Fight crime.
Unravel incidents... one byte at a time.

SANS GIAC
Practical Assignment
GCFA
Version 1.0

Steven D Dietz
September 15, 2002

Table of Contents

SANS GIAC CERTIFIED FORENSIC ANALYST
Assignment One-Option two: ISObuster as a forensic tool
Author Steven Dietz

Abstract:

There are many different tools in the forensic investigator's toolkit. ISObuster is a tool available to investigate data on CD-ROMs, DVDs, and binary images of CDs stored. This includes CDs or DVDs thought to be bad. The tool offers a flexible and intuitive interface to read and recover information even from supposed bad CDs.

Description and Scope:

The world of forensic analysis is very large. Not all forensic investigations are based on network intrusion or located on hard drive media. Forensic data investigation and recovery from other media is required as well. The forensic analyst needs a complete toolkit for all media types. The CD-ROM and DVD has become a common storage media for personal and business use. The .ISO and .bin extensions have become a standard for transporting and storing CD-ROM images on hard drives. The forensic investigator requires a consistent and testable tool to read and recover data from this form of media.

ISObuster is a potential forensic analysis tool for a Windows based analysis and a potential forensic recovery of the various types of CD-ROM and DVD image types. The tool can read and recover data from CDs that cannot normally be read. This product can read and recover data in the different formats of CD-ROMs. It can read and recover data from apparent bad CDs. The program can also extract specific files from good image files. The extraction can be granular extracting a specific file or general extracting raw data. This extraction can be at the block level or expand to the complete CD.

This tool would not be installed on a compromised device. It would be a tool installed on an investigator's PC to investigate optical media discovered and recovered during a forensic investigation. The media can originally be created on a Macintosh, Linux or PC. The program is agnostic in regards to the original hardware that created the CD.

The following table lists the data formats that ISObuster can read.

Image File types	CD Types
*.ISO (Nero, BlindRead, Creator)	CD-I
*.BIN (CDRWin)	VCD
*.IMG (CloneCD)	SVCD
*.CIF (Creator)	CD-ROM
*.FCD (Uncompressed)	CD-ROM XA

*.NRG (Nero)	DVCD,
*.GCD (Prassi)	DVD
*.P01 (Toast)	ISO-9660
*.C2D (WinOnCD)	UDF 1.02,1.5,2.01
*.CUE (CDRWin)	Motorola
*.CIF (DiscJuggler)	Big Endian
*.CD (CD-i OptImage)	Little Endian
*.GI (Prassi PrimoDVD)	Rock Ridge
*.DAO (Duplicator)	
*.TAO (Duplicator)	

Background information:

The tool ISObuster can be found and downloaded from a number of sites on the Internet. The product and description can be found at the web site: http://www.smart-projects.net/isobuster/. The newest version is version 1.0 which is the version used for this document. The product author is Peter Van Hove. The original intent of this product is to be used to access or recover CD media that may be bad. It can also be used to extract portions of data from good CDs.

Prior to May 2002, this product was considered a beta level product. When considering a forensic tool, the use of a beta level product should be avoided. The information that a program is a beta level product may be a way for a litigator to discount the validity of the results.

The web site offered general information about the program and the ability to download the two different versions of the program, International language support and English only. There was also a download link for the Help file. The two files I chose to download for the test were the help file and the International language support version of ISObuster.

Installation of the program is straightforward. The program is provided in a zip file. In the zip file is one file, an executable for extraction and setup. The setup program installs the program by default to **Drive letter:\Program Files\Smart Projects\ISObuster**. During the installation, it goes through the standard steps of location, use agreement and file type association. The file association is for all standard images of CD and DVD types. Please reference the previous table for potential associations.

The finished installed program is a stand-alone executable file that is 1.5 MB in size. In the ISObuster subdirectory there are five subdirectories created in addition to placement of the executable, IsoBuster.exe. The five subdirectories are: FAQ, Lang, Online, Plug-ins and Uninst. These directories provide the secondary support for the application. The program allows for potential plug-ins and references the various .dll files in the ***Drive letter*:\Program Files\Smart Projects\ISObuster\lang** directory for language support other than English.

The program source code is not readily available. The program was written in Borland C++ and must run in Win32 mode. The system is expected to have the standard ASPI .dlls as well as the communication dialog .dll files. The program is successfully run on Windows 2000 and Windows XP Professional. This program as a forensic tool depends on successful verification it makes no hidden system or trojan calls. It must also have the ability to verify that the data rendered from the CD was forensically identical to the original data. The mechanisms and tests to check these two issues are described below.

Validation Testing Processes:

The following describes the testing environment. All tests were conducted on the described device.

Device Base Hardware

Name and Type	**Description**
Hardware	HP Pavilion- XU155 384Mb P3-750 Notebook
Operating System	Windows 2000 Professional SP2
CD-RW	Mashita UJDA 330 1.53
DVD-ROM	Toshiba DVD-ROM SD-2502 1915

Software Test Suite

ISOBuster V1.00	**Test Program**
SysInternals Utility	PMON
SysInternals Utility	Process Explorer
CygWin Utility	Md5sum
Good CD #1	F.I.R.E. CD
Good DVD	**And Then There Were None** Movie DVD
Bad CD #1 (Buffer UnderRun)	Unknown Data
Bad CD #2 (Bad Index Table)	Unknown Data

The validation ISObuster could be used as a forensic tool was conducted in a two-phase test. The first phase is to run the program while monitoring the system for unknown calls. The programs used for this test are SysInternals PMON and Process Explorer. These programs work by displaying the processes and system calls. They also provide the ability to create a logged text file of all process activity. These monitoring programs are originally written for Windows NT. Their monitoring functions performed adequately in a Windows 2000 Professional environment. The strength of these utilities is they monitor undocumented calls and processes not normally visible.

The testing process of ISObuster was initiated by starting the PMON and Process Explorer utilities. The ISObuster application was then started. A test CD-ROM was then be opened, examined and a file extracted. The log files were

then reviewed. All other applications were stopped to avoid any unknown process calls.

The test results from the two programs showed no unusual system calls. The following is the result of the Process Explorer utility. There were no unknown handles as well. Opening a CD did not cause a change and open any undocumented calls or processes. The following table provides the results from Process Explorer.

Process:		**IsoBuster.exe PID :1592**					
Base	**Size**	**Description**	**Version**	**Time**			**Path**
0x240000	0x16000		*	12/7/1999	7:00	AM	F:\WINNT\system32\Unicode.nls
0x260000	0x2F000		*	5/4/2001	2:05	PM	F:\WINNT\system32\locale.nls
0x290000	0x41000		*	12/7/1999	7:00	AM	F:\WINNT\system32\sortkey.nls
0x2E0000	0x4000		*	5/4/2001	2:05	PM	F:\WINNT\system32\sorttbls.nls
0x400000	0x1DC000		1.00.0000.0001	5/5/2002	9:26	PM	F:\Program Files\Smart Projects\IsoBuster\IsoBuster.exe
0x970000	0x2000		*	12/7/1999	7:00	AM	F:\WINNT\system32\ctype.nls
0x1040000	0x4000		*	3/23/2002	7:32	PM	F:\WINNT\Registration\R0000000000cc.clb
0x70BD0000	0x4C000	Shell Light-weight utility Library	5.50.4807.2300	7/23/2001	8:16	PM	F:\WINNT\system32\shlwapi.dll
0x716F0000	0x8A000	Common Controls Library	5.81.4807.2300	7/23/2001	8:16	PM	F:\WINNT\system32\comctl32.dll
0x759B0000	0x6000	LZExpand/Compress API	5.00.2134.0001	12/7/1999	7:00	AM	F:\WINNT\system32\lz32.dll
0x76B30000	0x3E000	Common Dialogs	5.00.3103.1000	5/4/2001	2:05	PM	F:\WINNT\system32\COMDLG32.DLL
0x770C0000	0x23000	Offline Network Agent	5.00.2195.2401	5/4/2001	2:05	PM	F:\WINNT\system32\cscdll.dll
0x775A0000	0x85000		2000.02.3488.0000	10/30/2001	7:57	AM	F:\WINNT\system32\clbcatq.dll
0x778000 00	0x1D000	Windows Spooler Driver	5.00.2195.2780	5/4/2001	2:05	PM	F:\WINNT\system32\WINSPOOL.DRV
0x778200 00	0x7000	Version Checking and File Installation Libraries	5.00.2134.0001	12/7/1999	7:00	AM	F:\WINNT\system32\version.dll
0x778400 00	0x3C000	ClientSide Caching UI	5.00.2195.2959	5/4/2001	2:05	PM	F:\WINNT\system32\cscui.dll
0x779B0000	0x9B000		2.40.4517.0000	5/4/2001	2:05	PM	F:\WINNT\system32\OLEAUT32.DLL
0x77A500 00	0xF6000	Microsoft OLE for Windows	5.00.2195.4439	10/30/2001	7:57	AM	F:\WINNT\system32\OLE32.DLL
0x77D400 00	0x70000	Remote Procedure Call Runtime	5.00.2195.4266	10/30/2001	7:57	AM	F:\WINNT\system32\rpcrt4.dll

0x77DB0000	0x5C000	Advanced Windows Windows 32 Base API	5.00.2195.4453	10/30/2001	7:57	AM	F:\WINNT\system32\ADVAPI32.DLL
0x77E10000	0x64000	Windows 2000 User API Client DLL	5.00.2195.4314	10/30/2001	7:57	AM	F:\WINNT\system32\USER32.DLL
0x77E80000	0xB5000	Windows NT User API Client DLL	5.00.2195.4272	10/30/2001	7:57	AM	F:\WINNT\system32\KERNEL32.DLL
0x77F40000	0x3C000	GDI Client DLL	5.00.2195.3914	10/30/2001	7:57	AM	F:\WINNT\system32\GDI32.DLL
0x77F80000	0x7B000	NT Layer DLL	5.00.2195.2779	5/4/2001	2:05	PM	F:\WINNT\system32\NTDLL.DLL
0x78000000	0x46000	Microsoft ® C Runtime Library	6.01.9359.0000	10/30/2001	7:57	AM	F:\WINNT\system32\msvcrt.dll
0x782F0000	0x23F000	Windows Shell Common DLL	5.00.3502.4718	12/3/2001	5:35	PM	F:\WINNT\system32\SHELL32.DLL

The second phase of tests validated whether the data extracted matched the original files. This was done by conducting extracting the file from a known good CD-ROM and then run an md5 checksum program against the original file and comparing it to the file extracted. This test was run on two different CD-ROM devices. The md5sum utility is part of the CygWin utility set. The program was first run against the file on the optical media and then against the file after it was extracted from the optical media. Please reference the following table for a comparison of the files and the md5 results.

Md5 Checksum comparison

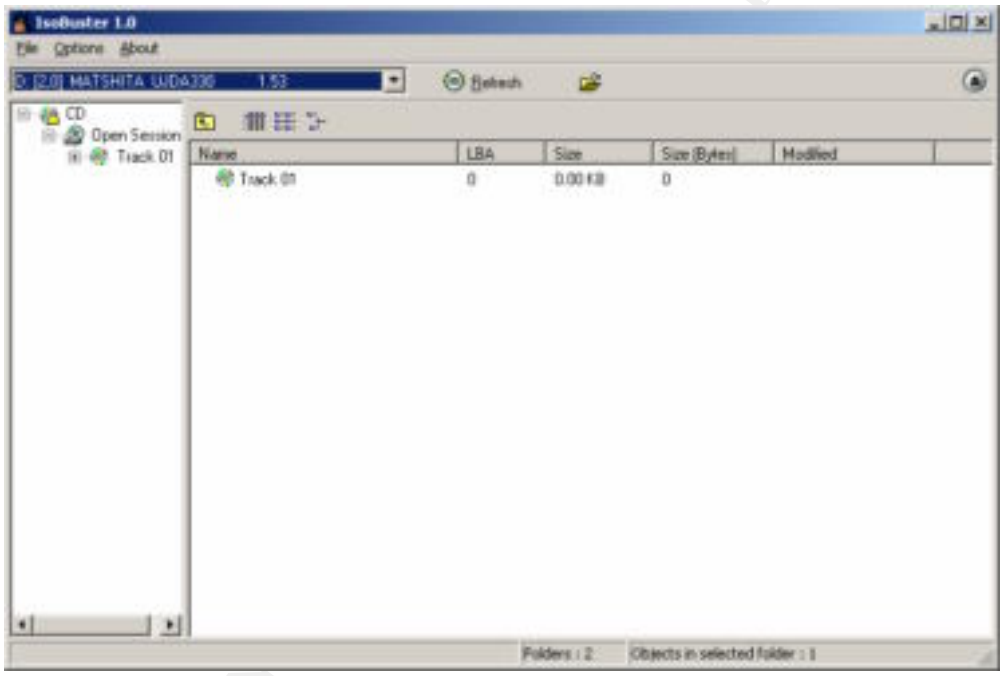
File name/CD Device	**Toshiba DVD-ROM SD-2502 1915**	**Mashita UJDA 330 1.53**
Custom.sh F.I.R.E CD	\a80f66dd15c0cbe483bcddb07f2c1db8 *e:\\images\\custom.sh	\a80f66dd15c0cbe483bcddb07f2c1db8 *d:\\images\\custom.sh
Custom.sh F.I.R.E Extract	a80f66dd15c0cbe483bcddb07f2c1db8 *custom.sh	a80f66dd15c0cbe483bcddb07f2c1db8 *custom.sh
video_ts.ifo DVD	\24dc0e02bffd98a4840177260056cef9 *e:\\video_ts\\video_ts.ifo	N/A
video_ts.ifo HD	24dc0e02bffd98a4840177260056cef9 *video_ts.ifo	N/A

The results from these two tests indicate this tool could be used as a reliable forensic tool. The next set of tests would indicate whether this tool could reliably recover data from media.

