

# Introduction to IPAudit

*Paul Asadoorian* 2005-07-11

IPAudit is a handy tool that will allow you to analyze all packets entering and leaving your network. It listens to a network device in promiscuous mode, just as an IDS sensor would, and provides details on hosts, ports, and protocols. It can be used to monitor bandwidth, connection pairs, detect compromises, discover botnets, and see whos scanning your network. When compared to similar tools, such as Cisco System's Netflow it has many advantages (see the SecurityFocus articles on Netflow, [part 1](#) and [part 2](#)). It is easier to setup than Netflow, and if you install it on your existing IDS sensors, there is no extra hardware to purchase. Since it captures traffic from a span port, it does not require that you modify the configuration of your networking equipment, or poke holes in firewalls for Netflow data.

Packet analysis tools like IPAudit help fill the gaps left by an IDS system or an IPS system. How does it do this? An IDS looks for certain signatures or behavior and can alert and log. An IPS looks for the same anomalies but can prevent the attack. Both of these technologies can greatly increase the security of your network -- however, what happens if they miss an attack? How would you know? Even if the IDS sensor matches a packet, a machine can still become compromised. When this happens how do you tell what happened on the network after the compromise? IPAudit can help fill the gaps, in addition to providing you with useful information about your network beyond specific security events. It is most often used by universities where its primary role is to identify who is using the most bandwidth. The author of this article finds it to be useful for all organizations; in fact, many corporate customers will also recognize the benefits and incorporate it into their security tool arsenal.

## Installation and configuration

IPAudit is a perl-based application written by John Rifkin at the University Of Connecticut. It can be downloaded from [Sourceforge](#) and is licensed under the GNU GPL. IPAudit is a command line tool that uses the libpcap library to listen to traffic and generate data. The IPAudit-Web package includes the IPAudit binary in addition to the web interface that creates reports based on the collected data. Using the Web package is recommended, as it gives you a slick graphical interface complete with traffic charts and a search feature.

You will need to have a Linux or Unix system setup with the libpcap library installed. The latest version can be downloaded from [tcpdump.org](#). In addition to libpcap, you will need Perl, Apache, GNUplot, and a perl module called "Time::ParseDate". Refer to your Linux distribution's documentation for more information on how to install these packages (here's a tip: In Debian Linux, execute the command 'apt-get install libtime-modules-perl' to install

Time::ParseDate). Once you have installed these packages you are ready to begin installing IPAudit:

**Step 1** - Become root on your system and create a user called "ipaudit". It will need a valid shell and home directory (typically /home/ipaudit, which will be used in this article for simplicity). Now switch to the newly created "ipaudit" user.

**Step 2** - Download and unpack the ipaudit-web tarball:

```
$ tar zxvf tar zxvf ipaudit-web-1.0BETA9.tar.gz
```

**Step 3** - Change to the compile directory:

```
$ cd ipaudit-web-1.0BETA9/compile
```

**Step 4** - Execute the configure script and run make:

```
$ ./configure  
$ make
```

**Step 5** - Become root and execute the make install commands:

```
$ su -  
Password:  
# make install  
# make install-cron  
# exit (Leave root and become ipaudit user again)  
$
```

**Step 6** - Now you will need to edit /home/ipaudit/ipaudit-web.conf

```
#  
LOCALRANGE=127.0.0  
#  
  
#  
INTERFACE=eth1  
#
```

Change the LOCALRANGE variable to your local subnet on the inside of your network. Also be certain to set the INTERFACE variable to the interface that you have setup to capture the desired traffic on your network.

**Step 7** - Add the following lines to your Apache httpd.conf file if they do not already exist:

```
<Directory /home/*/public_html>
AllowOverride All
Options MultiViews Indexes Includes FollowSymLinks
Order allow,deny
Allow from all
</Directory>

<Directory /home/*/public_html/cgi-bin>
Options +ExecCGI -Includes -Indexes
SetHandler cgi-script
</Directory>
```

Note that your Apache server may already contain configuration similar to the above for the "/home/\*/public\_html" directory. If you do not plan to use the Userdir module for anything other than IPAudit, it is suggested that you comment out the original configuration and replacing it with the configuration above.

Your Apache server will need to support SUEXEC, Mod\_Perl, and Mod\_Userdir. Once you have modified the Apache configuration restart your Apache server. For more details on the IPAudit-Web installation, refer to the INSTALL file located in the installation directory of that package. It contains more information about the required Perl module Time::ParseDate, SUEXEC, and password protecting your IPADUIT-Web installation. Since it requires just moderate Google hacking skills to find other peoples IPAudit installations, protecting IPAudit with a password would be a very good idea.

