Global Information Assurance Certification Paper

Virtually Free Network Security Software
*For the *nix disinclined*

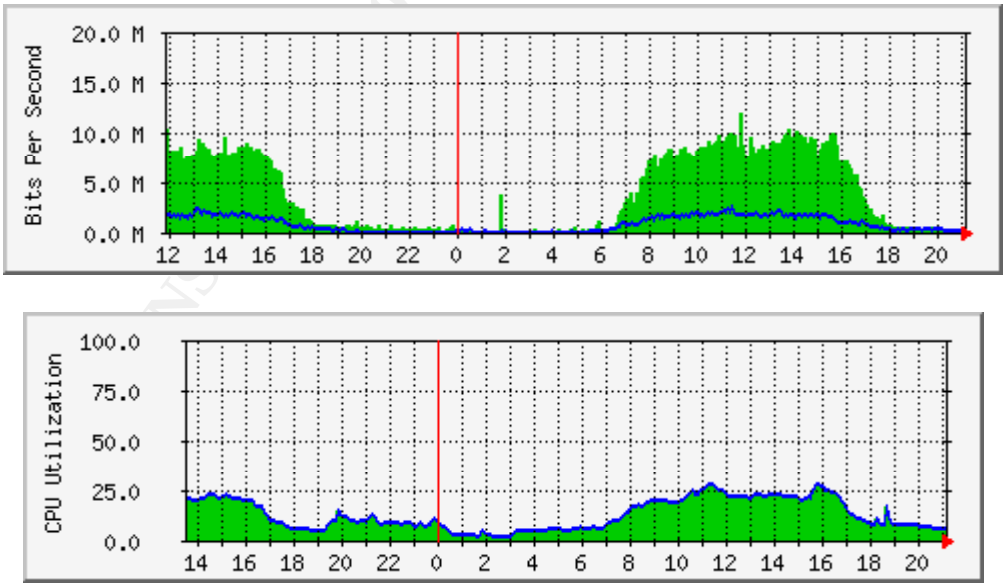
Dennis W. McHugh
Version 1.2e

I. DETECTION TOOLS

In my very brief but exciting history of detecting and defending against network attacks, as well as analyzing networks for irregularities, I have amassed a toolkit consisting of some free software. Some of the tools that have become a part of my toolkit have provided me with the ability to detect or verify different attacks and vulnerabilities as well as giving me information necessary to report the attacks to the proper authorities.

MRTG

The most useful product I have taken advantage of is MRTG. Originally, I was tasked to setup MRTG to look at IfInOctets and IfOutOctets (inbound and outbound traffic) per interface of our backbone links. By itself, the data is meaningful; however, it becomes even more useful from a security perspective when combined with CPU utilization. Rather then one graph of inbound and outbound traffic, now I had two graphs in 5-minute increments that could alert me to abnormal traffic patterns or traffic patterns that cause unusually high CPU utilization.



In a normal MRTG graph, you are presented with two values: IfInOctets and IfOutOctets. These are collected per interface and the MRTG program requests them via SNMP (Simple Network Management Protocol) and graphs the results as above.

These are bit values of the traffic traversing the interface in each direction. These two values can help security administrators discover possible attacks entering and exiting their network. For instance, if your normal traffic pattern is an inbound 4mb per second and an outbound 500k per second, any significant increase to this pattern would lead you to conclude that some node(s) either inside or outside is(are) continually requesting or sending more data. This is especially useful when one rises or falls while the other doesn't change. This could be all you need to alert you to a condition.

You may be saying "so what! We've done that, but I can't see a pattern that small since the interface is Fast Ethernet and the graph is set properly at 100mbps." To help you see the patterns, you can add maximal values to your graphs to see how far beyond the five-minute level the respective utilizations peaked by adding the WithPeak value to the interface info in the MRTG config file. Additionally, you don't have to set the graph at what the interface actually is capable of. For instance, if you have a T-1 link that operates normally below 400k per second, you can reset the horizon to 600k per second by changing the MaxBytes value in the MRTG config file. When you do this, the five-minute changes are more obvious. If you have a Fast Ethernet link to your firewall, but you aren't loading more than 3mb per second across the link, the resulting pattern will be more difficult to decipher unless you lower the horizon to around 6mb per second. As long as the peaks aren't beyond the horizon, the graph will come out OK. You can always move the horizon up or down, depending upon the average bandwidth utilization. In our enterprise we have several 100mb links, which are set at 10mb per second to monitor the average daily peak of 8mb per second.