Functional Aspects of ISObuster

The CD hardware device type is automatically sensed when the program is loaded. The media can then be inserted and identified. This identification process may take any where from 15 seconds to two minutes. If the media is not readily recognized after two minutes, the refresh CD icon can be clicked. If the media is blank or recovery is not possible, the media will not be read or it will be listed as blank. Insertion of the optical media to be investigated is straightforward. Once the media is recognized the lower window indicates the presence of the media.

Please reference the screen capture below as an illustration.



The previous screen capture shows there are three titled options at the top of the program window. The **File** option allows the user to either open an image file such as an .ISO image or exit the application.

The **Options** option provides three choices. The first choice in the **Options** drop down is **Communication**. This choice is dependent upon whether the device uses an ASPI or SPTI as the communication interface.

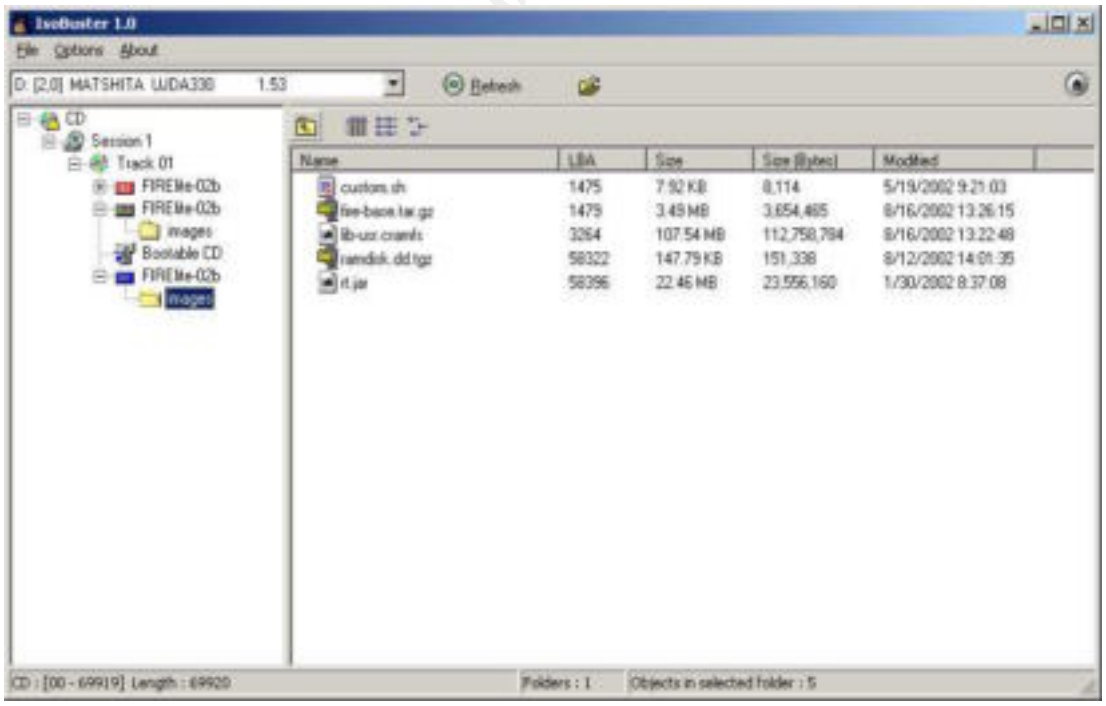
The second choice is **File Systems**. This selection allows the user to change the type in a number of different variables. These variables are **Scan Options** or **Text Conversion**. The scan options are Big Endian/Motorola vs Little

Endian/Intel. This option can usually be left as the default. This becomes more important for the ability to read and translate some of the non-ISO images or non-ISO 9660 optical media. The **Text Conversion** offers the choice of the two defaults of ANSI or Unicode. The Unicode selection can be changed on demand. The default is Unicode page type 437.

The third **File Systems** choice is **Languages**. If the version downloaded is the International version, the various language choices are available.

The **About** Option is to provide a registration capability or provide version number. As stated previously, the Help file is a standalone option. It is not integrated into the system. The Help file provides assistance primarily for standard use and recovery.

The CD information is displayed by clicking on the CD session icon in the lower left hand window. If there is more than one type, it will be displayed as well. The F.I.R.E. CD information is illustrated in the screen capture below. It shows that the CD is a bootable CD, the ISO information, the RR information, and the Joliet information. The ISO icon information displays the standard 8.3 filename structure. The Joliet extension displays long file name (LFN) structure. The Bootable CD icon displays the boot files and information. The RR icon provides the Rock Ridge extension information. This information is displayed by clicking on the color sub-icon. Right clicking on the sub-icon displays the properties of the specific image type.

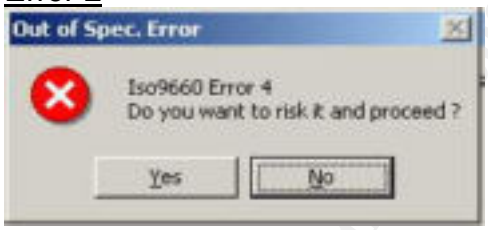


If there is a problem in reading the media, an error message will appear. Example error messages are displayed below. The first error is the most common. This will display for Buffer Underrun CDs. Clicking on the OK button usually allows for a successful display of the media's contents. The following three errors indicate a more serious issue. The second error is also common. It indicates the ISO system error may be unreadable. The third and fourth errors indicate errors in the index or directory of the media. Error three indicates the extended Joliet image information is unreadable. Error four indicates problems with the ISO image directory

It may be necessary to click OK multiple times on error messages to begin an investigation of the media. Recovery may be possible only in ISO format and not Joliet. It may be possible in the end to only display the contents of the media. Recovery of the contents may require manual recovery of the logical blocks of the file.

Error 1

Error 2



Error three

Error four

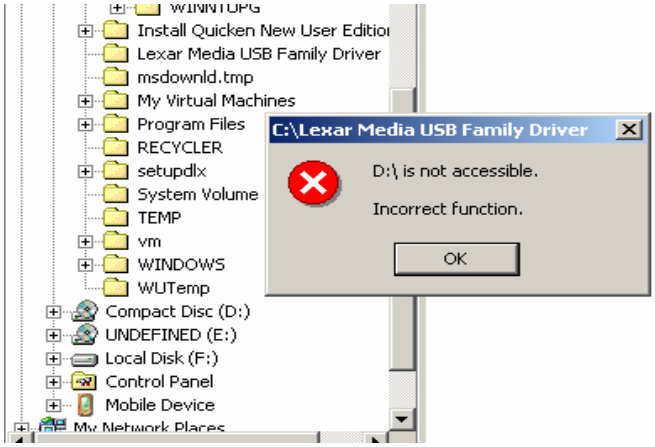
Functional Forensic Testing:

The previous test results in this document validated it could be used as a forensic tool. This first set of tests did not provide direct forensic verification that it could retrieve data from media thought to be bad. There was a need to create a credible testing process with nearly identical media. This was done by comparing known original files with the identical files recovered from "bad" media. Due to limitations of equipment, a quantitative test was not completed. An effective quantitative test needed to be completed with multiple independent devices and larger quantities of media.

The process chosen was to compare two known bad CDs. A file on bad CD number one and bad CD number two was chosen and then compared to the original file located on the hard drive. This test was run twice, with the media located once in the DVD-ROM and then the CD-RW drives. The recovered file was extracted and then completed an MD5 checksum. The checksum results would be compared between the recovered file and the original file.

Three CDs were used for the extraction test. The first CD was the baseline test media. It was normal without errors. The second test CD was the result of a Buffer Underrun. The third test CD was the result of stopping the CD burn process prior to completion. These provided media similar to what might be reviewed during a forensic investigation of acquired media.

In each instance of the test, the test CD was checked to see if it could be read by the DVD-ROM and/or the CD-RW drive. The results for each device had the following error message from Microsoft Explorer.



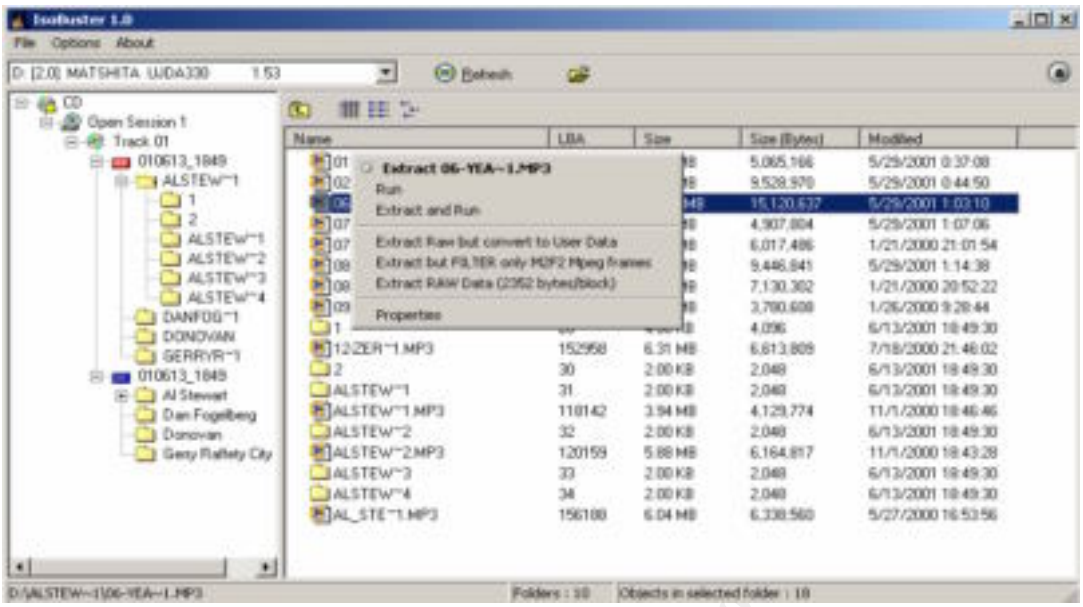
The PC was restarted prior to each test with no network connection. Only five programs were running during the test: Microsoft Explorer, Microsoft Paint, Cygwin md5sum, Microsoft NotePad and ISObuster. All other applications and non-critical system applications were stopped.

Example Process:

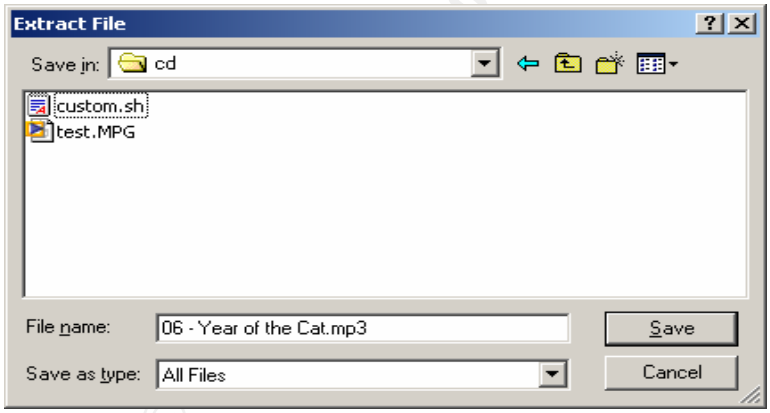
The following is a screen capture process that was done during the evaluation and test recovery from CD number two.

Step A – Test the value of opening the CD using Microsoft Explorer. The same value was received whether the CD was placed in the CD-RW or the DVD/CD unit.

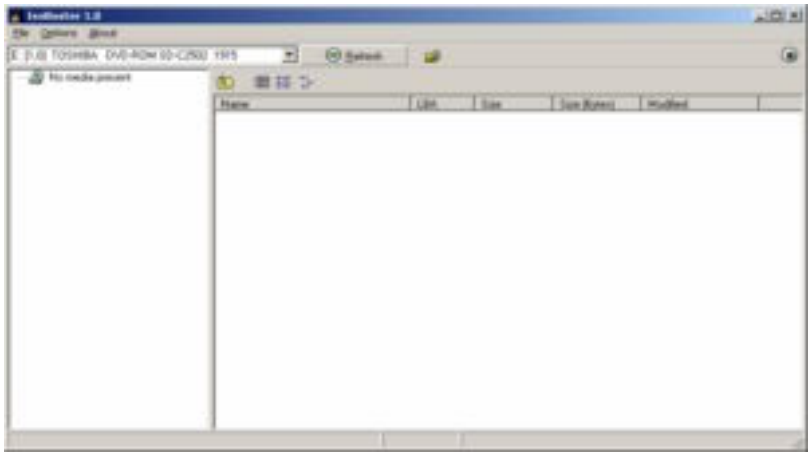
Step B – Start of ISObuster using Matshita CD-RW device. The CD was successfully read and files could be extracted. Both ISO and Joliet subfolders can be read.



