

# The Honeyynet

P R O J E C T

**Who we are and what we have  
done for you lately**

## Speaker

- Lance Spitzner
  - The full-time HoneyNet Project Pimp and Whip Cracker.
- Edward Balas
  - Sebek lead
- Rob McMillen
  - Honeywall Guru

## **Purpose**

To explain the HoneyNet Project, Honeynets, and demonstrate the latest technologies.

# Agenda

- The Project and Research Alliance
- Honeynets
- Sebek2

# **Honeynet Project**

## **Problem**

*How can we defend against an enemy, when we don't even know who the enemy is?*

# **The HoneyNet Project**

- All volunteer organization of security professionals dedicated to researching cyber threats.
- We do this by deploying networks around the world to be hacked.

# **Mission Statement**

To learn the tools, tactics, and motives of the blackhat community, and share the lessons learned.

## Goals

- Awareness: To raise awareness of the threats that exist.
- Information: For those already aware, to teach and inform about the threats.
- Research: To give organizations the capabilities to learn more on their own.

## **Project History**

- The group informally began in April, 1999 as the [Wargames] maillist.
- Officially called ourselves the HoneyNet Project in June, 2000.
- Formed HoneyNet Research Alliance in January, 2002.

## Value of the Project

- Totally Open Source, sharing all of our work, research and findings.
- Everything we capture is happening in the wild (there is no theory.)
- Made up of security professionals from around the world.
- We have no agenda, no employees, nor any product or service to sell (*crummy business model*).

## **Project Organization**

- Non-profit (501c3) organization
- Board of Directors
- No more than two members from any organization.
- Diverse set of skills and experiences.
- Team works virtually, from around the world.

## **Honeynet Research Alliance**

Starting in 2002, the Alliance is a forum of organizations around the world actively researching, sharing and deploying Honeynet technologies.

*<http://www.honeynet.org/alliance/>*

## Alliance Members

- South Florida HoneyNet Project
- netForensics HoneyNet
- Azusa Pacific University
- Paladion Networks HoneyNet Project (India)
- Internet Systematics Lab HoneyNet Project (Greece)
- AT&T Mexico HoneyNet (Mexico)
- HoneyNet.BR (Brazil)
- Irish HoneyNet
- Norwegian HoneyNet
- UK HoneyNet

# Honeynets

# Honeypots

- A security resource whose value lies in being probed, attacked or compromised.
- Has no production value, anything going to or from a honeypot is likely a probe, attack or compromise.

## **Advantages**

- Collect small data sets of high value.
- Reduce false positives
- Catch new attacks, false negatives
- Work in encrypted or IPv6 environments
- Simple concept requiring minimal resources.

