# Converting your old PC into a Unified Threat Management Appliance, an Inexpensive and Ultimate Defense for Small Office and Home Office Networks

By

Ivan Paulo Barrera Lewis University

May 2, 2008

# Abstract

Computer security for small business and home office users are typically limited by inexperience and cost to an ad hoc and piecemeal collection of security devices. These include virus detection and prevention programs, firewalls, spam blockers, and a never ending stream of software updates and patch management. Not only does this effort waste productivity, it is also problematic as to whether these defenses are either complete or sufficient to ensure timely protection.

Organizations today are looking for an affordable, integrated and unified approach to network security. This thesis studies a Unified Threat Management software known as IPCop as a more effective and lower cost security alternative. The IPCop software is run on an older and perhaps discarded computer which is used exclusively as a network firewall to isolate a safe internal network from the dangerous external network. The thesis engineered a universal IPCop system and investigated the levels of security that it affords. The evaluation was performed using Nessus, Zenmap, Wireshark Attack, and a variety of real virus and spam attacks. The results from this study are discussed.

# **Table of Contents**

Introduction	Page 4
Review of Literature	
Unified Threats	Page 5
Unified Threat Management	Page 8
Additional Challenges facing today's SOHO Network	Page 9
Procedure	
Future Mode of Operation	Page 11
UTM Implementation Prerequisites	Page 13
UTM Core Installation	Page 14
Basic UTM Appliance Configuration	Page 20
Custom Configuration	Page 26
Proxy Server	Page 26
Content and URL Filter	Page 27
Anti-Malware System	Page 30
Monitoring System	Page 33
SMTD Filter	Page 34
HTTP Filter	Page 36
FTP Filter	Page 37
SPAM Filter	Page 37
Anti-Virus	Page 38
VPN	Page 39
Results	
UTM Implementation Tests	Page 42
Firewall Tests	Page 48
IDS Tests	Page 52
Anti-Spam and Anti-Malware Tests	Page 56
VPN Tests	Page 63
Web Filter Test	Page 65
Conclusion	Page 66
References	Page 68

#### Introduction

Unified Threats have becoming the new norm of internet attacks where Viruses, Spyware, Spam, Denial of Service attacks, break-in attempts, and other type of attacks join forces to target just about any computer connected to the internet. Hacker sophistication has greatly developed up to the point where exploiting multiple weaknesses in a system while performing a single attack can greatly increase the chances of success to gain unauthorized access to private data.

Large companies have implemented all sorts of security controls to prevent damage to their systems and deny access to theirs assets from unified threats. System Administrators utilize a variety of security products such as firewalls, anti-virus software, content filters, Virtual Private Networks, and other similar products in order to reduce or eliminate the impact of each individual threat. These companies spend hundreds of thousands or even millions of dollars in security infrastructure and personnel in order to procure and protect company data.

On the other hand, small business owners and home users do not have the ability to put in place such security infrastructures like big companies normally do. Most often these two types of users have two to three security products in place like Antivirus software, Spyware software, and a personal firewall/Router. Moreover, these users do not have dedicated security staff, so they acquire what they think is best to protect their computers.

These users try to configure the hardware and software themselves, thinking this will really provide the necessary security against internet attacks. As a result, small business owners and home users are the most vulnerable of all internet users to the danger from hackers and from unified threats.

To deal with such threats, Unified Threat Management Appliances (UTMs), also known as All-in-One Security boxes, have come out to the market and typically include all of the individual security controls that exist in big companies. These provide the same level of protection afforded by using separated security controls, but are combined within a single box. They are easy to use, easy to setup, and most features are self-managing.

The downside of these boxes is their price. Their typical cost goes from a couple of thousand dollars to about fifty thousand dollars or more. Certainly, a lot of small business owners and home users do not have the luxury to acquire one of these great security boxes.

To address the two main problems at hand - lack of budget and lack of knowledgeable security personnel - this project proposed that small business and home office (SOHO) users can build their own unified threat management appliance.

All that that was needed was an old and unwanted PC, an internet connection, some free software downloadable from the internet, and a little bit of guidance and patience. We

had made use of simple tools, as well as to test the effectiveness of our new UTM appliance implementation.

#### **Unified Threats**

The term Unified Threat defines the entire spectrum and variety of available attacks against an organization or against a SOHO user. Although this term does not say a lot about the specifics of external attacks, its meaning signifies devastation and chaos to the targeted system because it represents an attack specifically focused on the target system.

SOHO users are faced with all sorts of attacks ranging from viruses, worms, Spyware, Trojan horses, backdoors, to even illegal intrusion, denial of service, phishing, man-inthe-middle, and others. However, most of these users do not know what attacks they are facing, and thus it is important to let such users know the real menace they have been ignoring for so long.

According to the FBI/CSI 2006 Computer Crime and Security Survey (Figure 1), the percentage of US machines (including SOHO users) that were attacked by the top security threats of 2006 were viruses (65%), insider abuse of net (42%), unauthorized access to information (32%), Denial of Service (25%), system penetration (15%), abuse of wireless network (14%), and theft of proprietary information (9%) (Lawrence, Loeb, & Lycyshyn, 2006). There are others common attacks such as financial fraud, telecom fraud, website defacement, and sabotage, but these have decreased in impact due to higher controls put in place by organizations making it harder for hackers to use such methods.



Figure 1. Common attacks for 2006 (Lawrence, Loeb, & Lycyshyn, 2006).

Considering of all these categories, the top three which accounted for three quarters (74.3%) of total monetary loses were viruses, unauthorized access and theft of proprietary information.

Out of 313 respondents (Figure 2) from different companies that responded to this survey, the total loses due to common threats was \$52,494,290 (Lawrence, Loeb, & Lycyshyn, 2006). Virus contamination had a year loss of \$69,125 per company, followed by unauthorized access to information hitting hardest with \$85,621, and denial of service had with a lowest price tag than in the previous year with a monetary loss of \$20,872 per company.



Figure 2.Total losses due to threats in 2006 (Lawrence, Loeb, & Lycyshyn, 2006).

From these numbers we can deduce that even a single individual threat will cause great monetary loss to any organization, especially to SOHO users, by bringing down a business or personal operation without a chance of recovery from such attack. But these losses can be scaled to exponential proportions when threats join forces to become a whole new unified threat.

Organizations no longer face some of these threats as isolated attacks, but rather multiple attacks which then lead to a never ending vicious cycle: trying to protect assets by patching whenever possible. However, this strategy never actually catches up with the bad guys. Preventing and reducing the impact from any sort of attack and especially against the entire unified threat becomes a tedious collection of separate job functions for many system administrators, not to say the amount of resources it requires and the strong support from upper management (Lawrence, Loeb, & Lycyshyn, 2006).

#### **Unified Threat Management**

Unified Threat Management (UTM), a term coined by Charles Kolodgy of International Data Corporation (IDC), refers to the comprehensive set of security products and controls used to reduce impact on business networks from unified threats (SearchSecurity, 2006). Proper coordination of point solutions was problematic with so many components in place, but UTM was born to provide a centralized point of timely protection and assurance for cross-functionality among secure systems (Feingold, 2005).

UTM developed out of the conventional view that firewalls should protect against intrusion-related attacks and evolved into a new concept that integrates point solutions like web content filters, spam filters, IDS, anti-virus detection, and others solutions into an "all-in-one" defense system (SonicWALL, 2005).

UTM is composed of four basic principles:

- Cost: In UTM terms, a UTM solution must be affordable; meaning that all of the administration and maintenance costs related to UTM should always stay low compared to the value of the protected assets (SonicWALL, 2005).
- Well coordinated: Since unified threats attack many vulnerabilities at one, UTM must make sure that all UTM defense mechanisms coordinately protect any and every layer of a target at all times (SonicWALL, 2005).
- Well balanced: UTM most assure safe use of resources from each defense service in order to achieve maximum performance without sacrificing productivity or increasing costs (SonicWALL, 2005).
- Cross-functional: UTM must verify that each defense service works well in conjunction with the other security services so that attacks can be prevented before they enter the perimeter (SonicWALL, 2005).

UTM's mission is to simplify all of the above principles into creating a friendly environment for administration, reducing unnecessary expenses, and successfully securing user data (Feingold, 2005).

Today, any gap in security can be a major expense to any organization, large or small, but UTM can fill these gaps more effectively than point solutions can. UTM has changed the conventional firewall architecture and has turned it into an "armored" solution (Figure 3) constantly staying ahead of unified threats (SonicWALL, 2005).



Figure 3. Unified Threat Management (SonicWALL, 2005).

## Additional Challenges in Today's SOHO Network

SonicWALL<sup>®</sup> inc. is the world UTM Appliance leader and according to their publications, unified threats can also cause network slowdowns for SOHO users. This is especially true for high bandwidth programs which cause bottlenecks and open security vulnerabilities that can be exploited. These performance bottlenecks include chat sessions and many multimedia applications (SonicWall, 2005).

Another problem that SOHO owners face is using theses business systems for personal access to the internet. This causes a lost of productivity and resources. Internet usage has to be provided to the right individuals in order to serve business needs but this benefit has to be weighed against the risk that misuse can open access to unified threats. Many large companies are spending money on point solutions in hopes of protection against the latest external and internal attacks in order to improve access to internal resources and productivity. The first point solutions included IDS systems, anti-virus agents, secure VPN solutions for remote laptop users, and other stand alone products in hopes of eliminating as many unified threats as possible (SonicWALL, 2005).

Others point solutions include the security of access from wireless secured segments, content filters that restrict web access to "company approved " internet traffic, spam filters, and firewalls to block the latest threats. Patch management for these point solutions, as well as other infrastructure can be hard to maintain. The ideal way to apply patches would be to apply them to a single device at a convenient time. Considering today's threats, point solutions simply are not sufficient to protect financial assets, as great amounts of productivity are lost, not to mention the heavy burden and costs on deployment, administration, and maintenance related to them (SonicWALL, 2005).

According to the FBI/CSI 2006 Crime and Security Survey (Figure 4), insider attacks were also a big security concern for various companies (Lawrence, Loeb, & Lycyshyn, 2006). Even though most respondents do not see insiders as accounting for most of their organization's cyber losses, a significant number of respondents believe that lack of inside security (Figure 5) still account for a substantial portion of losses (SonicWALL, 2005).



Figure 4. Percentage of Inside Threats (Lawrence, Loeb, & Lycyshyn, 2006).



Figure 5. How a UTM blocks Internal and External Threats (SonicWALL, 2005).

#### **Future Mode of Operation**

As opposed to the usual, unsecured SOHO network, the test lab created a secured network environment using our UTM appliance as the main inline gateway, protecting the client's private network from the internet.

The future mode of operation (Figure 6) required users to share files and folders, and to still be able to print to the office network printer which has its own network card. The two previous machines still exist except that now a third machine was added. This machine was used as a backup for purely browsing the internet. Workstation 1 and workstation 2 used to be PC 1 and PC 2 which simulated common tasks and role-specific job functions for each machine.



Figure 6. Future Mode of Operation for SOHO UTM Appliance.

Connecting from the internet cloud, there was a laptop user, who like many SOHO environments, connected to the private network remotely. We did not know what kind of data the remote SOHO user sent, but we were sure that security was a number one priority due to data having business-related nature.

Our test lab, like a real SOHO environment, still gets its internet signal the same way, using the same modem and obtaining his assigned DHCP IP address from the ISP (64.14.75.20). The big difference was that our UTM appliance would now attempt to obtain the internet address from the ISP instead of the router provided by the ISP.

The scope of the security perimeter covered by our UTM appliance was to stop external attacks at any and all costs. The UTM appliance was supposed to help establish, into any SOHO environment, the three principles of the CIA model: confidentiality, integrity, and availability.

Integrity meant keeping the SOHO network and data safe from unauthorized changes or modifications. Confidentiality allowed the inner network to stay safe from outside intrusion leading to leakage of data. And finally, availability meant that the remote laptop, internet traffic, and any other required services used by the SOHO user, would be available with security in mind.

#### **UTM Implementation Prerequisites**

In order for our UTM appliance to perform as desired (Figure 7), we used IPCop, a hardened open source Linux distribution which works mainly as a stateful firewall system but can be customized according to each individual SOHO user needs. IPCop comes with Snort, an open-source intrusion detection system capable of sensing threats by using attack signatures updated regularly by the Snort team in a signatures database. Both systems can be downloaded from their websites at no cost.



Figure 7. IPCop and Snort IDS at the core of our UTM appliance.

According to the IPCop Quick Start Guide, there are some basic hardware and software requirements. For IPCop to run on our old pc, we needed at least a 386 processor with 32Mb of RAM and more than 300Mb of hard disk space, a CDROM drive to install IPCop, at least one network card (10/100 Ethernet card was better), and obtain a copy of the IPCop CD or download the .iso file and burn the image onto a CD (Oberlander, 2004).

The hardware we used for our UTM appliance was an Emachines 550 with a Pentium 3 500 MHz processor, 192 MB of RAM, 10 GB HDD, cd-rw, 250 kb cache, two 10/100 Mbps network cards, two Cat 5 UTP cables, one 10/100 Mbps 4 port switch, and a power cord. We tried to go for an old machine that could meet the above minimum requirements for installing IPCop thus saving the SOHO user unnecessary expenses.

