

Rhino Hunt Lab

Connor Shott
08.07.2022

Senario:

The city of New Orleans passed a law in 2004 making possession of nine or more unique rhinoceros images a serious crime. The network administrator at the University of New Orleans recently alerted police when his instance of RHINOVORE flagged illegal rhino traffic. Evidence in the case includes a computer and USB key seized from one of the University's labs. Unfortunately, the computer had no hard drive. The USB key was imaged and a copy of the dd image is on the CD-ROM you have been given.

In addition to the USB key drive image, three network traces are also available. These were provided by the network administrator and involved the machine with the missing hard drive. The suspect is the primary user of this machine, who has been pursuing his Ph.D. at the University since 1972.

Provided Evidence:

- c0d0093eb1664cd7b73f3a5225ae3f30 *rhino.log
- cd21eaf4acfb50f71ffff857d7968341 *rhino2.log
- 7e29f9d67346df25faaf18efcd95fc30 *rhino3.log
- 80348c58eec4c328ef1f7709adc56a54 *RHINOUSB.dd

Tasks

- Who gave the telnet/ftp account
- what is the username and password for the account
- what relevant file transfers appear in the network traces
- what happened to the computer hard drive and where is it
- what happened to the USB key
- what is recoverable from the DD image of the USB key
- is there any evidence connecting the USB key and network traces

Steps Taken

This section will explore how the answers were obtained and the process for solving this lab. Please skip to the next section for just the answers.

1. Resources

To complete this lab, some resources are necessary.

- The SIFT workstation available [here](#).
- The evidence zip file: [download](#).
- Install fcrackzip on the SIFT workstation
- Install the SecLists repo from <https://github.com/danielmiessler/SecLists>.

2. Getting Started

This walk-through assumes that the SIFT workstation is up and running. Information on setup and installation can be found [here](#).

2.1. Unzipping The File

In the terminal, navigate to `~/Desktop/cases` and run `mkdir Rhino` and `wget <evidence file URL>`. finally, run `unzip DFRWS2005-RODEO.zip` and four files will be extracted to the working directory.

```
$ ls
DFRWS2005-RODEO.zip  rhino2.log  rhino3.log  rhino.log  RHINOUSB.dd
```

2.2. RHINOUSB.dd

The first item of interest is the RHINOUSB.DD file. Running `file RHINOUSB.dd` gives the following output

```
$ file RHINOUSB.dd
RHINOUSB.dd: DOS/MBR boot sector, code offset 0x3c+2, OEM-ID "mkdosfs", sectors/cluster 8, root entries 512, Media descriptor 0xf8, sectors/FAT 248, sectors/track 62, heads 8, sectors 506848 (volumes > 32 MB), serial number 0x4092d9d1, label: " ", FAT (16 bit)
```

We can see that it is a boot sector device that has data on it. That data can be recovered in a couple of ways, I am going to use the photorec tool.

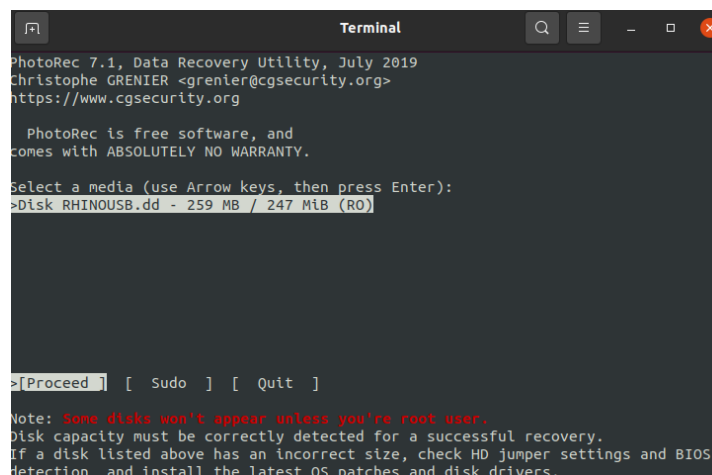
2.2.1. What Is Photorec

Photorec is a recovery tool used to recover file from hard disks, digital cameras, and CD-ROM devices. Full documentation can be found on the man page, or at their [website](#). We are going to use photorec since it outputs more information, although as we will see, not all of it is helpful.

