Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

A Security Analysis of System Event Logging with Syslog

An analysis of the system event logging protocol, syslog is discussed. A review of the problems with the syslog protocol are described. Theses security problems include the transmission of system log data in clear text, use of UDP for network transfer and storage of event data in clear text. A survey of some of the syslog replacements was done. The paper concludes with a discussion of how one might go about creating a reasonably secure logging infrastructure.

Copyright SANS Institute
Author Retains Full Rights

AD

A Security Analysis of System Event Logging with Syslog

Kenneth E. Nawyn
2003-05-28

Abstract
An analysis of the system event logging protocol, syslog is discussed. A review of the problems with the syslog protocol are descibed. Theses security problems include the tranmission of system log data in clear text, use of UDP for network transfer and storage of event data in cleartest. A survey of some of the syslog replacements was done. The paper concludes with a disucssion of how one might go about creating a reasonably secure logging infastructure.

© SANS Institute 2003, Author retains full rights

Introduction

A common task of a network, systems or security administrator is the analysis of system and device event log files. This task might be done to determine if a system or network intrusion has occurred or to determine the events which an intruder used to gain unauthorized access to a system. In many organizations this event information is collected and compiled using the "syslog" protocol, via the logger[1]

[1]Logger(1) manual entry, 4.2BSD User's Manual

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

program, the syslog system calls[2]. and the syslogd[3] daemon process. In this paper I will examine how the syslog protocol fails to provide for logging of event data in a confidential, integral and available fashion. I will show work being done to create a more secure syslog protocol, and I will provide recommendations for the deployment of syslog in a secure fashion in a networked environment.

History

The syslog protocol was introduced by the Computer Science Research Group (CSRG) at the University of California-Berkeley (UC-Berkeley) as part of the Berkeley Software Distribution of UNIX[4]. Syslog was designed to provide the ability to report system events. These events are collected by a process and then recorded in event log files on the local system, a remote system or both. Also included with the syslogd and system calls is the logger program allows a user to write entries to the syslogd daemon process from the command line interface or from a shell script. When the syslogd daemon process starts a socket is created in the /dev directory, (/dev/log) or in more recent releases of UNIX, the socket is created in the /var/run (/var/run/log) directory. A configuration file, /etc/syslog.conf is used to control how the information sent to the syslogd process is stored. Programs can also communicate with the syslogd daemon process using the syslog system call provided with UNIX systems. It is common at many sites to collect this event data both in a file or files on the local system and to also forward that data via a UDP socket across the network to a centralized "logging" server. This allows an administrator to monitor the log files at one location rather than trying monitor the log files on a large number of systems. As the number of systems managed per administrator increases, it becomes more common for sites to move to the centralized logging of syslog data.

As other types of devices have been introduced into a computer network, an increased dependence upon the collection of log data via syslog has occurred. It is very common for network switches, network routers, printers, disk storage units, tape libraries, and other devices to be configured to log data via the syslog protocol. This is facilitated by the fact that besides listening to the socket on the local filesystem (i.e. /dev/log, or /var/run/log) the syslogd process also binds to network port 514. On this network socket, the syslogd process listens for UDP packets containing event log information. As port 514 is in the range of 1 to 1023, on a UNIX system, the syslogd process must run with "super user" or "root" privilege. This fact may not have been a security issue at the time of the development but it is now a point of concern for security administrators. The reason for this concern is that programmatic errors can lead to elevation of privilege. This elevation of privilege can lead to unauthorized access to critical systems and devices.

Failings of the Syslog

The syslog protocol has many failings. In the next few sections I will review the protocol and explain how it is insecure.. I will also discuss other matters which effect how a security administrator should view the parts of the syslog event delivery process.

Confidentiality

[2]Syslog(3) manual entry, 4.2BSD Programmer's Manual
[3]Syslog(8) manual entry, 4.2BSD Systems Manager's Manual
[4]UNIX is registered trademark of the Open Group.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The data forwarded by the syslogd daemon to a remote syslogd process is sent in clear text. It is viewable using a network analysis tool such as tcpdump. This allows an intruder to view the events being reported by routers or switches and by tools such as Snort[5]. The intruder can then use this information to determine which systems they may wish to target.

