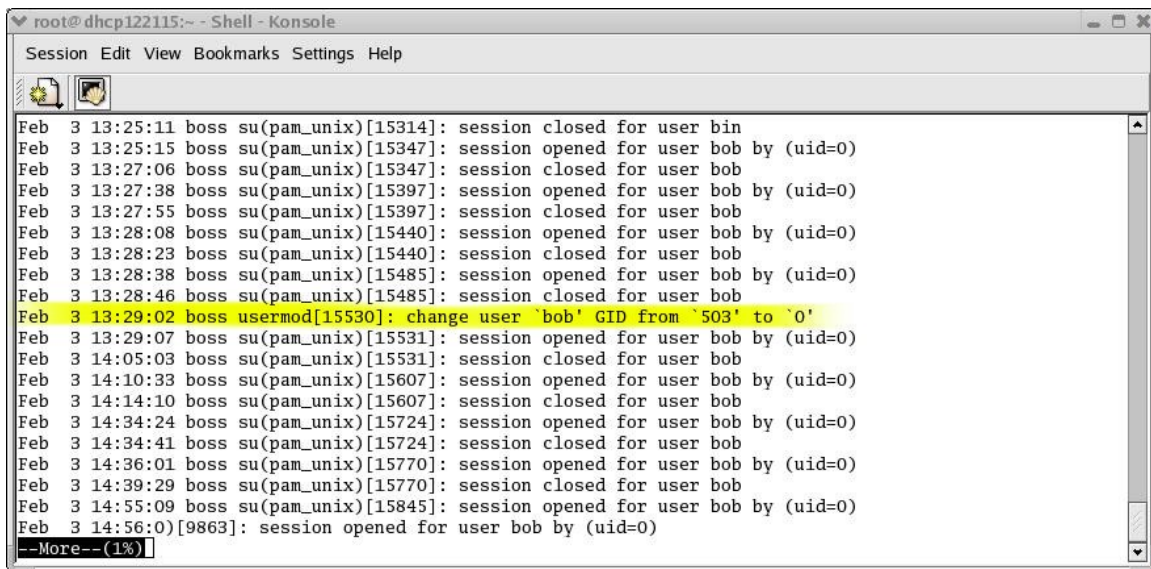
This appendix provides more evidence of the compromise gathered from the keyword files created by *UNRMread*. These results are gathered from the Mandrake box.

A screenshot of a terminal window titled "root@dhcp122115:~/PROJECT - Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". Below the menu bar are two icons: a yellow gear and a document. The terminal shows the command "[root@dhcp122115 PROJECT]# cat DEL-keyword.txt" and its output, which is a list of keywords: pam\_unix, exportfs, @cron, password, USER=, hostbased, COMMAND=, @authentication, @name=, Logfiles, Feb, user=, uid=, rhost=, shost=, ssh, nc-1.10, @rpm, rpg, sudo, adduser, bob, userdel, useradd, bossy, shady.user.org, 254.68.135.115, and opencat. The prompt "[root@dhcp122115 PROJECT]#" is visible at the bottom of the terminal.

```
[root@dhcp122115 PROJECT]# cat DEL-keyword.txt
pam_unix
exportfs
@cron
password
USER=
hostbased
COMMAND=
@authentication
@name=
Logfiles
Feb
user=
uid=
rhost=
shost=
ssh
nc-1.10
@rpm
rpg
sudo
adduser
bob
userdel
useradd
bossy
shady.user.org
254.68.135.115
opencat
[root@dhcp122115 PROJECT]#
```

**Figure 66: DEL-keyword.txt used with UNRMread**

Some new keywords found from analyzing the files from the RedHat box were added to the keyword file used with *UNRMread*.

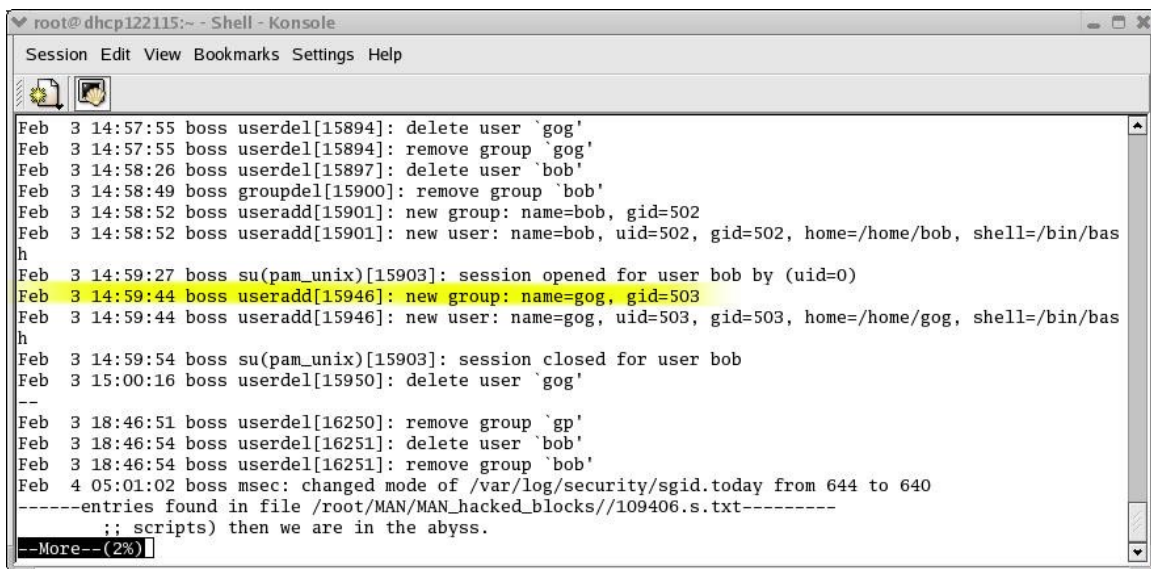


```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb 3 13:25:11 boss su(pam_unix)[15314]: session closed for user bin
Feb 3 13:25:15 boss su(pam_unix)[15347]: session opened for user bob by (uid=0)
Feb 3 13:27:06 boss su(pam_unix)[15347]: session closed for user bob
Feb 3 13:27:38 boss su(pam_unix)[15397]: session opened for user bob by (uid=0)
Feb 3 13:27:55 boss su(pam_unix)[15397]: session closed for user bob
Feb 3 13:28:08 boss su(pam_unix)[15440]: session opened for user bob by (uid=0)
Feb 3 13:28:23 boss su(pam_unix)[15440]: session closed for user bob
Feb 3 13:28:38 boss su(pam_unix)[15485]: session opened for user bob by (uid=0)
Feb 3 13:28:46 boss su(pam_unix)[15485]: session closed for user bob
Feb 3 13:29:02 boss usermod[15530]: change user 'bob' GID from '503' to '0'
Feb 3 13:29:07 boss su(pam_unix)[15531]: session opened for user bob by (uid=0)
Feb 3 14:05:03 boss su(pam_unix)[15531]: session closed for user bob
Feb 3 14:10:33 boss su(pam_unix)[15607]: session opened for user bob by (uid=0)
Feb 3 14:14:10 boss su(pam_unix)[15607]: session closed for user bob
Feb 3 14:34:24 boss su(pam_unix)[15724]: session opened for user bob by (uid=0)
Feb 3 14:34:41 boss su(pam_unix)[15724]: session closed for user bob
Feb 3 14:36:01 boss su(pam_unix)[15770]: session opened for user bob by (uid=0)
Feb 3 14:39:29 boss su(pam_unix)[15770]: session closed for user bob
Feb 3 14:55:09 boss su(pam_unix)[15845]: session opened for user bob by (uid=0)
Feb 3 14:56:00[9863]: session opened for user bob by (uid=0)
--More--(1%)
```