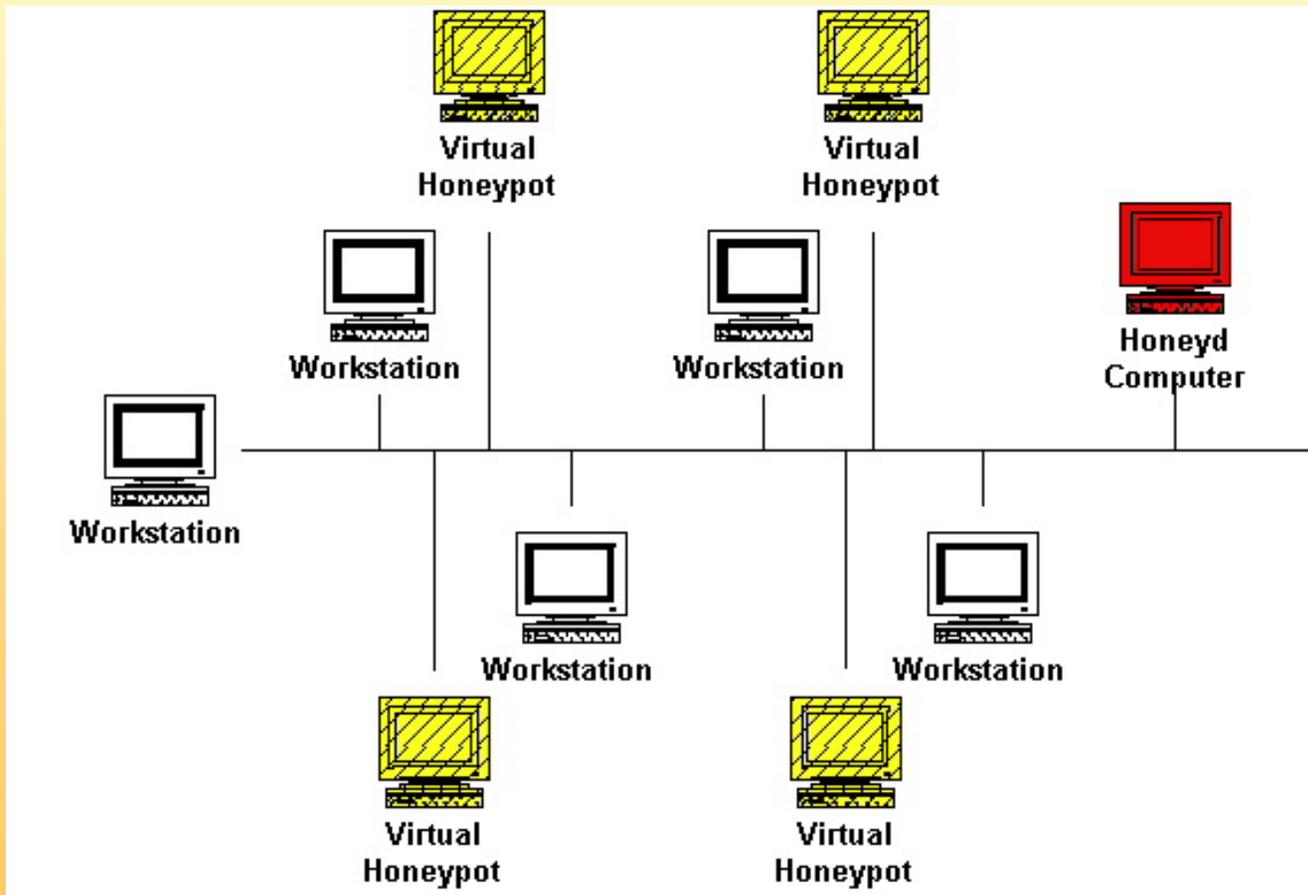
## **Disadvantages**

- Limited field of view (microscope)
- Risk (mainly high-interaction honeypots)

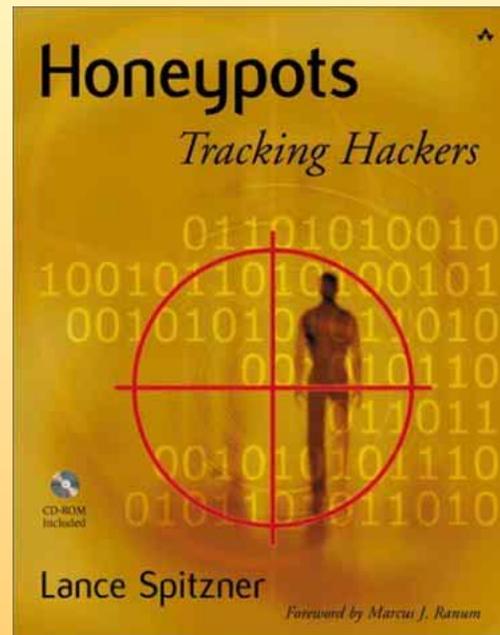
## **Examples of honeypots**

- Honeyd
- Specter
- ManTrap
- NetBait
- Honeynets

# Honeyd monitoring unused IPs



# Honeypots: Learn More



<http://www.tracking-hackers.com>

# Honeynets

- Nothing more than one type of honeypot.
- High-interaction honeypot designed to capture in-depth *information*.
- Its an architecture, not a product or software.
- Populate with live systems.

## How it works

- A highly controlled network where every packet entering or leaving is monitored, captured, and analyzed.
- Any traffic entering or leaving the Honeynet is suspect by nature.