Another tool that can be used is `foremost` which is simpler to use, but does not give as much data in this situation.

2.3. Using photorec

Using photorec from the command line is easy, and it is already installed on the SIFT workstation. To invoke it just type `photorec RHINOUSB.dd` and this screen will appear



```
PhotoRec 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

PhotoRec is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press Enter):
>Disk RHINOUSB.dd - 259 MB / 247 MiB (RO)

>[Proceed] [ Sudo ] [ Quit ]

Note: Some disks won't appear unless you're root user.
Disk capacity must be correctly detected for a successful recovery.
If a disk listed above has an incorrect size, check HD jumper settings and BIOS
detection, and install the latest OS patches and disk drivers.
```

Pressing enter on the highlighted option prompts us to choose a partition. We want the full disk

```
PhotoRec 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

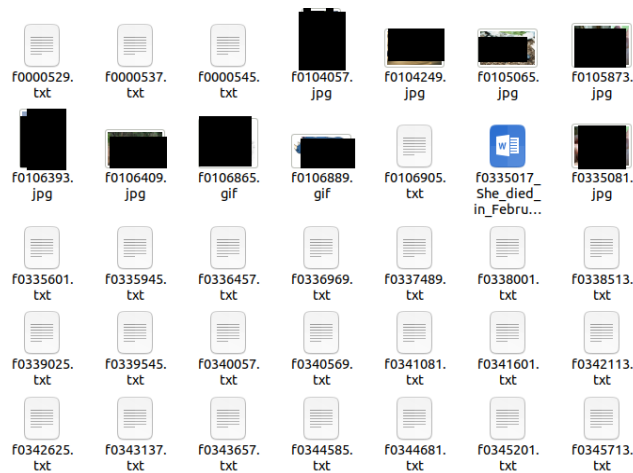
Disk RHINOUSB.dd - 259 MB / 247 MiB (RO)

Partition      Start      End      Size in sectors
> Unknown      0 0 1 1021 6 60 506848 [whole disk]
P FAT16        0 0 1 1021 6 60 506848

[Search] [Options] [File Opt] [Quit]
Start file recovery
```

the next screen asks for the file system type, select ext2/ext3 and hit enter. Next, photorec wants a directory to output the recovered files to. Leaving it on the "." directory will make a new file in the current directory, which for us is ~/Desktop/cases/rhino. Hitting C leads us to a report screen saying 134 files have been recovered. Since we never gave a name to the output file, it is by default recup_dir.1. Lets take a look at what was recovered!

2.4. Photorec Output



Along with a TON of similarly named text files, we find some .jpg files, and some gifs, and even a Microsoft Word document. The images have been redacted so you can experience their beauty for yourself, and they may not all be rhinos...

Looking at one of the .txt files at random, it is hundreds of lines all with CHARLIE on them. Inspection of more reveals that they are all long and all the lines contain the first two. Only two files break this, and they appear first in the file order. Opening up the Microsoft Word document shows that it is a diary of some form. There are a couple of interesting points towards the bottom.

Rhino pictures illegal? Makes me sick. I “hid” the photos...hehehehe. Apparently, if there are less than 10 photos, it’s no big deal.

OK. Things are getting a little weird. I zapped the hard drive and then threw it into the Mississippi River. I’m gonna reformat my USB key after this entry, but try not to destroy the good stuff. I need to change the password on the gnome account that Jeremy gave me. I can probably just do that at Radio Shack.

This answers two of our tasks! What happened to the USB drive, and where is the hard drive. There is also reference to a “ gnome ” account that someone named “ Jeremy ” supplied the writer with. That’s a good start! Lets take a break from the dd image and look at the log files, beginning with rhino.log.

3. Rhino.log

One of the first things to do when looking at the first of the .log files is to determine exactly what kind of log file we are working with. The easiest way to do this is by running the file command on the listed file.

```
$ file rhino.log
rhino.log: pcap capture file, microsecond ts (little-endian) - version 2.4 (Ethernet, capture length 65000)
```

So it’s a pcap file! That means we can use Wireshark to process it.