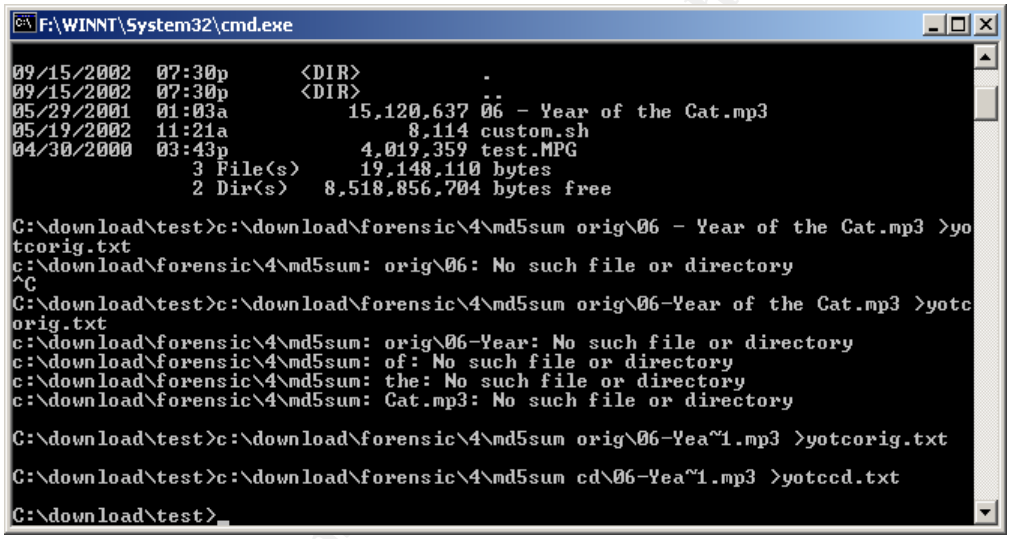
Step C – The file was extracted from the CD by right-clicking the file name to be extracted. A popup window appeared. Shown is an extraction pop-up Window in the ISO sub-folder. The Joliet pop-up differs by providing the Long File Name for the file. The file extracted from the ISO will not provide LFN support. The file was extracted from the Joliet subfolder to assure there would be identical naming of the original file and the extracted file. A file save window appears to direct saving the file to the desired location. The file was saved to the c:\download\test\cd subdirectory. The following screen capture illustrates this action.



Step D - When the same CD was inserted into the Toshiba DVD/CD unit, the following value was received. This indicated that it was not an effective testing device for this specific CD.



Step E – The CygWin MD5sum utility was run against the original file and the extracted file. It was discovered that the MD5sum would not work with LFN. Standard 8.3 names were substituted.



Step F – The MD5 Checksum value is verified. These result values would be copied into the results table.

This same process was performed for all tested CDs. The example process did not indicate any unusual issues other than the md5sum inability to handle long file names.

Analysis and Results:

Please review the following tables for the results of the recovery. Each table describes the results from the recovery attempts and comparison with the original file.

CD #1 F.I.R.E CD (Normal CD, No Errors)

File name /CD Device	Toshiba DVD-ROM SD-2502 1915	Mashita UJDA 330 1.53
Internet Explorer	Read OK copied to c:\download \test\orig \a80f66dd15c0cbe483bcddb07f2c1db8 *origdvd\\custom.sh	Read OK, copied to c:\download \test\orig \a80f66dd15c0cbe483bcddb07f2c1db8 *origdvd\\custom.sh
Original File This file was on the Hard Drive and not the DVD	Taken from ISO image downloaded from Sourceforge.net site \a80f66dd15c0cbe483bcddb07f2c1db8 *iso\\custom.sh	N/A
Original File Extracted using ISObuster	\a80f66dd15c0cbe483bcddb07f2c1db8 *origedvd\\custom.sh	\a80f66dd15c0cbe483bcddb07f2c1db8 *origecd\\custom.sh
Extracted File using ISObuster	\a80f66dd15c0cbe483bcddb07f2c1db8 *dvd\\custom.sh	\a80f66dd15c0cbe483bcddb07f2c1db8 *cd\\custom.sh

CD#2 *06 – Year of the Cat.mp3* (Error Buffer Underrun)

File name /CD Device	Toshiba DVD-ROM SD-2502 1915	Mashita UJDA 330 1.53
Internet Explorer	Not Accessible	Not Accessible
Original File		Taken from personal CD rip MP3 compilation fd5c0b5e230b949cf2ea2a7ed73c791c *orig\\06-Yea~1.mp3
Extracted File	Not Accessible	\fd5c0b5e230b949cf2ea2a7ed73c791c *cd\\06-Yea~1.mp3

CD #3 test.mpg (Error on Directory table, Joliet Entry bad- Error # 3)

File name /CD Device	Toshiba DVD-ROM SD-2502 1915	Mashita UJDA 330 1.53
Internet Explorer	Not Accessible	Not Accessible
Original File from personal files	\f5afbdb5b34bed29666b3eb22a9d4ef1 *orig\\test.mpg	\f5afbdb5b34bed29666b3eb22a9d4ef1 *orig\\test.mpg
Extracted	\f5afbdb5b34bed29666b3eb22a9d4ef1 *cd\\test.mpg	\f5afbdb5b34bed29666b3eb22a9d4ef1 *dvd\\test.mpg

Efficacy testing:

A total of ten CDs were tested using this methodology. The success rate was higher when the Matshita CD-RW device was used over the Toshiba DVD-ROM/CD reader. There was a successful extraction of data from the CD-RW device on seven out of the ten "bad" CDs. The success for the DVD/CD unit was less effective. It was successful on only four of the same ten CDs. This would indicate the device used would have a direct effect on the ability to read the data. There were three CDs that were unsuccessfully read by either device. All three returned the previously displayed error number one in addition to the error number four.

The same file test.mpg was extracted five times. Each time, the MD5 checksum was identical. The file was also tested for functionality. The images from the mpeg file were identical to the original. This is an important issue in that it shows consistency in results.

The results of the testing indicate that when the media can be recovered, ISObuster can recover the data and extract the identical data. The product is not 100% successful, but in real world forensic scenarios, the validation that the data was identical can be just as important.

Presentation Process:

The presentation of the value of the program ISObuster would focus on the ability to tie data through MD5 checksums to the original data. This would be presented by showing three critical steps:
The data extracted was the same as the data on the original CD. This was a critical step. If the CD used for evidence was from a pile of "bad" CDs, it would be necessary to prove that the data will be forensically reliable whether the file was extracted from a good CD or a "bad" CD. The same illustrative process was described in first set of validation tests.

The data recovery methodology should be reliable. Screen captures of the process as well as table results must be consistent and not use poor scientific procedures.
The data extracted is forensically sound. The extracted data may not always be directly compared to the original data. Results must be consistent during the extraction to illustrate that the file was recovered from the same CD.
This information when presented in a logical and concise manner would show that the application ISObuster did forensically recover the data.

Conclusions:

This application is a useful forensic tool when faced with optical media discovered in an investigation. It can be used to recover data from CDs potentially considered bad. It can recover the data and provide a provable mechanism that the data from the original or from the acquired CD is the same.

The program does not have the capability to recover all data from CDs thought to be unrecoverable. This is an effective forensic tool for a forensic investigator who does not have a large budget. This tool offers the forensic analyst a good Microsoft Windows based tool to evaluate and recover CD and DVD optical media data.

Citations:

CygWin Utilities, MD5sum
http://www.cygwin.com/

F.I.R.E.lite Forensic CD
http://sourceforge.net/forum/forum.php?forum_id=203500

Russinovich, Mark and Coswell, Bryce, PMON 1997
http://www.sysinternals.com/ntw2k/freeware/pmon.shtml

Russinovich, Mark. Process Explorer 2002, v5.25
http://www.sysinternals.com/ntw2k/freeware/procexp.shtml

Russinovich, Mark. LISTDLLs 2000, v2.23
http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml

Van Hove, Peter, ISObuster V1.0 The ultimate CD/DVD Data Recovery Tool
http://www.smart-projects.net/isobuster/

SANS GIAC CERTIFIED FORENSIC ANALYST
Practical Assignment – Part 2: Unknown Binary Analysis
Author Steven Dietz

Exercise Abstract:

An unknown binary was discovered on a Red Hat Linux version 7.3 server at ABC Enterprises. A forensic examination of the unknown binary was completed. The file was forensically examined for its purpose, history, and any other information for a potential criminal or legal investigation that would reference the discoveries. This was the exercise for the second part of the SANS Certified Forensic Analyst practicum.

Materials provided:

A compressed zip file called sn.zip. This file contained two compressed files, sn.dat and sn.md5.

Environment:

All analysis was completed on an HP Pavilion XU155 notebook. This notebook used Microsoft Windows 2000 Professional operating system as the base operating system. All tests and examination processes were conducted on a virtual network using VMWARE version 3.1. The internal virtual network was created on the 192.168.2.0/16 network. The forensic analysis virtual machine was using the Mandrake 8.0 operating system. This machine was assigned the 192.168.2.5 address. The secondary virtual machine was set up using a Red Hat Linux 7.3 OS. This was considered the compromised device. This machine was assigned the 192.168.2.2 address. The Windows 2000 machine was assigned the 192.168.2.1 address.

Scenario and Test Assumptions:

In a real investigation, there would be no assumptions, only facts of discovery. As this file was part of an exercise I made the following example assumptions and statements for this exercise.

The file sn.dat was discovered on the Red Hat virtual machine. It was discovered during a general review of the system. The location of the file was found under the directory **/usr/lotus/license**. This location was unusual in the fact that there was no installation of IBM/Lotus products on this machine. This was where the unknown binary was unzipped. Once this was complete, the investigation and analysis was considered to begin. The following was the process I undertook to discover the purpose of the file. The preparation and testing of the unknown binary was divided into two locations, the forensic read-only image of the

"compromised" Red Hat Linux v7.3 virtual machine and test directories on the Mandrake v8.1 virtual machine. The Windows 2000 Professional host was also used as an investigative tool.

Discovery and Initial Actions:

The Red Hat machine was only used for placement of the unknown binary. All investigative efforts took place on the Mandrake machine. All other efforts were stopped on the potentially compromised Red Hat Machine upon discovery of the file. This action was done to leave the unknown file and any other processes in place to minimize any potential compromise or notification to who placed the file. The device was also left running in order to not notify the potential intruder. Investigative testing was not done on the original machine. This was done to prevent any compromise of evidence on that device and make the original machine data inadmissible in most legal processes.

A packet capture session of the network protocol analyzer Ethereal was started and configured to monitor any inbound or outbound activity of the Red Hat device on the Windows 2000 machine. This information provided added detail in the event of any unauthorized access or use.

The first task and goal was to create and mount a forensic image of the Red Hat physical drive on the Mandrake virtual machine. The first step was login to the Red Hat virtual machine as an existing user, *sdd*. A CD-ROM mount was created to use a previously created forensic CD. This CD contained precompiled static binaries to use in any forensic investigation. The initial static binary used from the CD-ROM was SU. This provided the user *sdd* the ability to create a substitute shell as the user root.

The following steps completed this task. The program Cryptcat was run from the CD-ROM creating an encrypted data transfer tunnel between the Red Hat machine and the Mandrake machine. Cryptcat is a program identical to the network utility NetCat. NetCat is a UNIX utility to read and write data across network connections. The commands for Cryptcat and Netcat are nearly identical. The difference between the two programs is the ability for Cryptcat to create an encrypted tunnel. The program Cryptcat is also loaded on the Mandrake device to allow the transfer.

The Cryptcat program was used in coordination with the program DD. DD is a utility that allows the copying of data from one location to another. DD offers granularity from file to the complete physical disk. The commands DD and Cryptcat were also run from the forensic CD-ROM. The commands used to transfer the data are listed below:

A1:- Mandrake:

```
$ cryptcat -l -p 21756 > /home/sdd/temp/test.img
```

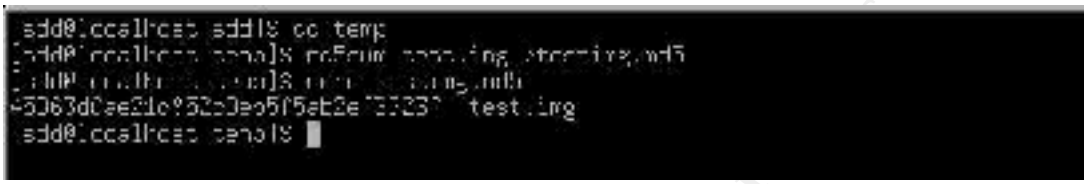
A2:- Red Hat:

$ dd if=/dev/sda1 | cryptcat 192.168.2.119 21756

An md5 checksum program called md5sum was run against the test.img file and redirected into a text file when the file transfer was complete. This was run on the Mandrake machine. The command to create the MD5 hash was:

$ md5sum /home/sdd/temp/test.img > testimg.md5

The resultant value is listed below.