Since the UTM appliance will be the main source of security for the SOHO internal network, it had to be hidden from prying eyes and curious people. Thus, we left it headless. The keyboard and monitor were used during the IPCop installation, but they

would have to be taken away from the UTM box in order to prevent anyone with access to the machine from being able to change something.

### **UTM Core Installation**

Once we had our UTM appliance hardware ready, we went ahead with the IPCop installation. The installation was easily done, as we simply followed the designated steps in the installation guide. Before running the setup, care was taken to make sure that all data had been saved since IPCop formatted the entire hard drive and assigned it its own operating system (Figure 8).

```
ISOLINUX 2.08 2003-12-12 Copyright (C) 1994-2003 H. Peter Anvin

Welcome to IPCop, Licensed under GNU GPL version 2.

PLEASE BEWARE! This installation process will kill all

existing partitions on your PC or server. Please be aware

of this before continuing this installation.

----- ALL YOUR EXISTING DATA WILL BE DESTROYED ----

---- ALL YOUR EXISTING DATA WILL BE DESTROYED ----

Press RETURN to boot IPCop default installation.

Or, if you are having trouble you can try these options....

Type: nopcmcia to disable PCMCIA detection

nousb to disable USB detection

nousborpcmcia to disable both PCMCIA & USB detection
```

Figure 8. IPCop formatting hard drive before installing itself (Walker, Goldshmitt, & Pielschmidt, 2004).

The UTM appliance needed two network cards in order to be able to inspect traffic in and out of its internal network and into the internet. IPCop divided the network segments connected to it into zones (Figure 9).



Figure 9. IPCop dividing the SOHO network into zones.

According to the installation guide for IPCop, the Green zone was the safe internal network, the Red zone was the internet, the Orange zone (or DMZ) was used to host public servers, and the Blue zone protected wireless devices (Walker, Goldshmitt, & Pielschmidt, 2004). For our purposes, we just used the Red and Green zones (Figure 10).



Figure 10. Choosing our IPCop zones (Walker, Goldshmitt, & Pielschmidt, 2004).

The Red zone used one of the required network cards to get its external connectivity (Figure 11), but not until the UTM machine was restarted and then the ISP assigned our new IP address. This card accepted traffic only coming from the internet.

IPCo	op v1.4.0 - The Bad Packets Stop Here
	RED interface
	Enter the IP address information for the RED interface.
	( ) Static ( ) DHCP ( ) PPPOE ( ) PPTP
	DHCP Hostname: ipcop
	IP address: Network mask: 255.255.0
	Ok
<tal< td=""><td>b&gt;/<alt-tab> between elements   <space> selects</space></alt-tab></td></tal<>	b>/ <alt-tab> between elements   <space> selects</space></alt-tab>

Figure 11. Setting up DHCP for the Red zone network card for internet connectivity (Walker, Goldshmitt, & Pielschmidt, 2004).

The UTM appliance then used a second network card (Figure 12) for the green zone which served as gateway for the internal computers that needed to communicate to the internet (or Red zone)

Enter the IP a GREEN interfac	ddress informa e.	tion for the	
IP address: Network mask:	255.255.2	55.0	
Ok	C	ancel	
	_		

Figure 12. Setting up the Green zone network card for the internal network (Walker, Goldshmitt, & Pielschmidt, 2004).

In a typical SOHO network environment, and the one we used here, a broadband router obtains its internet connection from the ISP and then shares this connection though NAT using a DHCP server, which then assigns private addresses to each individual computer. Along with our UTM appliance, however, IPCop comes with its own DHCP server, and thus there is no need to have two different subnets. As a result, we replaced the router and used a switch instead which allowed easier IP address assignment to all nodes, easier management, and thus easier to troubleshooting if problems came up.

We decided to use the private addressing scheme with network 192.168.1.0 in order to comply with internet standards and to not have our addresses collide with internet addresses. Our SOHO network allowed up to 254 single addresses. We knew we needed only a subset of them and could have used "Subnetting". However, to make the installation processes less complicated, we decided to apply the default mask (i.e. 255.255.255.0).

The UTM appliance was assigned gateway address 192.168.1.1 which created addresses in the range of 192.168.1.10 - 192.168.1.30, each IP address with a lease of about two hours (Figure 13). This gateway address also acted as the address for the proxy server.



Figure 13. DHCP setup for IPCop.

The domain we chose for our UTM appliance was x.localdomain. This will allow the SOHO user to connect to the internal network remotely, from the internet by using a secure VPN connection. A VPN connection requires a right side and a left side, thus we added "x" to as a qualifier to the domain name (i.e. localdomain) to denote the left side of the VPN connection. The SOHO user's machine would had used "y" to denote the right side of the VPN connection. The name ipcop1 was the actual UTM appliance's name.

After setting up the network and host names, the last part of the IPCop installation was setting up passwords for the administrative accounts (Figure 14). IPCop required an eight digit, alphanumeric password with at least one upper-case letter for each account password. We had set up a password for the "root" user, accessible only through shell commands. Then we had setup an "Admin" account password used only through the administrative web page. And last, we had setup a "Back up" account password needed only if IPCop crashes and we need to recover the system.



Figure 14. Setting up password in IPCop for root account (Walker, Goldshmitt, & Pielschmidt, 2004).

The good thing about having all these settings to configure was that we could easily change them later on the administrative page. If there is a need to match new network requirements (such as a new domain name, new static ISP address, etc), having all these settings available to tweak will allow us to change the configuration as needed.

There were some problems at the beginning of the IPCop installation, as one of the network cards was not working. Thus, after finishing the installation and trying to get access to the internet, we could not make the connection. First, we tried to redo the installation, thinking we missed something. After a couple more tries, we knew we had installed IPCop properly according to the installation guide. We replaced the network card and were then able to connect to the internet.

We also could not get the administrative page for IPCop which we should have gotten from the Green zone, thus we believed that the network card for the Green zone needed a replacement. We found a good old network card from a Pentium 1 machine - a 3Com 100 card - and we knew this was a good replacement for the broken card.

After fixing this issue, we were able to access the administrative page for IPCop at address <u>http://ipcop1:445</u>, which uses SSL and an X.501 certificate with RC4 encryption for authentication. We discovered that the Red zone card was getting the ISP's assigned address right at the administrative home page (Figure 15).

ipcop.localdomain
Connect Disconnect Refresh
Current profile: Local ISP
Connected (0d 0h 2m 23s) - Local ISP
IP Address:
IPCop's Hostname: u==
13:38:50 up 12 days, 20:38, 0 users, load average: 0.17, 0.28, 0.18

Figure 15. Administrative home page for IPCop (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

#### **Basic UTM Appliance Configuration**

As mentioned earlier, our UTM appliance contained the IPCop firewall and the Snort intrusion detection system at its core. In order to get our SOHO lab started, our next step required us to configure these basic systems in order for our UTM appliance to work to our future mode of operation specifications.

The first step in our basic UTM configuration began by making sure that IPCop received the ISP's assigned IP address. To do this, we went into the Network Status page (Figure 16), which showed the current status of the Red and Green network cards. But right before viewing the contents of the page, we needed to log in with the administrator login account for IPCop along with the previously set password for it.

🗅 IPC	op - Network Status Informat	🖸 🔀 ipcop Web server - Google Search 🛛 🔄
1	RED DHCP configuration	
	Hostname:	konstantin terterak teri bergi matatan kanan kanan
	Gateway	()First Bioshillon)
	Primary DNS:	A CARLES AND A CARLES
	Secondary DNS:	
	DHCP Server	0.0.0.0
	Default Lease Time	92.7686111111111 Hours
	Default Renewal Time	46.3841666666667 Hours
	Maximum Renewal Time	81.1725 Hours

Figure 16. Network Status page showing ISP's assigned address (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

The dialog listed the IP address obtained from the ISP, the mask, DNS server addresses, time of lease, and default gateway. Also, it showed the current DHCP leases from the address range previously configured for the green network zone. At this point, we had successfully acquired internet connectivity for the Red zone network card within our UTM appliance. We then tried to access the internet directly from each workstation on the internal network and we were able to access the yahoo homepage.

By default, IPCop firewall started running once we obtained the ISP's assigned IP address for the Red zone network card. We could verify this fact, however, by going to the top of the Network Status page (Figure 17) and observing how eth1 (Red Zone) network card showed its IP address assignment information.

) IP(	op - Networl	k Status Informat 🔯 🔀 ipcop Web server - Google Search 🛛 🔹
	eth0	Link encap:Ethernet HWaddr 00:01:03:D4:C8:13 inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
		UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:9005 errors:0 dropped:0 overruns:0 frame:0 TX packets:13133 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000
		RX bytes:1143238 (1.0 MB) TX bytes:8966422 (8.5 MB) Interrupt:11 Base address:0x4000
	eth1	Link encap:Ethernet HWaddr <b>Construction</b> inet addr: <b>Construction</b> Bcast:255.255.255.255 Mask: <b>Construction</b> UP BROADCAST NOTRAILERS RUNNING MTU:1500 Metric:1 RX packets:683801 errors:0 dropped:0 overruns:0 frame:0 TX packets:5124 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:49785913 (47.4 MB) TX bytes:843716 (823.9 KB) Interrupt:10 Base address:0xdf00
	10	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:16436 Metric:1 RX packets:9 errors:0 dropped:0 overruns:0 frame:0 TX packets:9 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0

Figure 17. Network Status page showing ISP assigned address information (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

Another way to make sure that the IPCop firewall was fully functional was to go to the firewall logs menu. Here, we could see how IPCop received incoming hits (or traffic)

whenever someone outside of the SOHO internal network wanted to connect to our UTM appliance, or whenever someone within the SOHO network browsed the internet or wanted to retrieve internet content, because the firewall logged every incoming connection. We had set the "clear firewall logs" period to happen every 2 weeks in order to prevent the hard drive fill it self with unneeded data, and to at least have some history in case there was a breach in the perimeter (Figure 18).

rewall l	og	6	3 G	ipcop Web server	- Goog	jle Sear	rch 💽			
ttings:										
			Month: 3 💌	March	•	Day:		<< >	>> Update	Export
g:										
ıl numl	ber of fire	wall hit	s for M	larch 03, 2008	3:13					
		Old	er						Newer	
me	Chain	Iface	Proto	Source	Src l	Port	MAC Ad	dress	Destination	Dst Port
01:20	INPUT	eth1	UDP	24.64.69.41	32	65 (	00:01:5c:2	23:d6:02	<u>67.175.17.185</u>	1027
01:20	INPUT	eth1	UDP	24.64.69.41	32	65 (	00:01:5c:2	23:d6:02	67.175.17.185	1026

 D7:33
 INPUT
 eth1
 UDP
 202.97.238.196
 34992
 00:01:5c:23:d6:02
 67.175.17.185
 1026

 11:10
 INPUT
 eth1
 UDP
 202.97.238.226
 32874
 00:01:5c:23:d6:02
 67.175.17.185
 1026

 11:10
 INPUT
 eth1
 UDP
 202.97.238.226
 32874
 00:01:5c:23:d6:02
 67.175.17.185
 1026

 11:10
 INPUT
 eth1
 UDP
 202.97.238.226
 32874
 00:01:5c:23:d6:02
 67.175.17.185
 1026

 11:10
 INPUT
 eth1
 UDP
 202.97.238.226
 32874
 00:01:5c:23:d6:02
 67.175.17.185
 1026

3265

33503

00:01:5c:23:d6:02 67.175.17.185

00:01:5c:23:d6:02 67.175.17.185

00:01:5c:23:d6:02 67.175.17.185

1028

1026

1026

01:20

02:56

04:59

INPUT

INPUT

INPUT

eth1

eth1

UDP

24.64.69.41

eth1 UDP 202.97.238.197 34619

UDP 221.208.208.89

Figure 18. IPCop firewall logs page (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

There are configuration options for our IPCop's Firewall like Port Forwarding (which is useful for web servers in the DMZ), and the External Access option used for remote SSL (non-VPN) connectivity from the internet. Since we do not have a web server and we are not remote connecting into IPCop at the moment, we will only make use of the firewall options.

According to the IPCop installation guide, the only available option under the Firewall options menu which disables ping requests on any interface is the "Disable ping response" (Walker, Goldshmitt, & Pielschmidt, 2004). For security reasons we disabled pings on the Red zone network card to prevent damage from port scans or any other type of attack using pings coming from the internet (Figure 19).

🔷 IPC	op - Firewall options	(	🖸 🚺 ipcop We	eb server - Google (	Search 💽		
D	Firewall	(†	FIREWALL OPTI	ONS +	-	The bad pad	ckets stop h
<b>20P</b> 4.18	SYSTEM & STA	ATUS 🔹	NETWORK *	SERVICES &	FIREWALL *	<b>VP</b> Ns	* Locs
	Firewall options Disable ping resp © No	s ponse					
	C Only RED C All Interfaces				Save		

Figure 19. IPCop firewall options page (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

Once we had our firewall running, we went ahead and configured our IDS created by Snort. By going to the services menu, we noticed that the IDS was turned off on both the Green and Red zone network cards, and this was because the IDS had not being started yet. Our first step in setting up the Snort system required a registered account with Snort. We went to www.snort.org to sign up for an account.