When you combine the flow with the CPU utilization, you have a virtually free warning system. Albeit, not an early warning system, but maybe earlier than the routers crashing! For instance, denial-of-service (DOS) TCP synchronize (SYN) attacks directed through a router will raise ingoing and outgoing utilization through the path as well as greatly increase router CPU utilization, as many small SYN packets can be sent at a tremendous rate through the router. Cisco System's white paper, "Defining Strategies to Protect Against TCP SYN Denial of Service Attacks" states:

> The TCP SYN attack exploits this design by having an attacking source host generate TCP SYN packets with random source addresses toward a victim host. The victim destination host sends a SYN ACK back to the random source address and adds an entry to the connection queue. Since the SYN ACK is destined for an incorrect or non-existent host, the last part of the "three-way handshake" is never completed and the entry remains in the connection queue until a timer expires, typically for about one minute. By generating phony TCP SYN packets from random IP addresses at a rapid rate, it is possible to fill up the connection queue and deny TCP services, such as e-mail, file transfer, or WWW to legitimate users. (Cisco Systems, The TCP SYN Attack, 4.html)

In this scenario, I've graphed the switchport that the server is attached to. As the attacked device runs out of resources, the inbound graph stays high, but the outbound lowers back

to normal or even below normal. That's when the calls start coming in.
You can run MRTG graphs for all of your servers in a switched environment, one graph per port, which will reveal the attacker's target. This may not be the best form of intrusion detection, but it is enough to know that something abnormal is going on and what the target is. You take that pattern of normality that you hopefully see every day and compare. Collecting the patterns is the job of the system administrator using tools like MRTG. CERT Security Improvement Module 7 states:

> Data about network activities (traffic, performance, etc.) can be collected from a variety of sources such as
>
> - administrator probes (ICMP pings, port probes, **SNMP** queries)
> - log files (routers, firewalls, other network hosts and devices)
> - alert reports
> - error reports
> - network performance statistics reports
> - the outputs of tools used to support in-depth analysis
>
> (Cert.org, Security Improvement Practices, P094.html#2)

As an example, our enterprise network uses Cisco's TCP intercept to intercept these half-open conversations between the sender and receiver in order to control these unwanted resource hogs. In my observations, when TCP Intercept is active, CPU utilization can double while the incoming and outgoing graphed values rise only slightly. TCP Intercept does not allow any SYNs to connect directly to a target device. It proxies for the destination host by creating a SYN to the destination host, SYN ACKs back to the sender and waits for the final ACK. If it doesn't receive one within its timeout, it drops the conversation. If it receives the ACK within the timeout, it merges the two connections into one. The unusually high CPU utilization without corresponding interface data on our MRTG graph is an indicator to us that intercept is active (aggressive mode), and therefore we may be under attack. We can verify at the router console.

In another instance, imagine a 60% increase in inbound and outbound traffic across your Internet connection. Router CPU utilization is higher, but normal in conjunction with the increase of traffic. This could be an FTP bounce attack. You are being used to send or retrieve data from another location.

Again, a local Ethernet segment is running at maximum capacity inbound, degrading the outbound and slowing connections. How about a broadcast storm, or malfunctioning Ethernet device, or broadcasting worm?

