ICEB 2002 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-10-2002

The Study and Implementation of Network-Based Auditing System with Session Tracking and Monitoring

Yu-Jen Chen

Wenchuan Hsieh

Yi-Hsien Chiu

Chen-Hwa Song

The Study and Implementation of Network-Based Auditing System with Session Tracking and Monitoring

Yu-Jen Chen
Department of
Information Management
Chang-Gung University
cyr@mail.cgu.edu.tw

Wen-Chuan Hsieh
Department of
Information Management
Shu-Te University
wch@mail.stu.edu.tw

Yi-Hsien Chiu
Department of
Information Management
Shu-Te University
dean_thinker2@pchome.com.tw

Chen-Hwa Song
CCL Labs
Industrial Technology
Research Institute
chsong@itri.org.tw

Abstract

With consideration of the increasing importance of auditing system and the present auditing systems' incapability of performing packet reassembling analysis, this research attempts to develop a "network-based auditing system with session tracking and monitoring" to assist network administrators to analyze and rearrange the packets into separate session groups. This developed system is able to reveal every single step of the unauthorized activities. As a result, the administrators can investigate each network session and its transferred data more efficiently, and reduced greatly the time for auditing data analysis. In addition, the event reconstruction simulates the actual event occurred at that time; this feature provides network administrators with more detailed and realistic insight concerning vulnerabilities in network security that need to be fixed. Also, this system keeps track of all network events, and collects related information in a set of auditing files (log files). Moreover, the collected records and reassembled files can serve as evidences in tracing cyber-crimes and as references for recovery process.

Keywords: network-based auditing system, session tracking, session reassembling, cyber-crime

1. Introduction

Ever since computer's birth, its popularity grows with its increasing importance. In fact, it has become an indispensable tool in today's modern society. Internet's advent creates a new medium that allows messages and data to be transmitted globally. As Internet progresses, more and more crimes are gradually extending its domain from human communities to this virtual dimension. And, network security has become a critical issue for users around the world. Nowadays, hackers vandalizing servers, violating privacy and stealing confidential information have become the new type of criminal act, which often causes victims to suffer from great damages. Hence, effective management and sound defensive facilities are crucial in ensuring network security. Techniques such as implementation of IDS and firewall have being widely applied to fortify and protect network system from intrusion. However, despite of putting in great effort in enforcing the defense measures, there is still no guaranteed security. Nevertheless, an auditing system can be installed to keep track of all system events and collect related information in a set of auditing files. These records served as evidences and references that can be utilized later in tracing cyber-crimes. Very often, the notion of getting caught can prevent hackers from actually carrying out the illegal act. Also, by analyzing auditing files, system managers are able to discern unusual activities. Presently, many types of intrusion detection system (IDS) have been proposed; however, those auditing files are kept in binary form [1] [2] [10] [12] and requires manual work to "read out", hence it requires complicated expertise and significant amount of time. In consideration of the increasing importance of auditing system and the need of improving efficiency of analysis on auditing files, this research attempts to develop a network-based auditing system, that has the ability of session tracking and monitoring. This developed system will rearrange those auditing data recorded by network-based IDS to reconstruct and simulate the real events occurred to overcome analysis difficulties. In addition, records and reassembled files can serve as evidences in tracing cyber-crimes and as references for recovery process.

2. Literature Research

2.1 Definition of Auditing System in Information Technology

"Auditing System" is a system that keeps track of system events and examines the audit files and audit trail actively, periodically or randomly to detect any trespasses and suspicious activities. All the information system on firewall, computer and server will records detailed information about each services provided. And, the collection of all these records, storing the information concerning system events, is called an audit file or log file. By studying the information extracted from audit files, system administrators are able to discern both implicit and explicit problems. Also, solutions can be devised in accords to the given analysis result in order to prevent similar problems form affecting system operation again.

Network auditing, simply, is a mechanism that targets on tracking network-specific events. Generally speaking, these detected activities are, likewise, stored into log files for reference in the future. Through network auditing, network administrator can investigate the data traveling between servers and clients, or monitor a specific connection and event. Such mechanism is capable of detecting and tracing any unauthorized activities, as well as potential hackers.

Therefore, administrators should examine these log files periodically. However, the analysis process is complicated and the audit files' content is difficult to understand; hence, the task is preferably to be completed with the assistance from analyzing tools.

2.2 Categories of Auditing System

Auditing systems can be categorized into "Host-Based" and "Network-Based", according to the source which they collect information from. Host-Based auditing system's main mission is to monitor the intrusion events from the host itself. For instance, user priority, file access and system manipulation are all monitored locally. Network-based auditing system treats each incoming and outgoing packet with different storing processes. The packet recording process and historical audit files often provide helpful information, such as debug process, system performance evaluation and evidence for illegal acts, in times of anomalies' occurrence. Network-Based auditing system captures packets stealthily; it gives no hint to the suspect in realizing the existence of this system. On the other hand, the audit files kept by Host-Based auditing system are likely to be modified by intruders intending to erase the crime evidence.

2.3 Present Auditing System

2.3.1 Host-Based Auditing System

- tripwire: tripwire is a "System File Integrity Inspection System", which finds out what are the files that have been modified. The system performance is satisfactory, and it occupies minimal space. Also, tripwire inspect the forged sessions coming from intruders. In addition, tripwire can re-adjust itself to accommodate various system configurations and provide flexible control in managing system file locations.
- Swatch: swatch is "Simple WATCHer" for short. It is the most popular host-based auditing system under Unix operating system. Swatch verifies the new-recorded events in the historical audit files to check whether they satisfy the requirements of predetermined configuration. Once the system event matches, Swatch will report to the user immediately. Moreover, it can even monitor log file of syslogd (a unix daemon), and respond with suitable solutions when problem arises.