3.1. What is Wireshark

Wireshark is a network traffic analysis tool. It will let us look in depth at network traffic, and see data that was sent. Further documentation can be found at their [website](#).

3.2. Findings in Rhino.log

One of the first places I check when looking at a pcap is the “ Protocol Hierarchy ” found under the statistics section of the Wireshark ribbon.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	50	100.0	4428	114	0	0	0
Ethernet	100.0	50	15.8	700	18	0	0	0
Internet Protocol Version 4	100.0	50	22.6	1000	25	0	0	0
Transmission Control Protocol	100.0	50	61.6	2728	70	0	0	0
File Transfer Protocol (FTP)	100.0	50	39.0	1728	44	50	0	0

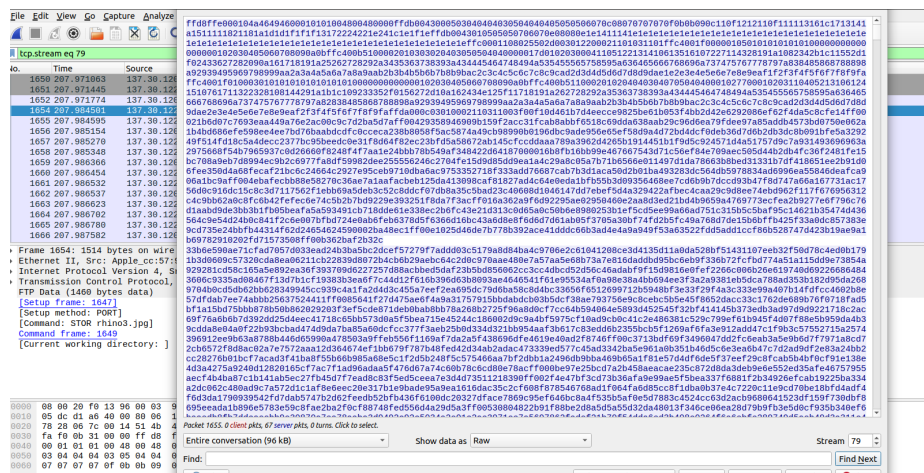
It looks like there was some FTP action going on, let’s take a closer look at that by applying a ftp contains rhinofilter since we know we are looking for rhinos.

ftp contains rhino						
No.	Time	Source	Destination	Protocol	Length	Info
1546	188.996081	137.30.122.253	137.30.120.40	FTP	71	Request: STOR rhino1.jpg
1550	189.033465	137.30.120.40	137.30.122.253	FTP	111	Response: 150 Opening BINARY mode data connection for rhino1.jpg.
1649	207.947603	137.30.122.253	137.30.120.40	FTP	71	Request: STOR rhino3.jpg
1653	207.979567	137.30.120.40	137.30.122.253	FTP	110	Response: 150 Opening ASCII mode data connection for rhino3.jpg.
1763	215.133258	137.30.122.253	137.30.120.40	FTP	71	Request: STOR rhino3.jpg
1767	215.158473	137.30.120.40	137.30.122.253	FTP	111	Response: 150 Opening BINARY mode data connection for rhino3.jpg.

Great, a lead! So, there are at least two files in this log that are of interest. Let's take a look at other packets, so we can try to get more information. Sorting by ftp-data leads to a good number of packets that appear to be from the above “rhino1” and “rhino3” jpg files. Lets take a closer look. Right-clicking on packet 1551 and selecting follow tcp stream takes us to some gibberish that from the first few lines appears to contain something having to do with Adobe Photoshop.



Lets see what we can do with this. Change the data to “show as raw”, then save as “rhino1.jpg” in a directory of your choosing. To keep things compartmentalized, I made a “rhino1log” folder in the working directory. If all goes well there will be an image in your file! Let's go back to the ftp-data filter and do the same thing for the “rhino3.jpg” file. It looks like the first ftp-data packet containing rhino3 is packet 1654. Following the same process of right click, follow tcp stream; then showing as raw data shows this:



saving the stream as “rhino3.jpg” into the same directory as the “rhino1.jpg” also shows an image!