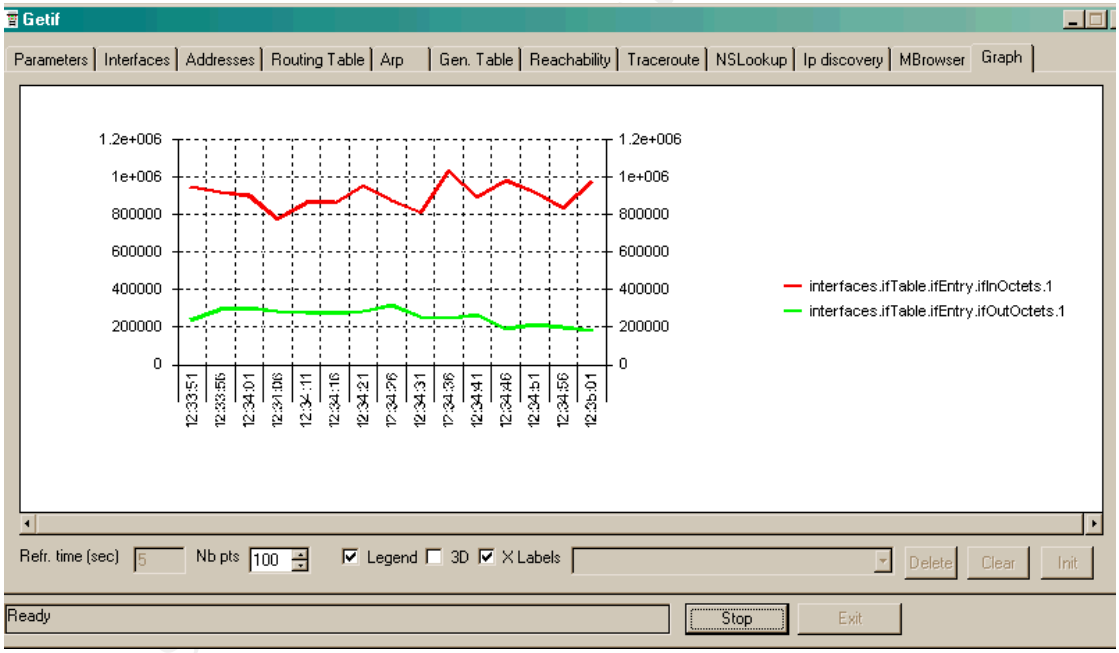
What I'm trying to illustrate is that this data actually leads you in particular directions, which can reveal the answer more quickly, possibly before damage is done.

The only thing that isn't free with MRTG is the time it will take you to set it up. Pay close attention to the script submissions by users, some of them are extremely helpful.

And yes, you can run ActivePerl on Windows NT or Windows 2000 to support MRTG. Visit Active Perl for this free download.

GETIF

A great (and free) SNMP reporting tool written by Philippe Simonet that can give you real-time (you can select down to 1 second interval) data on connections. Sometimes MRTG's 5-minute polling needs to be analyzed in light of the more frequent polling you can get from this program. GETIF allows you to poll interfaces in real time and watch the patterns. From the MIB browser tab, you can graph whatever you can find through the MIB (obviously integer values are desired) and be presented with a multi-colored graph of the data. With this you can look at things like interface errors, which may answer why that flapping port appears in your MRTG graph as merely a slow link.



PERSONAL FIREWALLS

Another great tool is the personal firewall. You don't need one on every desktop for this purpose (even though it is advisable to have one on every desktop). If you have a group of power users throughout your enterprise that have personal firewalls on their PCs, you could enlist this army to help you detect unusual network activity. I suppose this could be called a free distributed intrusion detection tool! If user A in network 10.1.1.0/24 reports to you that they are seeing a healthy number of connection attempts to TCP port

111 from a device within their subnet, with negative reports from other subnets, you know you probably have an active scan originating from that network, and destined only for that network (a broadcast scan). If you have unusual patterns traversing networks, it could lead you to the device being compromised or the device attacking others. The drawback to this scenario is that you must rely on the individuals to actually check and report the results.

Although unreliable, in smaller networks this may be adequate.

II. TESTING AND DATA COLLECTION

PORT SCANNERS

Two software tools I have found useful are CyberKit by Luc Neijens and Superscan from Foundstone. Since this paper is written for the **nix disinclined*, I'm not going to talk about NMAP, even though it pays to learn LINUX if only for that tool. Again, always seek management approval on the use of port scanners within your network and as a matter of course, don't scan any network you are not directly responsible for. Be careful with Superscan, or any other port scanner for that matter. I do know of an individual who ran several PING scans at once, throttled the scans to the fastest setting and then selected 10.0.0.0/8 as the network to scan, resulting in CPU overload on the core switch. Did I mention that the switch was the primary backbone campus switch? Ouch. Please only scan viable networks. With Superscan, you can create port lists or use their basic vulnerability list, and select precisely the network block you wish to scan.
You can also throttle the speed down so as not to create too much overhead for the systems you are scanning.

What you are generally looking for are open ports (services) that workstations or servers are allowing connections to. These scanners send a packet to the device to appear to initiate a conversation in order to get a response, which will determine if that port is open or closed. There is a lot more detail within this subject, but you would need a scanner like NMAP to take advantage of more sophisticated scans for devices that attempt to thwart scanners as well as providing TCP fingerprinting capabilities to determine operating systems.