Once the file image had an MD5 hash value created and stored, the drive image was mounted. The command to mount the drive image was:

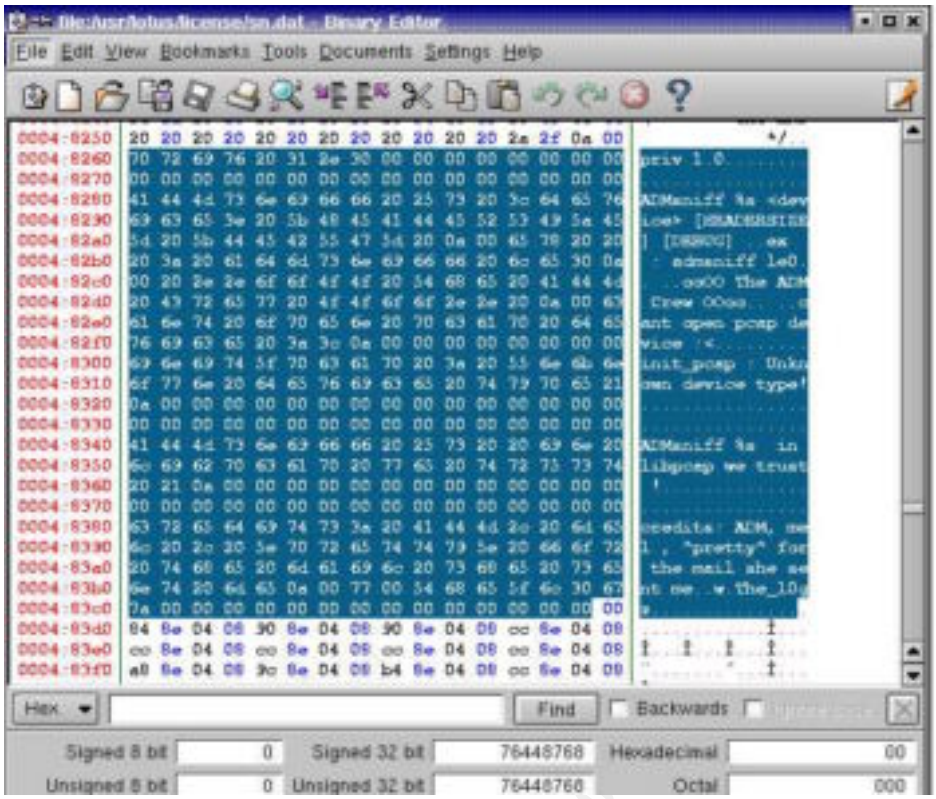
$ mount –ro,loop,nodev,noexec,noatime /user/temp/test.img /mnt/test

These variables mounted the file as a virtual drive volume in a read only mode with no execution possible, or adjustment of the access times. The direct analysis of the file sn.dat began. The analysis steps were completed on the Mandrake virtual machine.

Initial Analysis Steps:

This analysis was done on a clean standard build of both the Mandrake operating system and hard drive. It was important to remember that any execution analysis of a binary file would compromise that build. When the analysis was complete, the investigative machine was archived for evidentiary use. The operating system was considered contaminated and not reused.

There are many different processes to initially analyze an unknown file. The first process I did was view the file and investigate the contents. I used the application Mandrake Binary Editor. This was done on the read-only mount image. Any attempts to modify the data would be denied. The following text was discovered in the initial analysis. This text was modified for viewing ease. Please reference the following screen capture for original view. The important values were found at 0004: 8280 through 0004: 83cf



Text captured from unknown binary:
* The END */ priv 1.0
ADMsniff %s <device> [HEADERSIZE] [DEBUG] ex : admsniff le0
..ooOO The ADM Crew OOoo..
cant open pcap device :< init_pcap : Unknown device type!
ADMsniff %s in libpcap we trust ! credits: ADM, mel , ^pretty^ for the mail she sent me w The_l0gz

The initial text captured from the unknown binary indicated this program may be a program called ADMsniff. I went to the Internet and began a **Google** search for keywords discovered in the file. There were approximately 115 hits with the keyword search criteria of: admsniff, admcrew, and linux. I reviewed many sites for further information. The information at each site indicated ADMsniff was a packet sniffing program.

The web site COTSE.com http://www.cotse.com/tools/sniffers.htm as well as the SANS ID FAQ article: "Why your switched networks isn't secure" by Steven Sipes http://www.sans.org/newlook/resources/IDFAQ/switched_network.htm described ADMsniff as a: "Sniffer for SunOS and Linux".

The information discovered from the ADM crew web page at http://adm.freelsd.net/ indicated a potential hacking site maintained a group called ADMcrew. The link for the FTP site: http://adm.freelsd.net/ADM/ yielded a

link to a gzip file called ADMsniff.tar.gz. I downloaded a copy of this file from this site. This file had a creation date of 30-Jun-1999 18:10 and a stated size of 128k. I reviewed other sites for comparison. All sites offered the same version and size for download, V08 and 128K. The contents of the download file were source code and instructions for a packet sniffer based on libpcap. This corroborated the information discovered in the initial hex view. This information gave insight to the unknown binary, sn.dat, was a sniffer. The information in the source code was used to compare with to the discovered file.

The gzip file contained the following files.

File	Modified	Size
ip.h	5/7/1999 7:27am	486
tcp.h	1/19/1999 7:45am	1491
bpf.h	1/19/1999 7:45am	8447
pcap.h	1/19/1999 7:45am	4908
Makefile	5/30/1999 6:23am	729
libcap-0.4.tar	5/7/1999 7:33am	487,424
thesniff.c	5/11/1999 4:52pm	8432
Readme	5/30/1999 6:24am	1072

A comparison of text within ADMsniff source code file thesniff.c and text found in the sn.dat was identical in many instances. These both contained identical text

-Ascii text from sn.dat

ADMsniff %s in libpcap we trust ! credits: ADM, mel , ^pretty^ for the mail she sent me w The_l0gz

-Source code from the file thesniff.c found in the tarred gzipped file ADMsniff.tar.gz

```
printf ("ADMsniff %s in libpcap we trust !\n", VERSION);
 printf ("credits: ADM, mel , ^pretty^ for the mail she sent me\n");
```

One concern was the text information in sn.dat was a false lead placed in the file to mislead. One other concern was the file might be something in addition to the packet sniffer. If this were true, any compile of ADMsniff source code would not match the file size of sn.dat. There were other potential variables that would potentially cause file size to be different than a standard compilation of ADMsniff. This included the use of a newer libcap than provided with the original file. Another variable was the flavor of the operating system that sn.dat was compiled. Further research was necessary on testing the rogue file.

It was time to inspect other attributes of the file. The file attributes were inspected on the read-only Red Hat image volume. The following three commands were run to discover more information about the file: file, ls, and stat. The results of these commands were also redirected to individual text files. Please reference the following screen capture.

The following information was discovered about the file: sn.dat. The ls command indicated the sn.dat file was created on this machine on Aug 31,2002 at 14:06 CDT with a file size of 399124 bytes. The owner was root.

The file command indicated the sn.dat file was a stripped, statically linked, executable file. This indicated the file was a stand-alone executable program that had the debug information removed. The static linking would allow sn.dat not make any calls to existing system libraries.

The stat command reiterated the information provided from the ls command and identified that it was last modified, Thu, Apr 11,09:29:58 2002. The file was last changed, Aug 31, 14:06:01 and last accessed, Aug 31, 14:15:50. It identified the creation mode with the default value of 0666.

The permission value 0666 is the default value for all files created in Linux and Unix. The value of an executable file is normally 0555. A value of 0666 is not normally assigned to an executable file.

The group and user owner was root. This indicated the user had potentially compromised the root account by some means. The change date indicated the file was compiled elsewhere and not on this device. This device was created on August 14 2002 while the application was changed/compiled on April 11, 2002.

The MD5 checksum matched the provided checksum. The file sndat.md5 was also in the **/Lotus/license** directory. This was the file that all MD5 checksums were made against.

The test of the sn.dat as an executable was not conducted directly from the forensic image. The forensic image was mounted as read-only and noexecute. This required the sn.dat to be copied to the regular read/write volume of the Mandrake device. The file sn.dat and sndat.md5 were copied to the **/home/sdd/temp** directory

In an actual forensic analysis, this test would not be done on the same machine as the location of the forensic image. All attempts would be made to provide an air-gap between the potentially compromised device, the original forensic image and the test environment.

The location of the file in the /lotus/license directory indicated that the individual placed the file in an attempt to hide the file. The file was to be temporarily assumed and defined as a packet sniffer until proven otherwise. This decision was based upon the collated information from these pieces of evidence.

The text information discovered within sn.dat named it as the program ADMsniff The comparison of the text within the ADMsniff source code file thesniff.c and the text found in the sn.dat. These both contained identical text
-Ascii text from sn.dat
ADMsniff %s in libpcap we trust ! credits: ADM, mel , ^pretty^ for
the mail she sent me w The_l0gz
-Source code from the file thesniff.c found in the tarred gzipped file
ADMsniff.tar.gz

```
printf ("ADMsniff %s in libpcap we trust !\n", VERSION);
printf ("credits: ADM, mel , ^pretty^ for the mail she sent me\n");
```

The apparent attempt to hide the file in a non-binary program location and giving it a non-descript name

The time information provided from the forensic image was weighed against the activities of the system administrator who discovered the file. The system administrator who discovered the file did an md5 checksum of sn.dat redirected to a text file and zip the files to a file called sn.zip when it was discovered.

Zip is a compression program used to store files. It is a common industry solution for data compression and file storage.

The zip task changed the “change” time values displayed by the Stat command results. The stat results value for “access” time indicated that the file may have been in use at the time the file was discovered. This value was within ten minutes of the change time. According to the system administrator who created the zip file, he did not remember making any access to the file, only creating the md5 checksum file and copying the file into the zip file.

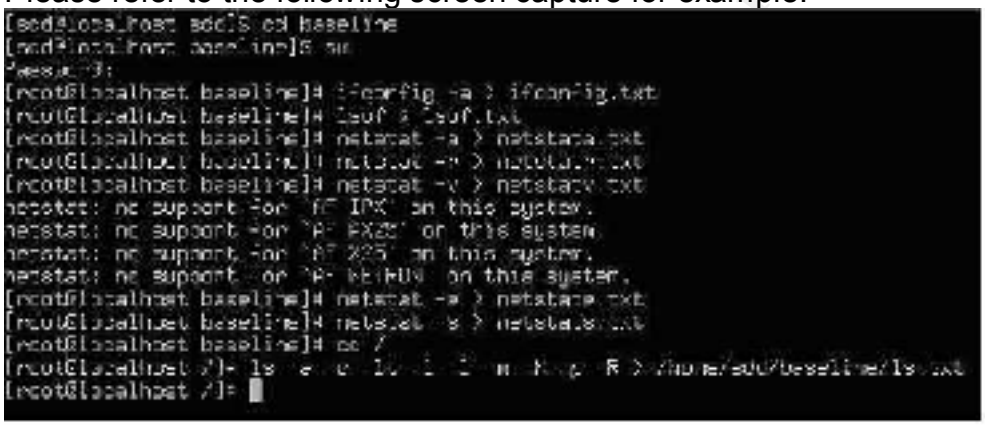
There were forensic evidence problems in the action taken by the system administrator. The system administrator should have left the file untouched upon discovery. The action of zipping the file had a direct effect of creating accurate MACtimes. Other information was to be used to corroborate its use. This information was documented in the forensic inspection file.

Unknown binary execution analysis:

The functional test of the sn.dat was made by copying the file from the forensic image to the directory **/home/sdd/temp**. An md5 checksum was run again and compared against the file after copying to verify copying the file did not modify the value. The cryptographic values were identical.

Prior to conducting the execution test of sn.dat, a baseline of the Mandrake device network status and configuration needed to be created. This was accomplished by running the following commands and redirected the text values to a group of text files. The commands IFCONFIG, LSOF, NETSTAT and LS were run with variable options to capture detailed information. The redirection of the output sent to specific unique named text files. This provided evidentiary information that would be directly referenced later.

Please refer to the following screen capture for example:



The focus for the text files was to baseline all files, directories, and network status. If the unknown binary was a packet sniffer, it would make port and interface changes, create new directories and files or changes to files and directories. When the file testing was complete, the same baseline commands were run again and placed in a separate data location.

An Ethereal packet capture session was begun and directed against the Mandrake device IP address by Windows 2000 base device. The protocol capture was to discover if sn.dat made any communication attempts to other sources when the process was started.

The application utility strace was also run and logged to identify any system calls about the executable was run. The network monitoring tools portsentry and logsentry were enabled to monitor if there was any unusual tcp/udp activity or unusual actions while the program sn.dat was executed.
The two tools, portsentry and logsentry are made by Psionic HTTP://WWW.PSIONIC.COM The web sites states: “PortSentry is a program designed to detect and respond to port scans against a target host in real-time”.