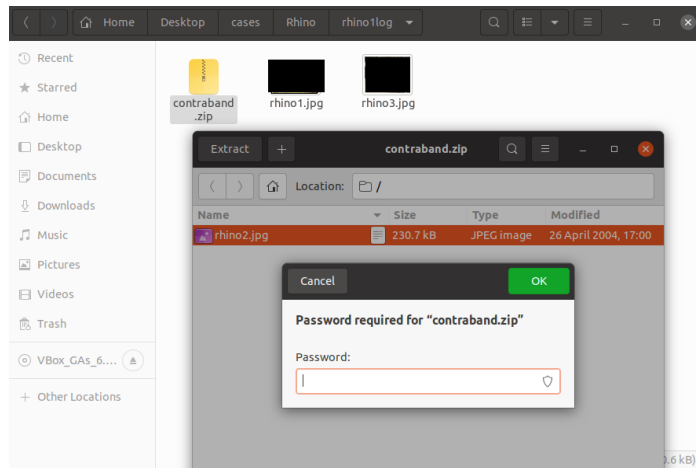
We have found a good chunk of info so far in this first log, but there is more! lets go back to the ftp-data filter and see if there is anything else of note. Look at that, there is!

No.	Time	Source	Destination	Protocol	Length	Info
1842	215.284176	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR rhino3.jpg)
1843	215.284257	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR rhino3.jpg)
1844	215.284337	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR rhino3.jpg)
1846	215.303567	137.30.122.253	137.30.120.40	FTP-DA...	593	FTP Data: 539 bytes (PORT) (STOR rhino3.jpg)
5652	485.745259	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5653	485.745416	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5655	485.745993	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5656	485.746075	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5658	485.746175	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5659	485.746255	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5661	485.746952	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5662	485.747034	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5663	485.747114	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5665	485.747929	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5666	485.748014	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)
5667	485.748094	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT) (STOR contraband.zip)

lets to the same process for this “ contraband.zip ” file that was transferred. This time we are following the stream for packet 5652, and saving as to the same directory as the jpg files.

3.3. Contraband.zip

We have another successful file recovery! This time it is a zip file. It looks like it contains another rhino image, but it is password protected.



3.3.1. Zip Password Cracking

Remember back at the start I had the SecLists repo installed, along with a tool called fcrackzip ? Well this is where they are going to come in handy! fcrackzip is a tool used to break passw ords for zip files. It takes in a dictionary, which we will pull from the SecLists repository. I used the “ 500-worst-pass-words.txt ” file from the password folder in SecLists. With that we are ready to get cracking!

The basic syntax for fcrackzip is

```
fcrackzip -u -D -p <dictionary path> <file to crack>.
```

The dictionary path is finniky sometimes, but always works for the current directory which is why we copied the password file. Running this command gives us the password fairly quick.

```
sansforensics@siftworkstation: /cases/Rhino/rhino1log
$ fcrackzip -u -D -p ./500-worst-pass-words.txt contraband.zip

PASSWORD FOUND!!!!: pw == monkey
```

putting this password into the extract window lets us grab the “ rhino2.jpg “ giving us three rhino images from this log! Where are these images being transferred from though? Lets go back to the pcap and do some more digging.

3.4. The Rest Of The Pcap

Lets start by taking a look at another piece of the pcap that hasn’t been touched yet, the tcp packets. Filtering by

`tcp contains rhino`

provides a number of packets to look at. Some of them, like packet 27 shows a login for a “ hugerhi-nolover@hotmail.com ” but this doesn’t seem to lead to anything interesting. Further down in packet 2600, there is a HTTP POST request linked to that rhinolover email. The body and recipient are of interest to us.

```
▼ Form item: "to" = "bighonkingrhino@hotmail.com"
  Key: to
  Value: bighonkingrhino@hotmail.com
▼ Form item: "cc" = ""
  Key: cc
  Value:
▼ Form item: "bcc" = ""
  Key: bcc
  Value:
▼ Form item: "subject" = "New rhino pics"
  Key: subject
  Value: New rhino pics
Form item: "body" = "
I just checked a few things on the gnome account on cook.cs.uno.edu.  I'm about to upload some
new rhino stuff.

▼ Check it out.

--John
"
```