Scan servers, routers, switches, and even workstations. For workstations, I'm looking for Trojans open on the workstations or other advertising malicious code, which means I scan particular ports. Information Security Magazine Online's Tech Talk from June of 2001 lists several well-known ports.

> …admins should be familiar with the ports commonly used by hackers to plant Trojan horses, distributed denial-of-service (DDoS) tools and malicious services. These ports include TCP 1,243, 6,667 and 27,374 (SubSeven server defaults); TCP 6,346 (Gnutella); TCP 12,345, 12,346 and 20,034 (NetBus); TCP 16,660 and UDP

18,753 and 20,433 (Shaft DDoS); and TCP 31,337 (BackOrifice). Port scanners are an invaluable tool in finding and determining the status of these often exploited ports. (Information Security Magazine Online, June 01 Tech Talk)

I use this tool (and NMAP) to give me a look at what those devices are supporting. With routers, there should be few, if any, services showing open or closed. For those of us who use Cisco routers and switches, all of those devices should not show open ports for small services like finger, echo, and chargen. You probably have an old code version if you see those. Check the open ports carefully. They should only be for legitimate services. Turn off, remove, or otherwise halt services that aren't necessary. There is a vulnerability in HTTP on Cisco routers, for instance, that may give an attacker control of the router. HTTP should not show as an open port on an older code revision.

The more ports that **don't** show up on a scan, the better!

ANALYZER AND ETHEREAL

There are few free Ethernet analyzers out there, but two come to mind: Analyzer, being developed at Politecnico di Torino, and Ethereal. My personal favorite is Analyzer.

After using MRTG or GETIF to locate the pattern, you definitely want to get a scan of the traffic. As long as you can hook up a laptop or other computer to a repeater on the offended segment, you can use these free Ethernet scanners to gather a trace of the information for both clarification and prosecution. The device must have an Ethernet NIC capable of promiscuous mode. Always seek MANAGEMENT APPROVAL on the use of analyzers within your network. You must have prior written approval to run an analyzer on ANY NETWORK. Verify and record the current time prior to starting the trace, and write down the start time and the time when you ended the trace.

After capturing some traffic, read the trace. I like to scan traces for major patterns first, and then drill down into the packet data. Some of the symptoms to look for are:

- Numerous SYN packets that don't seem to respond to the corresponding SYN/ACK.
- Attempted or failed connections to the same address or ports, or from the same address or ports.
- Physical errors. You could be chasing a pattern that is actually created by a faulty physical device.
- Unusually high percentage of traffic directed toward a single address or from a single address.
- Sequential connection attempts to addresses or ports. (10.1.1.1, 10.1.1.2, 10.1.1.3, port 1658, port 1659, port 1660)
- High number of network broadcast packets.
- Unusually high amount of a particular type of traffic, like UDP or ICMP packets. You could be misled here easily, so know what your traffic looks like. It's always

a good idea to have a scan when everything is normal to compare it to.

III. INFORMATION TOOLS

ARIN – INTERNIC – RIPE – APNIC

Not just Internet websites, but network information and autonomous system number lookups, as well as domain information. These are tools you use after the fact to find out where the attacks came from. This information can be used to inform the sysadmins or system coordinators about attacks coming from their networks.

NEOTRACE

Or, if you just want to point and click for the above information, then choose NEOTRACE. For a low purchase fee, you can enter the offending IP address, and receive all the pertinent information about where the connection came from, and the addresses and in some cases, phone numbers of the administrators of the systems the attack originated from.

DNS LOOKUP UTILITIES

Any of these free DNS lookup tools, like GHOST or Trumphurst DNS Lookup are very helpful in determining names to addresses as well as reverse lookups from Win95 or 98 systems. HexHostname, written by Hexnation and DNSMonitor, created by Christopher Prest are free utilities that will do name lookups based on IP address. Another excellent all-in-one postcardware utility is CyberKit by Luc Neijens. CyberKit can perform PINGs, traceroutes, whois lookups, time, and also has a port scanning utility. Always seek management approval on the use of port scanners within your network. You must have prior written approval to run a port scan on your network. Do not run scans of networks you are not responsible for or don't have written approval to do so. Port scans can be interpreted as attacks by some intrusion detection software and some analysts as well, and the analysts may take administrative action against you.