A very simple syslog configuration file (/etc/syslog.conf) is show below. A more common example of a syslog configuration file can be found in Appendix A.

```
[ken@sapphire ken]$ cat simple-syslog.conf
*.*             @loghost
```

This syslog configuration file forwards all syslog messages to the system named "loghost." The following session show how some one using the command tcpdump can collect data being transferred to a remote loghost. I used the strings command to extract the syslog data from the tcpdump capture file.

```
[root@sapphire tmp]# /usr/local/sbin/tcpdump -vvvv -w sans-
syslog dst host rose and port 514
tcpdump: listening on wlan0

16 packets received by filter
0 packets dropped by kernel

[root@sapphire tmp]# strings sans-syslog
9CU<6>kernel: device wlan0 entered promiscuous mode
<4>kernel: skb_p80211_to_ether: Host de-WEP failed, drs
<37>su(pam_unix)[1926]: authentication failure; lognams
<4>kernel: skb_p80211_to_ether: Host de-WEP failed, drx
4~ @
<37>su(pam_unix)[1927]: authentication failure; lognam
<4>kernel: skb_p80211_to_ether: Host de-WEP failed, dr
I<4>last message repeated 8 times
<38>su(pam_unix)[1928]: session opened for user root b
<4>kernel: skb_p80211_to_ether: Host de-WEP failed, dr
```

Integrity

The data stored by the syslog process is saved in clear text. It is easy to find the data from syslogd as it is saved in the /var/log directory on a new UNIX system or in the /var/adm directory on Solaris[6] or HP-UX[7] systems. The only technique used to control access to these files in the log directories are the standard UNIX permissions. The permissions are typically set so that only the "root" user has the ability to read from , or write to the log files. A listing of the /var/log directory from a RedHat[8] Linux system can be found in Appendix B. If an intruder gains access to a system which contains log files with "root" privilege, as is commonly the case, they will be able to modify these files. This will allow the intruder to reduce the chance of being discovered and also remove the record of his actions. If this happens, the system or security administrator will not be made aware of the intruder's presence. Tools to modify the files in either /var/adm or /var/log are commonly provided as part of most "rootkits."

Additionally when using a central loghost, an administrator is not able to determine if the log data has been

[5]SNORT Home Page, http://www.snort.org
[6]Solaris is a trademark of Sun Microsystems.
[7]HP-UX is a trademark of Hewlett-Packard.
[8]RedHat Home Page, http://www.redhat.com

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

modified at any time while being delivered to the syslogd daemon process or while it is traversing the network to the log host. As syslog uses UDP packets to deliver its event data there is no way of knowing if the syslog data is true or if the data has been modified. This failing is a direct result of the protocol which was chosen for network transport when the syslog protocol was designed.

Authenticity

As written the syslog protocol does not provide any tools to allow for authenticating data received by the syslogd process. In the example shown below the syslogd daemon has recorded the source hostname "sapphire".

```
May 23 23:15:43 sapphire su(pam_unix)[1926]: authentication failure;
logname= uid=500 euid=0 tty= ruser=ken rhost=  user=root
May 23 23:15:47 sapphire kernel: skb_p80211_to_ether: Host de-WEP failed,
dropping frame (-4).
```

Because there is no authentication feature in the the syslog protocol, the data sent in the previous example could easily be forged using readily available tools. Some common examples of these tools are "netcat" and its descendants, crypt-cat and aes-cat. These tools can be used to create syslog messages which contain source addresses which are either invalid or not correct. This "spoofing" of the source address compromises the integrity of the event log data. These tools along with other packet crafting tools can be used to flood a network with invalid event log data. This flood of data is commonly referred to as a "Denial of Service" (DOS) attack.

Another area of concern for the administrator is the "logger" program which is provided as part of the syslog tool set in many UNIX distributions. I have used the "logger" program torecord events when writing shell programs. While this facility is useful, it can also be used to "flood" the local syslogd process with messages. A message flood could easily cause DOS attack which could lead to the lose of important system event data. This will occur if the system is not able to write to the log files due to a exhaustion of disk space or in the case of a remote system, if the system is unable to process the large quantity of incoming syslog packets.