Once registered with Snort, we were able to have access to the Snort VRT certified rules which could then be downloaded from the Snort website into IPCop by using an oink code created during registration (Figure 20). According to Snort, oink codes allow you to get the latest updates for snort rules (Snort, 2007). The SourceForce Vulnerability Research Team (VRT) is an open source group dedicated to respond to the latest trends in intrusion attempts and vulnerabilities. Their rules (Figure 21) are very well accepted in the network security industry (Snort, 2007).



Figure 20. Snort VRT rules (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mountd access";)

Figure 21. Sample Snort Rule (Snort, 2007).

Once the newest Snort VRT rule set had been downloaded and applied to the IPCop system, we had activated the IDS on the UTM. The IDS can be activated on both the Green and Red zones, but on the Red zone it will fire unnecessary false positives because the firewall had been constantly hit. As a result, we decided to activate the IDS on the Green zone as its function was to work inline with the firewall to log and prevent any intrusions that did enter the internal network.

Once activated, we saw the IDS status changing from stopped to running, indicating that the IDS sensor on the green zone began to capture packets and match them against its attack signature database for any suspicious intrusions (Figure 22).

IP	Services		INTRUSION I	Detec	π <b>ο</b> n •			The bad	packe	ts :
<b>'COP</b> 1.4.18	SYSTEM 🔹 S	STATUS 🕓	NETWORK	*	SERVICES	*	FIREWALL *	<b>¥</b> ₽Ns	*	L
(	Intrusion Dete	ction Sys	tem:							
Í	Interfaces:						Status	:		
	GREEN Snort	eth0						RUNNIN	G	
	🗖 RED Snort eth	11						STOPPE	D	

Figure 22. Snort changing state from stopped to running (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

IPCop's Snort sensor, like any other IDS sensor only logged intrusions, but in the event that a real intrusion it will not prevent it like an Intrusion Prevention System (IPS) does. Fortunately, Snort VRT signature rules are updated every fifteen minutes, and so if a new vulnerability comes out, it won't be long until an update patches that vulnerability and thus prevent any illegal intrusion. Updating rules VRT rules on IPCop, however, had to be done manually. To verify that the intrusion detection system really worked, we needed to view the IDS logs located under the "IDS Logs" menu. If this menu appeared, this meant that the IDS ran fine (Figure 23).

IP	Logs			$\phi$	IDS LOGS			*		-	The bad	packe	ts st
<b>'Cop</b>	SYSTEM	*	STATUS	*	NETWORK	*	SERVICES	*	FIREWAI	.г ∗)	<b>VP</b> Ns	*	LOG
1	Setting	s:											
					Month: F	ebru	ary 💌	Da	iy:	<.	< <u>&gt;&gt;</u>		odate
	Log:												
	Total of r	numbe	er of Inti	<b>rusic</b> Ole	<b>on rules ac</b> der	tiva:	ted for Fe	ebrua	ary 16: O			Newer	r
ļ													

Figure 23. IDS Snort Logs Clancey, Goldshmitt, Kastner, Oberlander, 2004).

#### **Custom Configuration**

The next step for building our UTM appliance required custom configurations of IPCop by adding a content filter, an anti-virus scanner, an spam filter, various filters like ftp and junk traffic, and the addition of VPN access for the remote laptop user. These add-ons provided the necessary services needed by the UTM appliance to secure the SOHO internal network from unified threats.

#### **Proxy Server**

The first added configuration was the Squid proxy server that came pre-installed with IPCop but first needed to be activated. Squid helped users to access internet pages faster than with a regular connection thanks to its cache database that saved commonly used pages and thus accelerated each user's internet browsing experience. The proxy server was enabled by going to the services tab, Proxy (Figure 24).

Web proxy:			
Enabled on Green:	V	Upstream proxy (host:port): 🌢	
Transparent on Green:	V	Upstream username: 🍳	
		Upstream password: 单	
		Proxy Port:	800
Log Enabled:		Your extension_methods list: 🌑	
Disallow local proxying on blue/green networks:		or specify a list of destinations which are not to be proxied: •	
Cache management			
Cache size (MB):	2000		Repair cache
Min object size (KB):	0		Clear Cache
Max object size (KB):	40096		

Figure 24. Proxy service administrative page (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

In here, the "Enabled on Green" box had to be checked in order to allow IPCop to start the proxy. However, if this check box was marked alone, then each client machine needed to be configured in order to use the proxy service, and so "Transparent on Green" was also enabled. Transparent mode allowed client machines to use the proxy server without the need for any browser configuration on their end, which made client access to the internet much easier. Enabling Transparent mode just saved us the hassle of doing an extra step, but it also had to be activated before installing the URL Filter (a web content filter explained later) to blocked unwanted URLs. The cache size was changed from 500MB (default) to 2000MB in order to store a large number of commonly used web pages and thus speed up page retrieval whenever clients were browsing the internet. The maximum download file size was set to 10Mb, which is the default size for most email clients like yahoo and hotmail. The downside of this was that it prevented users from downloading large files, but this was reasonable since allowing this could slow down the network bandwidth.

We saved our changes and the proxy server was scheduled to start up immediately. The best way to know if the proxy server was running was to go to the Graphs tab, Proxy graphs (Figure 25).



Figure 25. Squid Proxy Access Graphs and statistics (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

In here, the status of the proxy server could be seen along with some statistics about how much traffic had been accessed, and time of day or even day of the year it was accessed. Another way to see if the proxy server worked fine was to go to the system status tab and see if the proxy server state was set to "Running".

#### **Content and URL Filter**

Once the client workstations had faster access to the internet, the SOHO owner decided to improve employee productivity by only allowing internet access to approved URLs or to websites with approved content, and the URL filter was used to fulfill this requirement.

URL filter, which is an add-on for IPCop (and can be downloaded from www.urlfilter.net), became the perfect tool for this job. According to URL filter, its URL and content filter will block unwanted websites by using blacklists or whitelists in conjunction with the Squid proxy server (Sondermann, 2004).

Before installing the URL filter, we needed to install Putty (an SSH remote client access application) in order to administer IPCop from one of the client workstations. Another application used during this implementation was WinSCP, which allowed files to be uploaded from a windows machine into a Linux machine. Once we downloaded URL filter and our two utility tools, the URL tar file was placed into a temp folder on the client workstation ready to be moved.

Before we logged into the UTM appliance with WinSCP from the client workstation, the SSH server needed to be enabled. This was done by going to the System tab, and then SSH (Figure 26). By checking the "SSH Access" check box, the SSH server provided access to the Green zone by default, but if we wanted to enable access from the internet we had to made sure that "Allow TCP forward" was checked (this was used in conjunction with Port Forwarding to allow remote access from the Red zone). It was too risky to leave Port Forwarding open for SSH, so it had to be closed whenever SSH was not used. The rest of the default options were accepted and saved the changes. One thing noted was that IPCop SSH server uses port 222 instead of the typical port 22.



Figure 26. SSH Server administrative age (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

With SSH activated, a WinSCP session was created with ipcop1 (Figure 27) using port 222 as SSH port, and the same username and password for IPCop's root user, and then accepted the rest of defaults.

niiser eogin	_		ن النہ	
Session - Stored sessions Environment - Directories SSH Preferences	Session Host name: [10001] User name: Troot Private key file:	Pojt number: 222 = Password:	Pot number: 222	
	Protocol Eile protocol:	CP		
Advanced options				
A <u>b</u> out Lan	guages L	.ogin <u>S</u> ave Close		

Figure 27. WinSCP connecting to IPCop using SSH (Prikryl, 2000).

While logged on to IPCop through WinSCP, we moved the URL filter tar file from the client workstation and then placed it into "/var/urlfilter" in IPCop. Later, Putty was used to unzip the tar file using the command "tar –xzf ipcop-urlfilter-1.9.1.tar.gz", and then we ran the installer using the command "ipcop-urlfilter-1.9.1/install". To make sure that the URL filter worked, we opened the administrative page and went to the services tab, URL Filter (Figure 28). We saw the URL Filter in the menu as expected.



# Figure 28. Services List tab within IPCop (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

### Anti-Malware System

The next configuration for our UTM appliance required us to include a way to stop any sort of malware (e.g. viruses, Spyware, worms, Trojan horses) from emails arriving to and originating from the SOHO network. This requirement also applied to internet browsing whenever any internal user could accidentally receive infected traffic, or when he or she could be pulled into an infected URL by a phishing attack. In order to successfully mitigate these risks, we had to find a system that could work fine under IPCop, that did not sacrifice available memory, and that could get the latest updates and virus definitions automatically. Copfilter was an un-official add-on and the only one available with these requirements.

According to Copfilter, some of its features include an email anti-virus and Anti-Spyware scanner, an internet anti-virus traffic scanner, monitoring, administration and management toolkit, daily and hourly updates, user notification emails, email reports, and many other software. Copfilter is free under a GPL license and was downloaded from copfilter.org. Copfilter, however, is not an official IPCop add-on and it says so right at their site. Memory requirements are more stringent on IPCop since it did take up more memory when Copfilter ran together with other services (Madlener, 2002).

In order for Copfilter to scan for malware, it needed various additional software packages including HAVP (an http antivirus transparent proxy that scanned dynamic and password protected traffic), ClamAV (a virus scanner with support for zip, gzip, and bzip and automatically updates itself), Renattach (a stream filter that can identify and rename potentially dangerous email attachments), Monit (a monitoring utility that communicates the administrator with alerts, updates, and reports), Frox (FTP transparent proxy filter), ProxSMTP (SMTP transparent proxy filter), and P3Scan (transparent POP3 filter).

We installed Copfilter in the same way as the URL filter by using SSH in conjunction with Putty and WinSCP. We first copied the latest version of Copfilter (version 0.84 Beta 3a at the time of this writing) using WinSCP into the "/root" directory of IPCop (Figure 29).

J 🖝   🕅 💷 .	] 💷 🖼 👽   💻   4	•🖬  ] ii ii	un en exa () "🖃 (f				
🛛 🥯 D 💌 🖉 😓 🗸	• ⇒ -  ] 🖻 🔯 🚮	🗿 🖾 📴	root	▼   ← → ⇒	- 🛛 🔁 🗖	🚮 👩  📾	te
D:\copfilter\copfilter			/root				
Name	Size 1	Гуре	Name 🛆	Size Cha	nged	Rights	Owr
💼	F	Parent directory	<b>È</b>	16.0	)4.2005 18:	rwxr-xr-x	root
copfilter-0.1.0be	ta1.tgz 10.476.984 F	PowerArchiver					
	9% Copying				×		
				( Constant	1		
	<u>~</u>		1-	Lancel			
				Minimize	1		
					- 1		
	File: D:\	.copfilter\copfilter\	copfilter-0.1.0beta1.tgz				
	Target: /ro	ot/		<u>Speed</u>			
					21		
	Transfer mode:	Binary	Resume: Ena	abled 5% 100:	*		
	Time estimated	: 0:00:41	Time elapsed: 0:0	10:04			
	Bytes transfere	d: 988 KB	CPS: 2461	KB/s			
	Disconne	ct when operation	finishes				
		or men operation					
•		Þ	•				F
0 B of 10.231 KB in 0	of 1		0 B of 0 B in 0 of 0				
🕫 F2 Rename 🛛	F4 Edit BF5 Copy	F6 Move	F7 Create Directory	🗙 F8 Delete	F9 Proper	ties 👖 F10	Quit
8.467 B	1.096 KB	🤵 🔒 aes	SFTP (v3)	0:00:16			

Figure 29. Copying Copfilter into IPCop using WinSCP (Madlener, 2002).

We proceeded to extract the package in the same directory using the command "xzyf Copfilter-0.84.3a.tgz", and then changed to the Copfilter 0.8.4.3a directory (Figure 30 and Figure 31) where we ran the "./install" command in order to run the installer.

root Gincon - %/confiltor - 0 1 00000 00 # /install
ndCebeck done
extracting uone
now executing /var/log/cop+ilter/0.1.0beta10a/setup_util -1
Copfilter installation Version 0.1.Obeta10a
WARNING :
This package is NOT an official ipcop addon. It has not been approved
or reviewed by the ipcop development team. It comes with NO warranty or
guarantee. so use it at your own risk.
This nackane adds firewall rules, proxies, filters, virus scanners
and precompiled biparies to your incon machine which is a
and precompting the set of the se
by security lisk : mere are probably several ways or breaking
or incruding copriler (pis do report ir you find any), so ir
security is an issue, up not install this package.
CONTINUE ? [Y/N]

Figure 30. Extracting the Copfilter package into IPCop (Madlener, 2002).