So it looks like the user John is using an account to access and upload more rhino material. Let’s see if we can find anything else related to this account in the pcap. Sorting by

`tcp contains gnome`

gives TELNET and ftp packets. The ftp ones are exactly what we were looking for.

tcp contains gnome						
No.	Time	Source	Destination	Protocol	Length	Info
1361	134.547422	137.30.120.40	137.30.122.253	TELNET	122	Telnet Data ...
1459	153.507872	137.30.120.40	137.30.122.253	TELNET	87	Telnet Data ...
1478	169.559115	137.30.120.40	137.30.122.253	TELNET	93	Telnet Data ...
1504	172.367891	137.30.120.40	137.30.122.253	TELNET	93	Telnet Data ...
1532	182.640647	137.30.122.253	137.30.120.40	FTP	66	Request: USER gnome
1534	182.644970	137.30.120.40	137.30.122.253	FTP	88	Response: 331 Password required for gnome.
1536	184.667754	137.30.122.253	137.30.120.40	FTP	69	Request: PASS gnome123
1538	184.748946	137.30.120.40	137.30.122.253	FTP	81	Response: 230 User gnome logged in.
1625	198.525443	137.30.122.253	137.30.120.40	FTP	66	Request: USER gnome
1627	198.529854	137.30.120.40	137.30.122.253	FTP	88	Response: 331 Password required for gnome.
1629	200.280951	137.30.122.253	137.30.120.40	FTP	69	Request: PASS gnome123
1631	200.357806	137.30.120.40	137.30.122.253	FTP	81	Response: 230 User gnome logged in.

Gnome is a username, and there is a simple password associated with it sent in plain text. How nice for us! Further down, in packet 5633 there are the beginning of some failed password reset attempts. That is a bunch of information pulled from the first log file, and everything of use I was able to find in it. Let’s move on now to the second log file.

4. Rhino2.log

Once again, running the `file` command on `rhino2.log` reveals that it is another pcap. Loading it up in Wireshark and looking at the protocol hierarchy shows that there were some interactions over IMAP, but most of the packets are HTTP.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	370	100.0	286660	23 k	0	0	0
Ethernet	100.0	370	1.8	5180	418	0	0	0
Internet Protocol Version 4	100.0	370	2.6	7400	597	0	0	0
Transmission Control Protocol	100.0	370	95.6	273968	22 k	343	273418	22 k
Internet Message Access Protocol	1.1	4	0.0	129	10	4	129	10
Hypertext Transfer Protocol	6.2	23	87.7	251437	20 k	13	4616	372
Line-based text data	1.4	5	1.8	5040	406	5	5611	452
JPEG File Interchange Format	0.3	1	53.4	153191	12 k	1	153484	12 k
CompuServe GIF	1.1	4	29.9	85810	6,926	4	86101	6,949

Lets use some search terms identified in the last pcap, starting with the “ gnome ” username. Running

`http contains gnome`

gives us six packets.

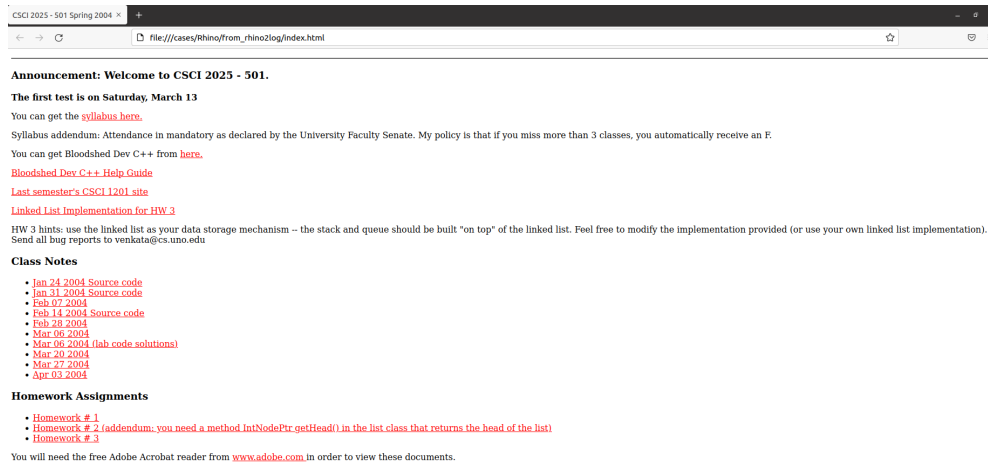
No.	Time	Source	Destination	Protocol	Length	Info
28	5.287376	137.30.123.234	137.30.120.37	HTTP	437	GET /~gnome HTTP/1.1
30	5.301396	137.30.120.37	137.30.123.234	HTTP	642	HTTP/1.1 301 Moved Permanently (text/html)
32	5.554353	137.30.123.234	137.30.120.37	HTTP	438	GET /~gnome/ HTTP/1.1
34	5.638951	137.30.120.37	137.30.123.234	HTTP	1033	HTTP/1.1 200 OK (text/html)
49	7.892558	137.30.123.234	137.30.120.37	HTTP	488	GET /~gnome/rhino4.jpg HTTP/1.1
217	14.008741	137.30.123.234	137.30.120.37	HTTP	488	GET /~gnome/rhino5.gif HTTP/1.1