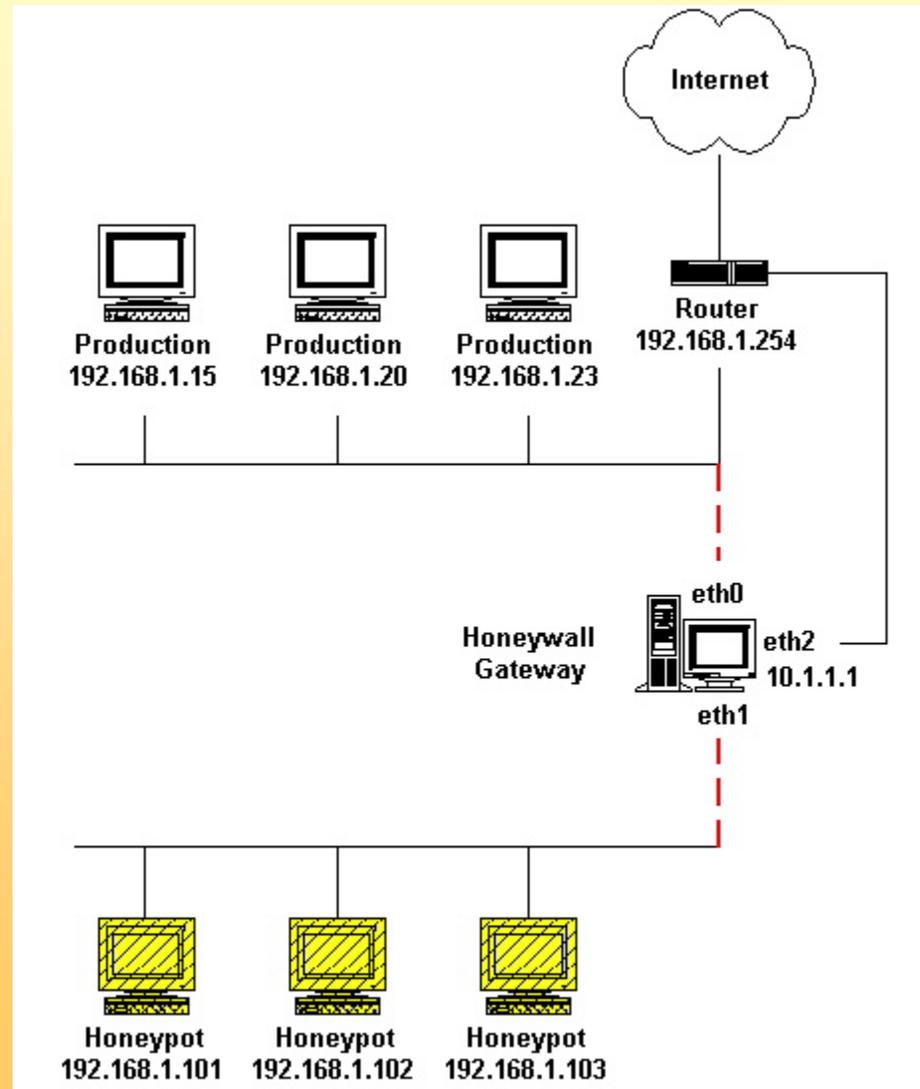
<http://www.honeynet.org/papers/honeynet/>

# Honeynet Requirements

- Data Control
- Data Capture
- Data Collection (for distributed Honeynets)