Continue ? [y/N]y Ok, now installing copfilter on your machine.. extracting tar file for /usr/local/bin helper scripts ok adding startup scripts to /etc/rc.d/rc.local ok adding copfilter startup scripts to /etc/rc.d/rc.firewall.local ok adding copfilter users and groups changing ownerships ok ok linking init scripts to /etc/rc.d/init.d/ ok modifying crontab ok linking unzip to /usr/local/bin ok linking wget to /usr/local/bin installing copfilter webgui for ipcop >= 1.4.4 installing copfilter webgui for email white/black lists ok ok ok inserting webgui entry into header.pl ok adding copfilter documentation to /home/httpd/html/copfilterdoc ok Modifying /etc/httpd/conf/httpd.conf and restarting httpd deleting link to copfilter installation directory from /root nk ok creating default link ok creating a new razor account (this could take a minute).. ok Copfilter 0.1.Obeta10a installation completed successfully ! Don't forget to: 1. Enter your Email Address and Smtp Server in the Copfilter webgui IPCop webqui -> Services -> Copfilter 2. Read the documentation: README (in webgui or /root/copfilter/doc) 3. Configure Copfilter webgui: configure AND press >>Save Settings<< in each section and then press >>Restart All Services to start all programs 4. If desired run tests from the webqui or from /root/copfilter/tests/make\_all\_tests.sh Pls report errors and questions to >>copfilter-main at lists dot sourceforge dot net<</pre> >>hello at test dot com<< is an email address and would mean hello@test.com or visit the official copfilter forum (link is at the bottom of the webgui) root@ipcop:~/copfilter-0.1.Obeta10a # 🚪

Figure 31. Running the Copfilter installer (Madlener, 2002).

After the install was done, we went ahead and verified that indeed Copfilter did get installed by going to the administrative GUI for IPCop where we could see the new tab for Copfilter (Figure 32). When we clicked it, we could see the status of all programs installed by Copfilter. We knew at first all of the services were turned off because they have not been enabled, but so far this showed us that at least all the packages were installed successfully.

ID	System	1	Номе	•			-	The bad	packe	ts stop here.
Сор	SYSTEM	* STATUS	* NETWORK *	SERVICES &	FIREWALL *	<b>VPNs</b>	*	LOGS	*	COPFILTER *
										STATUS
	1									EMAIL
		ipcop1.x.l	ocaldomain						-	MONITORING
		( in the second s								POP3 FILTER
			Conne	ct Discor	nnect Refre	esh				SMTP FILTER
			C	onnected (0d	1h 45m 14s)					HTTP FILTER
			•	IP Address: 63	7.165.170.5					FTP FILTER
			IPCop's Hostr	iame: c-67-165	-170-5.hsd1.il.co	mcast.net	t			ANTI SPAM
			15:48:27 up 1:	45, 0 users, loa	d average: 0.00	, 0.00, 0.	00			
				0	)					TESTS & LOGS
	ļ						_			

Figure 32. Copfilter available from IPCop GUI (Madlener, 2002).

Before Copfilter began scanning for malware, we needed to do some configuration within Copfilter. The very first thing we did was configure the Copfilter's Email section where all virus alerts, virus signature updates, and failed services notices went to whenever these happened. We entered the administrator's email address and the SMTP DNS server name that points to the administrator's email SMTP server.

#### **Monitoring System**

Next, we turned on Monit, the service that monitored all of the enabled services under Copfilter which would send all monitoring notices to the email address set in the Email section above (Figure 33).

IP	COPFILTER MONITORING The bad packets stop here.	-
1.4.18	System * Status * Network * Services * Firewall * YPNs * Logs * Copfilter *	4
1	Monitoring:	
	Monitor all enabled on  Monit Service Manager	
	<ul> <li>Note: <ol> <li>Monit will only monitor services that currently are running, as configured per webgui.</li> <li>If any monitored service fails, then monit will within 60seconds automatically restart that service and send a notification to Email address</li> <li>If any monitored service is stopped manually on the ipcop command line then monitoring for that service will stop, restarting the service manually will NOT enable monitoring again</li> <li>To re-enable monitoring of all enabled services, restart the monit services</li> <li>Autorization for Monit Service Manager: user admin and same password as in ipcop webgui</li> <li>SMTP-AUTH will not be used to send monit email notifications</li> <li>Monit will only start if you have configured an email address and if you have an active internet connection</li> </ol> </li> </ul>	

Figure 33. Monit management console (Madlener, 2002).

Monit also monitored these enabled services by providing its own web GUI with useful features like the status of each service, CPU usage, and memory used by each service. This Monit GUI could be accessed from "192.168.1.1:446"

#### **POP3** Filter

Next in line for our custom configurations was the POP3 scanning configuration. P3Scan was our transparent POP3 proxy filter in charge of detecting viruses whenever SOHO users received email. For our desired POP3 filter configuration we turned on as many options that allowed us to filter any type of outgoing and incoming malware (Figure 34)

ID	COPFILTER	🔶 РОРЗ	Filter	•			The bad	packets stop he	ere. 🔺
1.4.18	SYSTEM & STATUS	* NETW	'ORK 🔹	SERVICES &	FIREWALL &	<b>VP</b> Ns	* Locs	* COPFILTE	ER 🕸 🏹
	POP3 Scanning (P3Sc	an):							
	Enable P3Scan to filter in traffic on GREEN*	acoming	on 💌		Stop viru virus <u>not</u> Tag Spai <u>modify ti</u> <u>Rename</u> attachmo	us emails an <u>tifications</u> in: m in emails <u>he subject</u> + <u>dangerous</u> ents	d send stead + and + email	on 💌 on 💌	
	Add Copfilter <u>Comment</u> to Email Header Add "Copfilter Scanned" to Email (works only for formatted emails)	Disclaimer correctly	on 💌		Send a c <u>notificati</u> Use Cop and <u>Blac</u>	copy of virus on to <u>Email</u> filter <u>Whiteli</u> <u>klist</u> +++	address st	on 💌	

Figure 34. Configuring P3Scan for POP3 scanning (Madlener, 2002).

We needed to analyze incoming traffic on the green interface because this is where our clients resided. We added a Copfilter comment to each email header for testing purposes, we quarantined emails with a score greater than twelve, we arranged virus notifications instead of sending the actual virus to the user (this allowed to administrator to get a copy of the virus), we made bad attachments be renamed (for various types of malware), and a copy of a virus notification had to be sent to the administrator. Infected emails also had to be quarantined. We accepted the rest of defaults and saved our changes.

#### **SMTP Filter**

Since our SOHO users needed to be able to send secure email from the internal network, we also needed to enable ProxSMTP (the transparent SMTP proxy filter) in order to allow secure email departure and arrival from the internal email server (Figure 35).

1.4.18	System &	STATUS 🌸	NETWORK *	SERVICES	*   FIREWALL *	<b>VP</b> Ns	*   LOG!	6 * (	COPFILTER *	
	SMTP Scan	ning (ProxSMT	P):							
	Enable ProxS traffic on GRE	MTP to filter out EN*	going on	]	Stop viru opt. senc instead ( Tag Spar <u>modify th</u>	is emails ar I virus <u>notif</u> see below) m in emails <u>te subject</u> -	nd fications + and ++	on 💌		
				_	<u>Rename</u> attachme Send use	o <u>angerous</u> ents er a <u>virus n</u>	otification	on 💌		
	Add Copfilter to Email Head Add "Copfilter	<u>Comment</u> ler r Scanned" Discl	aimer	1	with infor originally the virus Send a c	rmation abo sent emai opy of viru	out the I containing s	on 💌		
	formatted em	ails)		1	notificatio	on to <u>Email</u>	address			-

Figure 35. Configuration for ProxSMTP SMTP proxy filter (Madlener, 2002).

To do this we enabled the SMTP filter for outgoing traffic on the green interface to prevent further virus spreading to other networks, and we attached a Copfilter email header to each outgoing email confirming that is was scanned. The SMTP filter was configured to stop virus-infected emails coming in or exiting from port 25 and rename dangerous attachments as well. The virus notifications went to the administrator whenever a user sent or received infected emails, and a copy of every virus was sent to the administrator so that isolated testing could be done and see whether a real virus threat existed.

Furthermore, the filter should quarantine infected emails for further scrutiny before deleting them, send a copy of quarantined emails to the administrator in the form of a report, sanitize emails from dangerous html tags, and quarantine emails which had score greater than 15 (on a scale of 5 being the least and 40 being the most dangerous).

Moreover, In order for Copfilter to successfully filter SMTP traffic, we needed to enable the port forwarding feature from within ProxSMTP, which opened up external access to SMTP port 25, thus allowed the SOHO's mail server to send and receive emails while being protected by using SMTP filtering. We turned on ProxySMTP for incoming traffic on the Red zone (this allowed email traffic going to the mail server to be screened with the SMTP filter first), and indicated the IP address where the email server resided (i.e. 192.168.1.11). We made it work so that external emails were redirected to the green zone because the SOHO owner does not have a dedicated mail server in the orange zone. Finally hit save to end configuration for the SMTP filter.

## **HTTP Filter**

Next, we activated HAVP proxy service. HAVP worked in conjunction with ClamAV and the Squid proxy server as a transparent proxy capable of filtering any type of http packet containing malware, exploits, or similar unified threats (Figure 36). This was very useful because the SOHO user, who knew very little about internet, could and would probably stumble into many infected websites, thus opening many security holes into the internal network.



Figure 36. HAVP http proxy filter configuration (Madlener, 2002).

To activate the HTTP filter, we went to Copfilter, HTTP filter. In here, we had turned on several options including the "Deny access to dangerous traffic" (i.e. exploits, phishing attempts, malware) that also started the HAVP proxy service. We enabled the filter to run in transparent mode so that no client configuration was required. Above all, we enabled the ClamAV virus scanner in HAVP for http scanning so that it prevented any viruses from entering though port 80.

Another activated feature of the http filter was Privoxy, a junk proxy that blocked ads, banners, jumping windows, pop-ups, Internet explorer exploits, and other unwanted content constantly appearing while browsing the Internet. Privoxy was enabled by checking the "on" box. We saved our changes and restart the service.

#### **FTP Filter**

One of the simplest services that we enabled was Frox, an ftp proxy filter. Like HAVP, Frox also used ClamAV to prevent SOHO users from downloading malware whenever users downloaded a file using FTP (Figure 37). The service was enabled by the "Deny access to virus infected FTP downloads on GREEN" option and then we saved our changes to enable this service.

1.4.18	SYSTEM ·	♦ STATUS	* NETW	'ORK 🔹	SERVICES	*)	FIREWALL *	¥PNs	* )	LOGS	* )	COPFILTER 🕸 🏹	<b></b>
ſ	FTP Scan	ning (frox):											
	<u>Deny acces</u> downloads	<u>is</u> to virus infe on GREEN*	ected FTP	on 🔽	·								
	<u>Note:</u> Frox uses C Be aware ti web browse downloadin deliver the	ClamAV (and I hat when you er or a comm g the file in th rest of the ftp	F-PROT if ins try to downl and line ftp u le backgrour file to your	talled) a oad an f utility) is id, and c client or	s a virus sc. tp file it mig hanging. Th only sends a oly after it h.	anner Iht app Iis is s I few t as full	bear that you o because th sytes of this f y downloaded	r ftp client e ftp proxy ile to your d and scan	(for ex r is alre ftp clie ned the	ample a ady nt. It will a ftp file.			
	* Restart of Skip Se	<sup>-</sup> frox service ervice Restart	required										
	Save set	tings (and res	tart service)										

Figure 37. Frox ftp scanning proxy configuration page (Madlener, 2002).

#### **Spam Filter**

Apart from secure email and safe Internet browsing, the SOHO users also desired freedom from unwanted emails or most commonly known as Spam. Copfilter came with an Anti-Spam system called SpamAssassin, which is a package that used a sophisticated scoring system to classify and eliminate Spam emails (Figure 38). P3Scan and ProxSMTP used SpamAssassin to detect Spam.

IP	COPFILTER		•			The bad p	ackets stop here.
<b>Cop</b> 1.4.18	SYSTEM & STATUS	* NETWORK *	SERVICES &	FIREWALL *	¥PNs -	LOGS	* COPFILTER * 🏹
ſ	AntiSpam (SpamAssa	ssin, P3PMail, Rule	sDuJour):				
	SpamAssassin Statistic	<u>s</u> <u>SpamAssassin S</u>	tatistics (text)				
	Year: 2007 💌 Month: 🗸 Year: 2007 💌 Month: 🗸	January 💌 Shi January 💌 Day:	ow    1 💌 Show	,			
				_			
	<u>SpamAssassin:</u>						
	Enable Spamassassin (necessary to detect spam)*	on 💌		<u>Copfilter Whit</u> <u>Copfilter Spar</u>	telist Manage m Digest Mar	r nager	
	Score required to identify email as spam (subject will be <u>modified</u> )*	6 -		<u>Razor, DCC, I</u> (improves red decreases pe	<u>DNSBL</u> * cognition, rformance)	off 💌	
	Send daily <u>spam digest</u>						_

Figure 38. Setting up SpamAssassin filter (Madlener, 2002).

We activated SpamAssassin by switching on the "Enable SpamAssassin" option. Then set the score required to identify email as Spam, ideally this had to be a low number to prevent every email from being marked as Spam. Also, we had set the daily Spam digest which notified the system administrator whenever Spam email was filtered to the quarantine bin.