**Step 8** - Check your installation

Open a web browser and go to:

<http://<your web server>/~ipaudit/>

If your installation was successful you should now see a screen like the one shown below in Figure 1.

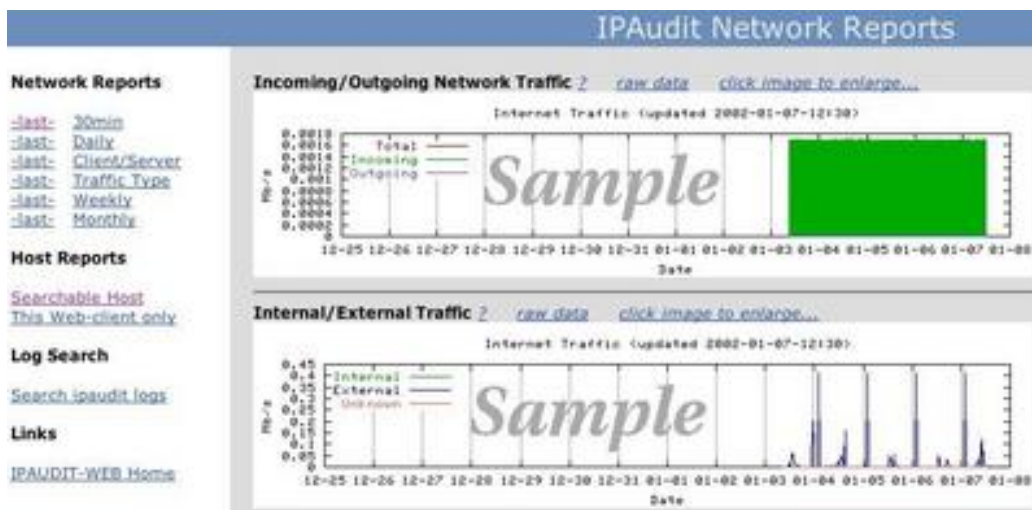


Figure 1. Running IPAudit's web interface for the first time.

You should make certain that the time on the server running IPAudit is correct and being kept up to date using NTP. Without accurate time, IPAudit will get confused if the time on the packets differs greatly from that of the system time.

After the first half hour mark, IPAudit will begin to graph all of your traffic and generate some reports. The screen should then look similar to the one in Figure 2.

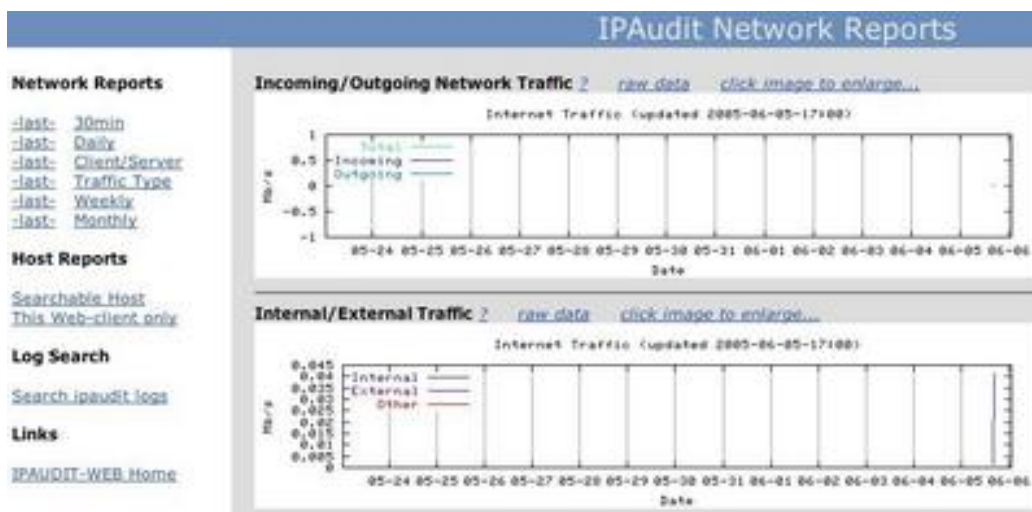


Figure 2. First graph appears after 30 minutes.