<http://www.honeynet.org/alliance/requirements.html>

# Honeynet - GenII



## Data Control - GenII

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 53
(msg:"DNS EXPLOIT named";flags:A+;
content:"|CD80 E8D7 FFFFFFFF|/bin/sh";
replace:"|0000 E8D7 FFFFFFFF|/ben/sh";)
```

<http://snort-inline.sourceforge.net>

**Sebek**

## **Its all about observation**

- How do you observe and intruder without their knowledge?
- In Cliff Stoll's adventures at Berkeley, he monitored the serial lines that the intruder used to access the system in question.
- Today's approach is similar, we now catch packets off the network.

## The complicating factor

- Network packet captures are a great source of information.
  - TCP session reassembly
  - FTP file extraction.
- What happens if the intruder encrypts his or her session?
- Do we brute force the encryption?
- Is there a easy way to circumvent the encryption?

## The Arms Race

1. Blackhats start to use session encryption to protect their communications
2. Honeynet operators installed trojaned shells to capture keystrokes
3. Blackhats installed their own shell as part of their rootkit
4. Honeynet operators started to look at Data Capture from within the kernel.

# Circumventing Encryption

- Sebek was born out of a desire to observe intruder behavior even when session encryption was in use.
- This meant either breaking the encryption or circumventing it.
- Sebek does not break the encryption it just does an end run around it.

## What does Sebek do?

- Operates inside kernel space, gathering forensic data of interest to researchers.
- Once gathered, it “invisibly” exports this data to a remote collector.
- Examples of the types of data it can collect:
  - keystrokes
  - secure file transfers using SCP
  - Burneye passwords
  - All kinds of other stuff too!

## Whats changes since last years talk?

- There are efforts to port Sebek to:
  - OpenBSD
  - XP
  - Solaris
- New packet hiding capabilities added
- Records all read data now (no more blind spots).
- Resides entirely in kernel space, almost entire rewrite of all code.
- New web based data analysis tool

## Who developed Sebek

- Sebek is a collaborative effort within the HoneyNet Project.
  - Edward Balas                      Linux & Data Analysis
  - Job de Hass                        Solaris
  - K2                                      Solaris
  - Mike Davis                         Windows XP
  - Dragos Ruiu                        OpenBSD
- Originally based on the Adore Rootkit and coded by Mike Clark for the linux OS.

## Sebek Design

- There are 2 components:
  - The client, which collects the data on the honeypot and exports it to the server.
  - The server, which records and processes the data exported by honeypots.

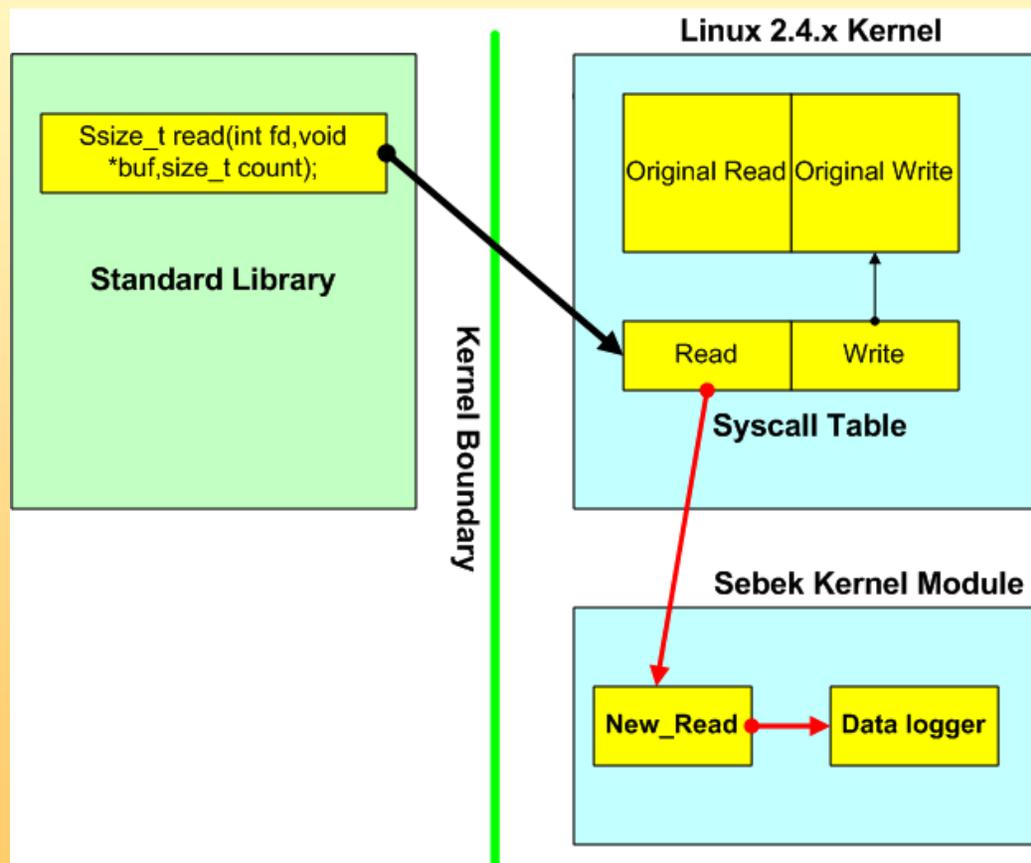
## The Sebek client

- For this presentation we will focus on the Linux version of Sebek.
- Sebek is currently implemented as a Linux Kernel Module(LKM).