#### Anti-virus

The last configuration for the Anti-Malware system involved enabling the antivirus scanner, with ClamAV as its virus protection engine, by going to the "Antivirus" section within the Copfilter menu. We changed the state of the service to "on", which allowed P3Scan and ProxSMTP to detect viruses (Figure 39).

<b>L</b> 1.4.18	COPFILTE System	R «	ANTIVIRUS	ervices & Fir	REWALL & VPNS	The bad pa	ckets stop here.	-
	Anti¥irus	(ClamA¥, A¥G	, F-Prot):					
	<u>ClamA¥:</u>	on 💌 *	Treat an encrypt archives as virus	ed es * on 💌	ClamAV Statistics	<u>Text</u> (immedia <u>Graph</u> (weekly u after one week)	te updates) updates, starts	
	Current vii ClamAV 0.9 Sun Apr 6 1 Enable Unof <u>Phishing Sic</u> update)	rus signatures 11.2/6635 11:29:31 2008 ficial ClamAV Inatures (+daily	: Autom © Ena C Disa	<b>atic update**:</b> bled every: abled	4 hours	Manual	update: te clamd now	
	F-PROT:	on 🔽		<u>بە</u> بە .				Ŧ

Figure 39. ClamAV configuration page (Madlener, 2002).

We configured this section so that it blocked encrypted files and treated them as viruses because SOHO users usually ignored that viruses could hide within an encrypted file. Encrypted emails were supported if the email server supported this feature, and also SpamAssassin had to recognize the sender's email address within the whitelist.

We enabled automatic updates every four hours and activated unofficial daily updates for ClamAV phishing signatures (useful for the novice Internet user and even for the savvy user). We finished the Antivirus section by saving our settings and then restarted the service.

#### VPN

Our next step involved configuring the VPN connection for the remote laptop user and we did this by using the existing VPN gateway in IPCop. This VPN gateway provided users with the latest secured-tunnel technology like IPSec joined together with 3DES (current encryption algorithms used by the U.S. Government) in order to hide data from prying eyes. Obviously the VPN client on the other end of the connection must also support the same encryption algorithm, and that's why IPCop supports a variety of other encryption algorithms like AES that with an excellent level of security and confidence to the VPN client. To configure the VPN service (Figure 40), we went to the VPN menu and selected VPNs within the IPCop administrative GUI



Figure 40. VPN configuration page (Support, 2007).

In here we configured our VPN service to allow the SOHO remote laptop user to establish a VPN session. We configured the global settings by entering our UTM appliance's Red IP address in the "Public IP" field (this is where the VPN client had to connect to). We then checked options like parsing, emitting, control, klips, DNS, and nat\_t, and finally checked "Enabled" to start the VPN service. With this done, we added a RoadWarrior VPN connection under "Connection Status and Control" (Figure 41), gave it a name in the localID field and saved our changes to enable the connection for this RoadWarrior account. A RoadWarrior connection is Client-to-Network connection where IPCop will consider this new VPN client as if it were within the internal network.

connection:			
Name:	First¥pn	Enabled:	N
Host IP address:	RED	Remote Host/IP:	
Local Subnet:	192.168.1.0/255.255.255.0	Remote subnet:	
Local ID: 🖜 (e.g.: @xy.example.com)	Accession of the second se	Remote ID: 单	
Dead Peer Detection action:	clear 💌 ?		
Remark: 单			
Authentication:			
Use a Pre-Shared Key:	F	ulu'illy:(#+	

Figure 41. Setting up a RoadWarrior connection (Support, 2007).

For the authentication section, we used a pre-shared key which required the client to be configured with the same key. This was ok since not a lot of users connected remotely to the green zone network using VPN and so there was no need to create a host certificate like in big companies where there are hundreds of users and where security can be compromised by the use of single paraphrase key.

In the advanced tab (Figure 42) we chose the type of encryption used by IPSec to preserve the confidentiality and integrity of transported packets. IPCop supported both IKE (Internet Key Exchange) and ESP (Encapsulation Security Payload) encryption and integrity modes. According to Wikipedia, IKE is a protocol used to setup security associations between communicating parties, and the ESP protocol is used to provide origin authentication, integrity, and confidentiality (Wikipedia, 2008).

According to Juniper Networks, we had to be careful not to use "Aggressive Mode", because even though it interchanges fewer messages between parties (three instead of six), it exposes Pre-Shared key in clear text (Juniper Networks, 2005). We accepted the default options as these algorithms provided enough protection.



Figure 42. Advanced settings for the RoadWarrior connection (Support, 2007).

To verify that our VPN RoadWarrior connection worked, we connected to our UTM machine remotely (Figure 43) from the internet using the GreenBow VPN client, which we downloaded from "www.thegreenbow.com".

Name	Туре	Common Name		Remark	Status	Action
FirstVpn	Host (PSK)				OPEN 🖑	🗹 🖉 🚺
Legend:	🗹 Enabled (click to disable)	📵 Show Certificate	🥖 Edit	🛱 Remove		
	Disabled (click to enable)	📊 Download Certificate	<b>ប័</b> Restart			

Figure 43. Connection status of the RoadWarrior connection (Support, 2007).

To confirm that the remote user did connect to the SOHO Green zone successfully, we looked for the status of the RoadWarrior connection (named earlier as FirstVPN) which showed an "OPEN" status as we desired. To do a final check we went to the network status menu and search in the current routing entry table (Figure 44) for the address of the remote desktop user which appeared in the table as connected to the green zone using IPSec. This concluded the final configuration for our UTM appliance.

Kernel IP rout	ing table						
Destination	Gateway	Genmask	Flags	Metric	Ref	Use Iface	
192.168.1.64		255.255.255.255	UGH	0	0	() ipsec0	
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0 eth0	
en la constante de la constante	\$0.0.0.0	255.255.252.0	U	0	0	0 ethl	
	0.0.0.0	255.255.252.0	U	0	0	① ipsec0     〉	
0.0.0.0		0.0.0.0	UG	0	0	0 ethl	

Figure 44. VPN session routing shown in the Network Status (Support, 2007).

#### **UTM implementation Tests**

For about three weeks, from February 17<sup>th</sup> to March 4<sup>th</sup> we let our UTM appliance sit out in the open for everyone in the Internet to attack us 24 hours a day in order to test basic features of our UTM appliance implementation. We tried to log every firewall hit and any intrusion attempt with our Snort sensor. During the first few days, we didn't see many things. However, as days progressed, we began to see impressive things happening every day, non-stop. One of these days was February 23<sup>rd</sup> when we decided to turn on the Red zone IDS sensor and see the sort of attacks we received rather than look at tons of hits appearing on the firewall log. We did not turn on the IDS sensor on the Red zone earlier because our main focus was to detect attacks activated in the Green zone first.

We left the UTM appliance on since about 9:15 pm on February 23<sup>rd</sup>, activated the Snort sensor on the Red zone network card. We then deactivated the Green zone sensor because we wanted to see attacks performed against our external IP address. Our internal sensor has not seen any external attacks yet, however, we did see a lot of firewall hits mostly

from China (Figure 45), Korea, and Russia. There was even one from Chile in South America.

Total of number	r of Intrusion rules act	ivated for Febr	uary 23: 21	
	Older		Newer	
Date:	02/23 01:12:54	Name:	MS-SQL Worm propagation attempt	
Priority:	2	Type:	Misc Attack	
IP info:	<u>61.132.223.14</u> :2885 ->	24.14.30.20:143	4	
References:	none found	SID:	<u>2003</u>	
Date:	02/23 01:12:54	Name:	MS-SQL version overflow attempt	
Priority:	3	Type:	Misc activity	
IP info:	<u>61.132.223.14</u> :2885 ->	24.14.30.20;143	4	
References:	none found	SID:	<u>2050</u>	
Date:	02/23 02:06:45	Name:	MS-SQL Worm propagation attempt	
Priority:	2	Type:	Misc Attack	
IP info:	<u>58.242.42.235</u> :1108 ->	24.14.30.20:143	4	
References:	none found	SID:	<u>2003</u>	
Date:	02/23 02:06:45	Name:	MS-SQL version overflow attempt	
Priority:	3	Type:	Mice activity	

Figure 45. Chinese attacker address detail from firewall logs (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

For the particular address/port entry of 58.242.42.235:1108 (Figure 46), I consulted the FAQs in the APNIC website, which handles addressing assignments for Asia about how to stop these addresses from hitting my network. According to APNIC, most of they time they only have details of the networks used but not about individual users because hackers constantly change IP addresses and APNIC do not have the legal ability to track down these individuals (APNIC, 2006).

58.242.42.235 (Reverse lookup failed) : whois.apnic.net					
% [whois.apni % Whois data	c.net node-2] copyright terms http://www.apnic.net/db/dbcopyright.html				
inetnum:	58.242.0.0 - 58.243.255.255				
netname:	CNCGROUP-AH				
descr:	CNC Group AnHui province network				
descr:	China Network Communications Group Corporation				
descr:	No.156,Fu-Xing-Men-Nei Street,				
descr:	Beijing 100031				
country:	CN				
admin-c:	CH455-AP				
tech-c:	CH455-AP				
remarks:	service provider				
mnt-by:	APNIC-HM				
mnt-lower:	MAINT-CNCGROUP				
mnt-lower:	MAINT-CNCGROUP-AH				
mnt-routes:	MAINT-CNCGROUP-RR				
status:	ALLOCATED PORTABLE				
changed:	hm-changed@apnic.net 20050603				
changed:	hm-changed@apnic.net 20050615				
changed:	hm-changed@apnic.net 20070301				
source:	APNIC				

Figure 46. Whois reverse lookup detail for an attack originated from China (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

So far we know we have been hit over and over again from 9:15 pm on February 23rd to about 3:15 pm on February 24th with a total of 688 firewall hits, and these are just the counted hits in a sixteen hour lap. We have had previous days where the UTM appliance logged attackers addresses for a small amount of firewall hits but this is actually the longest day we have left the UTM machine on.

The total number of intrusion rules activated just for this time frame from yesterday till today were twenty one; all external threats, half were MS-SQL version overflow attempts and the other half were MS-SQL worm propagation attempts as showed in Figure 47. The attacker or attackers seemed to be using the same IP address to perform each of the previous mentioned attacks repeatedly against IP address and to a specific port number or application within our internal network hoping to get a chance to penetrate any of our Green zone computers that are vulnerable to MS-SQL attacks.

For instance, the first intrusion rule appearing on the IDS log showed the name of the intrusion rule, that is, MS-SQL worm propagation attempt. According to Snort, this worm exploits a buffer overflow in the resolution service from MS SQL Server 2000, also compromising other machines very fast (Snort, 2007).

Total of number of Intrusion rules activated for February 23: 21					
	Older		Newer		
Date:	02/23 01:12:54	Name:	MS-SQL Worm propagation attempt		
Priority:	2	Type:	Misc Attack		
IP info:	<u>61.132.223.14</u> :2885	-> <u>24.14.30.20</u> :143	4		
References:	none found	SID:	<u>2003</u>		

Figure 47. IDS Name of Attack (Snort, 2007).

It was activated today at 01:12:54, which was very early we might say. It had a priority of 2 ("not too bad") according to the Administrative Guide (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

The IP information indicates the IP address used by the attacker was 61.132.223.14:2885 trying to access my IP address marked by an arrow. The type of attack was a "Miscellaneous" attack (i.e. no exact type has been designated by the VRT Snort team) as seen in Figure 48.

Total of numbe	r of Intrusion rules act	ivated for Feb	ruary 23: 21
	Older		Newer
Date:	02/23 01:12:54	Name:	MS-SQL Worm propagation attempt
Priority:	2	Type:	Misc Attack
IP info:	<u>61.132.223.14</u> :2885 ->	24.14.30.20:143	4
References:	none found	SID:	2003
Date:	02/23 01:12:54	Name:	MS-SQL version overflow attempt
Priority:	3	Type:	Misc activity
IP info:	<u>61.132.223.14</u> :2885 -> ,	24.14.30.20;143	4
References:	none found	SID:	<u>2050</u>
Date:	02/23 02:06:45	Name:	MS-SQL Worm propagation attempt
Priority:	2	Type:	Misc Attack
IP info:	<u>58.242.42.235</u> :1108 -> .	24.14.30.20:143	4
References:	none found	SID:	<u>2003</u>

Figure 48. IDS Attack detail (Snort, 2007).

Nevertheless, we did get an SID (Snort ID number) of 2003, and if we clicked on the SID (Figure 49) the Snort website popped up allowing us to identify the particular pattern of this attack from the Snort intrusion signatures database (Snort, 2007).

T.ORG	00	YOU	source ire
SIGNATURE DA	TABASE		
	By SID:	search	
	By Message:	search	
GEN:SID	1:2003		
Message Summary	MS-SQL Worm propagation attem This event is generated when an Server.	pt attempt is made by the "Slammer" worr	m to compromise a Microsoft SQL
Impact	A worm targeting a vulnerability i Service was released on Januar exploit a buffer overflow in the R nature of the vulnerability, the wo other machines very rapidly.	n the MS SQL Server 2000 Resolution y 25th, 2003. The worm attempts to esolution Service. Because of the orm is able to attempt to compromise	
Detailed Information	The Monitor Service provided by provided data in an SQL version	MS SQL and MSDE uses unchecked cli check function.	ent
	The worm attempts to evoloit a h	Iffer overflow in this version request	

Figure 49. VRT attack signature database detail (Snort, 2007).