**Figure 67: example result from keyword 'bob' on lazarus files part 2**

Shows user *bob*'s GID being change from 503 to 0 that of the root's group.



```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb 3 14:57:55 boss userdel[15894]: delete user 'gog'
Feb 3 14:57:55 boss userdel[15894]: remove group 'gog'
Feb 3 14:58:26 boss userdel[15897]: delete user 'bob'
Feb 3 14:58:49 boss groupdel[15900]: remove group 'bob'
Feb 3 14:58:52 boss useradd[15901]: new group: name=bob, gid=502
Feb 3 14:58:52 boss useradd[15901]: new user: name=bob, uid=502, gid=502, home=/home/bob, shell=/bin/bas
h
Feb 3 14:59:27 boss su(pam_unix)[15903]: session opened for user bob by (uid=0)
Feb 3 14:59:44 boss useradd[15946]: new group: name=gog, gid=503
Feb 3 14:59:44 boss useradd[15946]: new user: name=gog, uid=503, gid=503, home=/home/gog, shell=/bin/bas
h
Feb 3 14:59:54 boss su(pam_unix)[15903]: session closed for user bob
Feb 3 15:00:16 boss userdel[15950]: delete user 'gog'
--
Feb 3 18:46:51 boss userdel[16250]: remove group 'gp'
Feb 3 18:46:54 boss userdel[16251]: delete user 'bob'
Feb 3 18:46:54 boss userdel[16251]: remove group 'bob'
Feb 4 05:01:02 boss msec: changed mode of /var/log/security/sgid.today from 644 to 640
-----entries found in file /root/MAN/MAN_hacked_blocks//109406.s.txt-----
;; scripts) then we are in the abyss.
--More--(2%)
```

**Figure 68: example result from keyword 'bob' on lazarus files part 3**

Shows new user *gog* being added

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@dhcp122115 root]# more MANDRAKE-BLOCK_UNRMread_file_dump/MANDRAKE-BLOCK-COMMAND\_keyword.txt
-----entries found in file /root/MAN/MAN_hacked_blocks//1093399.1.txt-----
Feb 3 14:41:30 boss kernel: sock_release: fasync list not empty!
Feb 3 14:56:18 boss sudo:      bob : command not allowed ; TTY=pts/1 ; PWD=/home/bob ; USER=bob ; COMMAN
D=/bin/useradd
Feb 3 14:56:51 boss sudo:      bob : TTY=pts/1 ; PWD=/home/bob ; USER=root ; COMMAND=/bin/useradd
Feb 3 14:57:04 boss sudo:      bob : TTY=pts/1 ; PWD=/home/bob ; USER=root ; COMMAND=/bin/useradd gog
Feb 3 14:59:37 boss sudo:      bob : command not allowed ; TTY=pts/1 ; PWD=/home/bob ; USER=root ; COMMA
ND=adduser gog
Feb 3 14:59:44 boss sudo:      bob : TTY=pts/1 ; PWD=/home/bob ; USER=root ; COMMAND=/bin/useradd gog
Feb 3 15:01:00 boss CROND[15956]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
Feb 3 15:56:34 boss sudo:      root : TTY=pts/2 ; PWD=/root ; USER=root ; COMMAND=/bin/bash
Feb 3 16:01:00 boss CROND[16064]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
Feb 3 16:08:22 boss sudo:      jim : TTY=pts/1 ; PWD=/home/jim ; USER=root ; COMMAND=/bin/useradd
Feb 3 16:38:35 boss kernel: sock_release: fasync list not empty!
Feb 3 17:01:00 boss CROND[16126]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
Feb 3 17:38:19 boss sudo:      jim : TTY=pts/2 ; PWD=/home/jim ; USER=root ; COMMAND=list
Feb 3 17:43:38 boss sudo:      jim : TTY=pts/2 ; PWD=/home/jim ; USER=root ; COMMAND=/bin/useradd gp
Feb 3 18:01:00 boss CROND[16229]: (root) CMD (nice -n 19 run-parts /etc/cron.hourly)
--
--More--(21%)

```

**Figure 69: Example result from keyword 'COMMAND=' on lazarus files**

Shows new user *gog* failing to be added via *sudo*

```

root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb 4 17:48:22 boss nfs: rpc.mountd shutdown succeeded
Feb 4 17:48:26 boss kernel: nfsd: last server has exited
Feb 4 17:48:26 boss kernel: nfsd: unexporting all filesystems
Feb 4 17:48:27 boss nfs: nfsd shutdown succeeded
Feb 4 17:48:27 boss nfs: Stopping NFS services: succeeded
Feb 4 17:48:27 boss exportfs[22083]: /etc/exports [1]: No 'sync' or 'async' option specified for export "192.168.123.3:/". Assuming default behaviour ('sync'). NOTE: this default has changed from previous versions
Feb 4 17:48:27 boss exportfs: exportfs: /etc/exports [1]: No 'sync' or 'async' option specified for export "192.168.123.3:/". Assuming default behaviour ('sync').
Feb 4 17:48:27 boss exportfs: NOTE: this default has changed from previous versions
Feb 4 17:48:27 boss nfs: Starting NFS services: succeeded
Feb 4 17:48:27 boss nfs: rpc.nfsd startup succeeded
Feb 4 17:48:27 boss nfs: rpc.mountd startup succeeded
Feb 4 17:48:31 boss rpc.mountd: export request from 192.168.123.2
Feb 4 17:49:28 boss rpc.mountd: authenticated mount request from little.box.com:693 for / (/)
Feb 4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb 4 17:56:33 boss sshd: stop succeeded
Feb 4 18:01:0 dump request from 192.168.123.2
Feb 4 17:47:56 boss rpc.mountd: export request from 192.168.123.2
Feb 4 17:48:22 boss rpc.mountd: Caught signal 15, un-registering and exiting.
Feb 4 17:48:22 boss nfs: rpc.mountd shutdown succeeded
Feb 4 17:48:26 boss kernel: nfsd: last server has exited
--More--(47%)

```

**Figure 70: Example result from keyword 'Feb' on single unrm file**

Shows nfs being restarted

Shows successful '/' directory mount request from little.box.com

Shows netcat rpm installation

Shows the time at which the user killed all SSH sessions after suspecting being compromised