The graphs will get more interesting as time goes on and as IPAudit sees more traffic. A "spike" in the graph typically denotes an indication of a problem, such as a host sending out a DoS (Denial of Service) attack.

## General reporting

IPAudit's "Network Reports" are useful for many reasons. The thirty-minute and daily reports are exactly the same, except of course for the timeframe. By clicking on the link labeled "-last-" next to the "30min" link you will see the report for the last 30 minutes. At the top of the screen you can see general network statistics, which is good if you are trying to keep tabs on your total bandwidth utilization. This is followed by the busiest local hosts

report, which is good way to keep an eye on who is transferring the most data into or out of your organization, as shown in Figure 3.

Busiest Local Hosts				
IP	Host Name	Incoming (bytes)	Outgoing (bytes)	Total (bytes)
<a href="#">192.168.001.237</a>	-	75,353,878	67,232,083	142,585,961
<a href="#">192.168.001.045</a>	bud.mycompany.com	9,093,694	1,102,415	10,196,109
<a href="#">192.168.001.211</a>	-	7,269,574	1,275,000	8,544,574
<a href="#">192.168.001.022</a>	hanzo.mycompany.com	2,835,412	84,678	2,920,090
<a href="#">192.168.001.223</a>	-	711,662	73,614	785,276
<a href="#">192.168.001.015</a>	ns.mycompany.com	424,349	201,070	625,419
<a href="#">192.168.001.010</a>	ns2.mycompany.com	168,262	93,613	261,875
<a href="#">192.168.001.232</a>	-	176,063	46,217	222,280
<a href="#">192.168.001.238</a>	-	100,123	42,951	143,074

Elapsed time is 8 seconds.

Figure 3. Displaying the busiest hosts on the local network.

Your servers, such as SMTP mail servers, will typically be close to the top of this list (in addition to your P2P hosts, if you allow that application). Over time you will develop a baseline of your busiest hosts. When checked everyday, you may notice a new host occupying the top busiest host and this would be cause for you to investigate.

The busiest remote hosts report tells you who on the Internet you are transferring the most data to and from.

Busiest Remote Hosts				
IP	Host Name	Incoming (bytes)	Outgoing (bytes)	Total (bytes)
<a href="#">207.025.253.040</a>	dispsd-40-www3.boulder.ibm.com	14,019,752	289,638	14,309,390
<a href="#">069.170.243.096</a>	69-170-243-96.atl5fl.adelphia.net	5,566,213	6,770,587	12,336,800
<a href="#">068.015.034.115</a>	entropy.tmok.com	8,100,317	329,706	8,430,023
<a href="#">024.080.174.189</a>	S0106001217040deb.vc.shawcable.net	1,861,523	5,086,032	6,947,555
<a href="#">204.152.191.007</a>	mirrors1.kernel.org	6,587,080	170,039	6,757,119
<a href="#">062.058.237.015</a>	dslam15-237-58-62.adsl.versatel.nl	6,372,971	193,395	6,566,366
<a href="#">024.068.097.037</a>	S0106000625633845.gv.shawcable.net	6,144,556	208,611	6,353,167
<a href="#">070.028.234.144</a>	CPE000ea640812c-CM0012c9db37da.cpe.net.cable.rogers.com	3,434,800	2,782,488	6,217,288

Figure 4. Displaying the busiest remote hosts.

Typically these tend to be Akamai caching servers, Windows Update IP addresses, and other popular web sites like Google or Yahoo. If one of the sites listed resolves to something unfamiliar like [www.evil.com](#), it should be cause for alarm.

The next report is the, "Possible Incoming Scan Hosts," which shows the IP addresses of the hosts that connected, or tried to connect, to the most unique IP addresses on the local subnet. This report is useful to see who is scanning your network, and what ports they are scanning for.

Possible Incoming Scan Hosts		
IP	Host Name	Local Hosts Contacted
<a href="#">218.066.104.131</a>	-	77,605
<a href="#">061.129.051.046</a>	-	73,451
<a href="#">061.134.049.034</a>	-	60,119
<a href="#">211.169.240.201</a>	-	40,999
<a href="#">142.179.217.049</a>	s142-179-217-49.ab.hsia.telus.net	40,786
<a href="#">210.222.009.101</a>	-	40,726
<a href="#">203.146.102.212</a>	-	40,713

Figure 5. Top remote IP addresses scanning your network.

