

## MRTG for Intrusion Detection with IIS 6

Mark Burnett 2003-08-18

The Multi Router Traffic Grapher (MRTG) is a simple cross-platform tool that administrators have used for years to monitor network traffic loads. The concept is simple: it queries SNMP counters and creates HTML pages with live network graphs showing bytes coming in and bytes going out. MRTG can show much more than in and out traffic, it can graph any SNMP counter. Microsoft has a web site that [demonstrates](#) some of the many SNMP counters available on a Windows 2003 server.

But MRTG is also a very effective intrusion detection tool. The concept is simple: attacks often produce some kind of anomalous pattern and human brains are well-equipped to spot anomalous patterns, given some way to visualize those patterns. The MRTG does just that -- it gives you the big picture of your network traffic and it also slices it into different views, allowing you to see any counter trends for the last week, month, or year.

To use MRTG as an IDS we need to first determine which counters are effective attack indicators. The following table contains some example attacks along with the anomalies they produce:

| Attack   | Counter Anomalies  |
|--|--|
| An attacker uses a CGI scanner against a web site to potential find vulnerable CGI scripts.    | A sharp increase in HTTP 404 Not Found errors; an increase in HTTP requests per second.          |
| An attacker tries to brute-force accounts on a password-protected web site.                    | A sharp increase in HTTP 401 Authorization Required errors.                                      |
| A new worm appears on the internet.  | An increase in network traffic targeting a specific protocol.                                    |
| A new worm infects your server and begins to attack other servers.                             | An increase in outgoing network traffic targeting a specific protocol; an increase in CPU usage. |
| An intruder finds an anonymous writeable FTP and creates a warez download site on your server. | An increase in bytes of outgoing FTP or HTTP traffic.  |
| An attacker tries to exploit an SQL injection vulnerability in your web application.           | An increase in HTTP 500 Server Errors.   |

|   |  |
|---|--|
| A spammer finds your SMTP server allows relaying and uses it to spam a million e-mail addresses.                | A huge increase in outgoing SMTP traffic; an increase in outgoing DNS lookups; an increase in CPU usage.   |
| An intruder exploits a buffer overflow and installs various tools to increase control of the server or network. | An increase in processes running on the server; a small increase in CPU and memory usage; a small decrease in available disk space.  |
| An attacker tries to take down your web site with a DDoS attack.  | An increase in ICMP traffic; an increase in various IP errors; an increase in TCP connections; an increase in multicast traffic; an increase in general traffic coupled with a decrease in actual web hits; an increase in TCP packets without much increase in actual bandwidth used. |
| An attacker is just trying to break in using whatever means available.  | An increase in IDS alerts.   |

Looking through this list you will begin to see how certain counters emerge as very effective attack indicators. Hackers need server resources: CPU, RAM, disk space, network connections, and bandwidth. Hackers create processes, open network ports, create log entries, and generate errors, all of which you can monitor. A hacker's only viable offence is to spread out an attack so that it does not produce counters significantly above your network average.

Here are some techniques hackers use to avoid detection:

- Increasing the time between each probe and between each actual attack.
- Throttling bandwidth usage.
- Avoiding significant increases in CPU, RAM, or hard drive usage.
- Attacking from multiple locations to avoid too much traffic from a single host.
- Attacking when counters are high such as during peak traffic hours or during an internet-wide worm attack (note that some hackers try to conceal their attacks by creating their own floods of traffic, but this is only effective in helping to hide their origin; the victim is still very aware that an attack is transpiring).
- Attacking during weekends or holidays when no one is around to notice the increased resource usage.

Despite these techniques, it is extremely difficult even for a skilled attacker to avoid detection from all resource counters. Plus, there are plenty of hackers who are not even skilled enough to avoid detection

through the most obvious counters.

For an IIS 6 web server, it is clear what we need to monitor:

- Network traffic, including bandwidth, number of packets, and number of connections.
- Network protocol errors
- Web traffic, including number of users, number of bytes, number of requests, and number of errors.
- CPU, RAM, and disk usage.
- Processes and threads

## Installing MRTG on Windows 2003

Before using MRTG, you need to install SNMP on your server. From the Control Panel, select Add or Remove Programs and click on Add/Remove Windows Components. Highlight Management and Monitoring Tools and click on the Details button. From there, check the box before Simple Network Management Protocol. Click on OK then click on Next to proceed with installation.

After installing SNMP, you need to take a few steps to secure it. SNMP is by no means a secure protocol and should not be used over a network without some kind of encryption. Microsoft has an [article](#) that explains how to use SNMP with IPsec, but for the purposes of this article, we will only use SNMP locally. Just be sure to block UDP ports 161 and 162 at your firewall or using IPsec. Next, you need to set an obscure community string. From the Administrative Tools, select Services and double-click on the SNMP Service. From the Security tab, add a READ ONLY community name, which is roughly equivalent to a password. Although a community string is far from secure, you should still avoid using a common community string such as PUBLIC. Check the box to only Accept SNMP packets from these hosts and make sure that only localhost is in the list.