According to the Snort signature database web page, this attempt was made by the "Slammer" worm to compromise a Microsoft SQL server. The impact from this worm exploits a buffer overflow in the resolution service and it will also compromise other machines as well. The signature database goes on providing more detailed information, what systems are affected, attack scenarios (e.g. a worm), ease of attack, and corrective action (Figure 50).

viewer	📔 🐡 Snort - the de facto standard for 🔯
	other machines very rapidly.
Detailed Information	The Monitor Service provided by MS SQL and MSDE uses unchecked client provided data in an SQL version check function.
Affected Systems	The worm attempts to exploit a buffer overflow in this version request. If the worm sends too many bytes in the request that triggers the version check, then a buffer overflow condition is triggered resulting in a potential compromise of the SQL Server. This vulnerability is present in unpatched MS SQL Servers. The following unpatched services containing MS
	SQL or Microsoft Desktop Engine (MSDE) may potentially be compromised by this worm: * SQL Server 2000 (Developer, Standard, and Enterprise Editions) * Visual Studio .NET (Architect, Developer, and Professional Editions) * ASP.NET Web Matrix Tool * Office XP Developer Edition * MSDN Universal and Enterprise subscriptions
Attack Scenarios	This is worm activity.
Ease of Attack	Exploits for this vulnerability have been publicly published.
	A worm has been written that automatically exploits this vulnerability.
False Positives	None known. If you think this rule has a false positives, please <u>help</u> fill it out.



The most common intrusion attempts logged were MS-SQL version overflow attempts (SID 2050), ICMP ping Cyberkit 2.2 for windows, DNS Spoof query response with TTL of 1 min (SID 254), and the Worm propagation attempt previously discussed.

Date:	02/17 15:51:46	Name:	(portscan) TCP Portsweep	
Priority:	n/a	Туре:	n/a	
IP info:	<u>192.168.1.30</u> :n/a -> <u>;</u>	<u>128.242.125.9</u> :n/a	]	
References:	none found	SID:	n/a	
Dates	02/17 15:51:46	Name	(portscap) TCB Portswaap	

Figure 51. Possible Spyware trying to communicate with external address (Snort, 2007).

There were two or three days in the beginning of our UTM implementation where we found Spyware communicating or trying to attack other machines (Figure 51). This happened around February 17<sup>th</sup> and 18<sup>th</sup> right after getting IPCop to work on our UTM appliance. It could have bot trying to communicate to its mothership. We looked on the Snort sensor and we had noticed that a Double Decoding attack, a TCP Port Sweep, and a UDP Port Sweep surged from one of the workstations to an external address on the internet. We solved the problem by running the most recent scans for Spybot and Adaware on each workstation. Later on, we tried to not find traces of the Spyware trying to communicate with external addresses, but we could not find any IDS logs related to those incidents.

#### **Firewall Test**

To test the SOHO firewall configuration, we used the windows version of Nmap, called Zenmap, which was used to perform different port scan attacks so as to find any unfiltered ports within the IPCop firewall. This test helped us to deduce whether or not IPCop secured the internal network from anyone on the internet trying to find holes on the network perimeter. The first Nmap scans (Figure 52) we performed were regular scans on the red zone with no ports configured to allow external access from the internet.

Zenmap provided a GUI that made it easier to use Nmap on a windows machine. The GUI supported the same type of options as did the UNIX version.

The results from the regular scans on the UTM appliance showed that 1715 ports were scanned.

👁 Zenmap
Sc <u>a</u> n <u>T</u> ools <u>P</u> rofile <u>H</u> elp
New Scan Command Wizard Save Scan Open Scan Report a bug
untitled_scan1 🗙 Operating System Detection on Regular Scan on Karakara X
Target:  Profile: Regular Scan
Command: nmap -v -PN (
Hosts Services Ports / Hosts Nmap Output Host Details Scan Details
OS   Host W Starting Nmap 4.60 ( <u>http://insecure.org</u> ) at 2008-04- Initiating Parallel DNS resolution of 1 host. at 22:02 Completed Parallel DNS resolution of 1 host. at 22:02, Initiating SYN Stealth Scan at 22:02 Scanning (Starting Manap 4.60) ( <u>http://insecure.org</u> ) at 2008-04- Initiating Parallel DNS resolution of 1 host. at 22:02, Initiating SYN Stealth Scan at 22:02 Scanning (Starting Manap 4.60) ( <u>http://insecure.org</u> ) at 2008-04- Initiating Parallel DNS resolution of 1 host. at 22:02, Initiating SYN Stealth Scan at 22:02 Scanning (Starting Manap 4.60) ( <u>http://insecure.org</u> ) at 2008-04- Initiating Parallel DNS resolution of 1 host. at 22:02, Initiating SYN Stealth Scan at 22:02 Scanning (Starting Manap 4.60) ( <u>http://insecure.org</u> ) at 2008-04- Initiating Parallel DNS resolution of 1 host. at 22:02, Initiating SYN Stealth Scan at 22:04 (Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating SYN Stealth Scan at 22:04, <u>139.09</u> s elapsed ( Host Initiating

Figure 52. Nmap Output screen details (Insecure.org, 2000).

In the scan details page (Figure 53) for Zenmap, we could also see what ports were open, the number of ports filtered (exactly the same as the total number of ports) in IPCop, and the protocol used to scan for services if a particular port was open.

Ports / Hosts Nmap C	output Host Details Scan Details						
Command Info	Command Info						
Command:	nmap -v -PN						
Nmap Version:	4.60						
Verbosity level:	1						
Debug level:	0						
🗉 General Info							
Started on:	April 15, 2008 - 22:02						
Finished on:	April 15, 2008 - 22:04						
Hosts up:	1						
Hosts down:	0						
Hosts scanned:	1						
Open ports:	0						
Filtered ports:	1715						
Closed ports:	0						
🗆 Scan Info - Syn							

Figure 53. Zenmap regular scan details from our UTM configuration (Insecure.org, 2000).

The results from this screen were very good because this was exactly what we wanted for our UTM machine to do. We wanted to test if all ports were closed to outsiders as required and we accomplished this goal.

Using the same configuration we then performed intensive scans and obtained just about the same results: no ports were open. We later performed quick scans and obtained similar results. When we performed any scan, we had to set the stealth mode, as IPCop had automatically blocked any ping response, and so by using this method we were able to bypass this defense. So far, these first port scans have showed that when no ports are left open by the SOHO administrator, our UTM appliance did its job in preventing any intrusion to any port, regardless of what type of port scan attack was being used. When we scanned the UTM appliance from within the Green zone, however, we were able to detect 11 open ports (Figure 54).

👁 Zenmap							
Sc <u>a</u> n <u>T</u> ools <u>P</u> rofile <u>H</u> elp							
New Scan Command Wizard Sa	ave Sca	in Open	Scan Rep	੍ਹੇ ort a bug	<b>O</b> Help	\$	
Intense Scan on 🗰 🙀 🗶 🗙							
Target:			Y Pro	file: Inten	se Scan		Scan
Command: nmap -T Aggressive	e -A -v						
Hosts Services	Ports /	Hosts	Mmap Output	Host Details	Scan Details	1	
	•	Port 4	Protocol 4	State 4	Service 4	Version	•
An antipatric	۲	21	tcp	open	ftp-proxy	Frox ftp proxy	
0.50	۲	25	tcp	open	tcpwrapped		
	۲	53	tcp	open	domain	dnsmasq	
	۲	80	tcp	open	http-proxy	Squid webproxy	
	۲	81	tcp	open	http	Apache httpd	
	۲	110	tcp	open	tcpwrapped		
	۲	222	tcp	open	ssh	OpenSSH	
	۲	446	tcp	open	ssl	OpenSSL	
	۲	800	tcp	open	http-proxy	Squid webproxy	
	۲	995	tcp	open	tcpwrapped		
	۲	2121	tcp	open	tcpwrapped		

Figure 54. Detected ports from within Green zone by Zenmap (Insecure.org, 2000).

Zenmap detected ports 21, 25, 53, 80, 81, 110, 222, 446, 800, 995, 2121. These ports, however, allowed access from the internal network and going into the internet because the internal workstations used services on these "opened" ports to request data from the outside world. As a result, IPCop left these ports open from within the Green zone.

On the other hand, we did leave 3 ports open (25, 110, and 80) for external use on the red zone due to port forwarding to these services and ran the scan one more time, but then Zenmap was able to find the state of each port, the service the port ran, the MAC address of the computer hosting the service, and even the operating system (with a pretty good

guess) of the workstation that hosted the service (Figure 55). We were alarmed by these results.



Figure 55. Host scan details from Zenmap on the Red zone with ports open (Insecure.org, 2000).

If we were to leave these services (i.e. web/smtp/pop3) open on the green zone like they were, it would not be safe, because any Zenmap user could detect information just like in this scan and anybody could hack into the SOHO internal network. However, since we are just testing how IPCop ports are filtered, this was a normal result for this trial. In other words, it is not how we typically ran our network.

These results had showed us that anybody with tools freely available like Nmap and Zenmap could hack into just about any open port in the UTM appliance. Therefore, security precautions must be taken before allowing outsiders the use of these internal services.

#### **IDS Test**

In order to test the efficiency of the Snort IDS system within our UTM box, we had used Nessus, which is a vulnerability scanner with thousands of plug-ins capable of testing just about any type of vulnerability on a target system. From the start, our IDS system had successfully logged intrusion attempts. With the information provided by the Snort vulnerability database for each IDS entry, we were able to find whether the attack had entered the perimeter, and, if so what could be done to prevent further incidents.

These series of IDS tests had also showed what sorts of attacks the IDS system detected, the weaknesses the attack looked for, the severity of the attack, and more information related to the attack pictured in every IDS intrusion entry (Figure 56).

HAVP -	Server Down	🔄 🤤 IPCop - IDS log viewer 🛛 🖾	
	Total of number of I	ntrusion rules activated for April 15: 69 <u>Older</u>	
	Date: Priority: IP info: References:	04/15 22:03:59 2 <b>100 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 </b>	
	Date: Priority: IP info: References:	04/15 22:14:03 2 <del>907 - 14 - 15 - 15 - 15 - 15 - 15 - 15 - 15</del>	
	Date: Priority: IP info: References:	04/15 22:14:03 2 none found	

Figure 56. Snort IDS intrusion entry detail (Snort, 2007).

The IDS sensor was used to test the Green and Red zone. We first tested the Red zone by running Nessus directly from the internet and scanning our UTM appliance externally in search for vulnerabilities using the "Enable all plug-ins but dangerous" option (Figure 57).



Figure 57. Scanning our UTM machine for vulnerabilities using Nessus (Deraison, 2002).

Since we did not have any open ports for this test, Nessus could not find any vulnerabilities whatsoever. Just like in the firewall test when we did not find any open ports after running the intensive scan, this Nessus scan also created 1715 IDS entries on the IPCop IDS log for all of the attacks performed upon the red interface. This confirmed that if we did not open any ports, IPCop could successfully prevent any hardcore attack from Nessus.

In the second test on the Red zone, we decided to open some ports (e.g. 25, 110, and 80) providing them with external access and then ran Nessus again against the UTM appliance. This second time Nessuss was able to detect the previous opened service ports, as well as information that could exploit each service (Figure 58).

Start Time:	Mon Apr 14 21:00:33 2008
	3 Open Ports, 15 Notes, 0 Warnings
pop3 (110/tcp) 💈	Port is open Plugin ID : <u>11219</u>
8	A pop3 server is running on this port Plugin ID : <u>10330</u>
8	Synopsis :
	A POP server is listening on the remote port

Figure 58.	Nessus report of vulnerabilities found on the Red zone port forwarding open
	(Deraison, 2002).

The bad thing about these "open" ports was that they resided on the green zone allowing any one on the internet with malicious intent to be able to see these vulnerabilities and exploit the services accessed through these ports. The solution for these found vulnerabilities by the Nessus scanner simply required that these services had to either be put in the orange zone or else closed these ports so that no further damage could be allowed to penetrate into the internal network.

Furthermore, when we performed a Nessus scan on the UTM appliance from within the Green zone, we were able to see that 1593 alerts had been raised by the Snort sensor (Figure 59).

IP	Logs	IDS Logs	+						
<sup>1</sup> COP 1.4.18	SYSTEM & STATUS	* NETWORK *	SERVICES *	FIREWALL &	VPNs				
	-								
	Settings:								
			Ма	onth: April	~				
	Log:								
	Total of number of Intrusion rules activated for April 16								
	Date: Priority: IP info: References:	04/16 2 2 none fo	2:07:53						

Figure 59. Snort alerts found when running Nessus against the Green zone interface (Snort, 2007).

This informed us that the IDS sensor worked just liked the IDS sensor on the Red zone and it raised 1659 alerts.