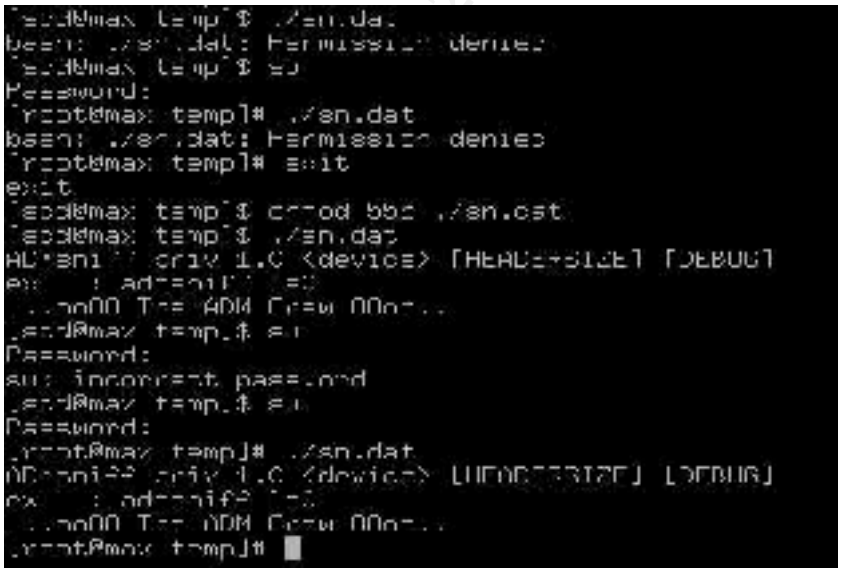
"LogSentry (formerly Logcheck) automatically monitors your system logs and mails security violations to you on a periodic basis." These tools offer additional monitoring capability of any unusual actions by an unknown binary.

There were two modes of testing to execute this application. The first attempt was made to execute it without the execution flag set This attempt was tried with a basic 500 level user and then as a user SU'd to act as the user root.

The second attempt was identical to the first except the sn.dat file would have the execution attribute changed to x. If the execution attempt was successful during the first attempt mode, the attempt in changing the execution attribute would still be attempted.

These tests verified two different issues. The first issue answered whether the program already had been run and executed. This seemed remote as the information described in the discovery and file attribute section. The second issue verified whether the program would run for someone without root equivalence. This information provided insight to how the binary executed in addition to be deployed.

The first portion of the screen capture below showed the results for the three attempts execute it with the execution flag not set. None of these were successful. This indicated two possibilities. Either sn.dat had not been executed yet or the execution flag was turned on or off by another program or done by manual effort. This information required research beyond the scope of this individual program analysis.



The second portion of the screen capture showed after changing the execution flag for the file to 555, the binary title information became available. The basic user *sdd* did not find or load the necessary pcap packet capture driver. When the

user was changed to root, there was success. The credits information and the motto both appeared. The successful execution indicated that sn.dat was the packet sniffer, ADMsniff.

The Ethereal packet captures showed there was no initial attempt by the program for external communication. External communication attempts by sn.dat were not fully validated until the program was run. The information displayed by the execution of sn.dat appeared to be a standard program requiring variable input. i.e. the network device to monitor.

When the STRACE with the – c variable was checked, the display information indicated the following system calls were made, Network, userid and groupid. This was compared in the next step when the network device was input as the variable.

```
[sdd@max temp]$ more stracec.txt
execve("./sn.dat", ["./sn.dat"], [/* 44 vars */]) = 0
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 75.99    0.000867         289         3           write
  6.66    0.000076          25         3           fcntl64
  5.78    0.000066          66         1           munmap
  2.54    0.000029          10         3           brk
  1.93    0.000022          22         1           old_mmap
  1.84    0.000021          21         1           getegid32
  1.49    0.000017          17         1           fstat64
  1.14    0.000013          13         1           uname
  0.96    0.000011          11         1           geteuid32
  0.88    0.000010          10         1           getuid32
  0.79    0.000009           9         1           getgid32
------ ----------- ----------- --------- --------- ----------------
100.00    0.001141                    17           total
[sdd@max temp]$
```

The Psionic Portsentry and Logsentry indicated no unusual activity. The logs in both instances were blank.

The next step was executing the sn.dat with the eth0 device variable. Once again, this was done at the 500 user level and then as a super user using SU.

The following screen capture illustrated the results.

```
[sdd@max sdd]$ cd temp
[sdd@max temp]$ ./sn.dat eth0
[sdd@max temp]$ su
Password:
[root@max temp]# ./sndat eth0
bash: ./sndat: No such file or directory
[root@max temp]# ./sn.dat eth0
ADMsniff priv 1.0  in libpcap we trust !
credits: ADM, mel , "pretty" for the mail she sent me
```

The external protocol analyzer captures showed there were no attempts for any unusual connection requests or SMTP processes. There were no system calls in the original source code for external communication as well. This would provide another parallel with the original program. There were indications that this was a single purpose program.

There was a need to corroborate this information and check for file changes. This testing involved running the baseline inspection commands again (LS, LSOF, IFCONFIG and NETSTAT) and then the results compared to the original baseline information. The commands were run in a separate terminal window from the terminal sn.dat sniffer was executed. These commands indicated the following change to the eth0 device. It had changed from a basic "BROADCAST NOTRAILERS RUNNING" to "BROADCAST RUNNING PROMISC MULTICAST". This information indicated the program had changed the adapter to promiscuous mode.

The original source code thesniff.c was referenced. It stated it would redirect captured traffic to a file called The_L0gz. There was no indication of where this file would be created. The code snippet below was referenced.

```
#ifndef COMPRESS
        filez = fopen ("The_l0gz", "w");
#else
        signal(SIGHUP,hup_handler);
        signal(SIGTERM,term_handler);
        filez = gzopen("The_l0gz","wb");
#endif
```

The Find command was executed from the root directory, there was only one instance of the file "The_l0gz" discovered on the mounted volumes. It was found in the same directory as the execution test sn.dat file.

An important note, it did not find the file in the forensic image that had been previously mounted. The absence of the file indicated that the file The_l0gz had either not been created or the file had been removed. This required further investigation on the forensic mounted volume.

There were protocol tests run to test the capability of sn.dat aka ADMsniff. The newly discovered "The_logz" file was reviewed and compared to the Ethereal logs during the same period.

The following processes were attempted across from the local address of 192.168.2.5 to the Windows 2000 Professional at 192.168.2.1 An ICMP Ping command was sent from one device to the other device. These commands were run from the command line of each device. A telnet session was attempted from both devices. An FTP session was attempted to each device. The following table compared what was logged on each session. The sn.dat did not seem to capture

any ICMP traffic in its The_l0gz file. Only telnet and the FTP session attempts were logged.

There was a need to conduct an extended test whether there were any hidden external communication programs in addition to the discovered ADMsniff. ADMsniff was run for an hour with a batch telnet attempts started and stopped to the Mandrake device from the Windows 2000 Pro device. There were no communication attempts made or identified. This lack of communication during the testing would not conclusively make it only a single purpose executable.

Comparison of log results between Ethereal and sn.dat

Device/ Process	sn.dat (ADMsniff)	Ethereal
ICMP Ping from 192.168.2.1	Not Captured	Captured
ICMP Ping from 192.168.2.5	Not Captured	Captured
FTP Session from 192.168.2.1	Captured	Captured
FTP Session from 192.168.2.5	Captured	Captured
Telnet Session from 192.168.2.1	Captured	Captured
Telnet Session from 192.168.2.5	Captured	Captured

The next step was to review the forensic image for The_l0gz file. The forensic image was strings searched for the existence of the The_l0gz file. The utility application Linux Disk Editor (LDE) was used to search for any occurrence of the text, The_l0gz.

Linux Disk Editor is a hexadecimal editor that can be used to hex edit a Linux volume.

The search found one occurrence of the string, The_l0gz. This text string was in the original sn.dat file. This indicated the file had not been created or all instances had been removed.

ADMsniff executable and sn.dat:

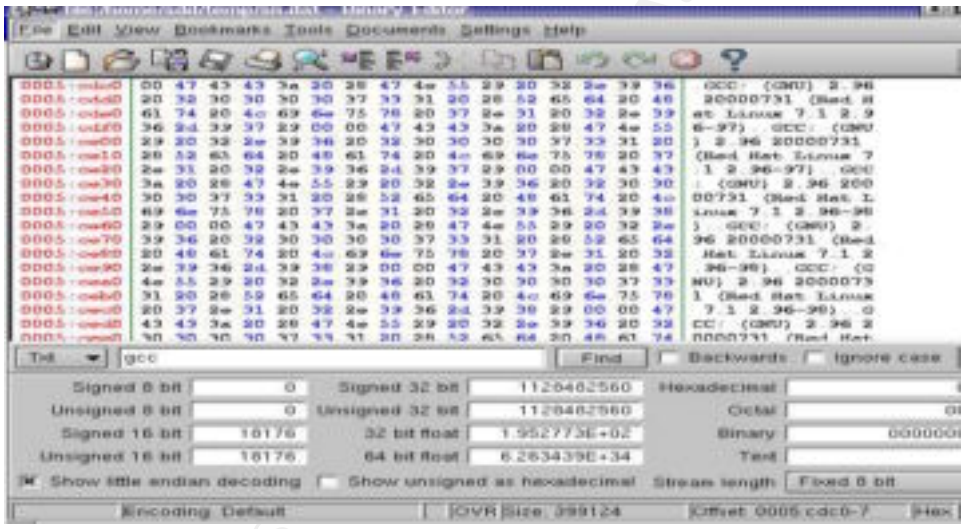
The final test and comparison of sn.dat and ADMsniff was to compare a compiled executable of each program. This required compiling the source code found at the ADMcrew FTP site. As stated previously, it was possible that sn.dat and

ADMsniff were the same file but be different due to different compilation of libraries, Operating Systems, and similar issues. It would require a machine functionally identical to the original to be able to get a similar value.

The compilation of ADMsniff used the standard gcc that is bundled with Mandrake 8.1 and the information provided in the ADMsniff.tar.gz file. This information included the instructions in the Readme file. As stated in the Readme, it was necessary to build a zlibc.

GCC is the Gnu C Compiler program provided with most Linux builds. Zlibc is a zip compression library to provide compression capability to programs compiled in C.

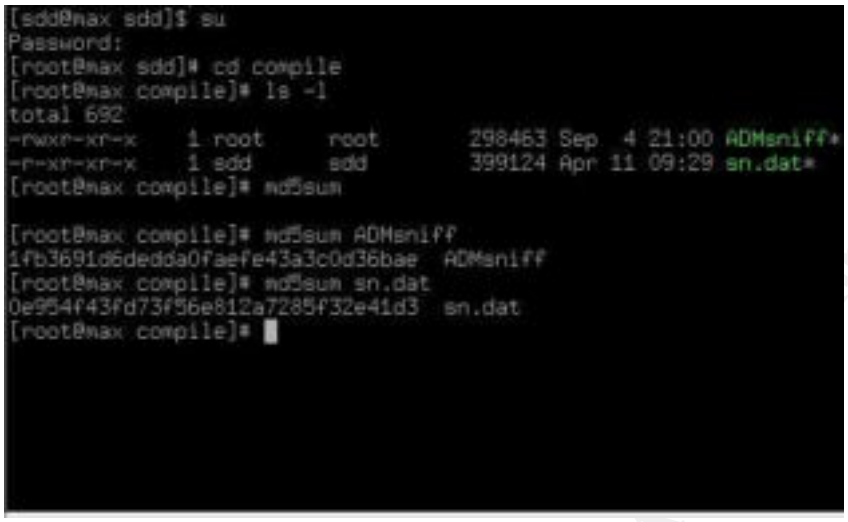
I used the application Mandrake Binary Editor program again to search for the version of GCC that compiled sn.dat This search provided both GCC version plus the potential version that compiled the program. The information provided indicated that it used GCC version 2.96 and probably compiled on a Red Hat Linux 7.1 device. This value was found at 0005:cdd0



The program zlibc was downloaded and then installed from the site http://zlibc.linux.lu/ . The version from the site was version 0.9j. This was an example of a potential variable that any compile of ADMsniff might not match sn.dat.

The original discovered executable, sn.dat, was a static build with the debug information stripped away. This was another important variable in creating a comparable copy of ADMsniff. The Clean command strip was run to remove debug info from the executable. The Makefile instructions were modified to allow zlibc calls and the “-static” variable to make it a standalone binary and create a potentially successful executable.

As predicted, the ADMsniff file compiled did not have the same size, or md5hash value. The screen capture below compared the compiled ADMsniff and sn.dat in both size and MD5 hash values. Functional comparisons did not provide adequate proof that this was the program ADMsniff.