- The first version of Sebek was based on the Adore Rootkit. This is no longer true.
- Sebek records any data that is accessed via the `sys_read` call.
- Can record all `sys_read` activity or just keystrokes.

## How does it get access to this data?

- It changes the `syscall_table`, so when someone does a read system call, our code is executed.
- All data accessed via the `sys_read` call is recorded.
- After the data is recorded it is immediately exported to the collector.

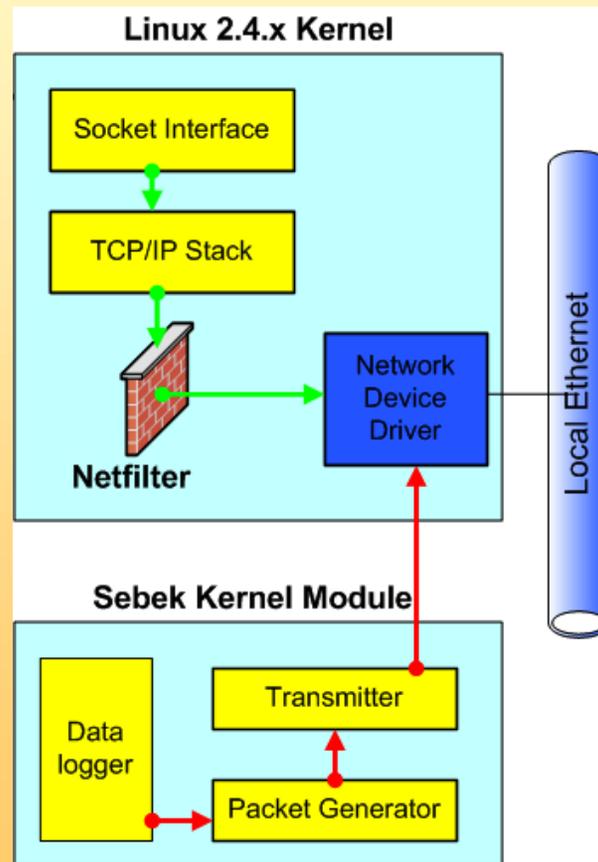
# Data Capture Illustration



## Data Export

- Sebek creates its own ethernet frames
- Sends them directly to the ethernet driver
- The packets bypass all filtering and packet capture code paths.
- This makes it impossible for a user on the honeypot to see the Sebek packets leaving the host.

# Data Export Illustration



## Data Export part 2

- At install time, the module also installs its own version of the raw socket implementation.
- This new implementation has a feature to silently drop packets which have a matching magic value in the packet header.
- This allows Sebek to not only hide its own packets but also packets from other honeypots.

## The captured data

- The data is exported to a collector using using UDP.
- Each packet contains exactly one Sebek record.
- The packets are generated entirely in Sebek, and require a static IP to MAC mapping.

# The Data Export Format

- Sebek packets have two components:
  - a fixed length head
  - a variable length payload.
- The head contains 10 fixed length fields.
- The payload is binary data of the length defined in the head.

# Packet Header Format

<b>Magic</b>	u_int32_t	identifies packets to hide
<b>Version</b>	u_int8_t	version number
<b>Counter</b>	u_int32_t	packet counter
<b>Time_Sec</b>	u_int32_t	timestamp in seconds
<b>Time_Usec</b>	u_int32_t	residual microseconds
<b>PID</b>	u_int32_t	process ID
<b>UID</b>	u_int32_t	user ID
<b>FD</b>	u_int32_t	file descriptor number
<b>COM</b>	char[10]	command name
<b>Len</b>	u_int32_t	payload length

## Other features

- Sebek can still modify the `syscall_table` even if it's symbol isn't exported to kernel modules.
- The Sebek LKM is removed from the linked list of modules making it invisible to the system.
- No log files or any user space modifications required.

## Install Tips

- Delete the tarball from the system after install, or better yet burn it on cd and unmount the cd after install.
- Make sure the shell isn't recording the commands you issue to install Sebek.
- If you installed from the file system, you may want to consider a way of scrubbing that data from the FS buffer in `/proc/kcore`
- Set the correct Dest MAC address or the ethernet frames will be sent to every switch port.

## **The Server Side**

- Captures Sebek packets.
- Uploads them into a mysql database
- Data is examined using the soon to be available web based analysis front end or a script that dumps keystroke logs.