```
root@dhcp122115:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Feb 4 17:47:09 boss exportfs[22020]: /etc/exports [1]: No 'sync' or 'async' option specified for export "192.168.123.3:/".
. Assuming default behaviour ('sync'). NOTE: this default has changed from previous versions
Feb 4 17:47:26 boss rpc.mountd: dump request from 192.168.123.2
Feb 4 17:47:56 boss rpc.mountd: export request from 192.168.123.2
Feb 4 17:48:22 boss rpc.mountd: Caught signal 15, un-registering and exiting.
Feb 4 17:48:22 boss nfs: rpc.mountd shutdown succeeded
Feb 4 17:48:26 boss kernel: nfsd: last server has exited
Feb 4 17:48:26 boss kernel: nfsd: unexporting all filesystems
Feb 4 17:48:27 boss nfs: nfsd shutdown succeeded
Feb 4 17:48:27 boss nfs: Stopping NFS services: succeeded
Feb 4 17:48:27 boss exportfs[22083]: /etc/exports [1]: No 'sync' or 'async' option specified for export "192.168.123.3:/".
. Assuming default behaviour ('sync'). NOTE: this default has changed from previous versions
Feb 4 17:48:27 boss exportfs: exportfs: /etc/exports [1]: No 'sync' or 'async' option specified for export "192.168.123.3:/".
Feb 4 17:48:27 boss exportfs: Assuming default behaviour ('sync').
Feb 4 17:48:27 boss exportfs: NOTE: this default has changed from previous versions
Feb 4 17:48:27 boss nfs: Starting NFS services: succeeded
Feb 4 17:48:27 boss nfs: rpc.nfsd startup succeeded
Feb 4 17:48:27 boss nfs: rpc.mountd startup succeeded
Feb 4 17:48:31 boss rpc.mountd: export request from 192.168.123.2
Feb 4 17:49:28 boss rpc.mountd: authenticated mount request from little.box.com:693 for / (/)
Feb 4 17:52:51 boss rpm: [RPM] nc-1.10-18 installed
Feb 4 17:56:33 boss sshd: stop succeeded
Feb 4 18:01:0 dump request from 192.168.123.2
Feb 4 17:47:56 boss rpc.mountd: export request from 192.168.123.2
Feb 4 17:48:22 boss rpc.mountd: Caught signal 15, un-registering and exiting.
Feb 4 17:48:22 boss nfs: rpc.mountd shutdown succeeded
Feb 4 17:48:26 boss kernel: nfsd: last server has exited
```

**Figure 71: Example result from keyword ‘Feb’ on lazarus files**

Shows nfs being restarted

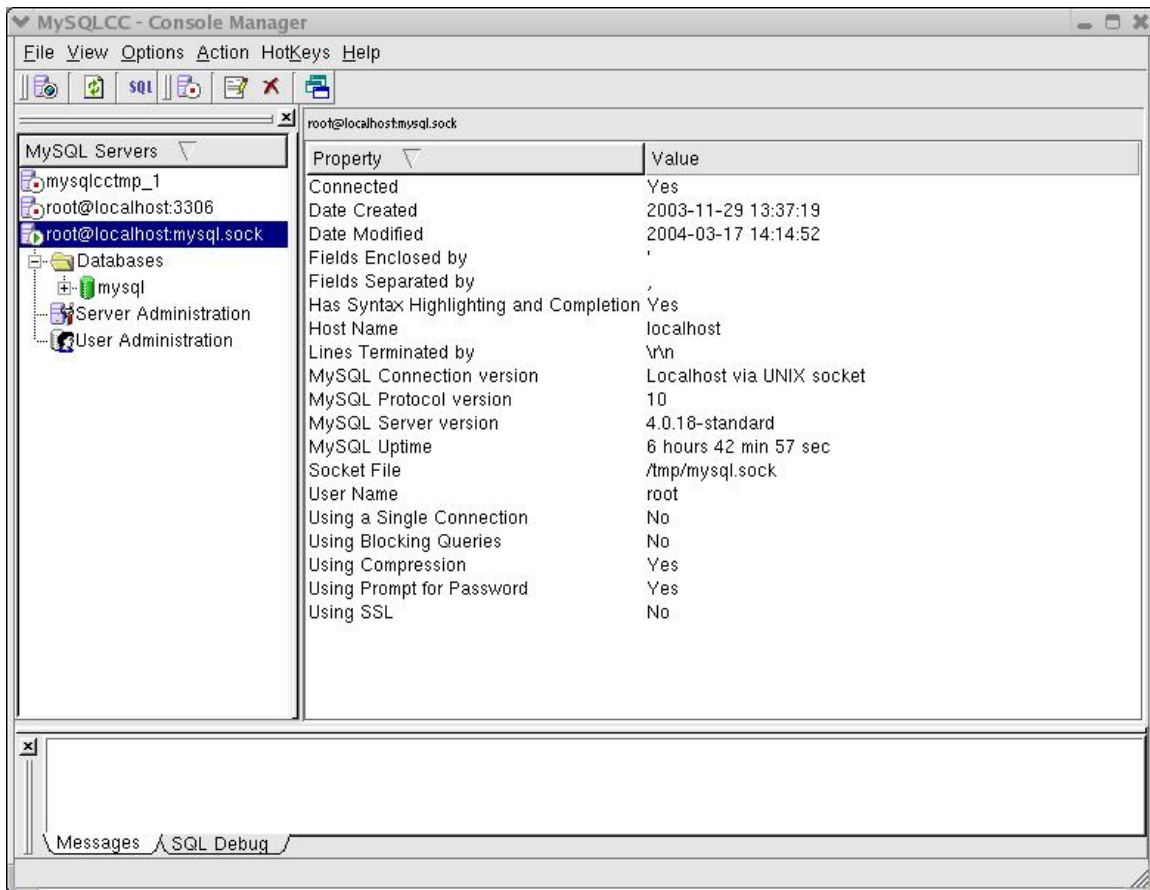