Programmatic Errors

In addition to the problems described in the previous sections, another area that any administrator must address are programmatic errors in the source code used by a vendor to create the syslogd, logger and other syslog processes. A search of the Common Vulnerabilities and Exposures Database[9] hosted by Mitre[10] shows 17 entries which contain references syslog. A search of the CERT[11] web site shows 74 entries which contain references to syslog. Many of these programmtic errors can result in a malicious user being able to gain "root" access on a system or being able to crash a system due to these faults. This means that a systems or network administrator must constantly track all bug and security mailing lists to determine if a problem has been discovered with a system which runs syslog in their domain. If a problem is found, the administrator must

[9]CVE web site, http://www.mitre.org/cve/
[10]Mitre web site, http://www.mitre.org
[11]CERT web site, http://www.cert.org

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

determine what risk that problem poses to the systems in their domain and then based upon their conclusion remedy the situation in a timely fashion. Other sources of this information include the "Bugtraq[12]" mailing list and the "Full Disclosure[13]" mailing list and equipment vendor's security mailing lists.

IETF RFC 3164, The BSD Syslog Protocol

While many systems and devices have used the "syslog" protocol to log system data and activity, no reference for the syslog protocol existed until this Internet Engineering Task Force (IETF) Request for Comment (RFC) was written. The version of the syslog protocol described in this RFC is based upon the authors observation of systems which implement the BSD Syslog protocol. In addition to describing the protocol, the author of this RFC has included comments which describe some of the problems which have been observed in the operation and use of the BSD syslog protocol. Any one who is interested in learning more about the syslog protocol or implementing version of it should start by reading this document.

Attempts at Improving Syslog

Many people have recognized the failings in the syslog protocol and the syslogd process. In this section, I will discuss work that has been done in an attempt to improve syslog tools. I will also point to tools and other resources which may assist the reader in improving the security of their site's logging environment.

Modular Syslog

Modular Syslog is project run to improve syslog being run by Core Security Technologies[14]. The modular syslog tool is available for download at Core Wisdom – Secure Logging[15] web page. The goal of this syslog replacement project is to provide an administrator with a version of syslog with which the administrator can verify the integrity of the event log data syslog has archived. The authors of modular syslog have added input and output modules to the syslogd daemon process. Two of the modules provided with the modular syslog software distribution allow the administrator to sign the event log data via the PEO-1 and L-PEO algorithms. If the event log data is modified an administrator these modules will allow an administrator to determine if the archived syslog data has been modified during network transfer or storage.

Modular Syslog also provides other modules which may be of interest to the security or systems administrator. Two of these modules provide the ability to submit system event log data to either a MySQL[16] or PostgreSQL[17] database. Lastly modular syslog provides a module which allows an administrator to use regular expressions to filter the log data.

In addition to "Modular Syslog" Core Security Technologies also provides "slogger." Slogger provides similar functionality to Modular Syslog in a Microsoft Windows environment. This tool is designed to replace the Windows Event Log facility and provide the same guarantee of data integrity for Windows

---

[12]Bugtraq Mailing List, http://archives.neohapsis.com/archives/sf/bugtraq
[13]Full Disclosure Security Mailing List, http://lists.netsys.com/mailman/listinfo/full-disclosure
[14]Core Security Technologies web site, http://www.corest.com
[15]Core Wisdom -Secure Logging web site, http://www1.corest.com/products/corewisdom/CW01.php?
[16]MySQL web site, http://www.mysql.com
[17]PostgreSQL web site, http://www.postgresql.org

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Event Log data that Modular Syslog provides in the UNIX environment.

NSYSLOG

The nsyslogd was developed by Darren Reed. It is available for download at the nsyslog[18] page maintained by Mr. Reed Nsyslog addresses the data integrity issue present in the current versions of syslog by replacing the use of UDP packets for data transfer with the use of Transmission Control Protocol (TCP) packet. TCP provides for reliable delivery of the log data as it requires all network packets which are sent to be acknowledged. Data confidentiality during network transfer is also provided by the ability to encrypt the nsyslog network packets using the Secure Socket Layer[19] (SSL) protocol. Nsyslog remains compatible with the current the syslog implementation on many UNIX system as it can be configured to listen for UDP packets on network port 514. Unfortunately, it does not work on the widely deployed LINUX[20] operating system. This is due to the non-standard use of the /dev/klog socket by the Linux syslog implementation to report events from the Linux kernel. From the age of the web page, it also does not appear that any additional development work is being done on this package.