2.3.2 Network-Based Auditing System

The mostly utilized Network-based Auditing Systems are as follows:

- TCPdump: It is currently most popular network-based auditing system. TCPdump is an "OSI Certified Open Source Software" [6]. Therefore, any user can download this tool without charge. TCPdump captures the packets traveling on network for analysis. It is able to monitor network users' (especially the potential hackers) every single activity. However, despite of its convenience and fame, understanding the generated auditing data may rather be difficult.
- Snort: Likewise, snort another free IDS software supported by Open Source. This system has the capability of monitoring the computers of a same domain. It searches through the captured packet data, and perform analysis on the contained information. Snort compares attach pattern with deterministic rules to detect various kinds of attacks. Its core source code is an altered version of tcpdump; hence, the audit files generated also troubled many users.

All in all, analyzing the audit files generated by the present auditing system, especially the network-based, requires the knowledge of network specialist. Due to the fact that packets are directly stored as in raw format, analysis process becomes inconvenient and time-consuming. Also, analysts have to one by one investigate each packet in the audit file. In attempting to solve this problem, this research proposed to introduce the concept of session into the analysis procedure.

3. The Proposed System

The present network-based auditing systems are often incapable of performing session analysis. Moreover, records accumulated for a long period of time usually lack in organization and have no meaningful structure. Therefore, it posts a considerable difficulty for the analysts. If these records can be rearranged prior to the analysis process, the complication of the work will be greatly reduced. As a result, analysts would, then, be able to find out unauthorized activities more efficiently. In consideration of the problem and needs, this research proposed "Network-Based Auditing System with session tracking and monitoring".

3.1 System Framework

The proposed system followed the traditional network-based IDS framework. In order to capture the transmitting packets, the system must be located between intranet and gateway. As soon as the system is activated, Ethernet interface card will immediately be placed into promiscuous mode, starting to collect the packets. First, the captured packets are automatically processed by network card driver and system kernel. Then, they are sent to the proposed auditing system for further analysis. The system framework is shown in below:

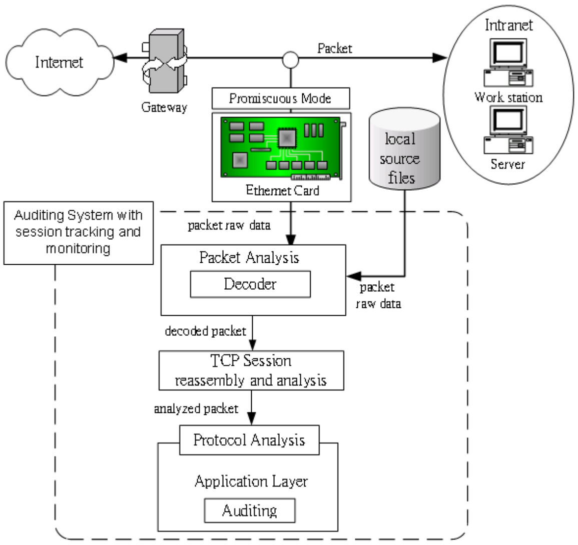


Figure 1. System framework

According to figure 1, the input raw packet data can either be obtained from local source files or packets captured on network. While system is operating on on-line capturing mode, it will keep collecting the packets traveling.

After the packets have been captured they are analyzed and categorized immediately. Meanwhile, the captured raw packets will also be stored in libpcap format (section 4.1) locally for reference in the future. Afterwards, these raw packets will be reassembled, preparing for restoring actual network events and activities.

First of all, raw packet data are sent to packet analyzer module for structure analysis in order to determine individual headers. In addition, the decoder contained within is responsible for converting raw data into understandable textual or numerical format.

The early stage of the analysis focuses only on packets' structure and characteristics. Consequently, the result is simply a collection of raw packets ordered by precedence; therefore, no distinct relationship is established between them. In fact, this is the major problem that creates the complication on analyst's work. In order to overcome this obstacle, it's necessary to design a calculation to rearrange the packets into individual session groups. TCP session analysis module is specifically designed to serve that purpose.

Finally, the analyzed packets will arrive at protocol analysis module. This module is the last module prior to packets' entry into application layer. It determines the packets' protocol type and decides how they will be treated for the later process. At this stage, analysis process will vary in accords to the protocol. This is where the TCP session groups from the previous module are combined into files if applicable. In the end, the analysts will be able to see the actual event be restored. After analytical processes have completed their task, each packet will then be summarized in textual form and stored into an audit file.

3.2 TCP Session Analysis and Reassembling

Usually, data will be chapped locally into several pieces prior to its transmission when its length is considered too long. Then, they will be reassembled into its original form after arriving the destination. Hence, packet reassembling is the process of reconstruction and rearrangement of packets base on their individual protocol, characteristics and sequence.

According to figure 2, in TCP layer, the communicating sides must synchronize with each other before any connection is established. This synchronization process is also known as "Three-Way handshaking". Thus, this special interaction procedure signifies each beginning of a new session. Likewise, prior to the end of a session, both sides will inform each other that they are about to close the transmission by sending FIN signal. In other words, every TCP session can be determined by "Three-Way handshaking" and FIN signal. After the beginning and the end of each session are defined, the rest of the packets will be grouped in accords to their IP, port , acknowledge and sequence number.