It is good to check this table everyday when monitoring a network. It is useful to take the most common ports that the network is being scanned for and research them. The following web sites are useful when determining what applications correspond to the ports attackers are scanning for:

- [SANS](#) - From the main page you will find a port search function. This reads from the [Dshield database](#). This page also offers the most up-to-date information on port-scanning trends and general blackhat activity.
- [The official port assignments database](#).
- [Google](#) - When in doubt, use Google to find information about ports, for example "tcp port 6881" to check for known Trojans that frequently use a given port.

Port-scanning activity is sometimes due to a new network scanning tool being released (like scannssh), or a new virus or worm that is being circulated. Having this information, it is good to warn the user population if the threat warrants that level of notification. An administrator can then target his notifications towards specific groups. For example, if the network is being scanned for MySQL instances, you should notify the server group and tell them to make certain they have applied all relevant patches and to not expose their servers to the Internet if it can possibly be helped. Oftentimes, you can correlate vulnerability or exploit releases with the portscanning attacks on your network. While you probably have a firewall that blocks these attempts, what if the firewall becomes mis-configured because of a firewall policy change? These reports allow you to react to threats against your network in an informed manner, adding another layer to your network security infrastructure.

Possible outgoing scan hosts are listed next. While the possible incoming scan hosts can be used for proactive measures, the following outgoing scan hosts report is more useful for reactive measures.

Possible Outgoing Scan Hosts		
IP	Host Name	Remote Hosts Contacted
<a href="#">192.168.001.045</a>	bud.mycompany.com	6,634
<a href="#">192.168.001.015</a>	ns.mycompany.com	19
<a href="#">192.168.001.010</a>	ns2.mycompany.com	14
<a href="#">192.168.001.237</a>	-	12
<a href="#">192.168.001.223</a>	-	2
<a href="#">192.168.001.211</a>	-	2
<i>Elapsed time is 8 seconds.</i>		

Figure 6. Outgoing scan hosts are useful for discovering Trojaned machines.

If I find hosts that are on the inside of the network scanning outwards, it is usually an indication that a machine has become compromised with a worm or virus, and in some cases an actual attacker has taken control of the host and is using to scan for other machines. When you begin to check the reports on a regular basis you will be able to develop a baseline and know what is normal on your network with regards to the number of hosts contacted in a given day. Some hosts need to contact numerous other unique hosts, such as SMTP relay and DNS servers. However, a typical user's workstation usually does not normally contact upwards of 1,000 different hosts on the Internet.

The port that your local hosts are scanning for is significant as well. A machine scanning out to the Internet on port 445 (Windows CIFS) or 6667 (IRC) should raise a red flag and cause you to investigate it as if it were compromised. Port 445, SMB CIFS, is a common port being scanned for on the Internet due to the number of vulnerabilities associated with it. IRC is typically used as a communications mechanism for compromised machines, more commonly known as botnets. However, a machine scanning out on ports 6881 (BitTorrent) or 6346 (GNUTella) would be an indication that the host is running a P2P networking application, which commonly scans the Internet looking for other P2P enabled hosts. The policy within your organization should dictate if this is acceptable behavior or not.

The busiest host pairs table is the final report. It lists which hosts had the largest single transfers between them. It's a good idea to take a look at this list and make certain that the transfers seem normal or not. Normal behavior would be someone downloading a Linux ISO image, whereas less normal behavior could be someone downloading pirated media from an already compromised host.

Going back to the main IPAudit page, you will notice even more reports that you can run. The client/server report can be useful for monitoring who is running the following services on your network:

- HTTP Servers
- Mail Servers
- SSH Servers
- Telnet Servers
- HTTPS Servers

I typically check these reports on a weekly basis to get an idea of who is running what

server services on the network. A red flag could be a user workstation that ends up in the top ten SMTP servers listing. This could indicate that the host has been infected and is being used to distribute SPAM. The listing of HTTP servers is useful to see not only who may be running legitimate web servers on your network, but it can also be an indication of anyone tunneling other protocols with HTTP and running it over port 80 or 443 TCP. Since IPAudit only looks at IP and transport layer information, it will not distinguish between actual HTTP traffic and tunneled traffic (which can actually be good in this case).

The traffic type, weekly, and monthly reports all contain summary information about your network. They should be checked weekly to get an overview of what networking protocols are in use, and which hosts transmit and receive the most data. Host reports contain much of the same information as the weekly and monthly reports, except on a per host basis.

The log searching feature is an excellent way to find certain traffic types using multiple criteria, as shown below in Figure 7.