IV. CONCEPTUAL DEVICES

PING SERVER

This is not a utility, but a concept. If, for instance, you are attempting to deny ICMP echo-request and echo-replies (PING), but still want to provide your users and administrators the ability to verify connectivity to the internet, think about installing a server which users can connect to and initiate PINGs. You can set up a device on or outside your firewall (I use our DMZ) that hosts a website that users can send ICMP Pings to either

preprogrammed addresses, or they can enter the addresses they wish to send the packets to (in some programs) via a web interface. You can limit via access-lists only echo replies to be sent to this IP address from the outside, thereby protecting it against most attacks, but allow your inside users to connect. There are a variety of programs and scripts you can run to accomplish this. I use AspPing, offered by Serverobjects. This is a free ASP object that uses preprogrammed scripts. You can change the number of PINGs per file so that you can have a script that runs extended PINGs as well. You can PING by name (using the DNS settings on the server) to quick test your DNSs, and by IP address. Replies show up as HTML as shown below:

Reply from www.yahoo.com bytes=32 time=50ms TTL=119
Reply from www.yahoo.com bytes=32 time=40ms TTL=119

Although not a security utility, this does freely provide you with some peace of mind as far as limiting ICMP is concerned, accomplishing a possible security objective while providing a service that your sysadmins can use.

References:

Cisco Systems White Paper. "Defining Strategies to Protect Against TCP SYN Denial of Service Attacks." URL:
http://www.cisco.com/warp/public/707/4.html (11 August 2001).

McNealis, Martin. "Product Bulletin - Public #576
The Cisco IOS™ TCP Intercept." 2 October 2000. URL:
http://www.cisco.com/warp/public/cc/pd/iosw/iore/prodlit/576_pp.htm (9 August 2001).

Oetiker, Tobias. "MRTG – Multi Router Traffic Grapher." URL:
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/ (29 July 2001).

Carnegie Mellon University. "Monitor and inspect network activities for unexpected behavior." Detecting Signs of Intrusion. 25 April 2001. URL:
http://www.cert.org/security-improvement/practices/p094.html#2 (10 August 2001)

Kessler, Gary C. "Plugging Leaky Holes." Information Security Magazine Online. June 2001. URL:
http://www.infosecuritymag.com/articles/june01/columns_tech_talk.shtml (14 August 2001)

Carnegie Mellon University. "CERT® Advisory CA-1997-27 FTP Bounce."
8 March 1999. URL:
http://www.cert.org/advisories/CA-1997-27.html (6 July 2001)

Hobbit. “The FTP Bounce Attack.” URL:
http://www.geocities.com/SiliconValley/1947/Ftpbounc.htm (9 August 2001)

Download Locations for Programs or Services Noted. As of 11 AUG 2001:

MRTG – Multi Router Traffic Grapher.” URL:
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/ (29 July 2001).

GETIF by Philippe Simonet. URL:
http://www.wtcs.org/snmp4tpc/files/tools/snmp/getif/getif.zip

Instructional materials for this program can be found at WTCS.ORG. URL:
http://www.wtcs.org/snmp4tpc/testing.htm#GETIF (11 August 2001)

ACTIVE PERL, by ACTIVESTATE. URL:
http://aspn.activestate.com/ASPN/Downloads/ActivePerl/index/

NEOTRACE. URL:
http://www.neotrace.com

CYBERKIT. URL:
http://www.cyberkit.net/

SUPERSCAN. URL:
http://www.foundstone.com/rdlabs/proddesc/superscan.html

ARIN WHOIS. URL:
http://www.arin.net

INTERNIC WHOIS. URL:
http://www.internic.net/whois.html

RIPE WHOIS. URL:
http://www.ripe.net/perl/whois

APNIC WHOIS. URL:
http://www.apnic.net/

HexHostname. URL:
http://www.mofunzone.com/downloads03.htm

DNS Monitor. URL:

http://www.mofunzone.com/downloads03.htm

Analyzer, being developed at Politecnico di Torino. URL:
http://netgroup-serv.polito.it/analyzer/

Ethereal. URL:
http://www.ethereal.com/

AspPing, by Serverobjects.com. URL:
http://www.serverobjects.com/products.htm

ZoneAlarm, by Zonelabs. URL:
http://www.zonealarm.com