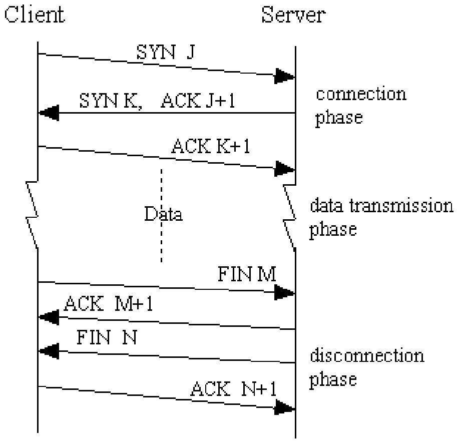


Figure 2. TCP session [13]

Once the session groups have been completely analyzed, reassembly takes place. In this research, attempts are made to simulate the reconstruction process at application level. For HTTP, the restoring process focuses on the commands and files transmitted. For example, a client sent a "GET" request to a web server for an index page file. Then, the server responds with a HTTP message followed by series of packets containing the requested file data. The request and respond packets provide crucial information in the reassembly process. For instance, the name of the requested file can be referenced from the "GET" command line in the request message, and content length can be found in HTTP message. Finally, with the given information and sequence number, the requested file will be generated using the data contained in the packets.

4. System Design and Demonstration

The proposed system, "Network-Based Auditing System with session tracking and monitoring", is developed under Windows and Linux operating system, using interface library WinPcap and Libpcap respectively. Libpcap is a network capture library developed by Network Research Group (NRG) of the Information and Computing Sciences Division (ICSD) at Lawrence Berkeley National Laboratory (LBNL) in Berkeley, California. Libpcap and several other packet analyzing tools are often included as part of the operating system as auxiliaries.

4.1 System Design

As what has been mentioned, Libpcap is an interface library, which provides packet capturing function for other applications. It has defined a file format, which the implementing application should apply to. Libpcap file consists of several sections including "File Header", "Record Header" and raw packet data; and no paddings in between. Each section header provides information pertaining to its content.

In a libpcap file, the section of the first 24 bytes is called "File Header", which is followed by many packet records. And, each record consists of 16 bytes "Record-Header" and the contained packet raw data. In other words, a libpcap file has only one "File-Header" but many packet records. A conceptual libpcap log file structure diagram is shown below.

In a libpcap file, the start of each packet record can be determined by record headers. Likewise, this header provides several important information, such as packet length, capture data and time. Applications must recognize the headers in order to access the raw packet data the follows.

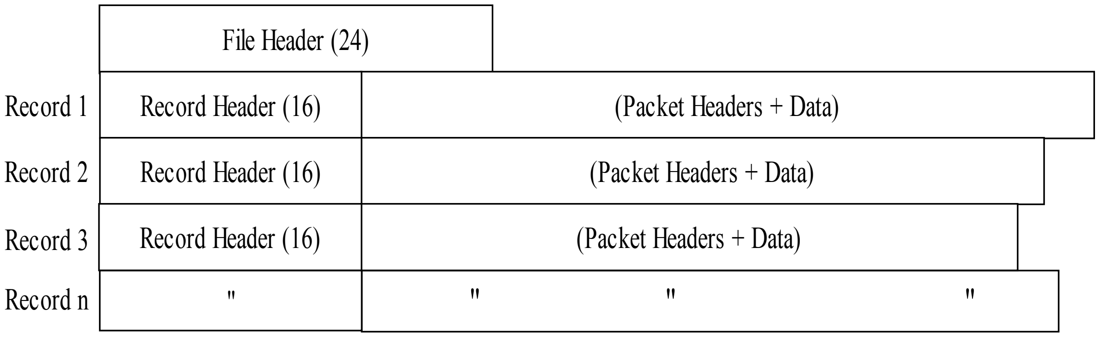


Figure 3. Libpcap file structure

Table 1. File-header format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MN				MajV		MinV		TZO				TSA			
LLT				SNP											

Table 2. File-header fields detail

Field	Length (byte)	Meaning
MN	4	Magic Number is a 4 bytes long hex: "0x1a2b3c4d". The main purpose is for the application, that is reading this file, to determine whether the writing pattern is big-endian or little-endian.
MajV	2	Major Version Number
MinV	2	Minor Version Number
TZO	4	Time Zone Offset
TSA	4	Time Stamp Accuracy
LLT	4	Link Layer Type
SNAP	4	Snap Length: The maximum length of the captured packet data.

Table 3. Record-header format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TS								CPL				ACL			

Table 4. Record-header fields detail

Field	Length (byte)	Meaning
TS	8	Time Stamp: The first 4 bytes represents the total seconds from the date of 1970/1/1 00:00:00 up to the time of packet being captured. The next 4 bytes are microsecond.
CPL	4	Captured Length: The length of the packet captured (can be configured).
ACL	4	Actual Length: The actual length of the packet captured.

4.2 System Requirements

This system is written in Python; hence, it also supports cross platform. Presently, this system has been tested to operate both on Linux and NT environment. There are a few software requirements. First of all, needs to install Python interpreter version 2.2, and libpcap library for Linux and Winpcap library for NT. There is no required for extra hardware requirement besides Ethernet card. However, a machine that is set up for IDS is, of course, the faster the better, and preferably equipped with larger storage space. That is, high performance hardware is likely to carry out its mission more efficiently.

4.3 Scenario: Monitor HTTP As An Example

In order to provide convenient and efficient control, this system has a graphical user interface. Once the system is activated, the user enters the main executing window immediately. In figure 4, sections marked A, B, C, D and E are parameter section, session tracking, packet summary, detailed packet structure tree and raw packet data view. Every system function can be started from this window, which is the core of the whole system. This system works in two different modes: real-time capturing and analyzing mode. Users can directly switch between modes with a mouse click.