We can see that two of the packets, 49 and 217, are http GET requests for rhino images. Inspecting the packets further, the full request URI is `http://www.cs.uno.edu/~gnome/<rhino image>`. This is important as this URI corresponds to the domain mentioned in the email recovered from “ rhino.log ” meaning that there is something fishy about that website. Let’s try exporting all http objects through the file toolbar and going to `export objects: http`

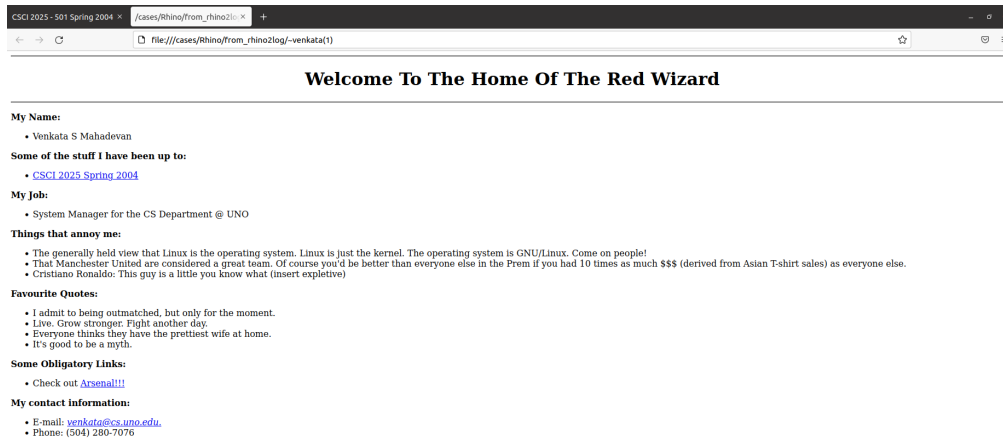
Packet	Hostname	Content Type	Size	Filename
30	www.cs.uno.edu	text/html	304 bytes	~gnome
34	www.cs.uno.edu	text/html	772 bytes	~gnome
37	www.cs.uno.edu	image/gif	148 bytes	blank.gif
45	www.cs.uno.edu	image/gif	309 bytes	image2.gif
46	www.cs.uno.edu	image/gif	216 bytes	back.gif
215	www.cs.uno.edu	image/jpeg	153 kB	rhino4.jpg
312	www.cs.uno.edu	image/gif	85 kB	rhino5.gif
345	www.cs.uno.edu	text/html	306 bytes	~venkata
350	www.cs.uno.edu	text/html	1,388 bytes	~venkata
362	www.cs.uno.edu	text/html	2,270 bytes	index.html

“ from_rhino2log ” in the working directory.

The “ ~gnome ” file gives a 404 error, while the “ ~gnome(1) ” file takes us to a page that holds links to the rhino images. This indicates that the URI identified above did indeed provide the rhino images. Moving into the “ index.html ” file, we see a course syllabus.