SYSLOG-NG

The syslog-ng project is hosted by Balabit IT[21]. The home page for the syslog-ng project can be found at "Products Syslog_ng[22]" on the Balab IT web site. The goal of this project is to create a new syslog tool set which extends the current protocol to meet the needs of the current networked environment. Some of the extensions created in this tool include the ability to control how often data is written to a local disk using the “sync” option. The ability to control whether data is transmitted via UDP or TCP network packets. The ability to submit event log data to a database. One of the changes which the author of syslog-ng has made is to change the format of the syslog-ng.conf file. While this change is necessary to provide for the additional functionality in syslog-ng, it may take an administrator some time to convert to this new configuration file format. The author of syslog-ng has also worked to design this syslog replacement to work well in environments in which many level of event data forwarding occur. These changes assist in the deployment of this tool in firewalled or partitioned environments.

The author of syslog-ng has also done considerable work to allow for greater granularity in filtering of event data. This is being done due to increased volume of event data being generated in the current system and network environment. This ability will allow the administrator to segregate events so that they only monitor those events they deem important or filter the events by importance. More information on the working of syslog-ng can be found in an article which the author published in the Linux Gazette.

---

[18] Nsyslog web site, http://coombs.anu.edu.au/~avalon/nsyslog.html.
[19] **SSL 3.0 Specification,** http://www.netscape.net/eng/ssl3/
[20] LINUX is a trademark of Linux Torvalds
[21] Balab IT web site, http://www.balabit.hu/
[22] Balab IT Products Syslog-NG web site, http://www.balabit.hu/products/syslog_ng/

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

A growing community of users have deployed syslog-ng in their environments. From these users, a useful "Frequently Asked Questions" (FAQ) web page has been created. This FAQ covers topics such as configuration file options, SSH[23] forwarding of syslog-ng data, using syslog-ng with stunnel[24] and forwarding event data syslog-ng to a database. The FAQ can be found at Campin.net[25].

Honeynet Project

The collection of event log data from a Honeypot or Honeynet is very important. The members of the Honeynet Project[26] have done significant work on protecting the data collected by syslog facility on the systems which this group deploy. Of concern to the Honeynet project is protecting the confidentiality of the log host and the data which resides on the log host. This has been done due the need of the project to maintain an accurate log of all events leading up to and after a ttarget system (honeypot) has been compromised. To this end this group uses the following technique to collect syslog data. First they configure syslog on the honeypot to send the syslog event data to an unused network address or network broadcast address where the honeypot resides.. This hides the IP address of the syslog server from the intruder. Then on a system connected to the network which contains the target host (honeypot), Snort is run. The network interface for the logging host is not configured as being active on that network segment. A filter for Snort was created to listen for all syslog broadcasts on this inactive network interface and transfer the event data to a log file. This technique of logging is being called "Stealthful Logging". More information on how to configure Snort for this use can be found on the Honeynet Tools[27] web page. Addtionally Eric Hines from Fatelabs[28] has written two papers on this topic. These papers are "Complete Reference Guide to Creating a Secure Remote Syslog Server[i] and "Flying Pigs: Snorting Next Generation Secure Remote Log Servers over TCP". Much discussion of this technique has occurred on the Honeynet mail list. A searchable archive of this mailing list is maintained at Neohapsis Mailing List Archives[29]. Additional information on the setup of a "Stealth Logging" host can be found in this article from Linux Journal Magazine, Paranoid Peguin: Stealthful Sniffing, Intrusion Detection and Logging[30].

IETF RFC 3195[31], Reliable delivery for Syslog

This IETF RFC describes a reliable delivery mechanism for the syslog data. One of the stated goals of the authors of this RFC was to maintain backward compatibility with the protocol described in RFC 3164. This was accomplished by using the BEEP[32] protocol to provide a tunnel in which the syslog event data can be secured while being delivered. Additionally BEEP provides for authentication of the transmitting

[23] SSH web site, http://www.openssh.com
[24] Stunnlet web site, http://www.stunnel.org
[25] Syslog-NG FAQ, http://www.campin.net/syslog-ng/faq.html
[26] Honeynet web site, http://www.honeynet.org
[27] Honeynet Projects web site, http://project.honeynet.org/papers/honeynet/tools/
[28] Fatelabs web site, http://www.fatelabs.com
[29] Honeypot mailing list archives, http://archives.neohapsis.com/archives/sf/honeyposts/
[30] Paranoid Peguin Column, http://www.linuxjournal.com/article.php?sid=6222
[31] IETF RFC 3195, http://www.ieft.org/rfc/rfc3195.txt
[32] IETF RFC 3080, http://www.ietf.org/rfc/rfc3080.txt

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

host but creating the ability to require the host to authenticate itself using a certificate when sending the event data via this protocol. Versions of this syslog will continue to accept data via the current programmatic interface but will use the facilities in BEEP to reliably deliver the event data to a remote host.