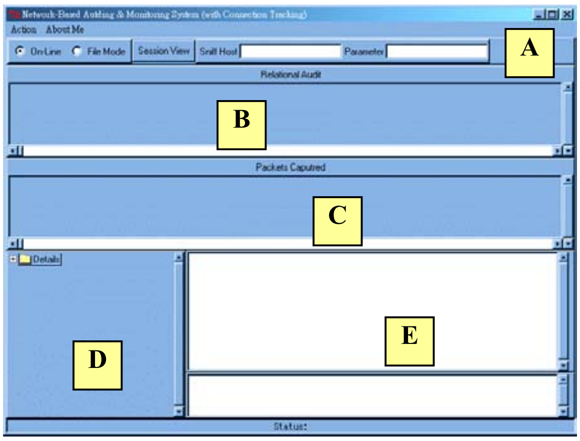
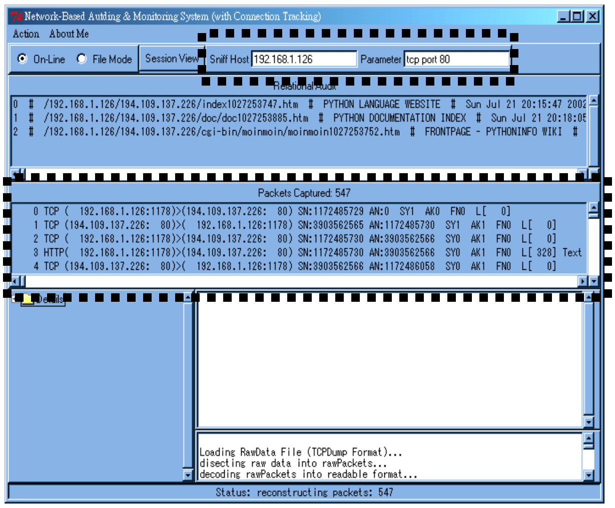


Figure 4. Main executing window

When it is placed into real-time capturing mode, the system will start to capture every incoming and outgoing packet. Of course, network administrator can also enter specific parameters and focus on a particular IP or protocol.

As shown in figure 5, packets that satisfied the defined requirements, will be captured. Filtering out the uninterested packets, the overall analysis accuracy and efficiency will be improved.


Network-Based Autding & Monitoring System (with Connection Tracking)
Action
About Me
File Mode
Session View
Relational Audit
Details

● On-Line ○ File Mode

Sniff Host 192.168.1.126 Parameter tcp port 80

```
0  ‡  /192.168.1.126/194.109.137.226/index1027253747.htm  ‡  PYTHON LANGUAGE WEBSITE  ‡  Sun Jul 21 20:15:47 2002
1  ‡  /192.168.1.126/194.109.137.226/doc/doc1027253885.htm  ‡  PYTHON DOCUMENTATION INDEX  ‡  Sun Jul 21 20:18:05
2  ‡  /192.168.1.126/194.109.137.226/cgi-bin/moinmoin/moinmoin1027253752.htm  ‡  FRONTPAGE - PYTHONINFO WIKI  ‡
```

Packets Captured: 547

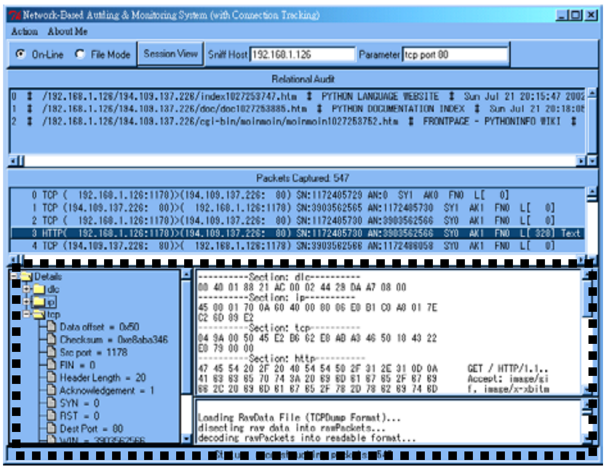
```
   0 TCP (  192.168.1.126:1178)>(194.109.137.226:  80) SN:1172485729 AN:0  SY1  AK0  FN0  L[   0]
   1 TCP (194.109.137.226:  80)>(  192.168.1.126:1178) SN:3903562565 AN:1172485730  SY1  AK1  FN0  L[   0]
   2 TCP (  192.168.1.126:1178)>(194.109.137.226:  80) SN:1172485730 AN:3903562566  SY0  AK1  FN0  L[   0]
   3 HTTP(  192.168.1.126:1178)>(194.109.137.226:  80) SN:1172485730 AN:3903562566  SY0  AK1  FN0  L[ 328] Text
   4 TCP (194.109.137.226:  80)>(  192.168.1.126:1178) SN:3903562566 AN:1172486058  SY0  AK1  FN0  L[   0]
```

```
Loading RawData File (TCPDump Format)...
disecting raw data into rawPackets...
decoding rawPackets into readable format...
```

Status: reconstructing packets: 547

Figure 5. Capturing packets that satisfied the requirements

After packets have been captured, the system immediately takes the responsibility to analyze their structure. The result will be shown in the packet summery section. To see more detailed information about a specific packet, users can simply click on a packet listed in summary section. Then, the complete packet structure (with consisted headers) and raw packet data (in hex format) will be shown (figure 6).