The vulnerability report generated from the Green zone Snort sensor, however, showed different results (Figure 60). The services that Nessus found (similar to Zenmap) were pop3(port 995), Amanda (port 10080, proxy with secure authentication), mdbs\_daemon (port 800), SSH (port 222, this could be exploit with x.11 session hijacking, and with medium risk factor), http (port 80), domain (port 53, UDP, vulnerable to cache snooping attacks, CVE-1999-0024), SMTP (port 25), FTP (port 21), and finally detected IPCop's operating system (Linux 2.6 Kernel).

Start Time:

Wed Apr 16 21:38:06 2008

12 Open Ports, 42 Notes, 5 Warning

pop3s (995/tcp) **@**Port is open Plugin ID : <u>11219</u>

It might be protected by some TCP wrapper

Plugin ID : 10330

amanda (10080/tcp)

Port is open Plugin ID : <u>11219</u>

Figure 60. Nessus report of vulnerabilities found from scan within the Green zone (Deraison, 2002).

#### **Anti-Spam and Anti-Malware Test**

To test the anti-spam and anti-malware filter mechanisms put in place on the internal SOHO network by UTM appliance, we first used the spam, virus, and spam-attachment test emails provided in Copfilter's Test section (Figure 61).



Figure 61. Copfilter virus, spam, and spam-attachment test emails (Madlener, 2002).

Here we could see how the SpamAssassin spam filter and the Antivirus filter had quarantined all three emails by sending notification emails for all three emails rather than sending the original virus or spam emails to the email server. Also, we could see these quarantined virus and spam emails being placed in the quarantine bin in IPCop (Figure 62).

Co	pfil	te	r log f	iles:						
Snan										
Span	n Quar	antir	ne - POP3 -	/var/log/co	pfilter/default/opt/p3scan/quara					
0	del		send	add	From:					
del	send	add	admin@iov;	From:	To:					
Span	n Quar	antir	e - SMTP -	/var/log/co	pfilter/default/opt/proxsmtp/qua					
			admin@joya	ayoro.com	admin@joyayoro.com					
			jagent@rou	te.monster.con	n rbpi18@yahoo.com					
			ExtraIncom	e@essexite.co	m rbpi18@yahoo.com					
			CecilSarger	nt63@yahoo.co	.uk CecilSargent63@yahoo.co.uk					
			CecilSarger	nt63@yahoo.co	.uk CecilSargent63@yahoo.co.uk					
			CecilSarger	nt63@yahoo.co	.uk CecilSargent63@yahoo.co.uk					
			emailsecuri	ty@byteplant.c	om admin@joyayoro.com					
			emailsecuri	ty@byteplant.c	om admin@joyayoro.com					
			emailsecuri	tv@byteplant.c	om admin@iovavoro.com					
				mailsecurity@byteplant.com admin@joyayoro.com						

Figure 62. Quarantine virus and spam emails in the quarantine bin (Madlener, 2002).

Although this seemed acceptable, we wanted to test our UTM appliance with real virus and spam emails to see if these could penetrate our UTM perimeter defenses. For this reason, we went to www.nospamtoday.com, a portal that provided an automated malware and spam security test suite which we used to test our anti-malware and anti-spam filters. This automated system had sent seven different emails containing one Eicar file (a viruslike file filtered by ClamAV), a GTUBE spam signature (filtered by SpamAssassin), an executable file (a malware-like file filtered by the SMTP filter), and 4 spam different emails containing disguised attachments (filtered by SpamAssassin in conjunction with ClamAV and the SMTP filter).

We reviewed the seven emails (Figure 63) in search of signs of UTM perimeter penetration, but all seven emails were SpamAssassin and ClamAV notifications from Copfilter of the real emails that were blocked, renamed, and classified as spam or virus infected emails. Copfilter recognized the virus files, attachments, or spam signatures contained in them.

🧐 Inbox - Outlook Express									
Eile Edit <u>V</u> iew <u>T</u> ools <u>M</u> essa	ge <u>H</u> elp					Ð			
	<u>40</u>		包		<b>S</b>				
Create Mail Reply Reply Al	Forward	Print Delete	Send/Recv	Addresses	Find				
🕸 Inbox									
Folders ×	!0 ?	From	Subject			Received 🗸 🔺			
🞲 Outlook Express	Ø	🖾 E-Mail Security Test	Test mail	5/7 (ID=0c5	Z!qEIa0imkWtT7v	/JctQ) 4/19/2008 2:—			
🖻 🇐 Local Folders	0	🖼 E-Mail Security Test	Test mail	6/7 (ID=0c5	Z!qEIa0imkWtT7v	/JctQ) 4/19/2008 2:			
- 🙀 Inbox (11)	0	E-Mail Security Test	*** renar	ned attachm	ent *** Test mai	l 4/7 ( 4/19/2008 2:			
	0	E-Mail Security Test	**** renar	ned attachm	ent *** Test mai	1/7 ( 4/19/2008 2:			
🖓 Sent Items		Madmin@joyayoro.co	n **** VIRUS	5: Eicar-Test-	Signature in SMT	P **** 4/19/2008 2:			
🚽 🗑 Deleted Items (4)		Madmin@joyayoro.com	n <u>Yeur</u> regi	Elear-Test-Sig	hature in SMIP and a	COPY - Test mail 2/7 (ID=0052)			
In the second se			**** urouc.	Ciación for ti		III Secu 4/19/2000 2: Tb 4/10/2000 0.45			
	•					•			
	From: E-Mail Security Test       To: admin@joyayoro.com         Subject: **** renamed attachment **** Test mail 1/7 (ID=0c5Z!qEIa0imkWtT7vJctQ)								
	You receive this mail because you registered for the NoSpamToday! email security check.								
	This mail contains a harmless executable attachment named "attached.bat".								
Contacts ▼ X Data Barrera	Even though it is harmless, it should have been removed (or replaced) by your attachment blocker.								
	Find out i	more here on how to p	rotect yourse	lt against un	wanted email att	achments:			
	http://ww	w.nospamtoday.com							
						v			
59 message(s), 11 unread	,			📃 Workin	ig Online				

Figure 63. Seven emails from nospamtoday's automated system (No Spam Today, 2003).

This was good news because our SOHO UTM appliance was ready to protect the internal mail server and workstations clients from any virus-infected emails, emails with other malware attachments in them, and above all from SPAM (Figure 64).

🚔 **** VIRU	JS: Eicar-T	est-Signatu	ire in SMTF	o ∗∗∗∗ Test r	nail 2/7 (ID	=0c5Z!qEI	a0imkWtT7vJctQ)			
<u> </u>	: <u>V</u> iew j	<u>F</u> ools <u>M</u> ess	age <u>H</u> elp							
<u>Q</u>	<b>⊈</b> √ <u>√2</u> <b>→</b> <del>→</del> <del>→</del> <del>↓</del> <b>↓</b>									
Reply	Reply All Forward Print Delete Previous Next Addresses									
From:       admin@joyayoro.com         Date:       Saturday, April 19, 2008 2:31 PM         To:       admin@joyayoro.com         Subject:       *** VIRUS: Eicar-Test-Signature in SMTP *** Test mail 2/7 (ID=0c5Z!qEIa0imkWtT7vJctQ)										
.=.=.=.										
Copfilter Instead o Virus nar	Copfilter detected a VIRUS in an email sent to you (SMTP)! Instead of the infected email this message has been delivered to you.									
Attachm	ent: eicar.:	zip								
Sender**: <u>emailsecurity@byteplant.com</u> Recipients: <u>admin@joyayoro.com</u> Subject: Test mail 2/7 (ID=0c5Z!qEIa0imkWtT7vJctQ) Date: Sat, 19 Apr 2008 21:31:18 +0200 Server: 67.165.170.5 Client: 78.47.119.33 Email File: proxsmtpd.3qY4hn Scanned on: ipcop1										

# Figure 64. Copy of virus email test sent from nospamtoday and filtered by the SMTP filter (No Spam Today, 2003).

For our last test, we went to the EICAR web site (www.eicar.org) to test for malware coming though HTTP or FTP traffic. There were four types of files that we downloaded through the HTTP protocol: eicar.com, eicar.com.txt, eicar.com.zip, and eicarcom2.zip (Figure 65). According to Eicar, most anti-virus products should detect all these test files as viruses (Eicar, 2006).We also tested the same four Eicar files using the FTP protocol.

🅘 ei	car   1	THE AN	TI-VIRUS	OR ANTI-	MALWARE	RE TEST FILE - Mozilla Firefox	
Eile	<u>E</u> dit	⊻iew	Hi <u>s</u> tory	<u>B</u> ookmarl	ks <u>T</u> ools	s <u>H</u> elp	
	•	> -	<b>e</b> 🛛		eie http://	://www.eicar.org/anti_virus_test_file.htm#	
🌮 Getting Started 🔝 Latest Headlines 🥥 IPCop - Main page 🐄 joyayoro.com 🔉 American Registry fo )							
0	IPCop	- Copfilt	er		eic eicar	r   THE ANTI-VIRU 😰 🛛 🤛 EICAR Test Files - Trend M	

	*					EUROPEAN	
е		ar		EIC	CAR ONLINE-UNIVERSITY	EICAR-FORU	
AB	OUT US	CONFE	RENCE	TASK FORCES	PRESS	cc	
			Downloa	ad area using	the standard protoc	col http	
	<u>eicar.com</u> 68 Bytes	<u>eicar.com.txt</u> 68 Bytes	<u>eicar_com.zip</u> 184 Bytes	<u>eicarcom2.zip</u> 308 Bytes			
		C	)ownload area	a using the se	cure, SSL enabled	protocol https	
	(Note: For the time being we make use of a self-signed certificate. You may be asked by your browser whet Depending on acceptance of this new service we may install a certificate coming from a tru Certificate Authority at a later point in time.)						
	eicar.com	eicar.com.txt	eicar_com.zip	eicarcom2.zip			
Done							

Figure 65. Eicar malware test files downloaded through HTTP and FTP method (Eicar, 2006).

For all Eicar files downloaded using the HTTP protocol, we had found that they were all successfully blocked by the HTTP filter (Figure 66).



Figure 66. HTTP filter blocking a malware Eicar file (Madlener, 2002).

The four Eicar files downloaded through FTP were also filtered by the FTP filter that denied access automatically to each Eicar file without a warning message (and it just froze the download) because this was the only option. The only bad thing missing about these tests and about Copfilter is that we had also tried to download these same four Eicar files using HTTPS protocol links, but unfortunately all four Eicar files were able to bypass the HTTP filter. The HTTP filter blocked malware coming through port 80 and not port 443 (SSL port for HTTPS). It seemed that we needed to focus on other add-ons that supported HTTPS protocol downloads like Dansguardian (a true web content filter which runs on Linux).

#### VPN Test

For our VPN Test, we connected to the SOHO internal network with our remote laptop using our previously configured RoadWarrior VPN connection together with the GreenBow VPN client software. To test the efficiency of the secure communications mechanism used between the VPN client and IPCop, we decided to use WireShark. WireShark is a network probe that allowed us to capture individual TCP and UDP packets to verify that indeed the AES 128-bit encryption algorithm was encrypting data for every packet inspected by our probe during the VPN session.

We initialized and opened our tunnel successfully from the internet and entered into the SOHO internal network (192.168.1.0). Once connected, we were able to ping the IPCop gateway (192.168.1.1) successfully as if we were hardwired internally to the Green zone. We had tried to access other workstations successfully and verified access to the IPCop's administration page from within the Green zone with equal success. Later, we stopped our probe and decided to verify the results from WireShark by looking at all of the packets, and specially the ESP packets. We were able to see that the GreenBow client successfully encrypted every data sent over the VPN tunnel by the remote user, as well as data received from IPCop (Figure 67).

<b>7</b> 1	Broadc	om Ne	tXtre	me G	igabi	t Eth	ernet	t Driv	er (M	Aicro	osoft	's Pa	icket	t Scl	nedul	er) : C	aptur	ing -
Eile	<u>E</u> dit	<u>V</u> iew	<u>G</u> o <u>G</u>	aptur	re <u>A</u> n	alyze	<u>S</u> ta	tistics	Help	)								
		M 🔐	۱			×	2	9	Q	\$			T	⊉			Ð	Q
Eilte	r:													•	<u>E</u> xpre	ssion	<u>C</u> lear	<u>A</u> ppl
No.	•	Time			Source						Desti	natior	ı				Protoc	ol
	163	496.	9332	54	192.	168.	15.1	100			192	.168	3.15	.10	)1		ESP	
	164	496.	9333(	03	192.3	168.	15.1	100			192	.168	3.15	.25	5		NBNS	
	165	497.	68198	80	192.	168.	15.1	100			192	.168	3.15	.10	)1		ESP	
	166	497.	6820:	33	192.	168.	15.1	100			192	.168	5.15	. 25	5		NBNS	
	169	498.	43201	50	102.	168.	15.1	100			102	160	5.13 2 1 5	. 10	1		NPNC	
	169	499.	20447	72	192.	168.	15.1	100			192	.168	3.15	10	)1		ESP	
	170	499.	20452	21	192.	168.	15.1	100			192	.168	3.15	. 25	5		NBNS	
	171	499.	94768	81	192.	168.	15.1	100			192	.168	3.15	.10	)1		ESP	
	172	499.	94773	32 :	192.3	168.	15.1	100			192	.168	8.15	.25	5		NBNS	
	173	500.	69767	73	192.3	168.	15.1	100			192	.168	3.15	.10	)1		ESP	
	174	500.	69772	23	192.3	168.	15.1	100			192	.168	3.15	. 25	5		NBNS	
	175	501.	4484	50	192.	168.	15.1	100			192	.168	3.15	.10	)1		ESP	
	176	501.	44848	86	192.	168.	15.1	100			192	.168	3.15	. 25	5		NBNS	
	170	502.	10778	50	102.	160	15.	100			102	160	5.ID 2 1 5	. 10	1		LOP	
	170	502	04771	10	102.	168	15 1	100			102	168	2 1 5	10	11		INDINO E S D	
	180	502.	94777	72	192.1	168.	15.1	100			192	.168	3.15	. 25	5		NBNS	
		175	(1.5.0	las at				1.50	las at				- 42					
	-rane	1/5	(150	byt	es o		ire,	130	Dyt	es (	Cap	-ta-	ea)					
E E	ther	net I	т, ы	rc:	Comp	arco	5_40	:a2:	00 0	(00)	10:	a4 :	40:8	a2:0	<i>00)</i> ,	DST:	D-L1	Ink_
Ξ	Inter	net P	roto	col,	Snc	: 19	92.1	68.1	5.10	00 (	(192	.16	8.1	5.1	00),	Dst:	192.	168
	Encap	sulat	ing :	5ecu	rity	Pay	/loa	d										
	ESP	SPI:	0x41	ff3d	ld9e													
	ESP	Sequ	ence	: 77														

Figure 67. Encrypted ESP packets as seen by the network probe Wireshark (Combs, 1998).