Figure 7. Searching IPAudit's logs.

You can adjust your query to a specific time period, right down to the minute. The IP address can either be a host on the local network or the external network/Internet. The local port is relative to the local address space you specified in the IPAudit-Web configuration file, as is the remote port. The next two fields, Max Lines Displayed and Print Increment tell IPAudit how to print out the query. It is best to start with a low number for the line displayed the first time you run a query, just in case there are thousands of results which could take some time. The session size is a particularly useful field when trying to determine traffic type. Sometimes you want to distinguish between actual data transfers and just portscanning. By playing around with the values in these fields you can do just that (for example, suppose you want to know who actually connected to the MySQL server, not who scanned it). The protocol drop down menu allows you to choose between TCP, UDP, and ICMP. IPAudit tries to keep track of state by indicating whom the first talker was in the connection.

Overall, IPAudit has many useful features and many ways in which to look at your network traffic. The next section will go into more detail on how to use it to detect compromise

machines on your network.

## Detecting compromised hosts

Similar to an IDS, IPAudit is a historical account of your network traffic. If an exploit comes flying into your network and is picked up by your IDS, it happily logs it. When you go to check the logs you can see this event, including the full packet, and you may say, "Yup, that was an exploit alright, I wonder if it was successful?" IPAudit works in much the same way, except you can use it to detect all behavior exhibited by the potential compromised host after the exploit was launched. Here is an example:

```
snort: [1:2351:10] NETBIOS DCERPC ISystemActivator path overflow attempt
little endian unicode [Classification: Attempted Administrator Privilege
Gain] [Priority: 1]: {TCP} 192.168.1.237:4014 -> 192.168.1.223:135
```

```
snort: [1:2123:3] ATTACK-RESPONSES Microsoft cmd.exe banner
[Classification: Successful Administrator Privilege Gain] [Priority: 1]:
{TCP} 192.168.1.223:31337 -> 192.168.1.45:32768
```

The above Snort alerts indicate that 192.168.1.237 is trying to exploit 192.168.1.223 using a very common exploit that takes advantage of the MS03-026 RPC vulnerability (See the full [Snort rule documentation](#)). We then see a very obvious backdoor attempt, most likely a simple Netcat command such as "nc.exe -l -p31337 -e cmd.exe".

Using IPAudit, let's examine the victim host's traffic. I would first go to the IPAudit searchable host feature, enter the timeframe I want to look at, then the IP address. It produces a report as shown in Figure 8.

Local IP	Remote IP	Proto- col	Local Port	Remote Port	Incoming Bytes	Outgoing Bytes	Incoming Packets	Outgoing Packets	First Packet Time	Last Packet Time	First Talker	Last Talker
192.168.1.223	10.1.28.217	tcp	39402	445	0	54	0	1	14:30:01.0078	14:30:01.0078	L	-
192.168.1.223	10.1.28.212	tcp	39402	445	0	54	0	1	14:30:01.0080	14:30:01.0080	L	-
192.168.1.223	10.1.28.192	tcp	39402	445	0	54	0	1	14:30:01.0081	14:30:01.0081	L	-
192.168.1.223	10.1.30.99	tcp	39402	445	0	54	0	1	14:30:01.0081	14:30:01.0081	L	-
192.168.1.223	10.1.22.134	tcp	39402	445	0	54	0	1	14:30:01.0082	14:30:01.0082	L	-
192.168.1.223	10.1.25.138	tcp	39402	445	0	54	0	1	14:30:01.0083	14:30:01.0083	L	-
192.168.1.223	10.1.30.234	tcp	39402	445	0	54	0	1	14:30:01.0083	14:30:01.0083	L	-
192.168.1.223	10.1.17.189	tcp	39402	445	0	54	0	1	14:30:01.0084	14:30:01.0084	L	-
192.168.1.223	10.1.20.46	tcp	39402	445	0	54	0	1	14:30:01.0085	14:30:01.0085	L	-
192.168.1.223	10.1.22.17	tcp	39402	445	0	54	0	1	14:30:01.0085	14:30:01.0085	L	-

Figure 8. Search results for a certain timeframe and IP address.

The above data indicated that the host is portscanning for port 445. First, we see that the same source port is used to connect to multiple different destination hosts. In normal TCP communications, a different source port would be used when connecting to a different host. Second, we see many attempts to port 445 on a class B network, with little data being transferred. Also, if we look at the column labeled "First Talker" it indicates that the host on the local network initiated the connection. The "Last Talker" column is blank, telling us that 192.168.1.223 sent out the packets, but received no responses. These are all telltale signs of portscanning.