Network-Based Autding & Monitoring System (with Connection Tracking)
Action
About Me
File Mode
Session View
Relational Audit

● On-Line ○ File Mode

Sniff Host 192.168.1.126 Parameter tcp port 80

```
0  ‡  /192.168.1.126/194.109.137.226/index1027253747.htm  ‡  PYTHON LANGUAGE WEBSITE  ‡  Sun Jul 21 20:15:47 2002
1  ‡  /192.168.1.126/194.109.137.226/doc/doc1027253885.htm  ‡  PYTHON DOCUMENTATION INDEX  ‡  Sun Jul 21 20:18:05
2  ‡  /192.168.1.126/194.109.137.226/cgi-bin/moinmoin/moinmoin1027253752.htm  ‡  FRONTPAGE - PYTHONINFO WIKI  ‡
```

Packets Captured: 547

```
   0 TCP (  192.168.1.126:1178)>(194.109.137.226:  80) SN:1172485729 AN:0  SY1  AK0  FN0  L[   0]
   1 TCP (194.109.137.226:  80)>(  192.168.1.126:1178) SN:3903562565 AN:1172485730  SY1  AK1  FN0  L[   0]
   2 TCP (  192.168.1.126:1178)>(194.109.137.226:  80) SN:1172485730 AN:3903562566  SY0  AK1  FN0  L[   0]
   3 HTTP(  192.168.1.126:1178)>(194.109.137.226:  80) SN:1172485730 AN:3903562566  SY0  AK1  FN0  L[ 328] Text
   4 TCP (194.109.137.226:  80)>(  192.168.1.126:1178) SN:3903562566 AN:1172486058  SY0  AK1  FN0  L[   0]
```

```
Details
  dlc
  ip
  tcp
    Data offset = 0x50
    Checksum = 0xe8aba346
    Src port = 1178
    FIN = 0
    Header Length = 20
    Acknowledgement = 1
    SYN = 0
    RST = 0
    Dest Port = 80
    WIN = 3903562566
```

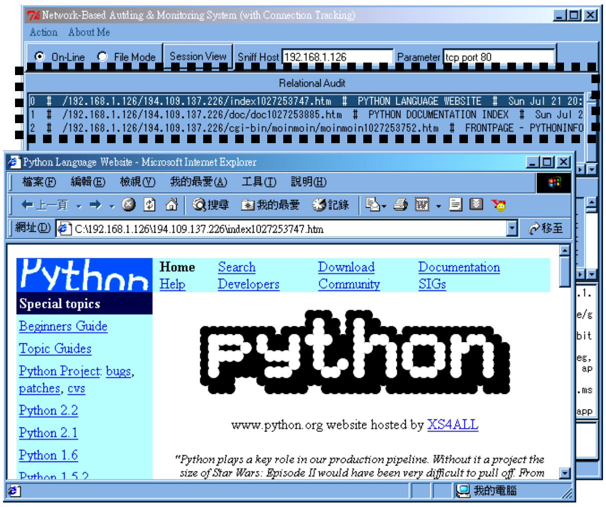
```
----------Section: dlc----------
00 40 01 88 21 AC 00 02 44 28 DA A7 08 00
----------Section: ip----------
45 00 01 70 0A 60 40 00 80 06 E0 B1 C0 A8 01 7E
C2 6D 89 E2
----------Section: tcp----------
04 9A 00 50 45 E2 D6 62 E8 AD A3 46 50 18 43 22
E0 79 00 00
----------Section: http----------
47 45 54 20 2F 20 48 54 54 50 2F 31 2E 31 0D 0A      GET / HTTP/1.1..
41 63 63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69      Accept: image/gi
66 2C 20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D      f, image/x-xbitm
```

```
Loading RawData File (TCPDump Format)...
disecting raw data into rawPackets...
decoding rawPackets into readable format...
```

Status: reconstructing packets: 547

Figure 6. Packet’s full detail

This system is also capable of performing session tracking and packet reassembling. That is, the collected packets are processed and reassembled by the system to generate files; for example, a web page. These files will be listed in chronological order in the session tracking section. Users can directly view the file by selecting the listed item. As in HTTP, the files are most likely to be the web pages or graphics. In fact, with this mechanism, the administrators can see the restoration of the actual event occurred and what exactly the monitored users have done. Indeed, this feature can allow administrators to manage the network much more effectively. As shown in figure 7, 8 and 9, the pages browsed by the monitored user are all captured and listed.


Network-Based Autding & Monitoring System (with Connection Tracking)
Action
About Me
○ On-Line
○ File Mode
Session View
Python Language Website - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
記錄
網址(D)
移至
我的電腦

Sniff Host 192.168.1.126 Parameter tcp port 80

Relational Audit

```
0  #  /192.168.1.126/194.109.137.226/index1027253747.htm  #  PYTHON LANGUAGE WEBSITE  #  Sun Jul 21 20:
1  #  /192.168.1.126/194.109.137.226/doc/doc1027253885.htm  #  PYTHON DOCUMENTATION INDEX  #  Sun Jul 2
2  #  /192.168.1.126/194.109.137.226/cgi-bin/moinmoin/moinmoin1027253752.htm  #  FRONTPAGE - PYTHONINFO
```