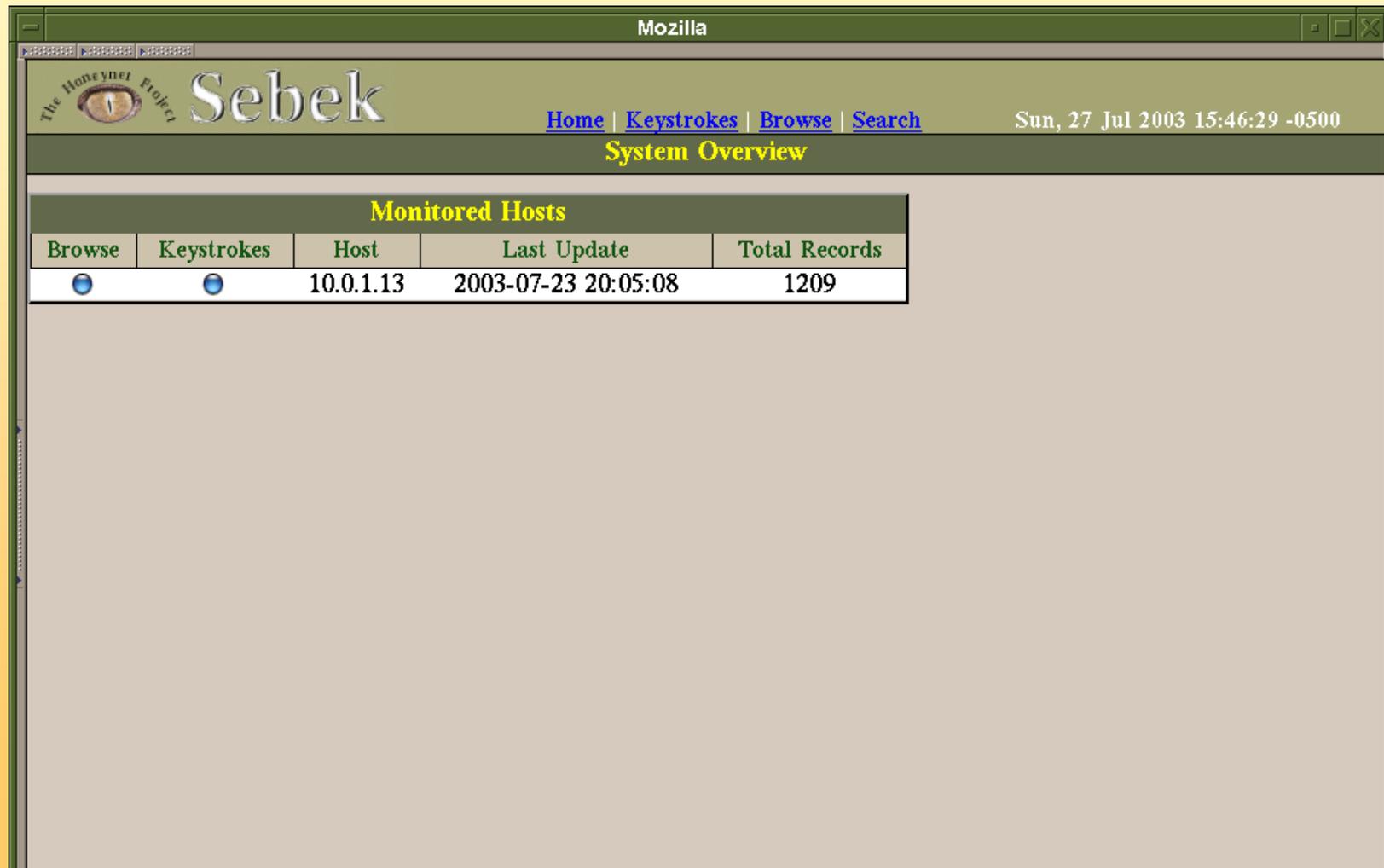
## Data Analysis

- Example shows a non-root user copying a file to his home directory.
- The file is a Burneye protected copy of a ptrace exploit.
- The user runs the binary and gains root access.

## **Analysis Questions**

- Can we recover the SCPed file using the web interface?
- Can we determine the password used to run the Burneye binary?
- Can we determine exactly when the user gained root access?

# Main Page: All hosts summary



The screenshot shows a Mozilla browser window displaying the Sebek web interface. The page title is "Sebek" and the date is "Sun, 27 Jul 2003 15:46:29 -0500". The main heading is "System Overview". Below this is a table titled "Monitored Hosts" with the following data:

Browse	Keystrokes	Host	Last Update	Total Records
<a href="#">Browse</a>	<a href="#">Keystrokes</a>	10.0.1.13	2003-07-23 20:05:08	1209

# Looking at Keystrokes

Mozilla

The HoneyNet Project **Sebek** [Home](#) | [Keystrokes](#) | [Browse](#) | [Search](#) Sun, 27 Jul 2003 15:46:40 -0500

**Keystroke Summary View for IP: 10.0.1.13**

Details	IP	PID	UID	COMMAND	FD	DATA
<input type="radio"/>	10.0.1.13	1318	0	sh	0	[2003-07-23 20:04:33]# ls [2003-07-23 20:04:34]# less messages [2003-07-23 20:04:52]# cd /etc [2003-07-23 20:04:54]# mkdir ... [2003-07-23 20:04:57]# ls
<input type="radio"/>	10.0.1.13	1323	0	less	3	[2003-07-23 20:04:35]# \000 [2003-07-23 20:04:50]# q
<input type="radio"/>	10.0.1.13	1321	0	w	6	[2003-07-23 20:04:09]# w\000
<input type="radio"/>	10.0.1.13	1271	500	bash	0	[2003-07-23 20:03:29]# ho[BS] [BS] who [2003-07-23 20:03:33]# w [2003-07-23 20:03:43]# ./malware [2003-07-23 20:03:47]# chmod ux[BS] +x mal [2003-07-23 20:03:52]# ./mal
<input type="radio"/>	10.0.1.13	1312	500	w	6	[2003-07-23 20:03:33]# w\000
<input type="radio"/>	10.0.1.13	1271	500	bash	3	[2003-07-23 20:03:24]# [BS] [BS]
<input type="radio"/>	10.0.1.13	1304	500	tput	3	[2003-07-23 20:03:24]# \000
<input type="radio"/>	10.0.1.13	1305	500	wc	0	[2003-07-23 20:03:24]# [BS]
<input type="radio"/>	10.0.1.13	1307	500	tput	3	[2003-07-23 20:03:24]# \000
<input type="radio"/>	10.0.1.13	1302	500	tput	3	[2003-07-23 20:03:24]# \000
<input type="radio"/>	10.0.1.13	1252	0	mingetty	0	[2003-07-23 20:03:16]# blackhat
<input type="radio"/>	10.0.1.13	1263	0	sshd	7	[2003-07-23 20:02:07]# \000\000\000
<input type="radio"/>	10.0.1.13	1264	500	scp	0	[2003-07-23 20:02:07]# C0664 38802 malware [2003-07-23 20:02:09]# \000
<input type="radio"/>	10.0.1.13	1263	0	sshd	3	[2003-07-23 20:02:09]# \000
<input type="radio"/>		0		sshd	4	[2003-07-23 20:02:02]# SSH-2.0-OpenSSH_3.1n1



# Using the SCP decode option

Mozilla

The HoneyNet Project **Sebek** [Home](#) | [Keystrokes](#) | [Browse](#) | [Search](#) Sun, 27 Jul 2003 15:46:52 -0500

**Details**

IP: 10.0.1.13 Command: scp bash sshd  
 PID: 1264 UID: 500 0

View as: sys\_read Data Context