Shows successful ‘/’ directory mount request from little.box.com

Shows netcat rpm installation

Shows the time at which the user killed all SSH sessions after suspecting being compromised

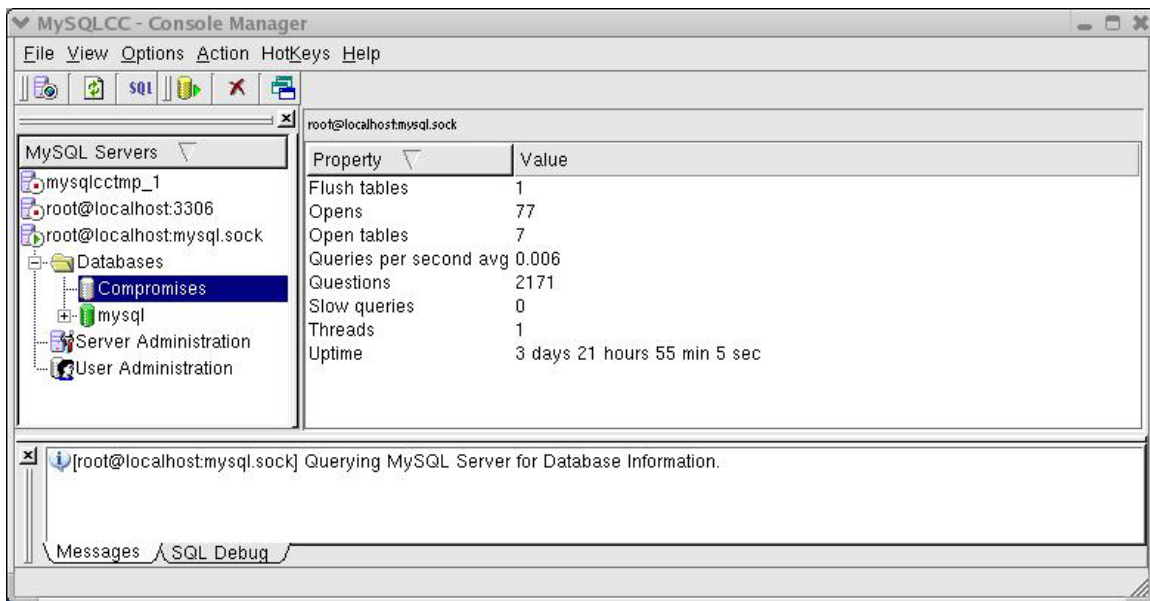
## APPENDIX E - MySQLcc reinforcement

This appendix provides screen shots gathered from *MySQLcc* after the options presented in *Sections 2.5 – Simple-MySQL* script were performed. They are offered as visual reinforcement to the correctness of *Simple-MySQL* since I was finally able to get it to work.



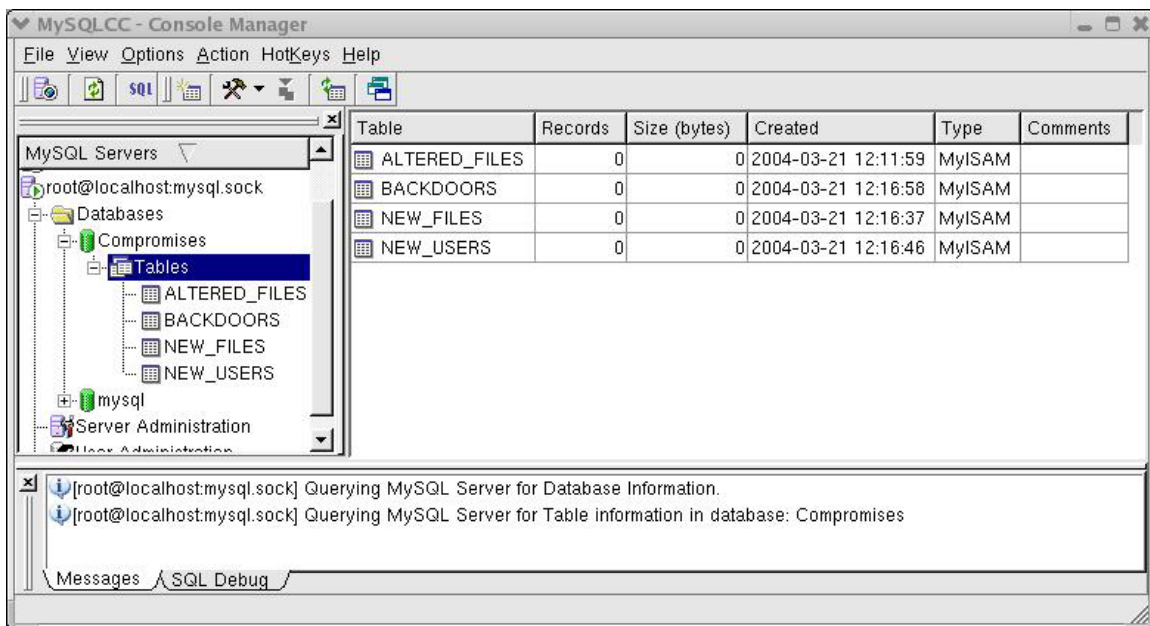
**Figure 72: Setup before Simple-MySQL is executed**

Presents the initial database setup. The highlighted row in the left-side panel is the account that is currently connected. As presented it should be noted that a database named 'mysql' already exists. This database was actually created during the installation of the MySQL database. The right-side panel gives some specifications of the current connection.



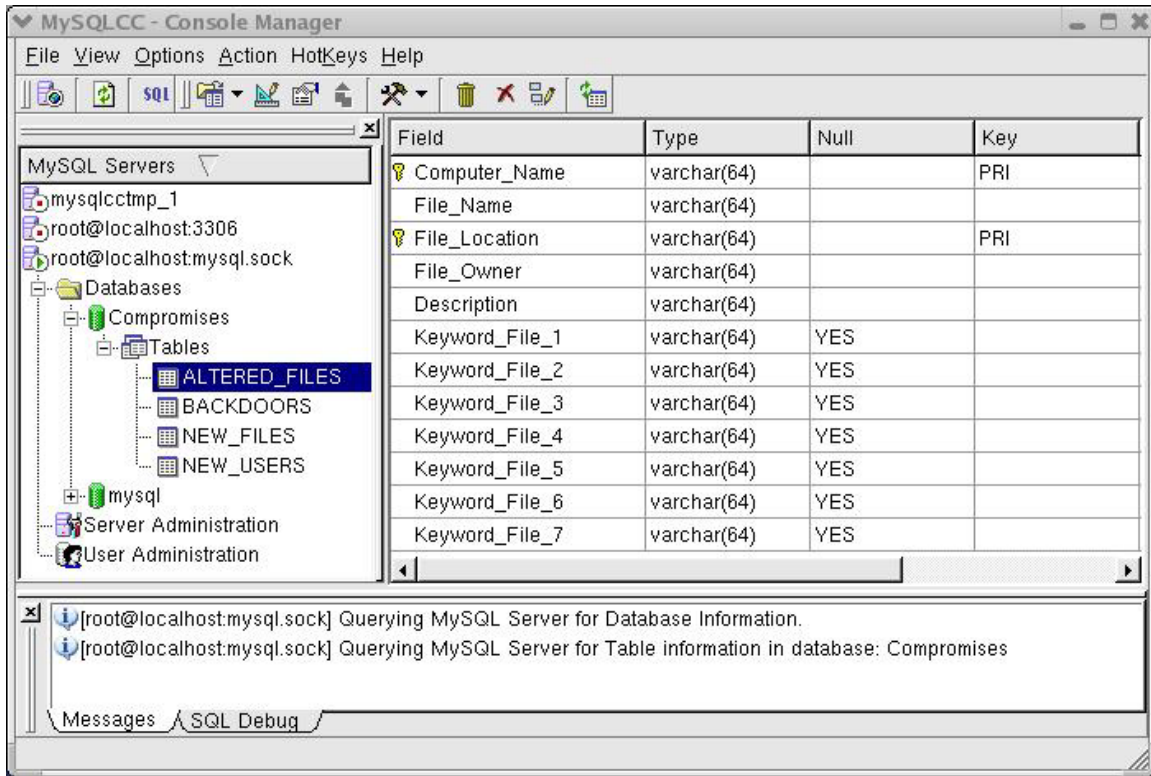
**Figure 73: Creating a database – MySQLcc view**

As made apparent in the above figure the 'Compromise' database was created and added to the database list for the user. Next the user has to create the tables to store the evidence.



**Figure 74: Showing tables – MySQLcc view**

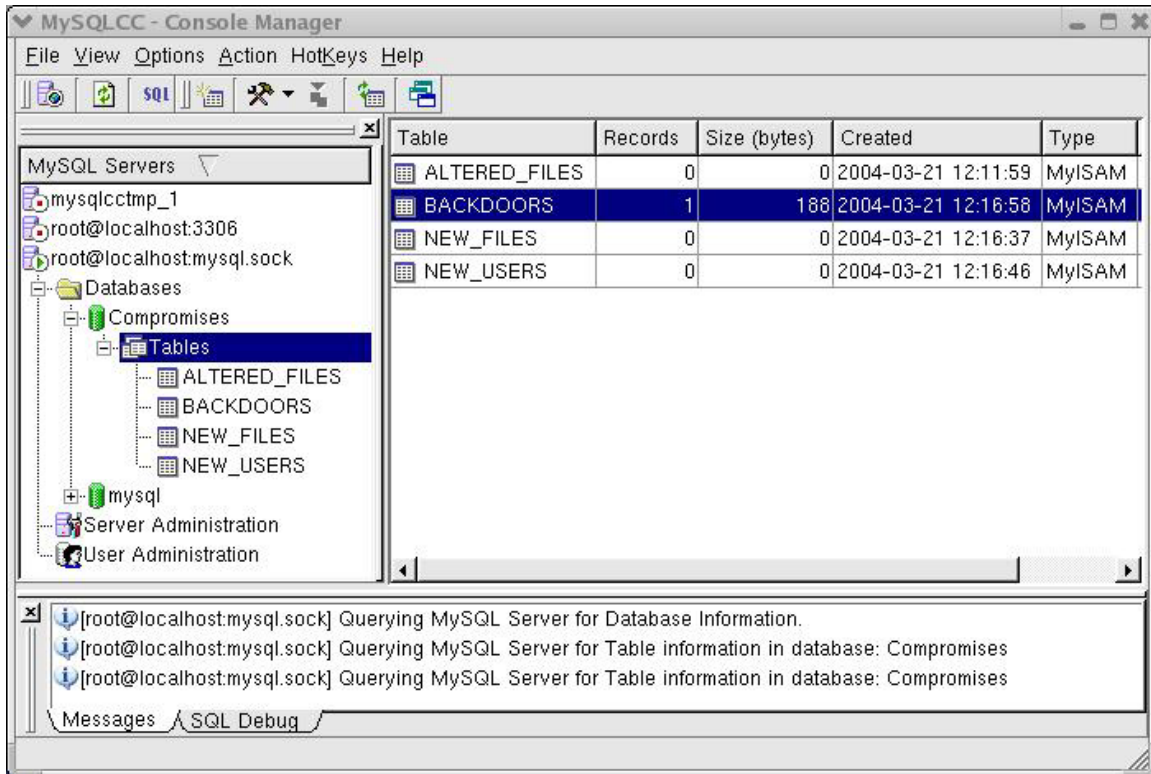
Shows that all four tables were created in the Compromises database by the Simple-MySQL script and the MySQLcc display is a little more descriptive.



**Figure 75: Column descriptions of a table – MySQLcc view**

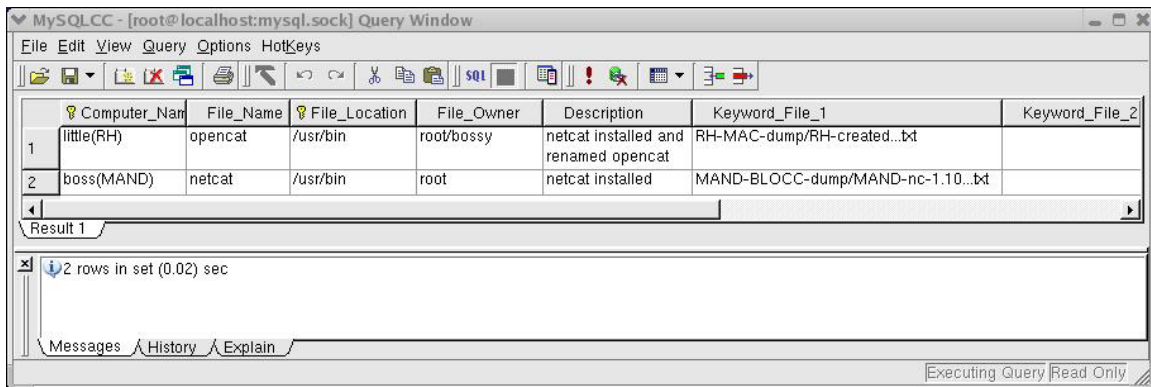
Shows the tables were constructed correctly with all predefined field present.





**Figure 76: Inserting a record – MySQL view**

Shows that a record was successfully entered into the BACKDOORS table.



**Figure 77: Viewing records – MySQL view**

Shows all fields contain the correct contents from the form read in by Simple-MYSQL.

## APPENDIX F - Project Scripts

This appendix provides the five project scripts GRcompare, DELread, MACread, UNRMread and Simple-MySQL.

### GRcompare