C:\192.168.1.126\194.109.137.226\index1027253747.htm

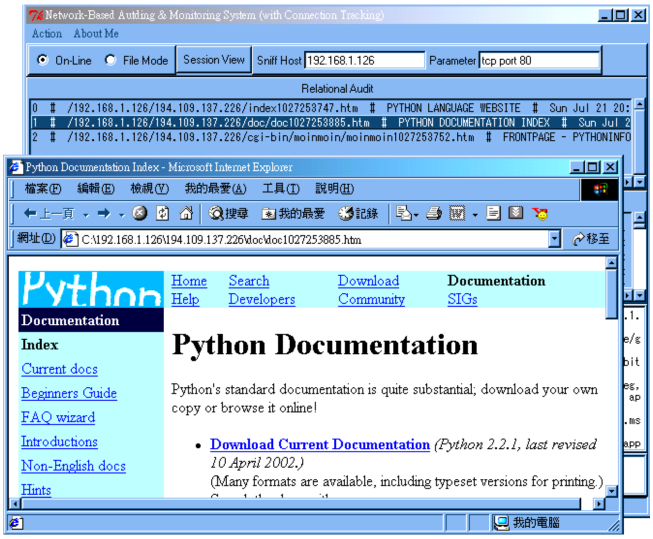
Python | Home | Search | Download | Documentation | Help | Developers | Community | SIGs

Special topics: Beginners Guide | Topic Guides | Python Project: bugs, patches, cvs | Python 2.2 | Python 2.1 | Python 1.6 | Python 1.5.2

www.python.org website hosted by XS4ALL

"Python plays a key role in our production pipeline. Without it a project the size of Star Wars: Episode II would have been very difficult to pull off. From

Figure 7. The result of the reassembled packets: web pages browsed by the monitored user


Network-Based Autding & Monitoring System (with Connection Tracking)
Action
About Me
○ On-Line
○ File Mode
Session View
Python Documentation Index - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
記錄
網址(D)
移至
我的電腦

Sniff Host 192.168.1.126 Parameter tcp port 80

Relational Audit

```
0  #  /192.168.1.126/194.109.137.226/index1027253747.htm  #  PYTHON LANGUAGE WEBSITE  #  Sun Jul 21 20:
1  #  /192.168.1.126/194.109.137.226/doc/doc1027253885.htm  #  PYTHON DOCUMENTATION INDEX  #  Sun Jul 2
2  #  /192.168.1.126/194.109.137.226/cgi-bin/moinmoin/moinmoin1027253752.htm  #  FRONTPAGE - PYTHONINFO
```

C:\192.168.1.126\194.109.137.226\doc\doc1027253885.htm

Python | Home | Search | Download | Documentation | Help | Developers | Community | SIGs

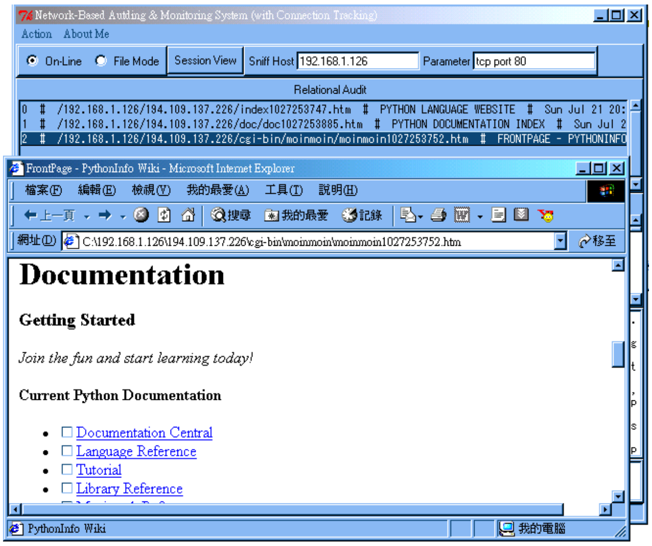
Documentation: Index | Current docs | Beginners Guide | FAQ wizard | Introductions | Non-English docs | Hints

Python Documentation

Python's standard documentation is quite substantial; download your own copy or browse it online!

- Download Current Documentation *(Python 2.2.1, last revised 10 April 2002.)* (Many formats are available, including typeset versions for printing.)

Figure 8. Web pages browsed by the monitored user (continuation)


Network-Based Autding & Monitoring System (with Connection Tracking)
Action
About Me
○ On-Line
○ File Mode
Session View
FrontPage - PythonInfo Wiki - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
記錄
網址(D)
移至
PythonInfo Wiki
我的電腦

Sniff Host 192.168.1.126 Parameter tcp port 80

Relational Audit

```
0  #  /192.168.1.126/194.109.137.226/index1027253747.htm  #  PYTHON LANGUAGE WEBSITE  #  Sun Jul 21 20:
1  #  /192.168.1.126/194.109.137.226/doc/doc1027253885.htm  #  PYTHON DOCUMENTATION INDEX  #  Sun Jul 2
2  #  /192.168.1.126/194.109.137.226/cgi-bin/moinmoin/moinmoin1027253752.htm  #  FRONTPAGE - PYTHONINFO
```

C:\192.168.1.126\194.109.137.226\cgi-bin\moinmoin\moinmoin1027253752.htm

Documentation

Getting Started

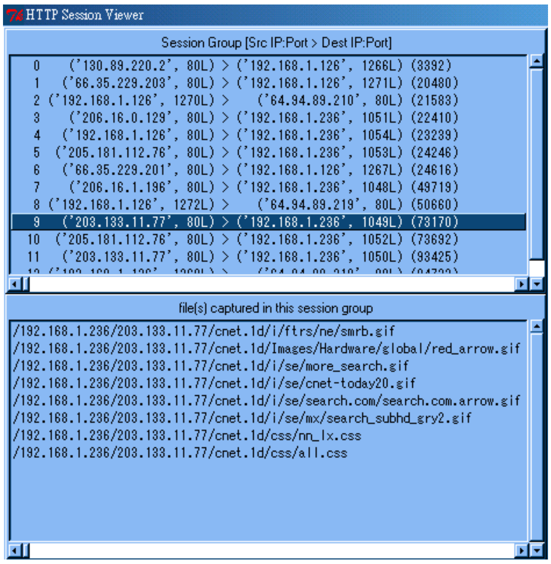
Join the fun and start learning today!