Legal Considerations:

A packet sniffer is used in a normal work environment to identify and troubleshoot network problems at the protocol level. ADMsniff is a type of packet sniffer similar to most basic TCP/IP protocol packet sniffing packages. The surreptitious use of a packet sniffer would be used for three primary purposes:

- UserID/password capture and compromise,
- Raw data acquisition for data theft
- Network/system configuration mapping for future compromises.

Any use of a packet sniffer in an unauthorized manner would be in violation of both state and federal laws. The federal laws are commonly known as the Wiretap Act (USC Title 18, sections 2510-2520) and the Electronic Communications Privacy Act (USC Title 18, sections 2701). Both make the unauthorized use of a packet sniffer a federal offense. The unauthorized placement and use of a file like ADMsniff or sn.dat would also be a violation of state law of Kansas Statute Electronic Trespass Act (State Law No. 21-3755). This law deals will unauthorized computer trespass and misuse.

The use of a sniffer without authorization is defined for each federal law. Violation of the Wiretap act would include fines and potential imprisonment of up to five years per violation. Important sections of Wiretap Act state under Section 2511:

The Pen and Register sections of the Electronic Communications Privacy Act state under Section 2701 that violations of inappropriate or unlawful access to stored headers or communication registers would include fines and imprisonment of up to two years per violation.

The Kansas Electronic Trespass act makes the unauthorized access or trespass a Class D state felony or a Class A misdemeanor dependent upon intent with fines and imprisonment of up to one year per citation.

Unknown Binary summary:

The file, sn.dat, was a stand-alone executable file that was statically linked with debug information removed. The file size is 399124 bytes. It was identified as a stand-alone version of the packet sniffer, ADMsniff. The owner of the file was root when the file was placed on the inspected device, Aug 31, 2002. The file was last modified and potentially compiled on Apr. 11, 2002. This indicated the file was created prior to placement on the device. This was due to the creation date of the device was August 14,2002.

The file was validated as ADMsniff by analysis comparison of the source code discovered on the Internet and downloaded with the ASCII contents of the sn.dat file. This information was reiterated by the results message when the file was successfully executed.

The verification sn.dat was a packet sniffer was proven during execution testing of the binary. The log file for sn.dat created was the same name as created for ADMsniff. These programs both created a log file called, The_l0gz. This identical file creation was another verifier that ADMsniff and sn.dat were the same program.

There was no indication that the program communicated to an outside source from the results of the packet captures of the Ethereal protocol analyzer. These logs did not indicate that the person who placed the files in the **/lotus/license** directory made any attempt to access this file during the investigation.

The placement of this file in the **/lotus/license** directory indicated that it was an attempt to obscure its purpose by its location. There was no direct proof that this file had yet been executed. The lack of the log file on the forensic image does not mean that the file had been executed. Further inspection at the hexadecimal level of the forensic image did not produce any evidence that the file had been run.

There is no qualitative proof that sn.dat is only a protocol analyzer. In the same vein, the file size comparison between the self-compiled version of ADMsniff and the discovered sn.dat is not adequate proof that these are not the same programs. The difference in libraries and operating system versions make precise comparison difficult without access to the original device that sn.dat was compiled. The proof for this would require creating the ADMsniff on a Red Hat Linux device version 7.1 along with the same zlibc version used.

The person who placed the file sn.dat on the Red Hat machine without authorization was liable to the Kansas Electronic Trespass Act. If the file had been executed, the person would have been violating both the Wiretap Act and the Electronic Communication Privacy Act. The file presence on the original Red Hat device required involvement of law enforcement if further action was pursued.

There were no clear indicators of who was the actual owner of the file. The use of the user account root to create and place the file indicated that the system was been compromised. Other files and devices were likely to be discovered on this system.

Interrogation questions:

If the person who placed the file on the device was discovered, the following questions would be posed.

1. What is a sniffer?
This question while innocent may give an indicator of the level of knowledge of the user. It provides a basis and ground level question for further in depth questions. An answer in ignorance, indicates that this person may have little in depth knowledge and be a dangerous script kiddie.

2. Are you aware that use of this file without prior consent would be in violation of both federal and the state laws of Kansas?
This question would provide a person under interrogation with a basic understanding of the potential danger that his actions were more than just an inquisitive act. The information gathered would be considered violations of the Kansas Electronic Trespass Act, the Federal Pen and Register Act and the Wiretap act at minimum.

3. What type of machines do you now have access and what are the operating systems for each machine?
This is another general question that can lead to further in depth questions. The file sn.dat was compiled for a Linux build. It could be run or accessed from a variety of devices. The indication of a Linux device currently in possession would give the basis to pursue a forensic investigation of this personal machine.

4. What type of machines and operating systems did you have access or possessed in April 2002?
The sn.dat file was compiled April 11, 2002. If any of those machines was a Red Hat 7.1 Linux device, this may have been the device on which this was compiled. An appropriate follow-up question would be: Where is this computer and the hard-drive now?

5. Do you know anyone here at ABC Enterprises? If so, who are they and where do they work?
This attempt appeared to be a work in progress. Most compromises occur due to an internal employee. This involvement may be with intent or unintended. It would be important to identify any linkage between this attempted compromise and any internal employee.

6. What are your programming skills and in what Languages?
The program was compiled with a GCC compiler within Linux. The more programming skills and skill in the programming language C would indicate the complexity of the program sn.dat and whether there are any unknown features.

Citations:

ADMcrew. Readme, ADMsniff.tar.gz, 30, May,1999
URL: ADMsniff.tar.gz

COTSE, The Church of the Swimming Elephant, Computers Professional Reference, ADMsniff link
http://www.cotse.com/tools/sniffers.htm

Icove,David. Seger,Karl. VonStorch,William. Computer Crime Sebastapol, O'Reilly & Associates, Inc, 1995, 77

Kansas Statute No. 21-3755, Kansas Electronic Trespass Act
http://www.kslegislature.org/cgi-bin/statutes/index.cgi/21-3755.html

Sipes, Chris. "SANS ID FAQ: Why your switched networks isn't secure" 10, September 2000
URL: http://www.sans.org/newlook/resources/IDFAQ/switched_network.htm

USC Title 18. Chapter 119, Section 2511, Chapter 119 - Wire and electronic communications interception and interception of oral communcations
http://caselaw.lp.findlaw.com/casecode/uscodes/18/parts/i/chapters/119/sections/section_2511.html

USC Title 18. Chapter 121, Section 2701, Chapter 121 - Stored wire and electronic communications and transactional records access
http://caselaw.lp.findlaw.com/casecode/uscodes/18/parts/i/chapters/121/sections/section_2701.html

Zlibc, V.9 read-only compressed file-system emulation
http://zlibc.linux.lu/

SANS GIAC CERTIFIED FORENSIC ANALYST
Dangers and Pitfalls of the Wiretap Act for the System Administrator
Author Steven Dietz

Abstract:

The Wiretap Act is a Federal law that network administrators and system administrators potentially violate every day. The group of laws that make up the various aspects of the Wiretap Act does offer exceptions to the monitoring and capturing of data. The danger consists of ignorance of the law and potential intentional misuse of the capabilities provided. The use of banners offers some legal protection but should not be considered a security blanket. The banner must be properly placed and worded.

History and Background:

The Wiretap Act is the common name for United States Code Title 18 Chapter 119 sections 2511-2527. It is also known as Title III Wiretap Act of 1968. It has its basis from the original Communications Act of 1934. The 1934 law under Section 605 - Unauthorized Publication or Use of Communications, prohibits any person or party involved in sending or receiving communications from exposing or publishing any part of the communication contents. There are exceptions and permits for disclosure if a legitimate subpoena is provided. Any information from an illegal wiretap is inadmissible and may not be introduced as evidence in federal or state courts.

The original Communications Act of 1934 has been modified or amended many times with the changes of technology and expectations. The current Wiretap Act was based upon five specific laws that directly affect system administrators' actions and restrictions.

These five laws are:

1968 Wiretap Act Title III	USC 18 Title III. Chapter 119, Section 2511-2520
Stored Communications Act	USC 18 Chapter 121, Section 2701
1986 Electronic Computer Privacy Act	USC 18 Section 2701-2711
Pen and Register Act	part of ECPA, USC 18 Section 3121
The Patriot Act of 2001: Section Computer Fraud and Abuse Act	USC 18 Section 1030

The 1968 Wiretap Act was part of the Omnibus Crime Control and Safe Street Act, Title III code. It expanded and better defined the scope and limitations of the original 1934 Communications Act Law. This law defined more explicitly the powers in regards to the use of a wiretap. There was a major limitation to this law. It only took into account the content of the communications. There was no

mechanism for exceptions in use or the ability to utilize logs and header information.

The 1986 Electronic Computer Privacy Act (ECPA) attempted to address the legal holes and problems created in the 1968 Wiretap Act. It provided for exceptions in use as well as the ability to monitor or access information such as logs or data header information. This log or header monitoring information was contained in two separate sections, the Stored Communications Act and the Pen-Register Statute. Both of these sub-section statutes provided better controls for evidence gathering of information such as use logs or electronic data headers. There were seven types of exceptions provided for within the original ECPA.

The Patriot Act of 2001 extended the power for enforcement and scope within the Wiretap Act. It placed a higher expectation of self-enforcement and notification of misuse to law enforcement. It also expanded the capability of law enforcement to conduct wiretaps. These increased expectations on the system administrator place increased responsibility of notification of illegal activity during any monitoring or analysis of data capture by a system administrator.

There are also state laws that have a direct relationship to the Wiretap Act. These state laws may actually make it more restrictive and require more extensive notification. There are 44 states with laws directly related to the Wiretap Act. Some states such as Maryland require both parties be notified and have given consent. The issue of consent is more fully discussed in the section on banners. In the state of Kansas, the Statute KSA 22-2514 - 22-2516 bases its law on the Wiretap Act and provides for consent of one party

System Administration Issues:

Each time the system administrator uses a protocol analyzer (sniffer), reads the information from a intrusion detection scan, or stores the information of firewall or internet use logs, the potential for violation of the Wiretap Act is an issue. There are the exceptions as mentioned above. The exceptions do not protect the system administrator who with intent uses or exposes the information disclosed from the results of a sniff. This intent can even be harmless. Two examples are listed below. Each is based on a real situation. The question is, which of the two are violating the Wiretap Act?

In situation one, the system administrator, reviews the logs of an IRC session capture from an IDS log. The contents of the logs reveal one side of a conversation about a steamy date. The system administrator deduces the person's identity and tells a co-worker the details.

In situation two, the system administrator, analyzes the logs of a protocol decode. In the text portion of the decode, there is a portion of an email describing

a planned assault against a coworker. The system administrator takes the information discovered and provides it to management in personnel.

The answer to both situations is the same. Both system administrators are violating the Wiretap Act. In the first scenario, the system administrator is protected under the 'provider' exception. This exception is discussed below. His actions of communicating the non-pertinent information to someone made him no longer exempt. The difference in situation two is if there were appropriate banners or notification agreements, the system administrator would be exempted and protected. The potential protection of the consent issue of banners is discussed below.

Exceptions:

The exceptions to the various aspects of the Wiretap Act provide for seven different exception types. There are three types of exceptions in regards to corporate monitoring or data management acquisition. The other exceptions do not normally apply to network data communication or they are directed towards law enforcement personnel. Some of these other exceptions may become valid if there is voice over IP (VOIP) on the corporate network. The telephony exceptions may also become more important as voice and data technologies merge.

The first exception involves the use of a court order. The average system administrator will never be faced with this exception. This exception is found in section 2518 of the Wiretap Act. A corporate system administrator served with a court order has the legal expectation of complying and allowing a wiretap to take place.

The second exception is the provider exception. The provider exception is found in section 2511, subsection 2-a-i. This exception provides for the ability to monitor the network in the course of daily maintenance of systems. This would be the most common exception used. The basic concept of this exception is that it protects the technician troubleshooting of a network. The problem with this exception is that it is very limited in scope. It cannot be used as a justification for conducting an investigation of an individual.

The third exception is the 'consent' exception. This is found in section 2511 subsection 2-I. The Wiretap Act 'consent' exception is based on a consensual agreement by one or both parties to allow the interception of data for monitoring. The two common means of consent are contractual use agreements and/or banners posted prior or immediately following access. This will be more fully discussed in the banners section. The consent exception with a properly crafted and displayed banner allows an investigation to be initiated based on the need to protect the network.

Banners:

The banner or notification agreement could provide the system administrator some protection because of the consent exception. There are two common means of consent for corporate networks. These are notification agreements of monitoring and data capture within corporations and businesses, contractual use agreements and banners posted prior or immediately following access.

The contractual use agreement may take on many forms. These include acceptable use agreements for new employees, strictures and secondary notes in business contracts and other contractual agreements. The purpose is to build a binding agreement that includes monitoring and capturing capability during the course of maintenance on the network. This agreement should be regularly renewed and provide continued awareness. This type of contractual agreement works well where there is an existing business relationship.

In instances where there is no prior relationship or where there is a need for continual reminding, the notification banner is used. The banner provides a visual prompt that network use may be monitored and captured. The important issue in the banner is that it must provide explicit notice and state continued use implies consent to be monitored. The lack of a banner potentially removes protection for the exceptions provided in the Wiretap Act.