MRTG is a Perl script and a compiled C program. You will need to download and install [ActivePerl](#). You will also need to download the [most recent](#) version of MRTG (hint: grab the most recent .zip file). The MRTG files that were used for this article are [available here](#). Extract MRTG from the zip file to C:\Program Files\MRTG.

Create a MRTG directory under your Inetpub directory (but not in the wwwroot directory). Use the Internet Information Services Manager to create a new MRTG web site. If possible, use host headers or a unique IP address for the MRTG site, but if neither of these are an option, create a protected virtual directory under another exiting web site.

Set the MRTG site to not run scripts or executables and provide only Read access. For NTFS permissions, only allow access to those users who will need to monitor MRTG. If possible, also set IP restrictions for this web site to only allow selected network hosts.

Now, take [these files config.] and place them in your C:\Program Files\MRTG\Bin directory. Finally, copy the index.html file to your \Inetpub\MRTG directory.

Now, to test your configuration, type the following:

```
C:\ProgramFiles\MRTG>perl mrtg mrtg.cfg
```

If everything installed correctly, you should now have some files in your MRTG web directory. If not, retrace your steps or search the MRTG documentation for more information. You may also want to read the [tutorial](#) if you are having problems installing MRTG.

## Using Counters

In my [example config](#) file I used the following counters:

| Counter           | Source               |
|-------------------|----------------------|
| Bytes in and out  | SNMP                 |
| CPU Usage         | SNMP                 |
| Memory Usage      | SNMP                 |
| Disk Usage        | SNMP                 |
| ICMP Messages     | SNMP                 |
| TCP Connections   | SNMP                 |
| Processes/Threads | WMI/VBScript         |
| HTTP Connections  | LogParser/Batch File |
| HTTP Errors       | LogParser/Batch File |

Figure 1, below, shows an example of how the counters appear.