Currently work is being performed at the San Diego Supercomputer Center on the implement a version of "secure syslog" based upon RFC 3195. The current version of this syslog replacment can be downloaded from the home page for the SDSC Secure Syslog[33] project. This code is currently only available for installation on systems running UNIX or a UNIX compatible systems. It does not provide for deployment of this tool for event logging on routers, switches or other systems which may be deployed in an enterprise. Thus while it may fill a need, until this protocol is available on all platforms, it is only a partial solution.

Logging tools listed on loganalysis.org

Tina Bird and Marcus Ranum has created the loganalysis[34] web site devoted to the analysis of logging data. While event log file analysis is not part of this paper, this site does provide links to many of the programs which have been discussed in this paper. In the library area there are a number of topics which any administrator will find of use. First is the listing of syslog replacements for UNIX systems. Second is the list of syslog tools for non-UNIX operating systems. And another area of interest is area which contains information on creating centralized logging environments. Tina Bird is a leading authority on building centralized logging systems. Much of her work is available for your review.

Recommendations for Syslog Implementations

To implement an event logging environment which is secure, scalable and reliable is quite a difficult task. These steps include creating a secure repository for the event logs, creating a reliable and secure transport mechanism for the event log data and having the ability to verify and maintain the integrity of the systems sending and receiving event log data.

Confidentiality

Maintaining the confidentiality of the event log data starts at the origin. All systems which will be generating event log data should be constructed so that the security of that system meets current "Best Practices" for that platform. Keeping the event log data confidential during transit is very important. A technique to keep the event data confidential is to encrypt the data during transit. One way that this can be done is by using SSH port forwarding for network tranport of the event log data. Using this technique, the encryption facility in SSH will prevent an unauthorized party from sniffing the event log data. Another alternative would be to use STunnel to wrap the data in SSL protocol. If all of the systems on a network are not capable of running SSH or using STunnel to encrypt the data, encryption of the data can be performed by using IP Security Protocol (IPSEC). A positive side effect of moving a network to IPSEC is that now all data transferred between these hosts will be encrypted. This will make it much more difficult for an unauthorized user to view network traffic. The use of the "Stealth Logging" techniques discussed in Eric Hines papers and on the Honeypot mailing list will make it much more difficult for some one to determine where the syslog data is stored. While it can be argued that this is a form of "security through obscurity", I feel that it is a valuable technique for maintaining the confidentiality of a site's event log data. I also recommend that one consider the use of an encrypted filesystem for the filesystem which will contain the log data on the syslog host. Properly configured, encrypted log files should allow for data to maintain confidential even if some one gains unauthorized access to the event log server.

[33]SDSC Syslog web site, http://security.sdsc.edu/software/sdsc-syslog/
[34]Log Analysis web site, http://loganalysis.org

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Integrity

Maintaining data integrity is a difficult process using syslog. The UDP packets used to transmit the data between hosts provide no guarantee of packet delivery. Again one should consider replacing the current syslog tools on the systems at your site with one of the newer versions. The nsyslog project, syslog-ng project and the SCSC Secure Syslog project address this issue issue by replacing UDP packets with TCP packets. The "modular syslog" project uses one-way hashes of the data to allow an administrator to verify that that has not been modified during its transfer or storage. The Secure Syslog project addresses the issue of data integrity during network transfer by using BEEP.

Authenticity

Currently, Secure Syslog host certificates to authenticate a system during network data transfer to meet this requirement. Or one could use the authentication facilities in SSH forwarding to meet this requirement. Unfortunately if a system or device does not support either Secure Syslog or SSH, it will be hard to determine the authenticity of data received by the syslogd daemon process.

Availability

Availability of the event log server to accept syslog data is important. As the syslog protocol uses UDP for network delivery of data, there is no guarantee of data receipt. If it is possible to use one of the TCP based syslog replacements in your environment, I would recommend it. Additionally the administrator should configure the syslog server with sufficient disk space to accept an atypically large amount of event log data. This will reduce the chance of losing event data due to exhaustion of disk space. Monitoring of disk space usage on the filesystem or disk partition where this data is stored must be setup and notification of changes in disk space usage patterns should be made the adminstrator's who monitor the system. The data that is collected on the system should be archived on a regular basis based upon the policies in place at your site and also based upon local, state and federal regulations.