All the links contained in this webpage are of course broken, but there is some helpful information in here. The largest bit of information is the name Venkata listed in the cs.uno.edu email. This name corresponds to more files pulled from the pcap: “~venkata” and “~venkata(1)”. The first of these directs to a “file moved” screen while the second directs to a viable webpage



Most of these links are also broken. Once redirects to the Arsenal website, making sense of the Manchester United jabs higher up. It can be speculated that this Venkata person is in collusion with the distribution of rhino images from their site coming up with these images, and them being tied to the university identified earlier to be hosting these images.

Finally, two rhino images, “rhino4.jpg” and “rhino5.gif” were also recovered from the pcap with the http filter. Let’s go back to the pcap and take a look at those imap packets. There is nothing of note in the IMAP packets, packets 314 - 317.

The above is all I was able to pull from “rhino2.log”. While I believe the connection with Venkata could be explored further, we have found some rhino images and made important links to already established evidence. Let’s continue the investigation with the last log file.

5. Rhino3.Log

Taking the usual first step of checking the protocol hierarchy we find nothing interesting this time around, although it is good to know that there were GIFs sent, maybe there is something good in those.

Let's do the same thing we did with " rhino2.log " and check the http exports.

Text Filter:		Content Type:		All Content-Types
Packet	Hostname	Content Type	Size	Filename
28	www.google.com	text/html	15 kB	search?hl=en&ie=UTF-8&oe=UTF-8&q=
31	www.google.com	image/gif	1,033 bytes	nav_first.gif
40	www.google.com	image/gif	376 bytes	nav_current.gif
47	www.google.com	image/gif	373 bytes	nav_page.gif
50	www.google.com	image/gif	290 bytes	nav_next.gif
68	groups.google.com		89 bytes	groups?q=rhino.exe&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=N&tab=wg
95	groups.google.com		5 bytes	groups?q=rhino.exe&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=N&tab=wg
274	www.cs.uno.edu	application/octet-stream	145 kB	rhino.exe

Selecting save all, and using the GUI to make a new directory called " from_rhino3log " all the items sent over http were downloaded. Taking a look at the .gif files, there is nothing interesting. However, there are some strange named files. Let's see what they are.

```
sansforensics@siftworkstation: /cases/Rhino/from_rhino3log
$ file 'groups%3fq=rhino.exe&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=N&tab=wg'
groups%3fq=rhino.exe&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=N&tab=wg: ASCII text, with CRLF line terminators
sansforensics@siftworkstation: /cases/Rhino/from_rhino3log
$ file 'groups%3fq=rhino.exe&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=N&tab=wg: ASCII text, with CRLF line terminators
groups%3fq=rhino.exe&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=N&tab=wg: ASCII text, with CRLF line terminators
sansforensics@siftworkstation: /cases/Rhino/from_rhino3log
$ file 'search%3fhl=en&ie=UTF-8&oe=UTF-8&q=rhino.exe'
search%3fhl=en&ie=UTF-8&oe=UTF-8&q=rhino.exe: HTML document, UTF-8 Unicode text, with very long lines, with CRLF, LF line terminators
```

One of these is a webpage! But it cannot be viewed in its current form. But, using the cp command to copy the file into search.html the file can be viewed.

Go to Google Home

Web Images Groups News Froogle^{New!} more »

rhino.exe

Search

Advanced Search Preferences

Web

[SiteScan](#)
... Stop Running Processes: Kill these running processes with Task Manager:
rhino.exe sitescan.exe. ... **rhino.exe** sitescan.exe. Research. File Analyses: ...
[www.pestpatrol.com/pestinfo%5Cs%5Csitescan.asp - 30k - Cached - Similar pages](#)

[PDF\] BVS RHINO PC INTERFACE SOFTWARE](#)
File Format: PDF/Adobe Acrobat - [View as HTML](#)
Page 2, Page 3. BVS RHINO PC INTERFACE SOFTWARE INSTALLATION Copy the file "rhino.exe"
from the supplied disk to a directory on the hard drive of the computer. ...
[www.bvssystem.com/Tech/Manuals/Rhino1.0.pdf - Similar pages](#)