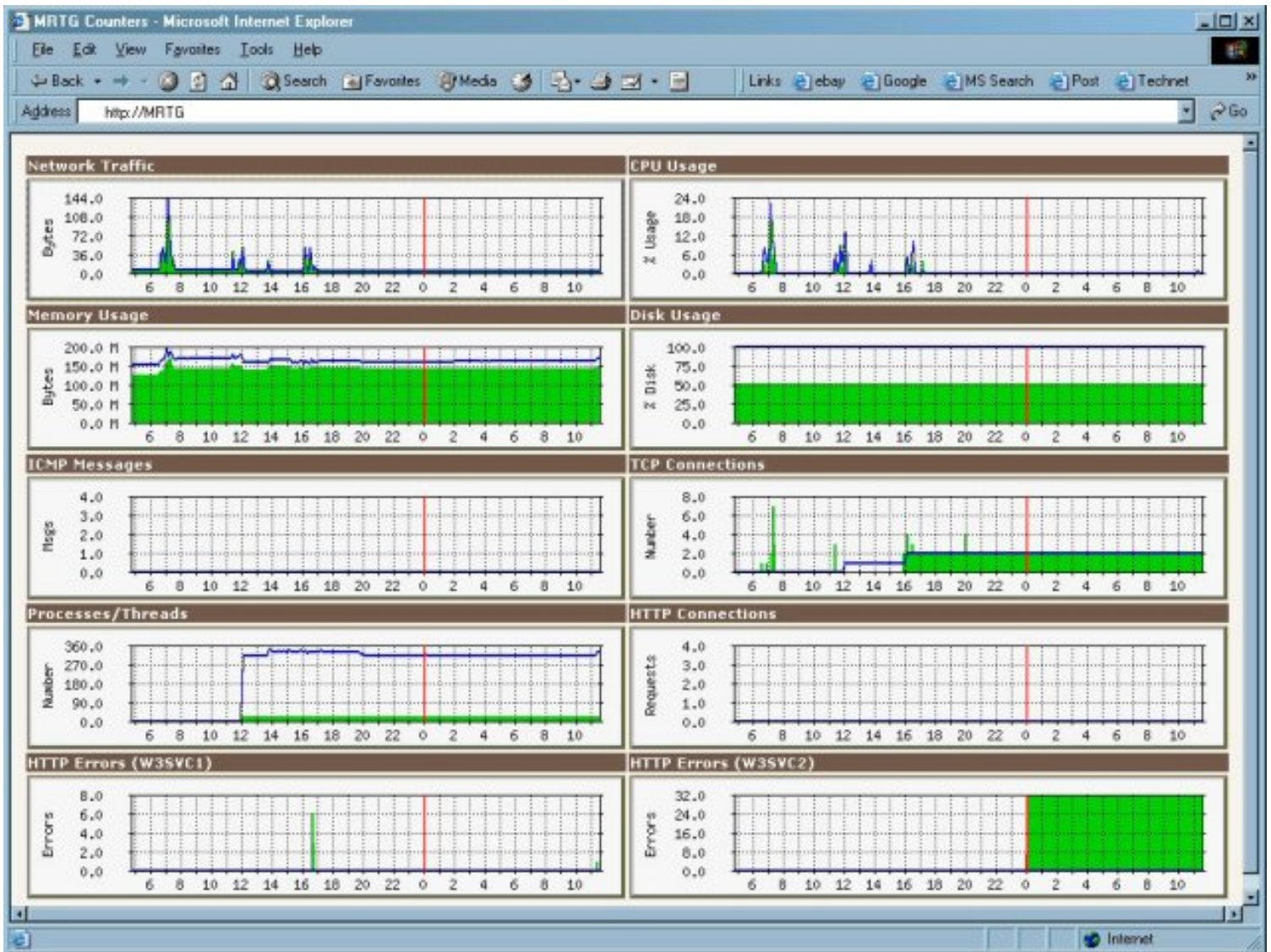


Figure 1 - MRTG screenshot

Note that although Microsoft provides a number of SNMP counters, I found they did not always work consistently or as documented. Nevertheless, MRTG allows you to pull counters from external applications, opening up a huge range of possible counters. The most obvious of these is using Windows Management Instrumentation (WMI) to pull from a vast array of information, including all performance counters. Unlike SNMP, Microsoft has put a great effort into supporting and documenting WMI. For example, to get process and thread information, I used the following script which could easily be modified to pull any information available through WMI:

```
Set oWService=GetObject("winmgmts:\\localhost\root\cimv2")
Set colItems=oWService.ExecQuery("SELECT * FROM Win32_PerfFormattedData_PerfOS_System",,48)
```

```
For Each Item in colItems
    Param1=Param1 + Item.Processes
    Param2=Param2 + Item.Threads
    Uptime=Item.SystemUptime
```

Next

```
WScript.Echo Param1
WScript.Echo Param2
WScript.Echo Uptime & " seconds"
WScript.Echo "LocalHost"
```

Another problem I had was getting detailed or custom web statistics through either SNMP or WMI. To solve that, I used Microsoft's LogParser tool to run custom queries from a simple batch file:

```
@for /f "tokens=1,2,3,4* delims=/ " %%i in ('date /t') do @set year=%%l&& @set month=%%j&&
@set day=%%k
@set logfile=c:\windows\system32\LogFiles\%1\ex%YEAR:~2,2%%month%%day%.log
@if exist %logfile% (
    @logparser "SELECT COUNT(*) FROM %logfile% WHERE (sc-status>= 400AND sc-status<500)
    AND TO_TIMESTAMP(date, time) > SUB(SYSTEM_TIMESTAMP(), TO_TIMESTAMP('5','m'))" -q
    @logparser "SELECT COUNT(*) FROM %logfile% WHERE (sc-status>= 500AND sc-status<600)
    AND TO_TIMESTAMP(date, time) > SUB(SYSTEM_TIMESTAMP(), TO_TIMESTAMP('5','m'))" -q
) ELSE (
    @Echo %logfile%
    @Echo 0
)
@Echo Unknown
@Echo %1
```

The batch file parses the output of the Date command to determine the most recent log file and then runs a query to pull the stats for the last five minutes. This could be further customized to pull statistics for a specific page or for hits coming from specific ranges of IP addresses. For example, you may want to count hits to your main page or even track worm attacks.

Because Microsoft's LogParser tool is so powerful, it can be a great source for MRTG counters. For example, you could write a query to count specific EventLog events such as failed logins. You could also track other files, such as the number of entries in a Snort log file or in the URLScan.log file.

As you build custom counters and scripts, keep in mind that you want to get the big picture and use those counters that best indicate an attack. Most attacks use resources so watching for anomalous resource usage can quickly lead to catching hackers.

## Relevant Links

<http://www.mrtg.org>

<http://www.wtcs.org/snmp4tpc/>

<http://snmpboy.msft.net/>

## About the Author

[Mark Burnett](#) is an independent security consultant and author who specializes in securing Windows-based servers. He is co-author of the best-selling book *Stealing the Network* (Syngress Publishing, ISBN: 1-931836-87-6). He has also co-authored or contributed to several other books, including *Special OPS: Host and Network Security for Microsoft, UNIX, and Oracle* (Syngress Publishing, ISBN: 1-931836-69-8); *Maximum Windows Security* (SAMS Publishing, ISBN: 0-672-31965-9); and *Dr. Tom Shinder's ISA Server and Beyond* (Syngress Publishing, ISBN: 1-931836-66-3). Mark is a regular contributor to many security-related magazines, newsletters, and web sites.

## More SecurityFocus Articles

View [more articles by Mark Burnett](#) on SecurityFocus.

[Privacy Statement](#)

Copyright 2006, SecurityFocus