A poorly worded banner can be problematic as well. Banners that unintentionally exclude or omit key words may in fact exclude the monitoring of an intruder. Please reference the banner listed below.

This computer system and the information it contains are private property and only persons authorized by the owner may access the system and information. If you are not properly authorized, you must exit the system immediately.
If you are an authorized user and continue your logon, you expressly agree to abide by all Corporate Information Security Policies. All activity on the system including e-mail, internet access and usage, and access to information stored on the system, is subject to monitoring, recording, and disclosure by the owner and its authorized representatives. By logging onto the system, you expressly consent to such monitoring, recording, and disclosure.

The statement in the banner, “If you are an authorized user and continue your logon”, may not protect the system administrator in any data capturing processes against an unauthorized external intruder on the network. The implied statement could be interpreted as, if you are unauthorized, you do not have to expressly agree to abide by the stated policy. A more appropriate version of the above banner could be rewritten as illustrated below to account for the unintentional exclusion.

This computer system and the information it contains are private property and only persons authorized by the owner may access the system and information. If you are not properly authorized, you must exit the system immediately.
If you continue your logon, you expressly agree to abide by all Corporate Information Security Policies federal and state law. All activity on the system including e-mail, internet access & usage, all access to information stored on the system, is subject to monitoring, recording, and disclosure by the owner and its authorized representatives. By logging onto the system, you expressly consent to any monitoring, recording, and disclosure.

This banner includes two changes. The first is the removal of the authorized user and change to a direct statement, "If you continue your logon". This includes all individuals accessing the system, authorized or unauthorized. The second change is the addition of being potentially bound by federal and state law in addition to potential tort action of violation of corporate policy. Variations of this banner can then be placed in a variety of locations. Message of the day, FTP and telnet banners are appropriate locations.

There are two additional limitations that should be considered. The first is the byte length limitation in some banner variables on systems. If there is a byte limitation of 256 bytes, the previous banner needs to be tightened and still maintain the important issues of consent to monitoring, scope of monitoring and promise of enforcement. The other limitation is placement. It may not be possible to place a banner upon access of all ports. This limitation can be ameliorated if an action of due diligence was made to common access ports.

Summary of Authority and Expectations for the Sys Admin:

The system administrator must be aware of both federal and state laws pertaining to network monitoring and release of information discovered while monitoring. This awareness should also be communicated to the personnel and legal departments.

Prior to any investigation, appropriate consent must have been placed in a location that was visually available during sign-on or in use of the network. The lack of a banner or other consensual notifiers places the company and the individual system administrator at risk to federal, state and tort litigation. The banner must be worded such that there is no misunderstanding.

Citations:

Computer Crime and Intellectual Property Section,Criminal Division, United States Department of Justice, Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations, July 2002
http://www.cybercrime.gov/s&smanual2002.htm

Icove,David. Seger,Karl. VonStorch,William. Computer Crime Sebastapol, O'Reilly & Associates, Inc, 1995, 77

Infosec Outlook, August 2000 Volume 1, Issue 5
http://www.cert.org/infosec-outlook/infosec_1-5.pdf

Kansas Statute No. 22-2514
http://www.kslegislature.org/cgi-bin/statutes/index.cgi/22-2514.html

Kansas Statute No. 22-2516
http://www.kslegislature.org/cgi-bin/statutes/index.cgi/22-2516.html

Rabinowitz, David , Privacy lawsuits on federal statutes protecting electronic data
http://www.healthlawtoday.com/privacy/electronic_communications.shtml

State of Maryland, 85 Opinions of the Attorney General ___ (2000)
[Opinion No. 00-020 (August 11, 2000)]
http://www.oag.state.md.us/Opinions/2000/00-020.pdf

Strang, Robert, Recognizing and Meeting Title III Concerns in Computer Investigations, March 2001, Cybercrime.Gov document
http://www.usdoj.gov/criminal/cybercrime/usamarch2001_2.htm

Salgado, Richard C., Federal Legal Issues & Monitoring Network Use, SANS2002 Technical conference, Session 1-9
http://www.sans.org/SANS2002/1-9_Salgado.pdf

USC Title 18. Chapter 119, Section 2500-2520, Chapter 119 - Wire and electronic communications interception and interception of oral communcations
http://caselaw.lp.findlaw.com/casecode/uscodes/18/parts/i/chapters/119/toc.html

USC Title 18. Chapter 121, Section 2700-2711, Chapter 121 - Stored wire and electronic communications and transactional records access
http://www4.law.cornell.edu/uscode/18/plch121.html

Wiretap Report 1998, US Federal Government
http://www.uscourts.gov/Press_Releases/wiretap_att.pdf

Appendix:

Pertinent sections of Federal and State laws

USC 18 Title III. Chapter 119, Section 2511
Interception and disclosure of wire, oral, or electronic communications prohibited

(1) Except as otherwise specifically provided in this chapter any person who -
(a) intentionally intercepts, endeavors to intercept, or procures any other person to intercept or endeavor to intercept, any wire, oral, or electronic communication;
(b) intentionally uses, endeavors to use, or procures any other person to use or endeavor to use any electronic, mechanical, or other device to intercept any oral communication when -
(i) such device is affixed to, or otherwise transmits a signal through, a wire, cable, or other like connection used in wire communication; or
(ii) such device transmits communications by radio, or interferes with the transmission of such communication; or
(iii) such person knows, or has reason to know, that such device or any component thereof has been sent through the mail or transported in interstate or foreign commerce; or
(iv) such use or endeavor to use (A) takes place on the premises of any business or other commercial establishment the operations of which affect interstate or foreign commerce; or
(B) obtains or is for the purpose of obtaining information relating to the operations of any business or other commercial establishment the operations of which affect interstate or foreign commerce; or
(c) intentionally discloses, or endeavors to disclose, to any other person the contents of any wire, oral, or electronic communication, knowing or having reason to know that the information was obtained through the interception of a wire, oral, or electronic communication in violation of this subsection;
(d) intentionally uses, or endeavors to use, the contents of any wire, oral, or electronic communication, knowing or having reason to know that the information was obtained through the interception of a wire, oral, or electronic communication in violation of this subsection; or
(e)(i) intentionally discloses, or endeavors to disclose, to any other person the contents of any wire, oral, or electronic communication, intercepted by means authorized by sections 2511(2)(a)(ii), 2511(2)(b)-(c), 2511(2)(e), 2516, and 2518 of this chapter, (ii) knowing or having reason to know that the information was obtained through the interception of such a communication in connection with a criminal investigation, (iii) having obtained or received the information in connection with a criminal investigation, and (iv) with intent to improperly obstruct, impede, or interfere with a duly authorized criminal investigation, shall be punished as provided in subsection (4) or shall be subject to suit as provided in subsection (5).

USC 18 Title III. Chapter 119, Section 2518
Procedure for interception of wire, oral, or electronic communications

(1) Each application for an order authorizing or approving the interception of a wire, oral, or electronic communication under this chapter shall be made in writing upon oath or affirmation to a judge of competent jurisdiction and shall state the applicant's authority to make such application. Each application shall include the following information:
(a) the identity of the investigative or law enforcement officer making the application, and the officer authorizing the application;
(b) a full and complete statement of the facts and circumstances relied upon by the applicant, to justify his belief that an order should be issued, including
(i) details as to the particular offense that has been, is being, or is about to be committed, (ii) except as provided in subsection (11), a particular description of the nature and location of the facilities from which or the place where the communication is to

be intercepted,
(iii) a particular description of the type of communications sought to be intercepted, (iv) the identity of the person, if known, committing the offense and whose communications are to be intercepted;
(c) a full and complete statement as to whether or not other investigative procedures have been tried and failed or why they reasonably appear to be unlikely to succeed if tried or to be too dangerous;
(d) a statement of the period of time for which the interception is required to be maintained. If the nature of the investigation is such that the authorization for interception should not automatically terminate when the described type of communication has been first obtained, a particular description of facts establishing probable cause to believe that additional communications of the same type will occur thereafter;
(e) a full and complete statement of the facts concerning all previous applications known to the individual authorizing and making the application, made to any judge for authorization to intercept, or for approval of interceptions of, wire, oral, or electronic communications involving any of the same persons, facilities or places specified in the application, and the action taken by the judge on each such application; and (f) where the application is for the extension of an order, a statement setting forth the results thus far obtained from the interception, or a reasonable explanation of the failure to obtain such results.
(2) The judge may require the applicant to furnish additional testimony or documentary evidence in support of the application.
(3) Upon such application the judge may enter an ex parte order, as requested or as modified, authorizing or approving interception of wire, oral, or electronic communications within the territorial jurisdiction of the court in which the judge is sitting (and outside that jurisdiction but within the United States in the case of a mobile interception device authorized by a Federal court within such jurisdiction), if the judge determines on the basis of the facts submitted by the applicant that –
(a) there is probable cause for belief that an individual is committing, has committed, or is about to commit a particular offense enumerated in section 2516 of this chapter;
(b) there is probable cause for belief that particular communications concerning that offense will be obtained through such interception;
(c) normal investigative procedures have been tried and have failed or reasonably appear to be unlikely to succeed if tried or to be too dangerous;
(d) except as provided in subsection (11), there is probable cause for belief that the facilities from which, or the place where, the wire, oral, or electronic communications are to be intercepted are being used, or are about to be used, in connection with the commission of such offense, or are leased to, listed in the name of, or commonly used by such person.
(4) Each order authorizing or approving the interception of any wire, oral, or electronic communication under this chapter shall specify -
(a) the identity of the person, if known, whose communications are to be intercepted;
(b) the nature and location of the communications facilities as to which, or the place where, authority to intercept is granted;
(c) a particular description of the type of communication sought to be intercepted, and a statement of the particular offense to which it relates;
(d) the identity of the agency authorized to intercept the communications, and of the person authorizing the application; and
(e) the period of time during which such interception is authorized, including a statement as to whether or not the interception shall automatically terminate when the described communication has been first obtained. An order authorizing the interception of a wire, oral, or electronic communication under this chapter shall, upon request of the applicant, direct that a provider of wire or electronic communication service, landlord, custodian or other person shall furnish the applicant forthwith all information, facilities, and technical assistance necessary to accomplish the interception unobtrusively and with a minimum of interference with the services that such service provider, landlord, custodian, or person is

according the person whose communications are to be intercepted. Any provider of wire or electronic communication service, landlord, custodian or other person furnishing such facilities or technical assistance shall be compensated therefor by the applicant for reasonable expenses incurred in providing such facilities or assistance. Pursuant to section 2522 of this chapter, an order may also be issued to enforce the assistance capability and capacity requirements under the Communications Assistance for Law Enforcement Act.

(5) No order entered under this section may authorize or approve the interception of any wire, oral, or electronic communication for any period longer than is necessary to achieve the objective of the authorization, nor in any event longer than thirty days. Such thirty-day period begins on the earlier of the day on which the investigative or law enforcement officer first begins to conduct an interception under the order or ten days after the order is entered. Extensions of an order may be granted, but only upon application for an extension made in accordance with subsection (1) of this section and the court making the findings required by subsection (3) of this section. The period of extension shall be no longer than the authorizing judge deems necessary to achieve the purposes for which it was granted and in no event for longer than thirty days. Every order and extension thereof shall contain a provision that the authorization to intercept shall be executed as soon as practicable, shall be conducted in such a way as to minimize the interception of communications not otherwise subject to interception under this chapter, and must terminate upon attainment of the authorized objective, or in any event in thirty days. In the event the intercepted communication is in a code or foreign language, and an expert in that foreign language or code is not reasonably available during the interception period, minimization may be accomplished as soon as practicable after such interception. An interception under this chapter may be conducted in whole or in part by Government personnel, or by an individual operating under a contract with the Government, acting under the supervision of an investigative or law enforcement officer authorized to conduct the interception.

(6) Whenever an order authorizing interception is entered pursuant to this chapter, the order may require reports to be made to the judge who issued the order showing what progress has been made toward achievement of the authorized objective and the need for continued interception. Such reports shall be made at such intervals as the judge may require.

(7) Notwithstanding any other provision of this chapter, any investigative or law enforcement officer, specially designated by the Attorney General, the Deputy Attorney General, the Associate Attorney General, or by the principal prosecuting attorney of any State or subdivision thereof acting pursuant to a statute of that State, who reasonably determines that –

(a) an emergency situation exists that involves -

(i) immediate danger of death or serious physical injury to any person,

(ii) conspiratorial activities threatening the national security interest, or

(iii) conspiratorial activities characteristic of organized crime, that requires a wire, oral, or electronic communication to be intercepted before an order authorizing such interception can, with due diligence, be obtained, and

(b) there are grounds upon which an order could be entered under this chapter to authorize such interception, may intercept such wire, oral, or electronic communication if an application for an order approving the interception is made in accordance with this section within forty-eight hours after the interception has occurred, or begins to occur. In the absence of an order, such interception shall immediately terminate when the communication sought is obtained or when the application for the order is denied, whichever is earlier. In the event such application for approval is denied, or in any other case where the interception is terminated without an order having been issued, the contents of any wire, oral, or electronic communication intercepted shall be treated as having been obtained in violation of this chapter, and an inventory shall be served as provided for in subsection (d) of this section on the person named in the application.