SCP File Transfer	Text / Keystrokes	Raw Data	UID	File Desc	Command	Start	End	Total Bytes
<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	500	0	scp	2003-07-23 20:02:07	2003-07-23 20:02:09	38823 bytes read
<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	500	3	bash	2003-07-23 20:02:07	2003-07-23 20:02:07	5249 bytes read
<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	500	3	scp	2003-07-23 20:02:07	2003-07-23 20:02:07	10172 bytes read
<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	0	7	sshd	2003-07-23 20:02:07	2003-07-23 20:02:07	3383 bytes read

**SCP File Transfer Decode for PID 1264 FD 0**

File Name: [malware](#)  
 Expected Size: 38802 bytes  
 Observed Size: 38802 bytes  
 Permissions: C0664  
 Lost 0 bytes

## Looking at the SCPed file

- We have now recovered a file named **malware** from PID 1264 FD 0.
- After downloading, we examined the file with strings.
- “**TEEE burneye - TESO ELF Encryption Engine**”
- This is a burneye binary

# Lets take a closer look at malware's activity

The screenshot shows a Mozilla browser window displaying the Sebek web interface. The page title is 'Sebek' and it includes navigation links for Home, Keystrokes, Browse, and Search. The search form has the following fields:

- IP address:
- UID:
- PID:
- Command:

A button labeled 'Fire it up gene!' is located below the search form.

The 'Summary View: Com malware' table shows the following data:

Details	IP Address	PID	UID	COMMAND	START	END
<input type="radio"/>	10.0.1.13	1318	500	malware	2003-07-23 20:04:00	2003-07-23 20:04:00
<input type="radio"/>	10.0.1.13	1315	500	malware	2003-07-23 20:03:56	2003-07-23 20:03:56





## Back to the Questions

- We were able to recover the file named **malware**, which was transferred using SCP.
- The password used to run **malware** was “**secret**”
- The blackhat user gained root access
  - Timestamp 2003-7-23 20:04:01
  - Process ID 1318
  - File Descriptor 0

## Current Limits

- Data export mechanism uses UDP and thus does not provide reliable stream transport.
- Nothing is perfect, and as such there are ways to detect Sebek on a host.
- Sebek often collects a good deal of data that is not of interest
- Being a kernel module if the system reloads then sebek isnt automatically reinstalled.

## The Future

- Ability to compile directly into kernel
- Make harder to detect
- anti-anti-Sebek techniques ;-)
- provide a better facility for users to express what data they want to collect.
- improved data analysis.

## Where Can I get Sebek

- Old version of Sebek can be downloaded from <http://www.honeynet.org/tools/sebek>
- For questions or comments contact Edward Balas
  - ebalas@iu.edu
- New version will be available in next couple of weeks.

## The Next Steps

### Bootable CDROM

- Boot any PC into a Honeynet gateway (Honeywall)
- Simplified interface
- Preconfigured logging to central system

### User Interface

- System management
- Data Analysis