Current Python Documentation

- □ Documentation Central
- □ Language Reference
- □ Tutorial
- □ Library Reference

Figure 9. Web pages browsed by the monitored user (continuation)

Other than regular analysis, network administrators can also utilize this system to determine individual session. In figure 10, each session in the session tracking window is grouped by its IP and port (both source and destination). Both total number of consisting packets and the sum of data length are noted behind each session group. After selected a session group from the list, the files that are generated in this session will be listed in the window shown below. Then, by clicking on a specific file, the users are able to see the packets that constitute it (figure 10, 11).



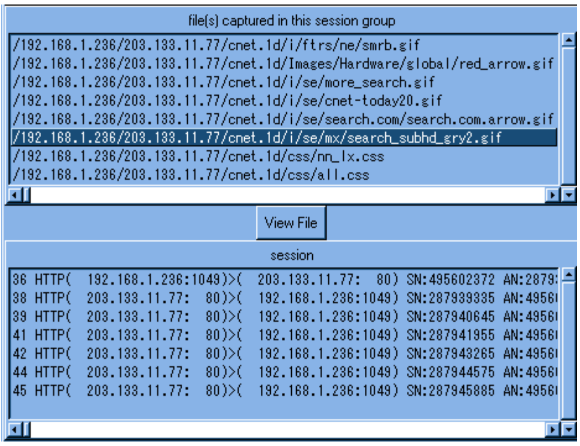
Session Group [Src IP:Port > Dest IP:Port]

```
 0    ('130.89.220.2', 80L) > ('192.168.1.126', 1266L) (3392)
 1   ('66.35.229.203', 80L) > ('192.168.1.126', 1271L) (20480)
 2 ('192.168.1.126', 1270L) >    ('64.94.89.210', 80L) (21583)
 3    ('206.16.0.129', 80L) > ('192.168.1.236', 1051L) (22410)
 4   ('192.168.1.126', 80L) > ('192.168.1.236', 1054L) (23239)
 5  ('205.181.112.76', 80L) > ('192.168.1.236', 1053L) (24246)
 6   ('66.35.229.201', 80L) > ('192.168.1.126', 1267L) (24616)
 7    ('206.16.1.196', 80L) > ('192.168.1.236', 1048L) (49719)
 8 ('192.168.1.126', 1272L) >    ('64.94.89.219', 80L) (50660)
 9   ('203.133.11.77', 80L) > ('192.168.1.236', 1049L) (73170)
10  ('205.181.112.76', 80L) > ('192.168.1.236', 1052L) (73692)
11   ('203.133.11.77', 80L) > ('192.168.1.236', 1050L) (93425)
```

file(s) captured in this session group

```
/192.168.1.236/203.133.11.77/cnet.1d/i/ftrs/ne/smrb.gif
/192.168.1.236/203.133.11.77/cnet.1d/Images/Hardware/global/red_arrow.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/more_search.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/cnet-today20.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/search.com/search.com.arrow.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/mx/search_subhd_gry2.gif
/192.168.1.236/203.133.11.77/cnet.1d/css/nn_lx.css
/192.168.1.236/203.133.11.77/cnet.1d/css/all.css
```

Figure 10. Selecting a desired session group

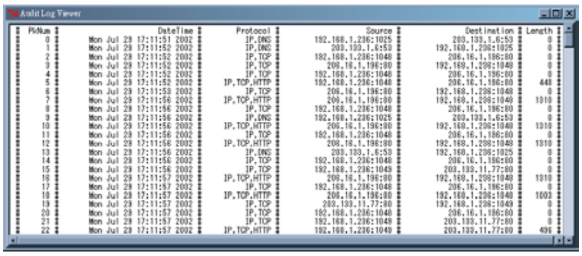


```
/192.168.1.236/203.133.11.77/cnet.1d/i/ftrs/ne/smrb.gif
/192.168.1.236/203.133.11.77/cnet.1d/Images/Hardware/global/red_arrow.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/more_search.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/cnet-today20.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/search.com/search.com.arrow.gif
/192.168.1.236/203.133.11.77/cnet.1d/i/se/mx/search_subhd_gry2.gif
/192.168.1.236/203.133.11.77/cnet.1d/css/nn_lx.css
/192.168.1.236/203.133.11.77/cnet.1d/css/all.css
```

```
36 HTTP(  192.168.1.236:1049)>(  203.133.11.77:  80) SN:495602372 AN:2879:
38 HTTP(  203.133.11.77:  80)>(  192.168.1.236:1049) SN:287939335 AN:4956
39 HTTP(  203.133.11.77:  80)>(  192.168.1.236:1049) SN:287940645 AN:4956
41 HTTP(  203.133.11.77:  80)>(  192.168.1.236:1049) SN:287941955 AN:4956
42 HTTP(  203.133.11.77:  80)>(  192.168.1.236:1049) SN:287943265 AN:4956
44 HTTP(  203.133.11.77:  80)>(  192.168.1.236:1049) SN:287944575 AN:4956
45 HTTP(  203.133.11.77:  80)>(  192.168.1.236:1049) SN:287945885 AN:4956
```