[Jigsaw Puzzles from BillyBear4Kids.com](#)
... Download 48 pc puzzle **rhino.exe** 9 pc puzzle **rhino.exe** 130 pc puzzle **h-rhino.exe**.
Download 48 pc puzzle **snake.exe** 9 pc puzzle **snake.exe** 130 pc puzzle **h-snake.exe**. ...
[www.billybear4kids.com/jigsaw-puzzles/download-pg2.html - 11k - Cached - Similar pages](#)

[Free YOUR Personalized Desktop Picture Postcards from ...](#)
BillyBear4Kids.Com Welcomes YOU! Free Your Personalized Desktop
Picture Postcards - PuterPals Click Here advertisement banner. ...
[www.billybear4kids.com/DesktopCritters/DesktopPicturePostcards6.html - 13k - Cached - Similar pages](#)
[More results from [www.billybear4kids.com](#)]

[RhinoScript Tutorial: Assigning scripts to Rhino's User Interface](#)
... b.) Rhino's "Scripts" folder. c.) Rhino's installation folder. d.) The folder
here **Rhino.exe** is located. Assigning the RunScript command to a button: ...
[www.rhino3d.com/scripting/tutorial/assign_to_buttons.htm - 14k - Cached - Similar pages](#)

[Rhino World](#)
... x = Shell("D:\rhino20\System\rhino.exe D:\rhino20\System\rhino2.ini", vbNormalNoFocus)
DoEvents or Sleep (2000) 'search for Rhino hwnd m = GetAllWindows(0 ...
[www.personal.kent.edu/~knamjesn/rhino/rhino_p1.html - 64k - Cached - Similar pages](#)

[購入からライセンス発行までの流れ](#)
... Select Rhino Execute: ie C:\Program Files\Rhinoceros\System\rhino.exe
というダイアログボックスが表示されますので、Rhino ...
[www.applcraft.com/nPower/license.html - 7k - Cached - Similar pages](#)

This appears to be a saved search page for “ rhino.exe ” it is currently unclear what the program does. As a sidetrack I attempted to disassemble “ rhino.exe ” with Ghidra, but was unsuccessful in getting anything helpful out of it. Maybe an addendum will be written exploring this exe. As for the other files from this dump, I could not find anything of use. For now, it’s time to get back to the pcap as there is more hidden!

Doing a filter for `http contains rhino` gives mostly the information we already had. However, it is worth noting that the “ rhino.exe ” file came from the “ ~gnome ” directory found in the last pcap. This links the exe file to the university website, to the rhino four and five images, and to the Venkata character.

```
Hypertext Transfer Protocol
> GET /~gnome/rhino.exe HTTP/1.1\r\n
  Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, i
  Accept-Language: en-us\r\n
  -----: -----\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)\r\n
  Host: www.cs.uno.edu\r\n
  Connection: Keep-Alive\r\n
\r\n
[Full request URI: http://www.cs.uno.edu/~gnome/rhino.exe]
[HTTP request 1/1]
[Response in frame: 274]
```

6. Conclusion

There is still more to be found and inspected. However, we have answers for all the tasks set to us that are found in the section below. A future update will look further at the “ rhino.exe ” as I believe there is more hidden in there. There is also another piece of information regarding the ftp server hiding in “ rhino.log ” that I encourage you to find for yourself.

Answers

6.1. Who gave the telnet/ftp account

The accused telnet/ftp account was provided by Jeremy

6.2. what is the username and password for the account

The username is **gnome** and the password is **gnome123**.

6.3. what relevant file transfers appear in the network traces

Relevant file transfers include:

- Rhino1.jpg
- Rhino3.jpg
- contraband.zip
- Rhino4.jpg
- Rhino5.gif
- rhino.exe

6.4. what happened to the computer hard drive and where is it

The document recovered from the DD image states:

"...I zapped the hard drive and threw it into the Mississippi River..."

6.5. what happened to the USB key

it was reformatted, possibly at RadioShack per the recovered Word document

6.6. what is recoverable from the DD image of the USB key

A Word document, some images, and a number of overwritten text files, along with two recipes can be recovered

6.7. is there any evidence connecting the USB key and network traces

" rhino2.jpg " recovered from the " rhino.log " pcap is the same image as " 00106395.jpg " recovered from the usb drive.

Thank you for reading!