```
#!/bin/sh
```

```
#This script uses code written by Ives Aerts for the purpose of doing a
#regressive directory comparison. It is used to to compare the results obtained
#from the Graver-Robber tool from the before and after images. As written this
#script will compare any two directories for files and links with the same name.
# Files found in one directory, but not the other will be placed in a file
#called 'os-missing.txt', where 'os' is the operating system. Files found in
#both directories are compared using the
# 'diff' command and the results are written to a file consisting of the file's
#basename folled by '_diff_file.txt'.
#Author: JaRay Jasper with code from Ives Aerts
```

```
***NOTE***
```

```
#All files are written to a directory assigned to the variable $FILE_DUMP
#located near the bottom of this script. This location can be changed by
#entering another path assignment for this variable.
```

```
# Determine parameter type
```

```
function gettype () {
    if [ -L $1 ]; then
        echo "softlink"
    elif [ -f $1 ]; then
        echo "file"
    elif [ -d $1 ]; then
        echo "directory"
    else
        echo "unknown"
    fi
}
```

```
# Check for parameter existence and write name of argument that does not
# exist in both directories to 'os-missing.txt'
```

```
function exists () {
    if [ -e $1 -o -L $1 ]; then
        return 0;
    fi
}
```

```

else
    echo "$1 does not exist " >> $FILE_DUMP/"$os"-missing.txt
    echo " " >> $FILE_DUMP/"$os"-missing.txt
    return 1;
fi
}

# Compare two files for similar content. If files differ write difference to a
# difference file. I.e. if the files being compared had the name 'profile' the
# file created for this would be 'os-profile_diff_file.txt'
function comparefile () {
    cmp -s $1 $2
    if [ $? -gt 0 ]; then
        fn=$(basename "$1")
        echo "-----`date`-----" >> $FILE_DUMP/"$os"-
"$fn"_diff_file.txt
        echo "$1 ---different from--- $2" >> $FILE_DUMP/"$os"-"$fn"_diff_file.txt
        diff -y $1 $2 >> $FILE_DUMP/"$os"-"$fn"_diff_file.txt
    fi

    return
}

# Compare two directories for similar content, except for the MACTIME
databases
# body and body.S.
function comparedirectory () {
    local result=0
    for i in `(ls -A $1 && ls -A $2) | sort | uniq`; do
        if [ "$i" = "body" ] || [ "$i" = "body.S" ]; then
            echo "not looking in $i"
        else
            if [ -f "$i" ]; then
                if [ -e "$1/$i" && -e "$2/$i" ]; then
                    comparefile $1/$i $2/$i
                fi
            fi
            compare $1/$i $2/$i || result=1
        fi
    done
    return $result
}

# Compare softlinks for the same pointer destination. Write differencing pointers
# to the file 'os-SOFTLINK_diff.txt'.

```

```

function comparesoftlink () {
    local dest1=`ls -l $1 | awk '{ print $11 }'`
    local dest2=`ls -l $2 | awk '{ print $11 }'`

    if [ $dest1 = $dest2 ]; then
        return 0
    else
        echo "different link targets $1 -> $dest1 ----- $2 -> $dest2" >>
        $FILE_DUMP/"$os"-SOFTLINK_diff.txt
        return 1
    fi
}

```

# compare a file, directory, or softlink

```

function compare () {
    (exists $1 && exists $2) || return 1;

    local type1=$(gettype $1)
    local type2=$(gettype $2)

    if [ $type1 = $type2 ]; then
        case $type1 in
            file)
                comparefile $1 $2
                ;;
            directory)
                comparedirectory $1 $2
                ;;
            softlink)
                comparesoftlink $1 $2
                ;;
            *)
                echo "$1 of unknown type"
                false
                ;;
        esac
    else
        echo "type mismatch: $type1 ($1) and $type2 ($2)."
        false
    fi

    return
}

```

# Check incoming parameters

```

if [ 3 -ne $# ]; then
    cat << EOU
Usage: $0 Dir1 Dir2 Operating-System
Compare directory trees:
    files are binary compared (cmp) and differences collected (diff)
    directories are checked for identical content
    soft links are checked for identical targets
Operating-System: i.e. REDHAT, Debian, etc.
EOU
    exit 10
fi

# Get the name of the script being ran
if [ "$0" = "./GRcompare" ]; then
    sn=GRcompare
else
    sn=$(basename "$0")
fi

#Store the OS in a variable
os="$3"

# Assign and Check/Create default file dumping directory
FILE_DUMP=/root/MYSQL/"$os"_"$sn"_file_dump

if [ ! -e $FILE_DUMP ]; then
    mkdir $FILE_DUMP
fi

echo "-----`date`-----" >> $FILE_DUMP/"$os"-missing.txt
compare $1 $2

# Remove any 'diff.txt' files that deal with the md5's created by Grave-Robber
rm $FILE_DUMP/*md5*
rm $FILE_DUMP/*MD5*

exit $?

```

## DELread

```
#!/bin/sh
```

```
#This script can be used to read the output from the ils and icat
#tools. Requires a directory as the first argument and the location of the
#keyword file. As written this script will except any directory and only a
#keyword file named 'DEL-keyword.txt'. You can generate your own keyword list
#or make additions or comment out entries in the file 'DEL-keyword.txt'. To
#comment out lines in the keyword file, place an '@' symbol without the quotes
#as the first character on the line. Use only one keyword per line in the file.
#Author: JaRay Jasper with code from Ives Aerts
```

```
***NOTE***
```

```
#All files are written to a directory assigned to the variable $FILE_DUMP
#located near the bottom of this script. This location can be changed by
#entering another path assignment for this variable.
```

```
# Check to see if arguments passed to the directorysearch exist
```

```
function exists () {
    if [ -e $1 -o -L $1 ]; then
        return 0;
    else
        # echo "$1 doesn't exist"
        return 1;
    fi
}
```

```
# Recursive directory search for files (just in case)
```

```
# File produced will contain the name of the keyword used. i.e. if the search
```

```
# word was 'USER' the file created for this would be 'os-USER_keyword.txt'
```

```
function filesearch () {
    for i in `(ls -A $1)`; do
        if [ -f $i ]; then
            fn=$(basename "$i")
            while read line; do
                if (file $i | grep -qia ELF); then
                    if (readelf -a $i | grep -qa $line); then
                        echo "-----entries found in file $i-----" >> $FILE_DUMP/"$os"-
"$line"_keyword.txt
                        # echo "in ELF $i"
                        `(readelf -a $i | grep -a $line) >> $FILE_DUMP/"$os"-
"$line"_keyword.txt`
                        cp "$i" $FILE_DUMP/"$os"-"$fn"
                    fi
                fi
            done
        fi
    done
}
```