Figure 11. The selected file's consisting packets

Each analyzed packet will be recorded in textual audit files, and these records will become important references for future verification (figure 12).



```
PktNum             DateTime       Protocol             Source          Destination Length
     0  Mon Jul 29 17:11:51 2002       IP,DNS  192.168.1.236:1025        203.133.1.6:53      0
     1  Mon Jul 29 17:11:52 2002       IP,DNS      203.133.1.6:53    192.168.1.236:1025      0
     2  Mon Jul 29 17:11:52 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
     3  Mon Jul 29 17:11:52 2002       IP,TCP     206.16.1.196:80    192.168.1.236:1048      0
     4  Mon Jul 29 17:11:52 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
     5  Mon Jul 29 17:11:52 2002  IP,TCP,HTTP  192.168.1.236:1048        206.16.1.196:80    448
     6  Mon Jul 29 17:11:53 2002       IP,TCP     206.16.1.196:80    192.168.1.236:1048      0
     7  Mon Jul 29 17:11:56 2002  IP,TCP,HTTP     206.16.1.196:80    192.168.1.236:1048   1310
     8  Mon Jul 29 17:11:56 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
     9  Mon Jul 29 17:11:56 2002       IP,DNS  192.168.1.236:1025        203.133.1.6:53      0
    10  Mon Jul 29 17:11:56 2002  IP,TCP,HTTP     206.16.1.196:80    192.168.1.236:1048   1310
    11  Mon Jul 29 17:11:56 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
    12  Mon Jul 29 17:11:56 2002  IP,TCP,HTTP     206.16.1.196:80    192.168.1.236:1048   1310
    13  Mon Jul 29 17:11:56 2002       IP,DNS      203.133.1.6:53    192.168.1.236:1025      0
    14  Mon Jul 29 17:11:56 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
    15  Mon Jul 29 17:11:56 2002       IP,TCP  192.168.1.236:1049      203.133.11.77:80      0
    16  Mon Jul 29 17:11:57 2002  IP,TCP,HTTP     206.16.1.196:80    192.168.1.236:1048   1310
    17  Mon Jul 29 17:11:57 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
    18  Mon Jul 29 17:11:57 2002  IP,TCP,HTTP     206.16.1.196:80    192.168.1.236:1048   1003
    19  Mon Jul 29 17:11:57 2002       IP,TCP    203.133.11.77:80    192.168.1.236:1049      0
    20  Mon Jul 29 17:11:57 2002       IP,TCP  192.168.1.236:1048        206.16.1.196:80      0
    21  Mon Jul 29 17:11:57 2002       IP,TCP  192.168.1.236:1049      203.133.11.77:80      0
    22  Mon Jul 29 17:11:57 2002  IP,TCP,HTTP  192.168.1.236:1049      203.133.11.77:80    496
```

Figure 12. Textual audit file

5. Conclusion and Future Improvements

Network security is becoming increasingly important. Besides firewall and intrusion detection measures, enforcing an efficient and effective security mechanism is the key to secure working environment. The importance of IDS should not be ignored. Auditing system will be the crucial monitoring technique towards cyber-crime. In this research, a "Network-Based Auditing System with session tracking and monitoring" is developed. Despite the packets are captured or retrieved from a raw packet source file, the actual monitored network events can be reassembled and replayed.

This system solves the problem of network administers in overcoming the difficulty in analyzing the audit files. When network attacks occur, the administrator can specify the query time and event to reveal the suspected network activities. These reassembled records will serve as references that can provide great help in tracing the intruders.

In the future research, the system is expected to support more protocols, such as dns, ftp, smtp, pop3 and etc. Also, a through evaluation should be done on the system operating under various network traffic stress in measuring its stability. If necessary, further improvements will be made on this auditing system's efficiency by transforming this centralized system to distributed system.

Acknowledgements

This research was sponsored by National Science Council, Republic of China (project number: NSC 90-2416-H-182-010).

Reference

[1] Biswanath Mukherjee, L. Todd Hoberlein, and Karl N. Levitt, "Network Intrusion Detection," IEEE Network, May./June, 1994

[2] Fan,Xiu-Wei et. "The detection and record of abnormal behavior in network", 第六屆資訊管理學術暨警政資訊實務研討會, 2002

[3] Ghosh, A.K.; Wanken, J.; Charron, F. ,"Detecting anomalous and unknown intrusions against programs", Computer Security Applications Conference, 1998 Proceedings 14th Annual, 1998

[4] Gregory B. White, Eric A. Fisch, and Udo W. Pooch, "Cooperating Security Managers: A peer-based Intrusion Detection System", IEEE Network, Jan./Feb. 1996

[5] http://netgroup-serv.polito.it/winpcap

[6] http://www.opensource.org/licenses/

[7] Jai Sundar, Jose Omar, David Isacoff, Eugene Spafford, and Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents", IEEE 1998

[8] Keven Richards, "Network based Intrusion Detection: A Review of Technologies", Computer & Security, 18 (1999)

[9] Lin, Feng-Ming et., "Intrusion Detection: a Network View", TANet2001 Conference, October 2001

[10] Lunt, T.F.; Jagannathan, R.; Lee, R.; Whitehurst, A.; Listgarten, S. "Knowledge-based intrusion detection", AI Systems in Government Conference, 1989 Proceedings of the Annual, 1989

[11] Nicholas Puketza, Mandy Chung, Ronald A. Olsson, and Biswanath Mukherjee, "A software platform for testing intrusion detection systems", IEEE software 1997

[12] Shen, Wen-Chu, "Monitoring and behavior analysis of network security systems" unpublished Master's Thesis, National Taiwan University Management Information Department, 2000

[13] TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994, ISBN 0-201-63346-9

[14] Teresa F. Lunt, "Real-time Intrusion Detection", IEEE 1989

[15] Vigna, G.; Kemmerer, R.A.," NetSTAT: a network-based intrusion detection approach", Computer Security Applications Conference, 1998 Proceedings 14th Annual