What if you want to see what happened in addition to the portscanning? If someone did in fact compromise this host then they most likely uploaded some sort of rootkit or IRC bot. Let's take the IP address of the machine that opened the backdoor on our victim host and see what other machines it connected to that day, as shown in Figure 9.

192.168.1.223	192.168.1.45	tcp	4000	9828	14.4k	21.0k	258	82	16:54:19.5093	16:59:43.1856	L	R
192.168.1.223	192.168.1.45	tcp	4000	9828	314.6k	1.83M	8953	5181	17:00:41.5510	17:30:00.2029	R	R
192.168.1.223	192.168.1.45	tcp	4037	8358	4.3k	1.5k	48	24	17:00:41.7394	17:00:43.1053	L	L
192.168.1.223	192.168.1.45	tcp	4038	8357	634	170	10	3	17:00:42.2716	17:00:42.2973	R	R
192.168.1.223	192.168.1.45	tcp	4039	8357	1.2k	170	10	3	17:00:42.4887	17:00:42.4965	R	R
192.168.1.223	192.168.1.45	tcp	4040	8357	144.9k	1.5k	112	29	17:00:42.6727	17:00:42.9222	R	R
192.168.1.223	192.168.1.45	tcp	4041	8357	582	170	10	3	17:00:42.9455	17:00:42.9715	R	R

Figure 9. Search showing potentially compromised IP connecting to other machines.

Here we see it connecting to our known victim host and transferring data on port 4000, among others.

After further analysis we see a similar transfer to another host on our network, 192.168.111.69, as shown in Figure 10.

192.168.1.223	192.168.1.45	tcp	4000	9828	14.4k	21.0k	258	82	16:54:19.5093	16:59:43.1856	L	R
192.168.1.223	192.168.1.45	tcp	4000	9828	314.6k	1.83M	8953	5181	17:00:41.5510	17:30:00.2029	R	R
192.168.1.223	192.168.1.45	tcp	4037	8358	4.3k	1.5k	48	24	17:00:41.7394	17:00:43.1053	L	L
192.168.1.223	192.168.1.45	tcp	4038	8357	634	170	10	3	17:00:42.2716	17:00:42.2973	R	R
192.168.1.223	192.168.1.45	tcp	4039	8357	1.2k	170	10	3	17:00:42.4887	17:00:42.4965	R	R
192.168.1.223	192.168.1.45	tcp	4040	8357	144.9k	1.5k	112	29	17:00:42.6727	17:00:42.9222	R	R
192.168.1.223	192.168.1.45	tcp	4041	8357	582	170	10	3	17:00:42.9455	17:00:42.9715	R	R

Figure 10. Similar traffic shown with another machine.

The backdoor port is different, but the host is in fact compromised in the same way as 192.168.1.223. This can be verified in the IDS logs:

```
snort: [1:2123:3] ATTACK-RESPONSES Microsoft cmd.exe banner
[Classification: Successful Administrator Privilege Gain] [Priority: 1]:
{TCP} 192.168.111.69:2143 -> 192.168.1.45:32768
```

Using IPAudit we can then continue to map the scope of the compromise. This includes all machines that have become compromised, which servers attacked them, which servers are controlling them via backdoors, and which IRC servers they logged into. We do this by modifying our search criteria to map connections between all hosts involved.

The incident described above was based on a real incident, but was also recreated in a lab. The real incident involved a dozen compromised computers, two IRC servers, an attacking host, and a remote shell host. It was all mapped using IPAudit and correlated with Snort.

## Conclusion

IPAudit is a great addition to your network monitoring. It provides reports that give you an overview of your network, inform you of security events, and report on anomalies. When used in conjunction with intrusion detection a security incident can be mapped out in a great deal of detail. Best of all, IPAudit is a free tool that is easy to setup and maintain. You will find it useful to install on all your IDS sensor installations.

## About the author

*Paul Asadoorian, GCIA, GCIH is the lead security engineer for a large university in the New England area where he designs, implements, and maintains intrusion detection systems, firewalls, and VPNs. He gives regular presentations in the academic community relating to network security. Paul is also the founder of [Defensive Intuition](#), a security company specializing in security auditing, penetration testing, and other security related services.*

[Privacy Statement](#)

Copyright 2006, SecurityFocus