```

        echo "$fn" >> $FILE_DUMP/"$os"-"$line"_inode_list.txt
    fi

    elif (file $i | grep -qia data); then
    if (od -s $i | grep -qa $line); then
        echo "-----entries found in file $i-----" >> $FILE_DUMP/"$os"-"$line"_keyword.txt
        #echo "in DATA $i"
        `(od -s $i | grep -a $line) >> $FILE_DUMP/"$os"-"$line"_keyword.txt`
        cp "$i" $FILE_DUMP/"$os"-"$fn"
        echo "$fn" >> $FILE_DUMP/"$os"-"$line"_inode_list.txt
    fi

    elif (cat $i | grep -qa $line); then
        echo "-----entries found in file $i-----" >> $FILE_DUMP/"$os"-"$line"_keyword.txt
        #echo "in TEXT $i"
        `(cat $i | grep -a -C 1 $line) >> $FILE_DUMP/"$os"-"$line"_keyword.txt`
        cp "$i" $FILE_DUMP/"$os"-"$fn"
        echo "$fn" >> $FILE_DUMP/"$os"-"$line"_inode_list.txt
    fi
done < $FILE_DUMP/temp_search_list.txt
fi
directorysearch $1/$i
done
return
}

```

# Search through directories if needed for all files

```

function directorysearch () {
    (exists $1) || return 1;
    filesearch $1
    return
}

```

# Check incoming parameters

```

if [ $# -ne 3 ]; then
    cat << EOU

```

Usage: \$0 Directory Keyword-file Operating-System

Search through deleted inodes recovered by ils&icat for keywords:

Keyword file must be named: DEL-keyword.txt

Operating-System: i.e. REDHAT, Debian, etc.

```

EOU
    exit 10
fi

# Make sure first argument is a directory
if [ ! -d $1 ]; then
    echo "$1 is not a directory!" 1>&2
    exit 1
fi

# Make sure second argument is a keyword file named 'DEL-keyword.txt'
kw=$(basename "$2")
if [ "$kw" = "DEL-keyword.txt" ]; then
    if [ ! -f $2 ]; then
        echo "$2 is not a file" 1>&2
        exit 1
    fi
else
    echo "$2 has the wrong file name" 1>&2
    exit 1
fi

# Get the name of the script being ran
if [ "$0" = "./DELread" ]; then
    sn=DELread
else
    sn=$(basename "$0")
fi

#Store the OS name to a variable
os="$3"

# Check for existing file dump directory and make one if it doesn't exist
FILE_DUMP=/root/MYSQL/"$os"_"$sn"_file_dump

if [ ! -e $FILE_DUMP ]; then
    mkdir $FILE_DUMP
fi

# Filter keyword file for comments, make a temporary file containing all the
# search words.
cat $2 | grep -v @ >> $FILE_DUMP/temp_search_list.txt

directorysearch $1

```



```
# Remove the temporary search list created to exclude commented entries  
# in the keyword file.  
rm $FILE_DUMP/temp_search_list.txt
```

```
exit $?
```

## MACread

```
#!/bin/sh
```

```
#This script can be used to search a file that contains the redirected
#output from the mactime tool. Requires as arguments the user created
#mactime file and the location of the keyword file. As written this script will
#except any file as the first argument and only a file named 'MAC-keyword.txt'
#as the second argument. You can generate your own keyword list
#or make additions or comment out entries in the file 'MAC-keyword.txt'. To
#comment out lines in the keyword file, place an '@' symbol without the quotes
#as the first character on the line. Use only one keyword per line in the file.
#Author: JaRay Jasper
```

```
***NOTE***
```

```
#All files are written to a directory assigned to the variable $FILE_DUMP
#located near the bottom of this script. This location can be changed by
#entering another path assignment for this variable.
```

```
# Parameter checking
```

```
if [ $# -ne 3 ]; then
```

```
    cat << EOU
```

```
Usage: $0 MACfile MAC_keyword_file Operating-System
```

```
Search through mactime file made by running mactime with output redirection
```

```
MACfile: file that has redirected output from running mactime
```

```
MAC_keyword_file must be named: MAC-keyword.txt
```

```
Operating-System: i.e. REDHAT, Debian, etc.
```

```
EOU
```

```
    exit 10
```

```
fi
```

```
# Make sure first argument is a file
```

```
if [ ! -f $1 ]; then
```

```
    echo "$1 is not a file!" 1>&2
```

```
    exit 1
```

```
fi
```

```
# Make sure second argument is named 'MAC-keyword.txt'
```

```
kw=$(basename "$2")
```

```
if [ "$kw" = "MAC-keyword.txt" ]; then
```

```
    if [ ! -f $2 ]; then
```

```
        echo "$2 is not a file" 1>&2
```

```
        exit 1
```

```
    fi
```

```
else
```

```

        echo "$2 has the wrong file name" 1>&2
        exit 1
    fi

    # Get the name of the script being ran
    if [ "$0" = "./MACread" ]; then
        sn=MACread
    else
        sn=$(basename "$0")
    fi

    #Store OS name in a variable
    os="$3"

    # Check for existing file dump directory and make one if it doesn't exist
    FILE_DUMP=/root/MYSQL/"$os"_"$sn"_file_dump

    if [ ! -e $FILE_DUMP ]; then
        mkdir $FILE_DUMP
    fi

    # Filter keyword file for comments, make a temporary file containing all the
    # search words, then delete the temporary file
    cat $2 | grep -v @ >> $FILE_DUMP/temp_search_list.txt

    # File produced will contain the name of the keyword used. i.e. if the search
    # word was 'USER' the file created for this would be 'USER_keyword.txt'
    while read line; do
        #find all files that modified and created, by default includes files that have
        #been accessed also.
        if [ "$line" = "m.c" ]; then
            if (cat $1 | grep -qaw $line); then
                echo "-----entries found in file $1-----" >> $FILE_DUMP/"$os"-
"$line"_MAC_keyword.txt
                `(cat $1 | grep -aw $line) >> $FILE_DUMP/"$os"-
"$line"_MAC_keyword.txt`
            fi
        elif [ "$line" = "..c" ]; then
            #find all files that have only been created (not modified) and remove files
            #that have only been accessed
            if (cat $1 | grep -qaw $line); then

```

```

        echo "-----entries found in file $1-----" >> $FILE_DUMP/"$os"-
created_MAC_keyword.txt

        `(cat $1 | grep -vaw m.c) >> $FILE_DUMP/temp_MAC_keyword.txt`
        `(cat $FILE_DUMP/temp_MAC_keyword.txt | grep -vaw ".a.") >>
$FILE_DUMP/"$os"-created_MAC_keyword.txt`
        rm $FILE_DUMP/temp_MAC_keyword.txt
    fi
    elif (cat $1 | grep -qa $line); then
#Uses the rest of the searchword found in the MAC_keyword file
        echo "-----entries found in file $1-----" >> $FILE_DUMP/"$os"-
"$line"_MAC_keyword.txt

        `(cat $1 | grep -a $line) >> $FILE_DUMP/"$os"-"$line"_MAC_keyword.txt`
    fi
done < $FILE_DUMP/temp_search_list.txt

#Create one big file that has only the unique entries from the other files
#Many of the other files will have repetitive entries.
cat $FILE_DUMP/*MAC* | sort | uniq >> $FILE_DUMP/"$os"-
UNIQ_MAC_ENTRIES.txt

# Remove the temporary search list created to exclude commented entries
# in the keyword file.
rm $FILE_DUMP/temp_search_list.txt

exit $?

```

## UNRMread

```
#!/bin/sh
```

```
#This script can be used to read the output from unrm and lazarus tools.  
#Requires a directory or file as the first argument and the location of the  
#keyword file. As written this script will except any directory/file and only a  
#keyword file named 'DEL-keyword.txt'. You can generate your own keyword list  
#or make additions or comment out entries in the file 'DEL-keyword.txt'. To  
#comment out lines in the keyword file, place an '@' symbol without the quotes  
#as the first character on the line. Use only one keyword per line in the file.  
#Author: JaRay Jasper with code from Ives Aerts
```

```
***NOTE***
```

```
#All files are written to a directory assigned to the variable $FILE_DUMP  
#located near the bottom of this script. This location can be changed by  
#entering another path assignment for this variable.
```

```
# Check to see if arguments passed to the directorysearch exist
```

```
function exists () {  
    if [ -e $1 -o -L $1 ]; then  
        return 0;  
    else  
        #      echo "$1 doesn't exist"  
        return 1;  
    fi  
}
```