(8)(a) The contents of any wire, oral, or electronic communication intercepted by any means authorized by this chapter shall, if possible, be recorded on tape or wire or other comparable device. The recording of the contents of any wire, oral, or electronic communication under this subsection shall be done in such a way as will protect the recording from editing or other

alterations. Immediately upon the expiration of the period of the order, or extensions thereof, such recordings shall be made available to the judge issuing such order and sealed under his directions. Custody of the recordings shall be wherever the judge orders. They shall not be destroyed except upon an order of the issuing or denying judge and in any event shall be kept for ten years. Duplicate recordings may be made for use or disclosure pursuant to the provisions of subsections (1) and (2) of section 2517 of this chapter for investigations. The presence of the seal provided for by this subsection, or a satisfactory explanation for the absence thereof, shall be a prerequisite for the use or disclosure of the contents of any wire, oral, or electronic communication or evidence derived therefrom under subsection (3) of section 2517.

(b) Applications made and orders granted under this chapter shall be sealed by the judge. Custody of the applications and orders shall be wherever the judge directs. Such applications and orders shall be disclosed only upon a showing of good cause before a judge of competent jurisdiction and shall not be destroyed except on order of the issuing or denying judge, and in any event shall be kept for ten years.

(c) Any violation of the provisions of this subsection may be punished as contempt of the issuing or denying judge.

(d) Within a reasonable time but not later than ninety days after the filing of an application for an order of approval under section 2518(7)(b) which is denied or the termination of the period of an order or extensions thereof, the issuing or denying judge shall cause to be served, on the persons named in the order or the application, and such other parties to intercepted communications as the judge may determine in his discretion that is in the interest of justice, an inventory which shall include notice of –

(1) the fact of the entry of the order or the application;

(2) the date of the entry and the period of authorized, approved or disapproved interception, or the denial of the application; and

(3) the fact that during the period wire, oral, or electronic communications were or were not intercepted. The judge, upon the filing of a motion, may in his discretion make available to such person or his counsel for inspection such portions of the intercepted communications, applications and orders as the judge determines to be in the interest of justice. On an ex parte showing of good cause to a judge of competent jurisdiction the serving of the inventory required by this subsection may be postponed.

(9) The contents of any wire, oral, or electronic communication intercepted pursuant to this chapter or evidence derived there from shall not be received in evidence or otherwise disclosed in any trial, hearing, or other proceeding in a Federal or State court unless each party, not less than ten days before the trial, hearing, or proceeding, has been furnished with a copy of the court order, and accompanying application, under which the interception was authorized or approved. This ten-day period may be waived by the judge if he finds that it was not possible to furnish the party with the above information ten days before the trial, hearing, or proceeding and that the party will not be prejudiced by the delay in receiving such information.

(10)(a) Any aggrieved person in any trial, hearing, or proceeding in or before any court, department, officer, agency, regulatory body, or other authority of the United States, a State, or a political subdivision thereof, may move to suppress the contents of any wire or oral communication intercepted pursuant to this chapter, or evidence derived therefrom, on the grounds that -

(i) the communication was unlawfully intercepted;

(ii) the order of authorization or approval under which it was intercepted is insufficient on its face; or

(iii) the interception was not made in conformity with the order of authorization or approval. Such motion shall be made before the trial, hearing, or proceeding unless there was no opportunity to make such motion or the person was not aware of the grounds of the motion. If the motion is granted, the contents of the intercepted wire or oral communication, or evidence derived therefrom, shall be treated as having been obtained in violation of this chapter. The judge, upon the filing of such motion by the aggrieved person, may in his discretion make available to the aggrieved person or his counsel for inspection such portions of the intercepted communication or evidence derived wherefrom as the judge determines to be in the interests of justice.

(b) In addition to any other right to appeal, the United States shall have the right to appeal from an order granting a motion to suppress made under paragraph (a) of this subsection, or the denial of an application for an order of approval, if the United States attorney shall certify to the judge or other official granting such motion or denying such application that the appeal is not taken for purposes of delay. Such appeal shall be taken within thirty days after the date the order was entered and shall be diligently prosecuted.

(c) The remedies and sanctions described in this chapter with respect to the interception of electronic communications are the only judicial remedies and sanctions for nonconstitutional violations of this chapter involving such communications

(11) The requirements of subsections (1)(b)(ii) and (3)(d) of this section relating to the specification of the facilities from which, or the place where, the communication is to be intercepted do not apply if -

(a) in the case of an application with respect to the interception of an oral communication -

(i) the application is by a Federal investigative or law enforcement officer and is approved by the Attorney General, the Deputy Attorney General, the Associate Attorney General, an Assistant Attorney General, or an acting Assistant Attorney General;

(ii) the application contains a full and complete statement as to why such specification is not practical and identifies the person committing the offense and whose communications are to be intercepted; and

(iii) the judge finds that such specification is not practical; and

(b) in the case of an application with respect to a wire or electronic communication -

(i) the application is by a Federal investigative or law enforcement officer and is approved by the Attorney General, the Deputy Attorney General, the Associate Attorney General, an Assistant Attorney General, or an acting Assistant Attorney General;

(ii) the application identifies the person believed to be committing the offense and whose communications are to be intercepted and the applicant makes a showing that there is probable cause to believe that the person's actions could have the effect of thwarting interception from a specified facility;

(iii) the judge finds that such showing has been adequately made; and

(iv) the order authorizing or approving the interception is limited to interception only for such time as it is reasonable to presume that the person identified in the application is or was reasonably proximate to the instrument through which such communication will be or was transmitted.

(12) An interception of a communication under an order with respect to which the requirements of subsections (1)(b)(ii) and (3)(d) of this section do not apply by reason of subsection (11)(a) shall not begin until the place where the communication is to be intercepted is ascertained by the person implementing the interception order. A provider of wire or electronic communications service that has received an order as provided for in subsection (11)(b) may move the court to modify or quash the order on the ground that its assistance with respect to the interception cannot be performed in a timely or reasonable fashion. The court, upon notice to the government, shall decide such a motion expeditiously.

USC 18 Title III Chapter 121, Section 2701

Unlawful access to stored communications

(a) Offense. - Except as provided in subsection (c) of this section whoever -

(1) intentionally accesses without authorization a facility through which an electronic communication service is provided; or

(2) intentionally exceeds an authorization to access that facility; and thereby obtains, alters, or prevents authorized access to a wire or electronic communication while it is in electronic storage in such system shall be punished as provided in subsection (b) of this section.

(b) Punishment. - The punishment for an offense under subsection (a) of this section is -

(1) if the offense is committed for purposes of commercial advantage, malicious destruction or damage, or private commercial gain -

(A) a fine under this title or imprisonment for not more than one year, or both, in the case of a first offense under this subparagraph; and
(B) a fine under this title or imprisonment for not more than two years, or both, for any subsequent offense under this subparagraph; and
(2) a fine under this title or imprisonment for not more than six months, or both, in any other case.
(c) Exceptions. - Subsection (a) of this section does not apply with respect to conduct authorized -
(1) by the person or entity providing a wire or electronic communications service;
(2) by a user of that service with respect to a communication of or intended for that user; or
(3) in section 2703, 2704 or 2518 of this title.

Kansas Statute No. 21-3755

Computer crime; computer password disclosure; computer trespass.

(A) As used in this section:
(1) "Access" means to instruct, communicate with, store data in, retrieve data from or otherwise make use of any resources of a computer, computer system or computer network.
(2) "Computer" means an electronic device which performs work using programmed instruction and which has one or more of the capabilities of storage, logic, arithmetic or communication and includes all input, output, processing, storage, software or communication facilities which are connected or related to such a device in a system or network.
(3) "Computer network" means the interconnection of communication lines, including microwave or other means of electronic communication, with a computer through remote terminals, or a complex consisting of two or more interconnected computers.
(4) "Computer program" means a series of instructions or statements in a form acceptable to a computer which permits the functioning of a computer system in a manner designed to provide appropriate products from such computer system.
(b) (1) Computer crime is:
(A) Intentionally and without authorization accessing and damaging, modifying, altering, destroying, copying, disclosing or taking possession of a computer, computer system, computer network or any other property;
(B) using a computer, computer system, computer network or any other property for the purpose of devising or executing a scheme or artifice with the intent to defraud or for the purpose of obtaining money, property, services or any other thing of value by means of false or fraudulent pretense or representation; or
(C) intentionally exceeding the limits of authorization and damaging, modifying, altering, destroying, copying, disclosing or taking possession of a computer, computer system, computer network or any other property.
(2) Computer crime is a severity level 8, nonperson felony.
(3) In any prosecution for computer crime, it is a defense that the property or services were appropriated openly and avowedly under a claim of title made in good faith.
(c) (1) Computer password disclosure is the unauthorized and intentional disclosure of a number, code, password or other means of access to a computer or computer network.
(2) Computer password disclosure is a class A nonperson misdemeanor.
(d) Computer trespass is intentionally, and without authorization accessing or attempting to access any computer, computer system, computer network or computer software, program, documentation, data or property contained in any computer, computer system or computer network. Computer trespass is a class A nonperson misdemeanor.

Last Updated: October 15th, 2018

Upcoming SANS Forensics Training

Mentor Session AW - FOR500	Washington, DC	Oct 22, 2018 - Oct 26, 2018	Mentor
Houston 2018 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	Houston, TX	Oct 29, 2018 - Nov 03, 2018	vLive
SANS vLive - FOR500: Windows Forensic Analysis	FOR500 - 201810,	Oct 29, 2018 - Dec 19, 2018	vLive
SANS Houston 2018	Houston, TX	Oct 29, 2018 - Nov 03, 2018	Live Event
SANS Gulf Region 2018	Dubai, United Arab Emirates	Nov 03, 2018 - Nov 15, 2018	Live Event
SANS Sydney 2018	Sydney, Australia	Nov 05, 2018 - Nov 17, 2018	Live Event
SANS DFIRCON Miami 2018	Miami, FL	Nov 05, 2018 - Nov 10, 2018	Live Event
SANS Rome 2018	Rome, Italy	Nov 12, 2018 - Nov 17, 2018	Live Event
SANS November Singapore 2018	Singapore, Singapore	Nov 19, 2018 - Nov 24, 2018	Live Event
SANS Paris November 2018	Paris, France	Nov 19, 2018 - Nov 24, 2018	Live Event
SANS Stockholm 2018	Stockholm, Sweden	Nov 26, 2018 - Dec 01, 2018	Live Event
SANS Austin 2018	Austin, TX	Nov 26, 2018 - Dec 01, 2018	Live Event
SANS San Francisco Fall 2018	San Francisco, CA	Nov 26, 2018 - Dec 01, 2018	Live Event
SANS Khobar 2018	Khobar, Kingdom Of Saudi Arabia	Dec 01, 2018 - Dec 06, 2018	Live Event
SANS Nashville 2018	Nashville, TN	Dec 03, 2018 - Dec 08, 2018	Live Event
SANS Frankfurt 2018	Frankfurt, Germany	Dec 10, 2018 - Dec 15, 2018	Live Event
SANS Cyber Defense Initiative 2018	Washington, DC	Dec 11, 2018 - Dec 18, 2018	Live Event
Cyber Defense Initiative 2018 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	Washington, DC	Dec 13, 2018 - Dec 18, 2018	vLive
Cyber Defense Initiative 2018 - FOR585: Advanced Smartphone Forensics	Washington, DC	Dec 13, 2018 - Dec 18, 2018	vLive
Cyber Defense Initiative 2018 - FOR572: Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response	Washington, DC	Dec 13, 2018 - Dec 18, 2018	vLive
Cyber Defense Initiative 2018 - FOR500: Windows Forensic Analysis	Washington, DC	Dec 13, 2018 - Dec 18, 2018	vLive
Cyber Defense Initiative 2018 - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	Washington, DC	Dec 13, 2018 - Dec 18, 2018	vLive
Mentor Session - FOR500	Phoenix, AZ	Jan 11, 2019 - Feb 15, 2019	Mentor
SANS Threat Hunting London 2019	London, United Kingdom	Jan 14, 2019 - Jan 19, 2019	Live Event
SANS Amsterdam January 2019	Amsterdam, Netherlands	Jan 14, 2019 - Jan 19, 2019	Live Event
SANS vLive - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	FOR610 - 201901,	Jan 21, 2019 - Feb 27, 2019	vLive
SANS Miami 2019	Miami, FL	Jan 21, 2019 - Jan 26, 2019	Live Event
Cyber Threat Intelligence Summit & Training 2019	Arlington, VA	Jan 21, 2019 - Jan 28, 2019	Live Event
Mentor Session - FOR585	Tampa, FL	Jan 24, 2019 - Mar 07, 2019	Mentor
SANS Security East 2019	New Orleans, LA	Feb 02, 2019 - Feb 09, 2019	Live Event
SANS London February 2019	London, United Kingdom	Feb 11, 2019 - Feb 16, 2019	Live Event