We also tried to open the RoadWarrior tunnel using incorrect credentials, but we were unable to initialize the VPN session.

#### Web Filter Test

For our web filter test, we decided to select ten content types to be blocked whenever internal users tried to access websites with the prohibited content. The first ten content types were Music, Chat, Warez, Spyware, Proxy, Violence, Guns, Videos, Games, and Adult. Once the undesired content types of websites had been selected, we enabled the filter by going to the Proxy tab, checked the "URL filter enabled" box and selected Save to start the web filter. Suddenly, we had encountered problems right when the URL filter was activated. All of the other services like IDS, virus and Spam filters, and other filters had already been activated previously and were working fine, but IPCop had frozen completely at that moment and could no longer run any service of any kind. After comparing the memory usage between using the URL filter alone vs. running it along with other services, we noticed that memory usage soared when the URL filter was used.

Apparently, the URL filter alone took about sixty percent above the out-of-the-box standard memory usage of twenty-five percent (with a total of 83 % of RAM used), and sometimes even reaching levels above ninety percent. This was not acceptable (Figure 68).

	Memory:	
	Size         Used         Free         Percent           RAM         192048         161064         30984           -/+ buffers/cache         63004         129044           Swap         32764         0         32764	entage 0 83% buffers 22880 32% cached 75180
	Disk usage:	
	DeviceMounted onSizeUsed/dev/root/3606M479M/dev/harddisk1 /boot16M4M/dev/harddisk2 /var/log10748M394M	Image: Tree Percentage           4 3091M           12M           25%           4 9809M           4%
Done	Inodes usage:	

Figure 68. URL filtering eating up available memory with no other active services (Sondermann, 2004).

This, however, did not include any other services available to run at the time of this test. We think this memory exhausting by the content filter was due to either a bug in the URL filter plug-in, or the memory requirements of the black lists used by the URL filter were so stringent that memory usage would have to increase so dramatically. Memory usage by the URL filter also increased if more content types were added to the black list as well. Sadly, we therefore had to abandon the URL filter test.

#### Conclusion

The results of testing clearly showed that the benefits of a custom UTM appliance definitely outweighed the costs and effort put into building it. Benefit was seen in all the areas, but especially in the anti-virus and anti-spam areas, where other UTM retail solutions and even individual enterprise point solutions have failed.

The test UTM solution was easy to install and easy to configure, but, above all, it was easy to manage. The cost of the UTM appliance ranged about one hundred and fifty dollars, which is a cost factor of 100 better than an equivalent UTM offered by companies like SonicWall, Fortinet, and Cisco ASA that cost more than twenty five hundred dollars or more. Unlike those solutions, most custom features are already integrated in our UTM appliance and best of all, our UTM system is 100 times less expensive!

Once configured, the UTM rarely needed an administrator to manage it (except when an add-on service went down) and it automatically updated itself for anti-virus signatures, anti-spam blacklists, and content filter blacklists without impacting the productivity of any of the internal systems that depended on these protections. Also, the UTM (through its various filters) successfully blocked the latest unified threats like malware, spam, penetration attempts, denial of service, spoofing, man-in-the-middle attacks, theft of proprietary information, unwanted web content, and much more.

In addition, the SOHO user did not have to worry about hiring security personnel or learning about security (although this could be useful) in order to protect the internal network, because the UTM appliance took care of all these security functions automatically. The UTM appliance provided the SOHO owner with an Intrusion Detection System that detected security incidents whenever they occurred and provided its best defensive effort to thwart those attacks. The appliance also provided handy tools to administer the UTM from any workstation within the LAN or from anywhere on the internet through secure SSL transmissions with peace of mind. VPN capability provided the remote laptop user with access to internal SOHO resources from any place on the internet using the best encryption algorithms, keeping data secure while traveling over unsecured routes.

Moreover, the UTM appliance accelerated internet browsing through the use of special proxy service that allowed client machines to have faster access to constantly visited pages, thus increasing job productivity. Statistical graphs and various log files were provided by the UTM appliance to oversee the performance of different services, or to enforce an acceptable use policy even when that policy required granular scrutiny over a user's daily activities.

The downside of the Unified Threat Management system, however, was a high utilization bottleneck. As more services were enabled on the UTM appliance, more system resources (i.e. memory, CPU, swap space) were needed to preserve UTM system stability. Our 192 MB RAM capacity was not sufficient enough to sustain all of the services to be tested at the same time. The low memory availability in the UTM appliance was also the primary reason the web content filter could not be tested. Instead, the minimum hardware requirements for the UTM appliance should be a 1.5 MHz CPU, 500 MB RAM, 250 KB cache, and enough hard disk space (e.g. 40 Gb or more disk space). The more memory there was the faster client machines accessed resources.

The next steps that a SOHO user can take with this project is to implement an Orange (DMZ) zone for the web, proxy, and email servers. Dynamic DNS can be enabled in IPCop to allow redirection for a hosted website in order to allow users to connect to the right IP address in the case this one changes. Tripwire can be installed on the web server to track for changes on the configuration files in the event someone penetrates the perimeter defenses and tries to deface the SOHO business website.

Another attractive option is to implement a Blue (wireless) zone using the previously explained VPN technology to provide secure (IPSec) access to internal resources. More antivirus filters like F-Prot and AVG can be added to Copfilter in order to provide security for future Linux clients, and additional security for already existing Windows clients.

There is no doubt that this extremely inexpensive SOHO UTM appliance (with better hardware) provides an enterprise-level UTM solution that surpasses SonicWall, SourceFire, and even Fortinet and Cisco ASA. Using open-source tools, we have developed a defense arsenal that offers tremendous security, as well as great peace of mind one at a price that is one one-hundreth of the cost of the commercial alternative.

#### References

Lawrence, G, Loeb, M, & Lycyshyn, W FBI/CSI 2006 Computer Crime and Security Survey. CSI/FBI, Retrieved February 13, 2008, from http://gocsi.com.

Walker, P, Goldschmitt, H, & Pielschmidt, S IPCop v1.4.0. IPCop, Retrieved January 22, 2008, from

https://sourceforge.net/project/showfiles.php?group\_id=40604.

Oberlander, Eric (2004). Quick Install Guide. IPCop, Retrieved January 18, 2008, from https://sourceforge.net/project/showfiles.php?group\_id=40604

Clancey, C, Goldschmitt, H, Kastner, J, & Oberlander, E (2004). Administrative Guide. *IPCop*, RetrievedJanuary 21, 2008, from https://sourceforge.net/project/showfiles.php?group\_id=40604

APNIC, (2006). Network Abuse FAQ series. Retrieved April 10, 2008, from APNIC Web site: <u>http://www.apnic.net/info/faq/abuse/index.html#2</u>

Snort, Sourcefire Vulnerability Research Team (2007). Sourcefire VRT Certified Rules. Retrieved March 12, 2008, from Snort Web site: <u>http://www.snort.org/vrt/advisories/vrt-rules-2007-01-09.html</u>

SonicWALL, (2005). Unified Threat Management: The Best Defense Against Blended Threats. *SonicWALL*, Retrieved February 16, 2008, from http://www.sonicwall.com/downloads/UTM\_WP.pdf

Wikipedia, (2008). Unified Threat Management. Retrieved March 1, 2008, from Wikipedia Web site: http://en.wikipedia.org/wiki/Unified\_Threat\_Management

SearchSecurity, (2006). Unified Threat Management Definition. Retrieved February 15, 2008, from SearchSecurity Web site:

http://searchsecurity.techtarget.com/dictionary/definition/what-is-unified-threatmanagement.html

Feingold, Richard (2005, March 7). SmartAdvice: Unified Management Is Next For Security. *InformationWeek*, Retrieved January 22, 2008, from http://www.informationweek.com/showArticle.jhtml?articleID=60405833

Juniper, Networks (2005). IKE Overview. Retrieved April 10, 2008, from Juniper Networks Web site: <u>http://www.juniper.net/techpubs/software/erx/erx51x/swconfig-routing-vol1/html/ipsec-config5.html</u>

Eicar, (2006). The Anti-Virus or Anti-Malware test file. Retrieved March 12, 2008, from Eicar Web site: <u>http://www.eicar.org/anti\_virus\_test\_file.htm</u>

Garcia, Andrew (2003, December, 15). A look at All-in-One Security Appliances. *eweek*, Retrieved March 12, 2008, from

http://www.eweek.com/c/a/Security/A-Look-at-AllinOne-Security-Appliances/

Frankin Jr, Curtis (2005, November, 10). All-in-One Security Appliances. *Network Computing*, Retrieved January 18, 2008, from <u>http://www.networkcomputing.com/172901783</u>

Messmer, E, & Hopfner, J (2007). All-in-one security devices face hurdles. *Computerworld*, Retrieved January 22, 2008, from

http://www.computerworld.com.my/PrinterFriendly.aspx?articleid=4624&pubid=4&issueid=106.

Fortinet, (2008). Fortinet Unified Threat Management Overview. Retrieved March 12, 2008, from Fortinet Web site: http://www.fortinet.com/products/fortigate\_overview.html

ChannelWeb, (2005). Unified Threat Management. *ChannelWeb*, Retrieved March 2, 2008, from

<u>http://www.crn.com/security/202404172;jsessionid=F4XSIENL0RZHSQSNDLP</u> <u>CKHSCJUNN2JVN?pgno=1</u>

Poeter, D, & Morejon, M (2007). Bake-Off: Unified Threat Management Appliances. *ChannelWeb*, Retrieved January 24, 2008, from

http://www.crn.com/article/printableArticle.jhtml;jsessionid=F4XSIENL0RZHSQ SNDLPCKHSCJUNN2JVN?articleId=202404182.

ByteandSwitch, Storage Network and Beyond (2004, Septerber 13). IDC: UTM Captures 12% of Market. Retrieved March 13, 2008, from ByteandSwitch Web site: http://www.byteandswitch.com/document.asp?doc\_id=60507

Web Applications Security Consortium, (2007). The Web Hacking Incidents Database. Web Applications Security Consortium, Retrieved March 20, 2008, from http://www.webappsec.org/projects/whid/statistics.shtml

Sondermann, Marco (2004). URL filter – The URL filter add-on. Retrieved March 11, 2008, from URL filter Web site: <u>http://www.urlfilter.net/</u>

Insecure.org, (2000). Zenmap. Retrieved March 20, 2008, from Insecure.org Web site: <u>http://nmap.org/</u>

Combs, Gerald (1998). Wireshark. Retrieved March 20, 2008, from Wireshark Web site: http://www.wireshark.org/about.html

Deraison, Renaud (2002). Nessus Security Scanner. Retrieved March 1, 2008, from Nessus Web site: <u>http://www.nessus.org/nessus/</u>

No Spam Today, Byteplant (2003). Free Spam Filter and Email Security Check. Retrieved April 4, 2008, from nospamtoday Web site: http://www.nospamtoday.com

Madlener, Markus (2002, November). Copfilter. Retrieved March 20, 2008, from Copfilter Web site: <u>http://www.copfilter.org/</u>

Prikryl, Martin (2000). WinSCP. Retrieved March 1, 2008, from WinSCP Web site: http://winscp.net/eng/download.php

Tatham, Simon (2006). Putty: A Free Telnet/SSH Client. Retrieved March 1, 2008, from PuTTY Web site: http://www.chiark.greenend.org.uk/~sgtatham/putty/

Support, TheGreenBow (2007). TheGreenBow IPSec VPN Client. *TheGreenBow*, Retrieved March 22, 2008, from <u>http://www.thegreenbow.com</u>

(This page was left blank on purpose)