```
# Search a single file only. This would be the entire unrm data file. It's forced to  
# be searched as a text file so garbage may recovered also.
```

```
function searchfile () {  
    while read line; do  
        if (cat $1 | grep -qa $line); then  
            echo "-----entries found in file $1-----" >> $FILE_DUMP/"$os"-  
"$line"_keyword.txt  
            `(cat $1 | grep -a -C 1 $line) >> $FILE_DUMP/"$os"-"$line"_keyword.txt`  
        fi  
    done < $FILE_DUMP/temp_search_list.txt  
    return  
}
```

```
#Searched block directory created by lazarus for specified keywords
```

```
function filesearch () {  
    for i in `(ls -A $1)`; do  
        if [ -f $i ]; then
```

```

fn=$(basename "$i")
if (ls $i | grep -q '!.txt') || (ls $i | grep -q 'x.txt') || (ls $i | grep -q 'a.txt'); then
:
else
while read line; do
if (cat $i | grep -qa $line); then
echo "-----entries found in file $i-----" >> $FILE_DUMP/"$os"-
"$line"_keyword.txt
# echo "in TEXT $i"
\ (cat $i | grep -a -C 1 $line) >> $FILE_DUMP/"$os"-
"$line"_keyword.txt`
cp "$i" $FILE_DUMP/"$os"-"$fn"
echo "$fn" >> $FILE_DUMP/"$os"-"$line"_block_list.txt
fi
done < $FILE_DUMP/temp_search_list.txt
fi
fi
directorysearch $1/$i
done
return
}

```

```

# Search through directories if needed for all files
function directorysearch () {
(exists $1) || return 1;
filesearch $1
return
}

```

```

# Check incoming parameters
if [ $# -ne 3 ]; then
cat << EOU
Usage: $0 Directory/file Keyword-file Operating-System
Search through unrm file or lazarus blocks directory for keywords:
Keyword file must be named: DEL-keyword.txt
Operating-System: i.e. REDHAT, Debian, etc.
EOU
exit 10
fi

```

```

if [ -d $1 ] || [ -f $1 ]; then
# Make sure second argument is a keyword file named 'DEL-keyword.txt'
kw=$(basename "$2")
if [ "$kw" = "DEL-keyword.txt" ]; then
if [ ! -f $2 ]; then

```



```

        echo "$2 is not a file" 1>&2
        exit 1
    fi
else
    echo "$2 has the wrong file name" 1>&2
    exit 1
fi

# Get the name of the script being ran
if [ "$0" = "./UNRMread" ]; then
    sn=UNRMread
else
    sn=$(basename "$0")
fi

#Store the OS name to a variable
os="$3"

# Check for existing file dump directory and make one if it doesn't exist
FILE_DUMP=/root/MYSQL/"$os"_"$sn"_file_dump

if [ ! -e $FILE_DUMP ]; then
    mkdir $FILE_DUMP
fi

# Filter keyword file for comments, make a temporary file containing all the
# search words.
cat $2 | grep -v @ >> $FILE_DUMP/temp_search_list.txt
else
    echo "$1 is not a directory or file"
    exit 1
fi

if [ -d $1 ]; then
    directorysearch $1
elif [ -f $1 ]; then
    searchfile $1
fi

# Remove the temporary search list created to exclude commented entries
# in the keyword file.
rm $FILE_DUMP/temp_search_list.txt

exit $?

```

## Simple-MySQL

```
#!/bin/sh
#This script provides an interface for automatically connecting to your MySQL
#database. It provides the user with the abilities to show databases, tables
#contents of one table, create a database, create 4 prespecified tables and to
#insert records into one of the four tables via a text file. It requires the
# name and password of the user who is allowed to access MySQL.

#The format of the text file to be used for record insertion is the same for
#the ALTERED_FILES, NEW_FILES and BACKDOORS. This is an example of
the file
#format.

# Redhat
# passwd
# /etc (for the NEW_USERS table this the uid field and requires an integer)
# root (for the NEW_USERS table this the gid field and requires an integer)
# new users bob and jim were added
# Redhat/Redhat-user=_keyword.txt
# @
# @
# @
# @
# @
# @
# @

#The record file the user has to create can only be 12 lines long and the
#after a table is created the describe option can be used to see what fields
#each table holds for better reference. The '@' is used as a null character
#the keyword_file fields only and as place holders, therefore are required
#to avoid possible troubles.
#Author JaRay Jasper

****NOTE****
#The mysql_path variable contains the path to the mysql executable and
#should be changed to the location the mysql executable on the user's
#computer. It's located near the bottom of this script.

# Shows the list of current databases for the user's account. Can be used
# anytime during the running of the script.
function show_db () {
    echo -n "List of current databases"
    echo ""
    echo ""
```

```

    $mysql_path -u $1 --password=$2 << EOF
show databases;
EOF
}

```

# Creates a new databases using whatever name the user enters. It then  
# shows the updated database list.

```

function create_db () {
    $mysql_path -u $1 --password=$2 << EOF
CREATE DATABASE $db;
EOF

```

```

    echo ""
    echo -n "List of current databases"
    echo ""
    echo ""

```

```

    $mysql_path -u $1 --password=$2 << EOF
show databases;
EOF
}

```

# Creates either the ALTERED\_FILES, NEW\_FILES or BACKDOORS table. It  
then

# then shows the updated list of tables in that database.

```

function create_table () {
    $mysql_path -u $1 --password=$2 << EOF

```

```

USE $db
CREATE TABLE $table_name( Computer_Name VARCHAR(64) NOT NULL,
File_Name VARCHAR(64) NOT NULL,
File_Location VARCHAR(64) NOT NULL,
File_Owner VARCHAR(64) NOT NULL,
Description VARCHAR(64) NOT NULL,
Keyword_File_1 VARCHAR(64),
Keyword_File_2 VARCHAR(64),
Keyword_File_3 VARCHAR(64),
Keyword_File_4 VARCHAR(64),
Keyword_File_5 VARCHAR(64),
Keyword_File_6 VARCHAR(64),
Keyword_File_7 VARCHAR(64),
PRIMARY KEY (Computer_Name, File_Location) );

```

```

show tables;
EOF

```

```

}

# Creates the NEW_USERS table. It then
# then shows the updated list of tables in that database.
function create_table_newuser () {
    $mysql_path -u $1 --password=$2 << EOF

    USE $db
    CREATE TABLE $table_name( Computer_Name VARCHAR(64) NOT NULL,
    User_Name VARCHAR(64) NOT NULL,
    UID INTEGER NOT NULL,
    GID INTEGER NOT NULL,
    Description VARCHAR(64) NOT NULL,
    Keyword_File_1 VARCHAR(64),
    Keyword_File_2 VARCHAR(64),
    Keyword_File_3 VARCHAR(64),
    Keyword_File_4 VARCHAR(64),
    Keyword_File_5 VARCHAR(64),
    Keyword_File_6 VARCHAR(64),
    Keyword_File_7 VARCHAR(64),
    PRIMARY KEY (Computer_Name, User_Name) );

    show tables;
    EOF
}

# Shows the list of tables found in a specified database.
function show_tables() {
    echo ""

    $mysql_path -u $1 --password=$2 << EOF

    USE $db
    show tables;
    EOF
}

# Shows the column descriptions of a table found in a specified database.
function show_columns() {
    echo ""

    $mysql_path -u $1 --password=$2 << EOF

    USE $db
    show columns from $table_name;
    EOF
}

```

```
}
```

```
# Used to keep the script running until the user enters '0' to exit the
# script. Keeps the user from having to continuously log in and out (running
# the script over and over). My preference is not to clear the screen after
# an input because I like to be able to scroll up and see what I was doing.
# You can enter the 'clear' command below the 'read' command and it will
# clear the screen for you after every time enter is hit.
```

```
function enter () {
    echo ""
    echo -n "Press Enter to choose again"
    read
```

```
}
```

```
# Reads the text file the user provides line by line and stores the lines in
# a variable for insertion into the table chosen by the user.
```