If it is deemed that the syslog host must always be available, you should consider building a server which implements some form of "High-Availability" (HA). How to do this should be based upon the operating system deployed on the host and the tools which are valuable for that operating system. If a reader wishes to learn more about this topic an Internet search for HACMP for AIX, Sun Cluster for Solaris, MC-ServiceGuard for HP-UX and the Linux High Availability Project would be a good starting point. How you can create an HA syslog server is beyond the scope of this paper.

Conclusion

While there is much work currently being done to develop a secure event logging tool based upon the syslog protocol much work continues to need to be done. First, there are many devices and products that continue to use the UDP based syslog protocol to transmit event log data. Second, work needs to be done to store the collected data in a secure format so that the confidentiality and integrity of the data is maintained. Lastly, in a large networked environment the amount of data being collected by an event logging system can overwhelm the ability for the syslog system or systems to accept that data. The use of high capacity storage devices and high performance network interfaces is needed. Overcoming these problems is certainly a difficult task and may require the development of a new event logging protocol.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix A syslog.conf

The syslog.conf provides the rules for delivery of the event messages received by the syslogd process. Here is a sample /etc/syslog.conf file as found on a RedHat 8.0 Linux system.

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                         /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                        /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                                      /var/log/secure

# Log all the mail messages in one place.
mail.*                                                          /var/log/maillog

# Log cron stuff
cron.*                                                          /var/log/cron

# Everybody gets emergency messages
*.emerg                                                         *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                                  /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                        /var/log/boot.log
```

© SANS Institute 2003, Author retains full rights

Appendix B Log Directory Permissions

Here is listing of the /var/log directory on a RedHat 8.0 system. While most of the log files are restricted to access just by the user “root” they still can be modified by a system intruder. If an administrator is concerned about the integrity of these files, a hash of each file must be maintained in a secure location. As some of these files change regularly that hash will be only be valid for that those files at the time the hash was computed. Committing these files to “write once” or “write only” media would impede an intruder's ability to modify these files.