## **Honeywall CDROM**

- Developed by Rob McMillen
- CDROM boots into Honeywall gateway

# User Interfaces

Honey Inspector v2 results - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Favorites

Address [https://216.80.71.109/cgi-bin/inspect2.pl?start\\_month=Jan&start\\_day=18&start\\_year=2003&start\\_hour=8&start\\_minute=8&end\\_mor](https://216.80.71.109/cgi-bin/inspect2.pl?start_month=Jan&start_day=18&start_year=2003&start_hour=8&start_minute=8&end_mor) Go

2003-01-18 15:42:16	TCP	202.107.52.170	34781	->	10.1.1.105	21	<a href="#">view, p0f, ARIN</a> (100)
<b>2003-01-18 15:45:18</b>	<b>TCP</b>	<b>202.107.52.170</b>	<b>53763</b>	->	<b>10.1.1.103</b>	<b>21</b>	<a href="#">view, p0f, ARIN</a> (651)
2003-01-18 15:45:18	TCP	202.107.52.170	53764	->	10.1.1.101	21	<a href="#">view, p0f, ARIN</a> (604)
<b>2003-01-18 15:45:18</b>	<b>TCP</b>	<b>10.1.1.101</b>	<b>1027</b>	->	<b>202.107.52.170</b>	<b>113</b>	<a href="#">view, ARIN</a> (100)
<b>2003-01-18 15:47:04</b>	<b>TCP</b>	<b>202.107.52.170</b>	<b>53996</b>	->	<b>10.1.1.101</b>	<b>21</b>	<a href="#">view, p0f, ARIN, Snort</a> (15k)
<b>2003-01-18 15:47:05</b>	<b>TCP</b>	<b>10.1.1.101</b>	<b>1028</b>	->	<b>202.107.52.170</b>	<b>113</b>	<a href="#">view, ARIN</a> (100)
<b>2003-01-18 15:50:41</b>	<b>TCP</b>	<b>202.107.52.170</b>	<b>54018</b>	->	<b>10.1.1.101</b>	<b>21</b>	<a href="#">view, p0f, ARIN, Snort</a> (16k)
<b>2003-01-18 15:50:42</b>	<b>TCP</b>	<b>10.1.1.101</b>	<b>1029</b>	->	<b>202.107.52.170</b>	<b>113</b>	<a href="#">view, ARIN</a> (100)
<b>2003-01-18 15:52:16</b>	<b>TCP</b>	<b>62.99.207.73</b>	<b>3068</b>	->	<b>10.1.1.101</b>	<b>80</b>	<a href="#">view, p0f, ARIN, plugin</a> (9k)
<b>2003-01-18 15:53:28</b>	<b>TCP</b>	<b>202.162.193.147</b>	<b>61115</b>	->	<b>10.1.1.101</b>	<b>22</b>	<a href="#">view, p0f, ARIN</a> (55k)
<b>2003-01-18 15:54:46</b>	<b>TCP</b>	<b>10.1.1.101</b>	<b>1030</b>	->	<b>212.15.64.41</b>	<b>80</b>	<a href="#">view, ARIN, plugin</a> (522k)
2003-01-18 15:54:46	ICMP	10.14.0.20	0	->	10.1.1.101	0	<a href="#">view, ARIN</a> (0)
2003-01-18 15:55:37	ICMP	10.14.0.20	0	->	10.1.1.101	0	<a href="#">view, ARIN</a> (0)
<b>2003-01-18 15:56:34</b>	<b>TCP</b>	<b>10.1.1.101</b>	<b>1031</b>	->	<b>205.158.62.27</b>	<b>25</b>	<a href="#">view, ARIN</a> (1k)
2003-01-18 15:57:35	UDP	64.56.227.36	1026	->	10.1.1.101	137	<a href="#">view, ARIN</a> (78)
2003-01-18 15:57:35	UDP	64.56.227.36	1026	->	10.1.1.103	137	<a href="#">view, ARIN</a> (78)
2003-01-18 15:57:35	UDP	64.56.227.36	1026	->	10.1.1.104	137	<a href="#">view, ARIN</a> (78)

Opening page [https://216.80.71.109/cgi-bin/inspect2.pl?start\\_month=Jan&start\\_day=18&](https://216.80.71.109/cgi-bin/inspect2.pl?start_month=Jan&start_day=18&) Internet

## **Honeynet Tools**

Find all the latest Honeynet tools for Data Control, Capture, and Analysis at the Honeynet Tools Section.

<http://www.honeynet.org/tools/>

<http://www.honeynet.org>

<project@honeynet.org>