```
function insert () {
    if [ -f $file ]; then
        count=0
        while read line; do
            case "$count" in
                0)
                    comp_name="$line" ;;
                1)
                    if [ "$flag" = "1" ]; then
                        user_name="$line"
                    else
                        file_name="$line"
                    fi
                    ;;
                2)
                    if [ "$flag" = "1" ]; then
                        uid="$line"
                    else
                        file_loc="$line"
                    fi
                    ;;
                3)
                    if [ "$flag" = "1" ]; then
                        gid="$line"
                    else
                        file_own="$line"
                    fi
                    ;;
            esac
            count=$((count+1))
        done
    fi
}
```

```

4)
  descp="$line" ;;
5)
  if [ "$line" = "@" ]; then
    key1=""
  else
    key1="$line"
  fi
  ;;
6)
  if [ "$line" = "@" ]; then
    key2=""
  else
    key2="$line"
  fi
  ;;
7)
  if [ "$line" = "@" ]; then
    key3=""
  else
    key3="$line"
  fi
  ;;
8)
  if [ "$line" = "@" ]; then
    key4=""
  else
    key4="$line"
  fi
  ;;
9)
  if [ "$line" = "@" ]; then
    key5=""
  else
    key5="$line"
  fi
  ;;
10)
  if [ "$line" = "@" ]; then
    key6=""
  else
    key6="$line"
  fi
  ;;
11)
  if [ "$line" = "@" ]; then

```



```

        key7=""
    else
        key7="$line"
    fi
    ;;
esac
count=$((count+1))
done < $file

if [ "$flag" = "0" ]; then
    $mysql_path -u $1 --password=$2 << EOF
USE $db
INSERT INTO $table_name
VALUES('$comp_name','$file_name','$file_loc','$file_own','$descp','$key1','$key2',
'$key3','$key4','$key5','$key6','$key7');
EOF
    else
        $mysql_path -u $1 --password=$2 << EOF
USE $db
INSERT INTO $table_name
VALUES('$comp_name','$user_name',$uid,$gid,$descp','$key1','$key2','$key3',
'$key4','$key5','$key6','$key7');
EOF
    fi
else
    echo "This file doesn't exist!!"
fi
}

```

# List the available tables the user can choose to create, show or insert  
# records into.

```

function table_menu () {
    echo ""
    echo "TABLE MENU"
    echo "1 - Altered files TABLE"
    echo "2 - New files TABLE"
    echo "3 - New users TABLE"
    echo "4 - Backdoors TABLE"
    echo ""
    echo "0 - EXIT AND DO NOTHING"
    echo -n "Enter choice: "
    read table
    return $table
}

```

# Shows the entire contents of one table. Because of the way MySQL outputs

```

# data this can get a bit messy. I tried to separate the information so
# it's displayed in a more readable fashion. Your best bet is not to use long
# path names and descriptions for entry into the table. While you are allotted
# up to 64 characters per line in the record text file, you may only want
# to use between 20 and 30.
function show_records () {
if [ "$table" = "3" ]; then
    $mysql_path -u $1 --password=$2 << EOF
USE $db
SELECT Computer_Name, User_Name, UID, GID FROM $table_name ;
EOF
else
    $mysql_path -u $1 --password=$2 << EOF
USE $db
SELECT Computer_Name, File_Name, File_Location, File_Owner, Description
FROM $table_name
EOF
fi

echo ""

if [ "$table" = "3" ]; then
    $mysql_path -u $1 --password=$2 << EOF
USE $db
SELECT Computer_Name, User_Name, Keyword_File_1, Keyword_File_2,
Keyword_File_3, Keyword_File_4 FROM $table_name ;
EOF
else
    $mysql_path -u $1 --password=$2 << EOF
USE $db
SELECT Computer_Name, File_Name, Keyword_File_1, Keyword_File_2,
Keyword_File_3, Keyword_File_4 FROM $table_name
EOF
fi

echo ""

if [ "$table" = "3" ]; then
    $mysql_path -u $1 --password=$2 << EOF
USE $db
SELECT Computer_Name, User_Name, Keyword_File_5, Keyword_File_6,
Keyword_File_7 FROM $table_name ;
EOF
else
    $mysql_path -u $1 --password=$2 << EOF
USE $db

```

```

SELECT Computer_Name, File_Name, Keyword_File_5, Keyword_File_6,
Keyword_File_7 FROM $table_name
EOF
fi

}

```

# Because of the various uses of the table menu. I decided to a function that  
# always calls the table menu function and based on the return this functions  
# does different things. One of which is calling the insert function which is  
# used to insert a record into a table as explained above.

```

function table_case () {
    table_menu
    case $table in
        1)
            table_name=ALTERED_FILES
            if [ "$choice" = "7" ]; then
                show_columns $1 $2 $db $table_name
            elif [ "$choice" = "5" ]; then
                create_table $1 $2 $db $table_name
            elif [ "$choice" = "6" ]; then
                flag=0
            else
                show_records $1 $2 $db $table_name
            fi
            ;;
        2)
            table_name=NEW_FILES
            if [ "$choice" = "7" ]; then
                show_columns $1 $2 $db $table_name
            elif [ "$choice" = "5" ]; then
                create_table $1 $2 $db $table_name
            elif [ "$choice" = "6" ]; then
                flag=0
            else
                show_records $1 $2 $db $table_name
            fi
            ;;
        3)
            table_name=NEW_USERS
            if [ "$choice" = "7" ]; then
                show_columns $1 $2 $db $table_name
            elif [ "$choice" = "5" ]; then
                create_table_newuser $1 $2 $db $table_name
            elif [ "$choice" = "6" ]; then
                flag=1
            fi
        ;;
    esac
}

```

```

        else
            show_records $1 $2 $db $table_name $table
        fi
        ;;
    4)
        table_name=BACKDOORS
        if [ "$choice" = "7" ]; then
            show_columns $1 $2 $db $table_name
        elif [ "$choice" = "5" ]; then
            create_table $1 $2 $db $table_name
        elif [ "$choice" = "6" ]; then
            flag=0
        else
            show_records $1 $2 $db $table_name
        fi
        ;;
    0) ;;
    *) echo "Please enter 1, 2, 3, 4 or 0"; enter
esac

if [ "$choice" = "6" ]; then
    if [ "$table" != "0" ]; then
        echo "Enter full path to file (just to be safe (;p) )"
        read file
        insert $1 $2 $db $table_name $file $flag
    fi
fi

}

# Only checks the number incoming parameters. If the wrong combination of
# username and password is entered the script won't run.
if [ 2 -ne $# ]; then
    cat << EOU
Usage: $0 mysql_user mysql_passwd
EOU
    exit 10
fi

# Path to the MySQL executable
mysql_path=/root/MYSQL/mysql/bin/mysql

# The interactive menu loop. Continuously runs and allows the user to do
# different tasks and type in things like the path to the record file
# until the option number '0' is chosen which will cause the script to end.
choice=

```

```

until [ "$choice" = "0" ]; do
    echo ""
    echo "DATABASE MENU"
    echo "1 - Show databases"
    echo "2 - Show tables"
    echo "3 - Show all records from 1 table"
    echo "4 - Create new database"
    echo "5 - Create new table"
    echo "6 - Insert record into table"
    echo "7 - Show the column description of a table"
    echo ""
    echo "0 - exit program"
    echo ""
    echo -n "Enter choice: "
    read choice
    echo ""
    case $choice in
        1)
            show_db $1 $2
            enter ;;
        2)
            echo "Enter database name that holds the tables"
            read db
            show_tables $1 $2 $db
            enter ;;
        3)
            echo "Enter database name that holds the table"
            read db
            table_case $1 $2 $db $choice
            enter ;;
        4)
            echo "Enter name for new database"
            read db
            create_db $1 $2 $db
            enter ;;
        5)
            echo "Enter database name to create table in"
            read db
            table_case $1 $2 $db $choice
            enter ;;
        6)
            echo "Enter database name"
            read db
            table_case $1 $2 $db $choice
            enter ;;
        7)

```

```
        echo "Enter database name that holds the table"
        read db
        table_case $1 $2 $db $choice
        enter ;;
    0) exit ;;
    *) echo "Please enter 1, 2, 3 or 0"; enter
esac
done
```