```
[root@sapphire var]# ls -l /var/log
total 1480
-rw-------    1 root     root        18027 May 23 22:35 boot.log
-rw-------    1 root     root        17016 May 18 21:44 boot.log.1
-rw-------    1 root     root            0 May  4 04:02 boot.log.2
-rw-------    1 root     root        22660 Apr 30 20:19 boot.log.3
-rw-------    1 root     root        12149 Apr 26 12:09 boot.log.4
-rw-------    1 root     root         9913 May 24 00:01 cron
-rw-------    1 root     root        13338 May 18 22:19 cron.1
-rw-------    1 root     root        14826 May 11 04:02 cron.2
-rw-------    1 root     root        13625 May  4 04:02 cron.3
-rw-------    1 root     root        12964 Apr 27 04:02 cron.4
drwxr-xr-x    2 lp       root         4096 Jan  8 05:59 cups
-rw-r--r--    1 root     root         6580 May 23 17:59 dmesg
drwxr-xr-x    2 root     root         4096 May 23 18:00 gdm
-rw-r--r--    1 root     root        63354 May 23 17:59 ksyms.0
-rw-r--r--    1 root     root        63354 May 23 16:49 ksyms.1
-rw-r--r--    1 root     root        63354 May 21 20:33 ksyms.2
-rw-r--r--    1 root     root        63354 May 18 21:13 ksyms.3
-rw-r--r--    1 root     root        63354 May 17 20:15 ksyms.4
-rw-r--r--    1 root     root        63354 May 16 19:50 ksyms.5
-rw-r--r--    1 root     root        63354 Apr 30 20:18 ksyms.6
-r--------    1 root     root     19136220 May 23 18:00 lastlog
-rw-------    1 root     root          415 May 22 04:02 maillog
-rw-------    1 root     root          830 May 18 22:19 maillog.1
-rw-------    1 root     root            0 May  4 04:02 maillog.2
-rw-------    1 root     root         1245 May  1 04:02 maillog.3
-rw-------    1 root     root          830 Apr 27 04:02 maillog.4
-rw-------    1 root     root        86440 May 24 00:11 messages
-rw-------    1 root     root        80243 May 18 21:44 messages.1
-rw-------    1 root     root         6377 May  9 21:07 messages.2
-rw-------    1 root     root       115726 May  3 23:28 messages.3
-rw-------    1 root     root        57071 Apr 26 20:22 messages.4
-rw-r--r--    1 root     root        16267 May 23 17:56 rpmpkgs
-rw-r--r--    1 root     root        16235 May 17 04:03 rpmpkgs.1
-rw-r--r--    1 root     root        16233 May 10 04:03 rpmpkgs.2
-rw-r--r--    1 root     root        16233 May  3 04:03 rpmpkgs.3
-rw-r--r--    1 root     root        16223 Apr 26 13:15 rpmpkgs.4
drwx------    2 root     root         4096 Apr  6 00:47 samba
-rw-r--r--    1 root     root        15267 Mar 18 19:24 scrollkeeper.log
-rw-------    1 root     root          607 May 23 18:00 secure
-rw-------    1 root     root          459 May 18 21:14 secure.1
-rw-------    1 root     root            0 May  4 04:02 secure.2
-rw-------    1 root     root          532 Apr 30 20:20 secure.3
-rw-------    1 root     root          374 Apr 26 12:09 secure.4
drwx------    2 root     root         4096 Feb 19 17:11 snort
-rw-------    1 root     root            0 May 18 22:19 spooler
-rw-------    1 root     root            0 May 11 04:02 spooler.1
-rw-------    1 root     root            0 May  4 04:02 spooler.2
-rw-------    1 root     root            0 Apr 27 04:02 spooler.3
-rw-------    1 root     root            0 Apr 20 04:02 spooler.4
-rw-r--r--    1 root     root            0 May 18 22:19 up2date
-rw-r--r--    1 root     root         2457 May 18 21:44 up2date.1
-rw-r--r--    1 root     root            0 May  4 04:02 up2date.2
-rw-r--r--    1 root     root         2275 May  1 06:59 up2date.3
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
-rw-r--r--    1 root     root          841 Apr 24 22:49 up2date.4
drwxr-xr-x    2 root     root         4096 Jul  1  2002 vbox
-rw-rw-r--    1 root     utmp        55680 May 23 18:00 wtmp
-rw-rw-r--    1 root     utmp        94848 Apr 30 20:20 wtmp.1
```

References

1.Balab IT, Balab IT Syslog-ng Web Site, http://www.balabit.hu/products/syslog_ng/
2.Bird, Tina and Ranum, Marcus, Loganalysis.org Web Site, http://loganalysis.org
3.Campi, Nathan, Syslog-ng FAQ, http://www.campin.net/syslog-ng/faq.html
4.Hines, Eric, "Complete Reference Guide to Creating a Secure Remote Syslog Server", Fatelabs, 2002-01-04, http://www.fatelabs.com/library/syslog.pdf
5.Hines, Eric, "Flying Pigs: Snorting Secure Remote Syslog-NG Servers", http://www.fatelabs.com/library/flyingpigs.pdf
6.Lonvick, C., BSD Syslog Protocol, IETF RFC 3164, August 2001, http://www.ietf.org/rfc/rfc3164.txt
7.logger(1) manual page, UNIX User's Manual, Berkeley, CA: USENIX Association, March 1984
8.Modular Syslog Web Site, http://www.corest.com/producs/corewisedom/CW01.php?
9.OpenSSH Project Web Site, http://www.openssh.com
10.Postal, J, User Datagram Protocol, http://www.faqs.org/rfcs/rfcs768.html
11.Reed, Dareen, Nsyslog Web Site, http://coombs.anu.edu.au/~avalon/nsyslog.html
12.Rose, Marshall T., BEEP, IETF RFC 3080, http://www.ieft.org/rfc/rfc3080.txt
13.Rose, Marshall T., BEEP, The Definitive Guide, Sebastipol, CA, O'Reilly and Associaties, March 2002.
14.Rose, Marshall,
15.Scheidler, Balazs, Syslog-ng, http://www.linuxgazette.com/issue43/scheidler.html, Linux Gazette Issue 34
16.SDSC Secure Syslog Web Site, http://security.sdsc.edu/software/sdsc-syslog/
17.Stunnel Project Web Site, http://www.stunnel.org
18.syslog(3) manual page, UNIX Programmer's Manual, Berkeley, CA: USENIX Association, March 1984
19.syslog(8) manual page, UNIX Systems Manager's Manual, Berkeley, CA: USENIX Association, March 1984

---

iHines, Eric, “Complete Reference Guide to Creating a Secure Remote Syslog Server” Fatelabs, 4 Jan 2002, http://www.fatelabs.com/library/syslog.pdf

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 21st, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced