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

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-in-the-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.

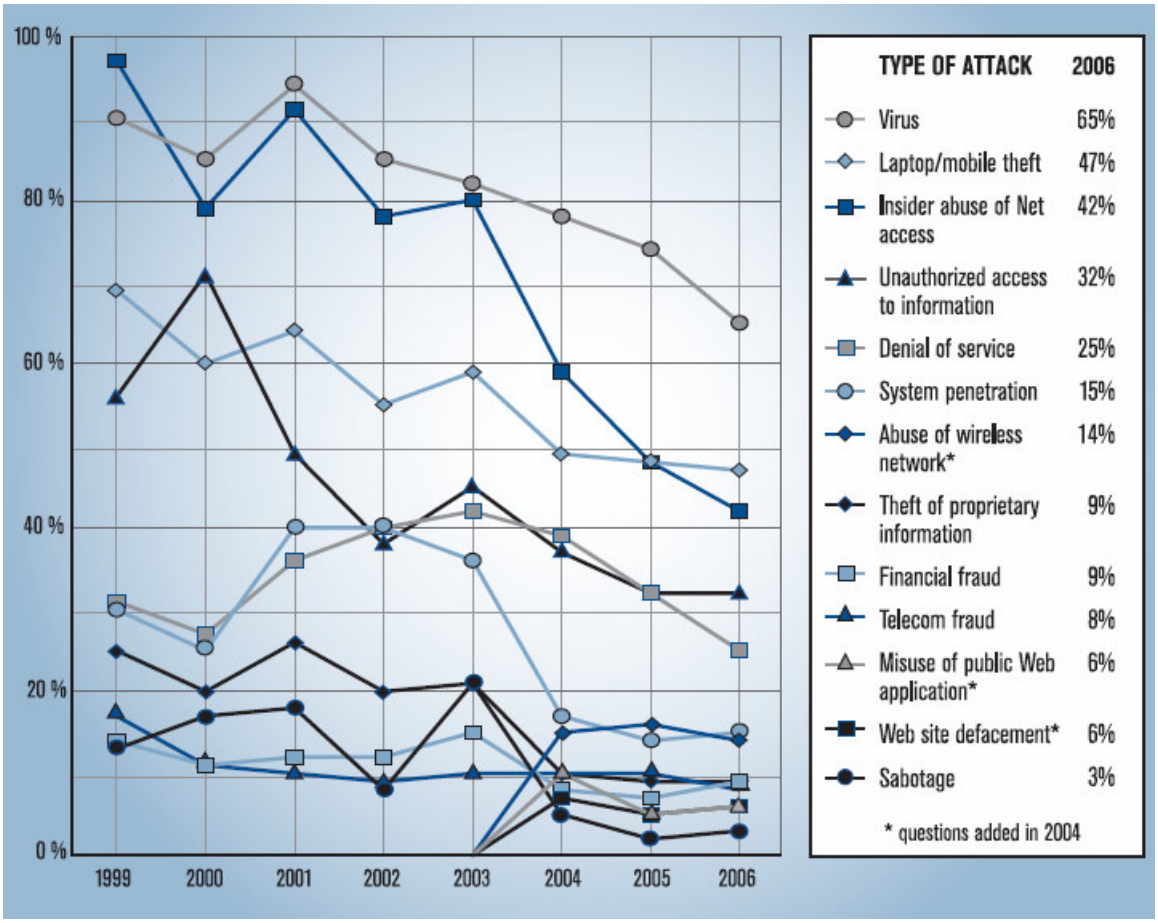


Figure 1. Common attacks for 2006 (Lawrence, Loeb, & Lycyshyn, 2006).

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.

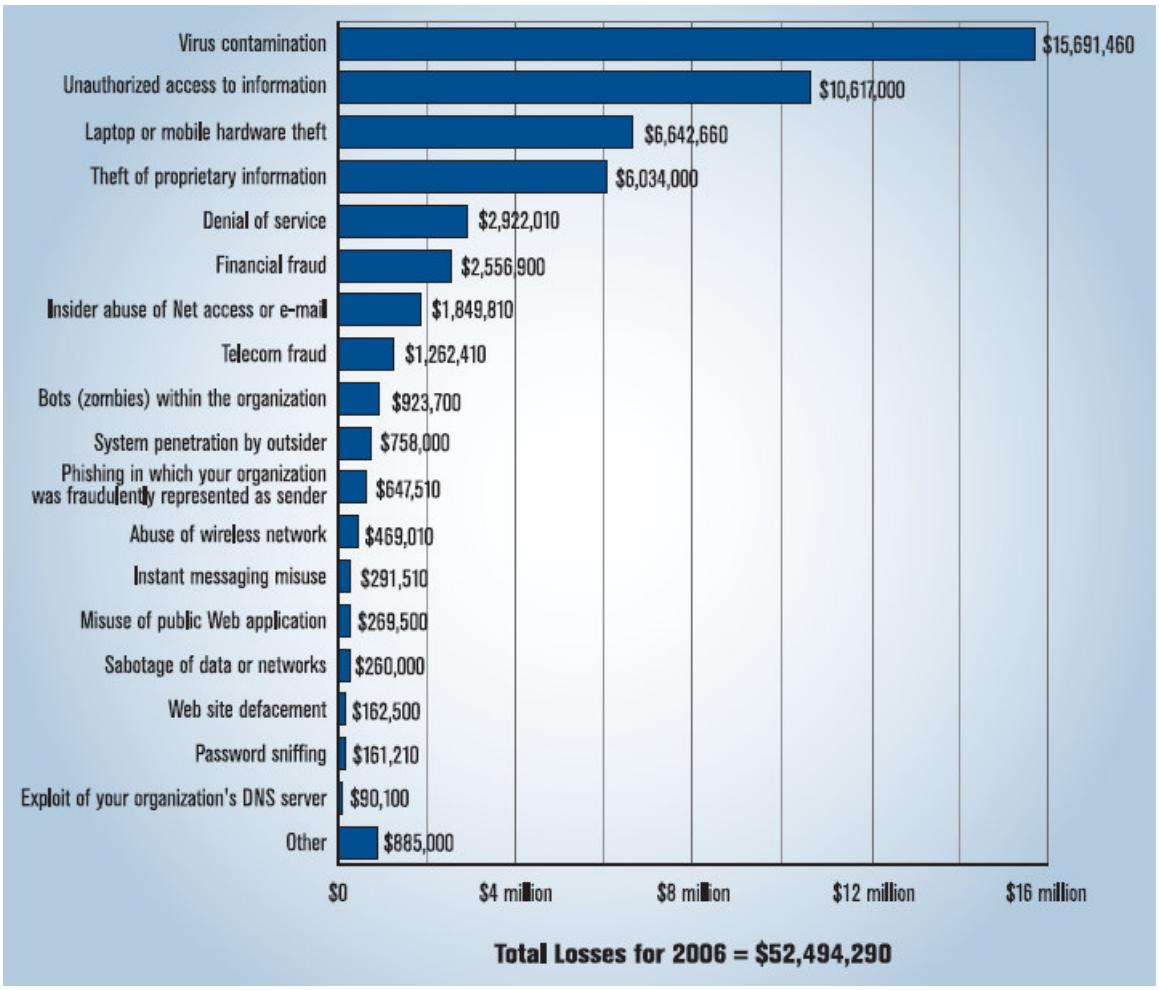


Figure 2.Total losses due to threats in 2006 (Lawrence, Loeb, & Lycyshyn, 2006).

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).

.

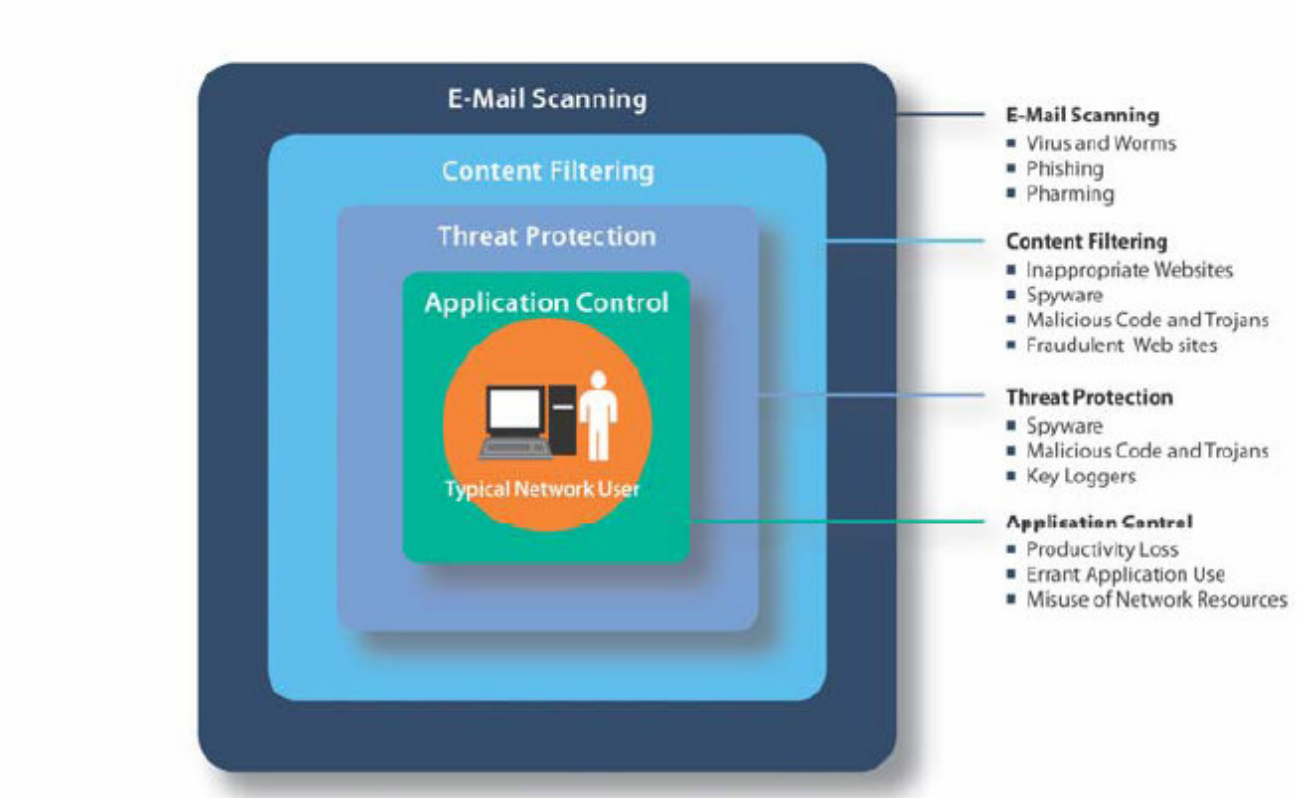


Figure 3. Unified Threat Management (SonicWALL, 2005).

Additional Challenges in Today's SOHO Network

SonicWALL® 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).

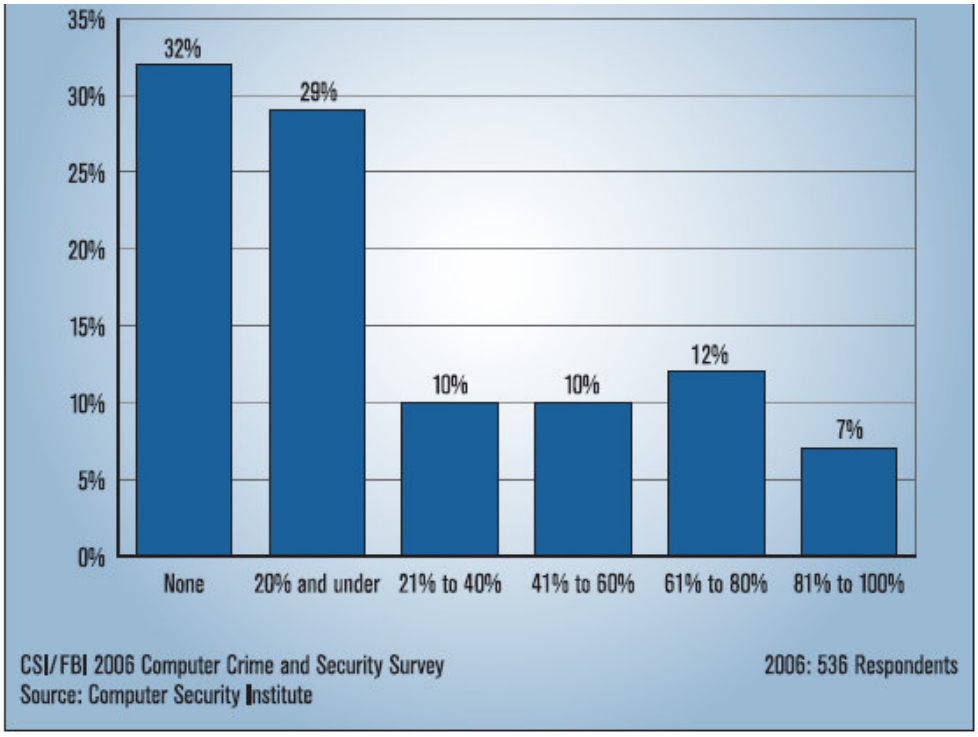


Figure 4. Percentage of Inside Threats (Lawrence, Loeb, & Lycyshyn, 2006).

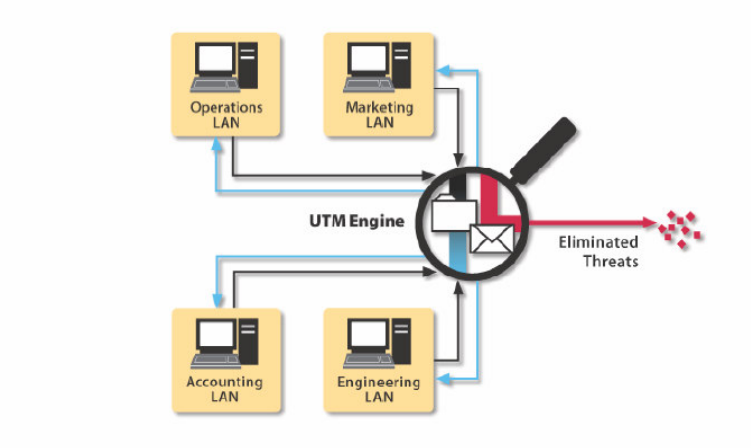


Figure 5. How a UTM blocks Internal and External Threats (SonicWALL, 2005).

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.

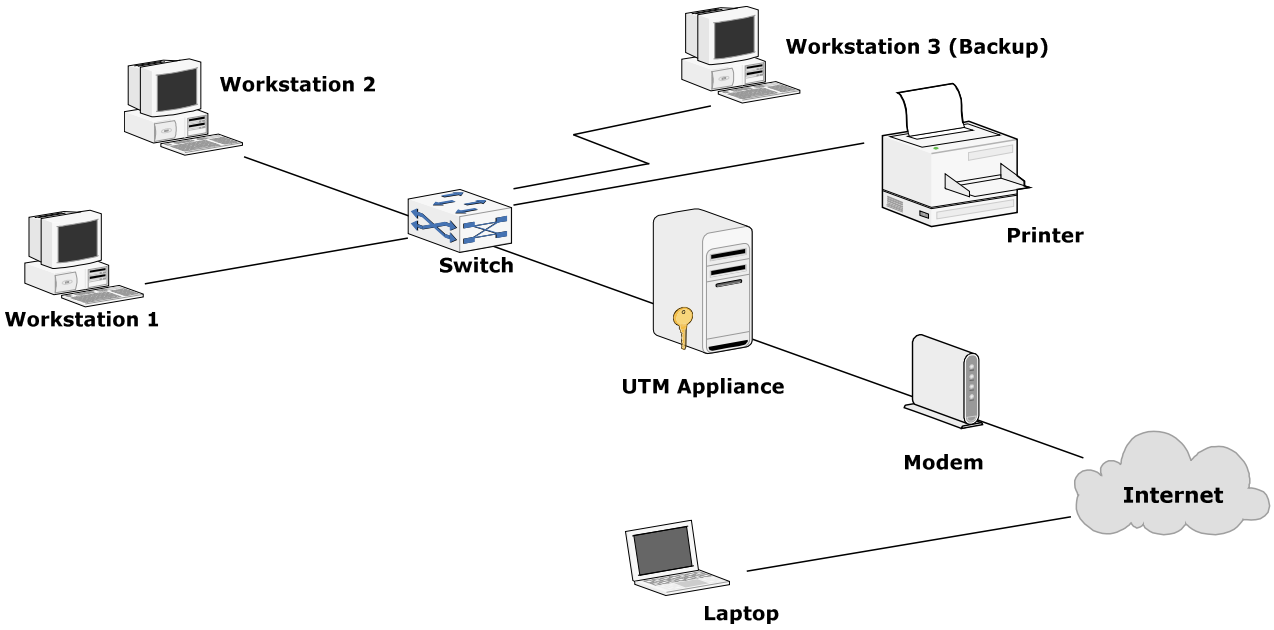


Figure 6. Future Mode of Operation for SOHO UTM Appliance.

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.

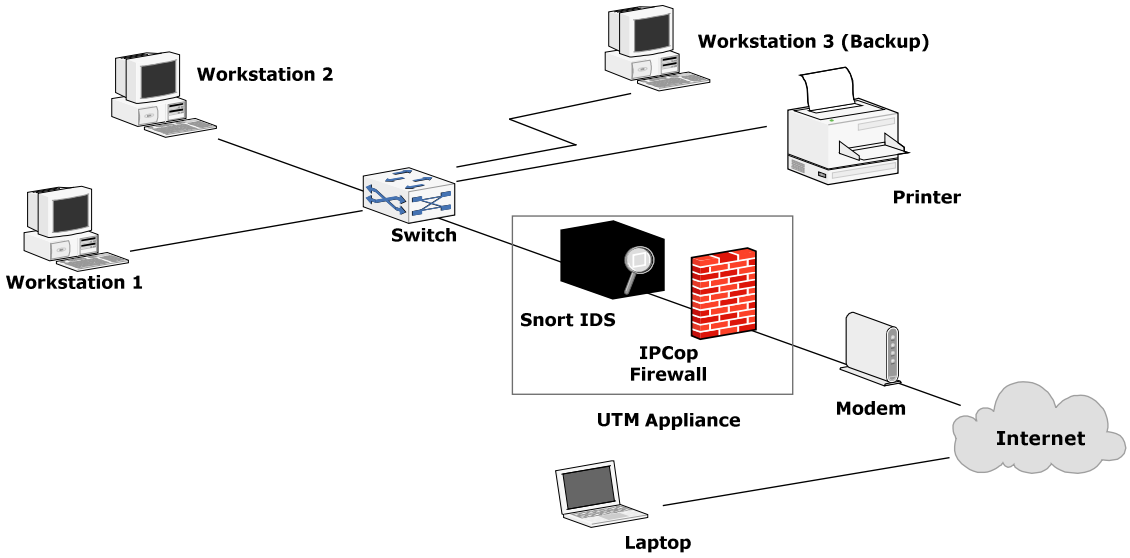


Figure 7. IPCop and Snort IDS at the core of our UTM appliance.

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).

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).

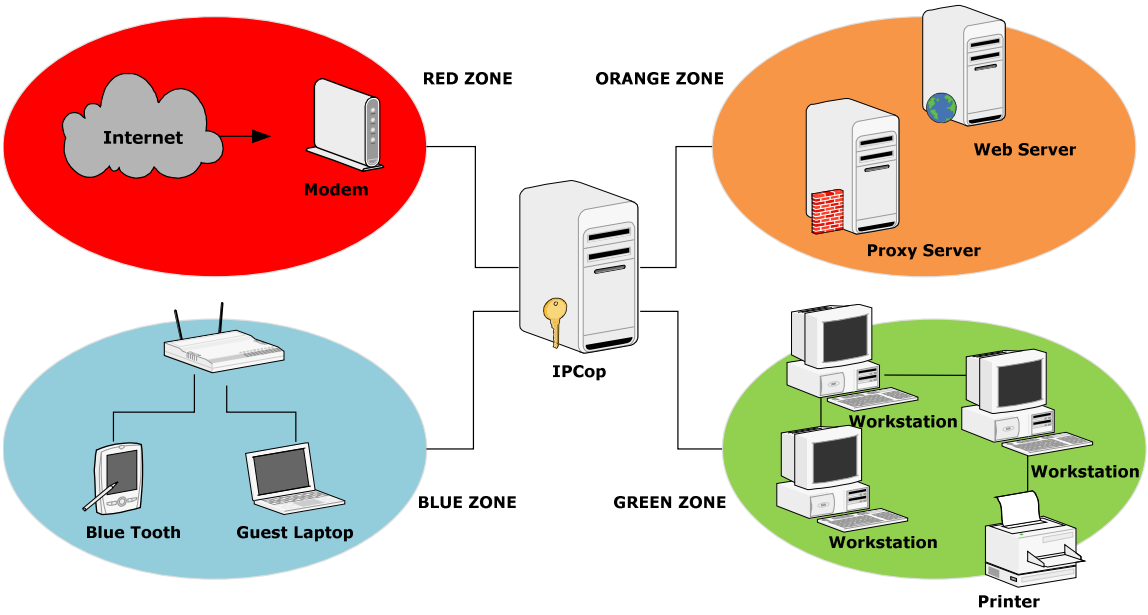


Figure 9. IPCop dividing the SOHO network into zones.

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).

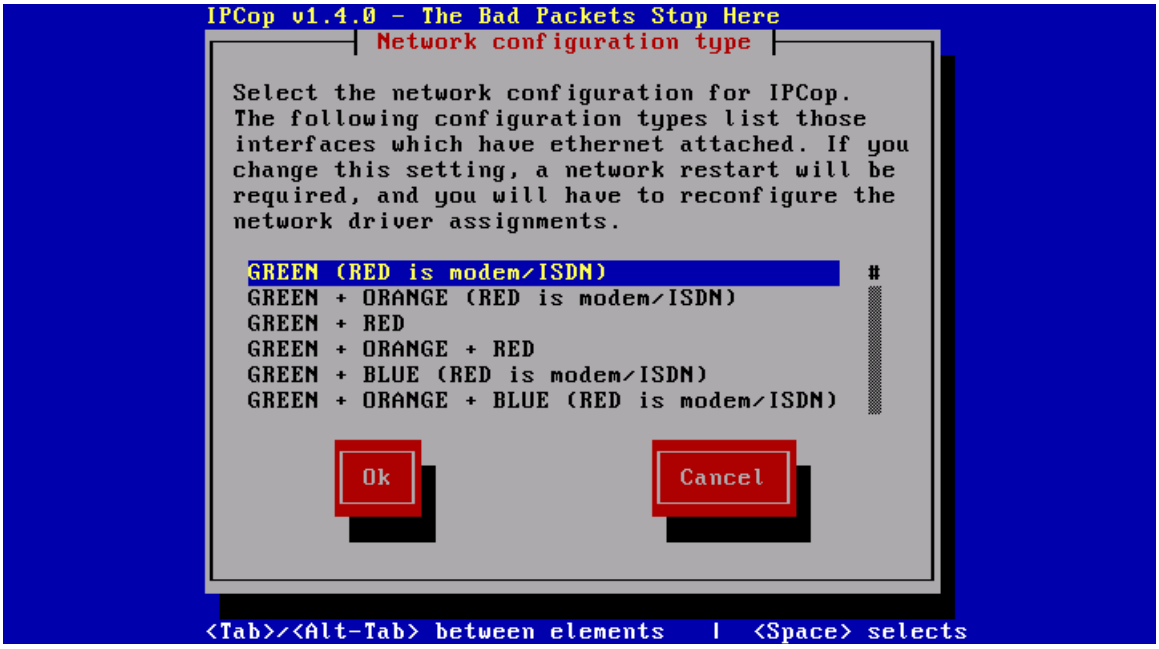


Figure 10. Choosing our IPCop zones (Walker, Goldshmitt, & Pielschmidt, 2004).

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.

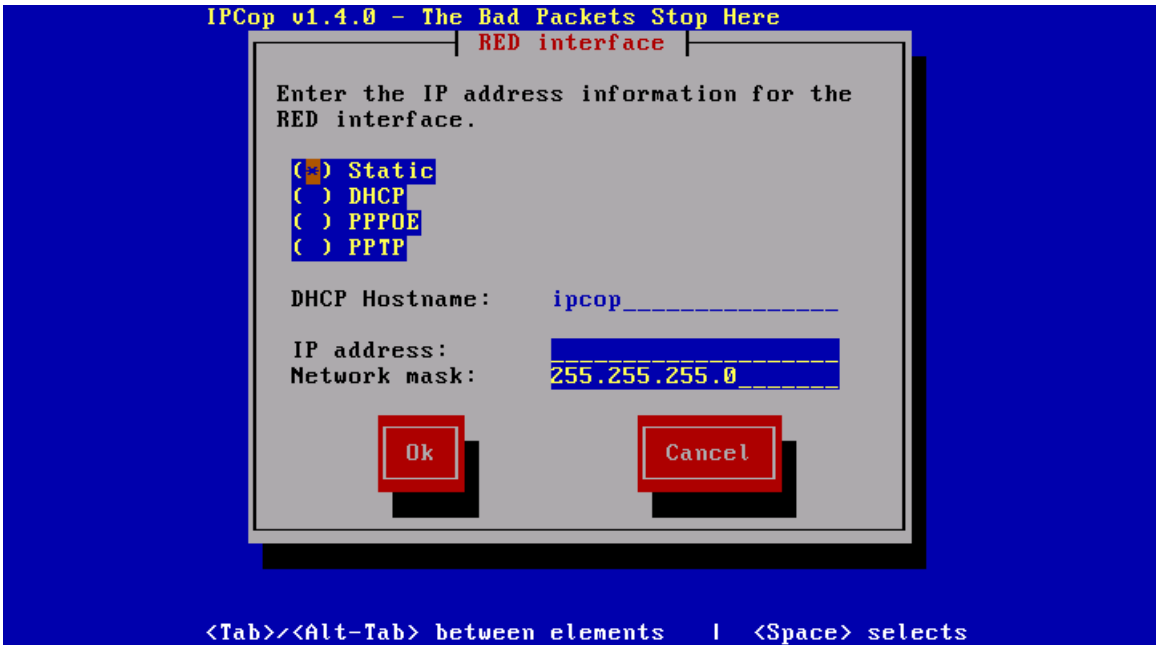


Figure 11. Setting up DHCP for the Red zone network card for internet connectivity (Walker, Goldshmitt, & Pielschmidt, 2004).

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)

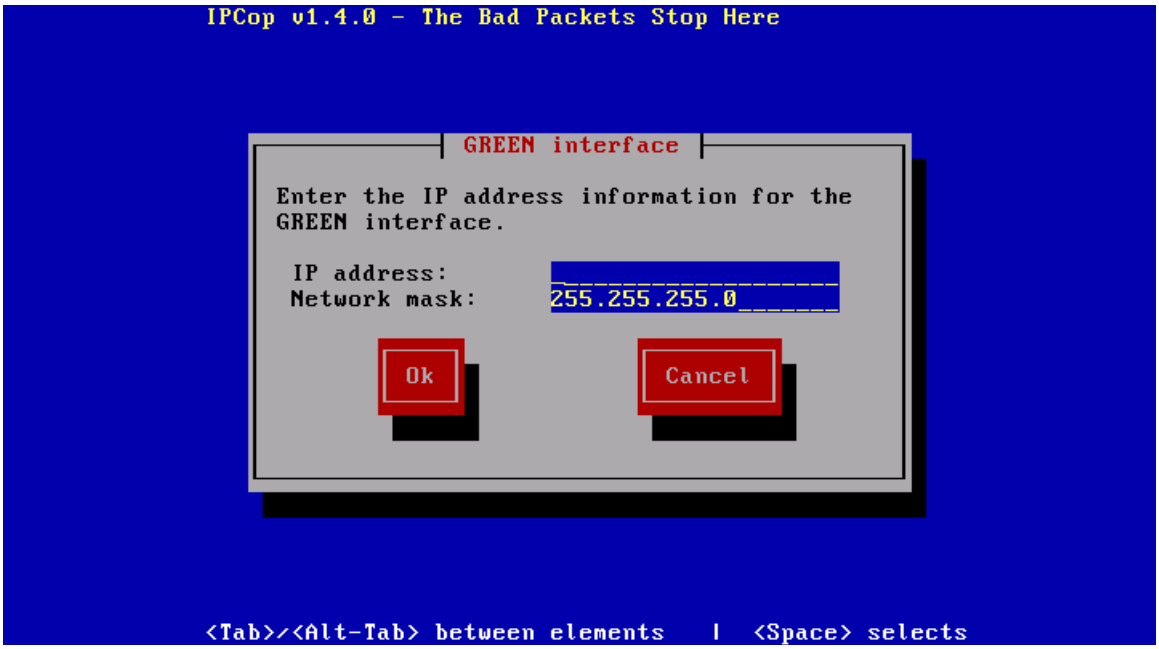


Figure 12. Setting up the Green zone network card for the internal network (Walker, Goldshmitt, & Pielschmidt, 2004).

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.

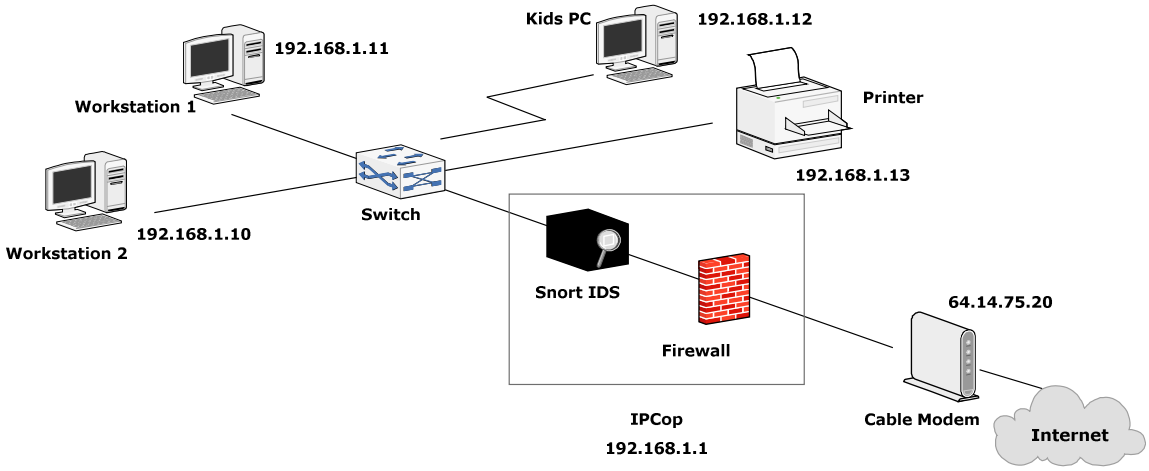


Figure 13. DHCP setup for IPCop.

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.

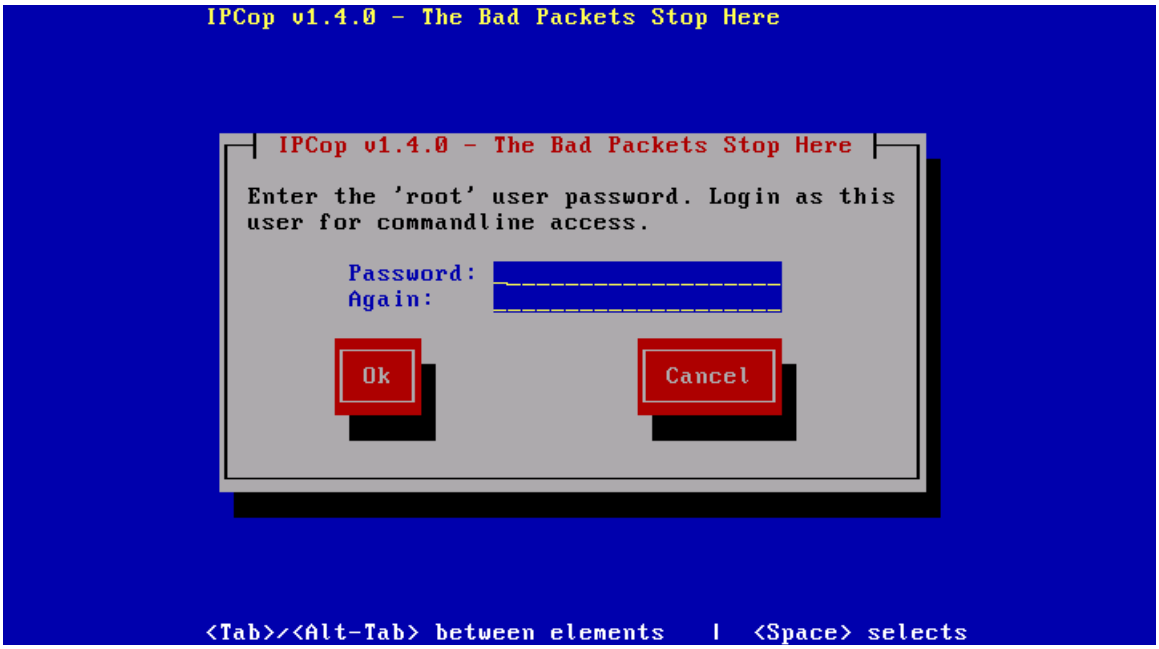


Figure 14. Setting up password in IPCop for root account (Walker, Goldshmitt, & Pielschmidt, 2004).

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 http://ipcop1:445, 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).

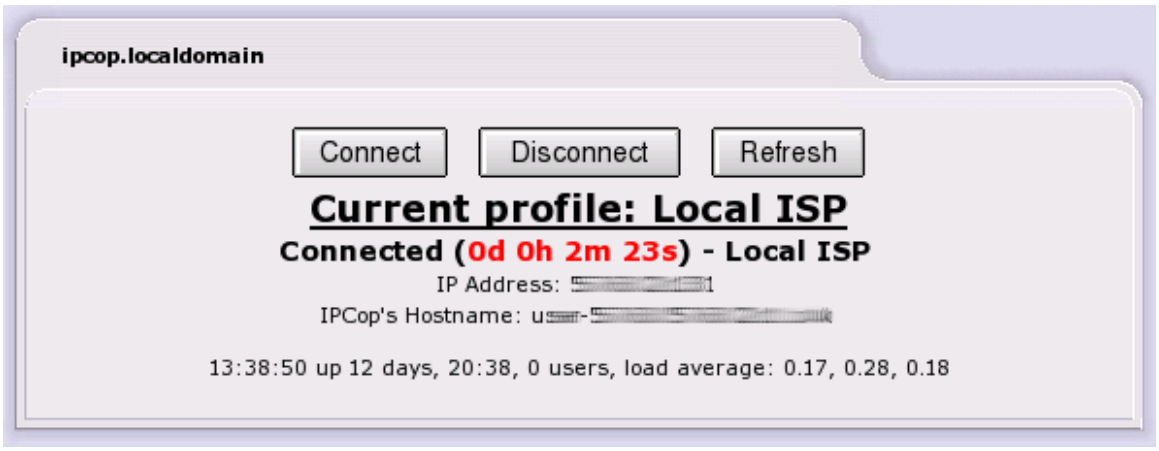


Figure 15. Administrative home page for IPCop (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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.

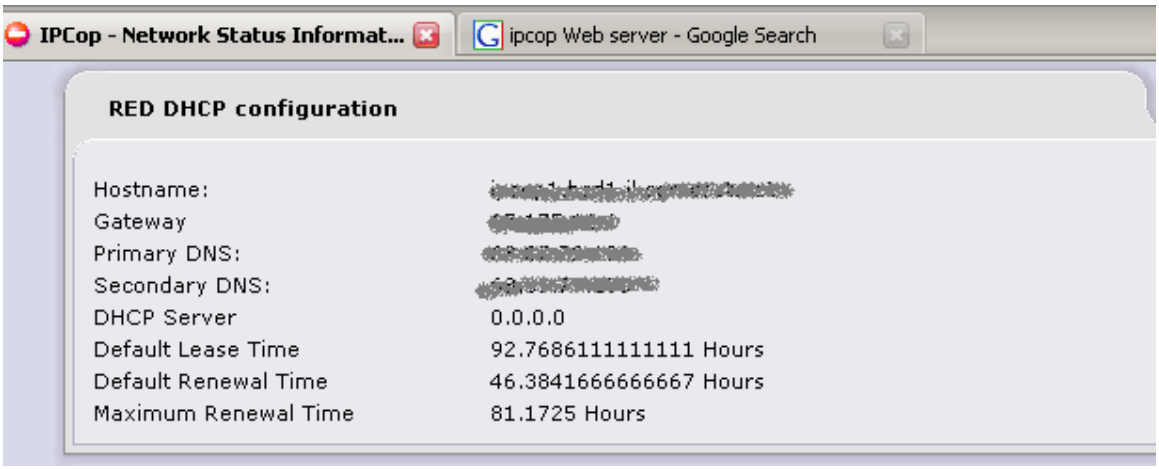


Figure 16. Network Status page showing ISP's assigned address (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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.

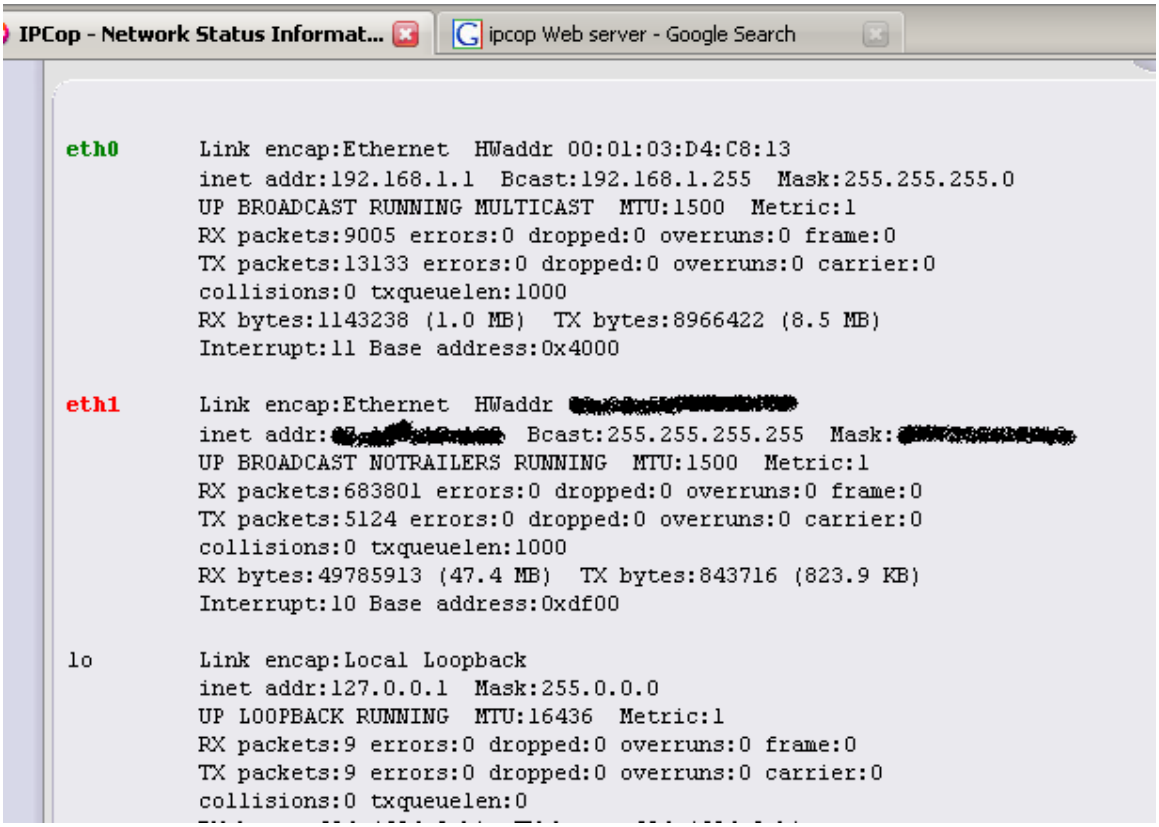


Figure 17. Network Status page showing ISP assigned address information (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

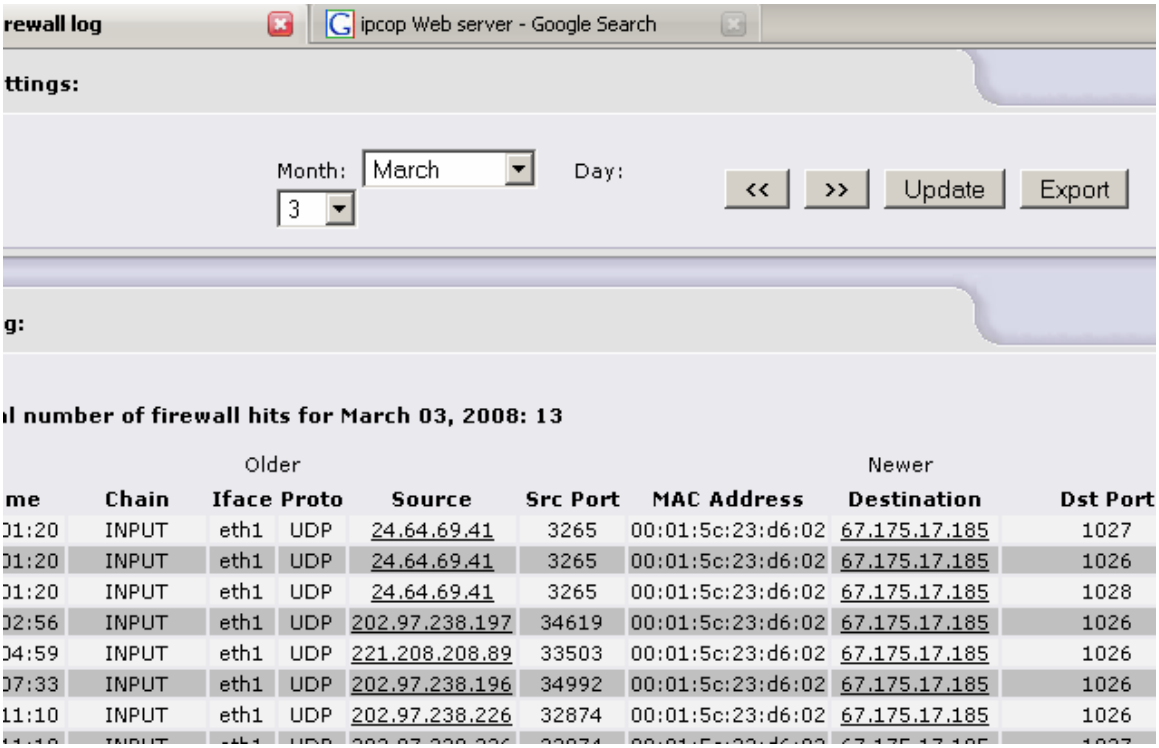


Figure 18. IPCop firewall logs page (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

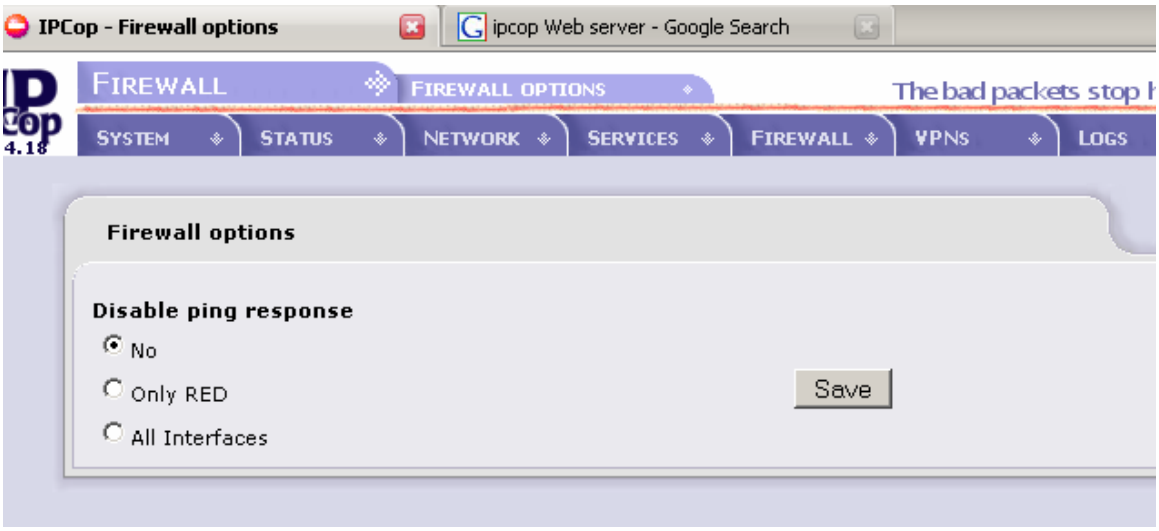


Figure 19. IPCop firewall options page (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

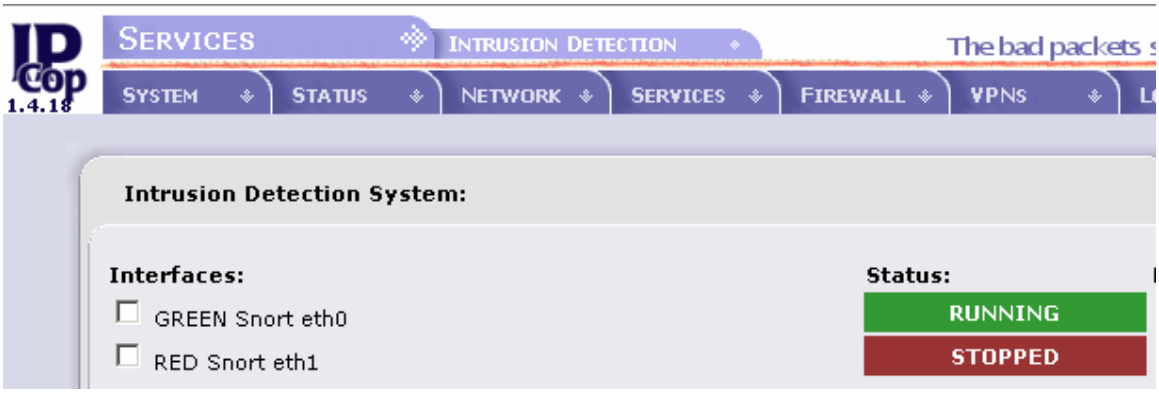


Figure 22. Snort changing state from stopped to running (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

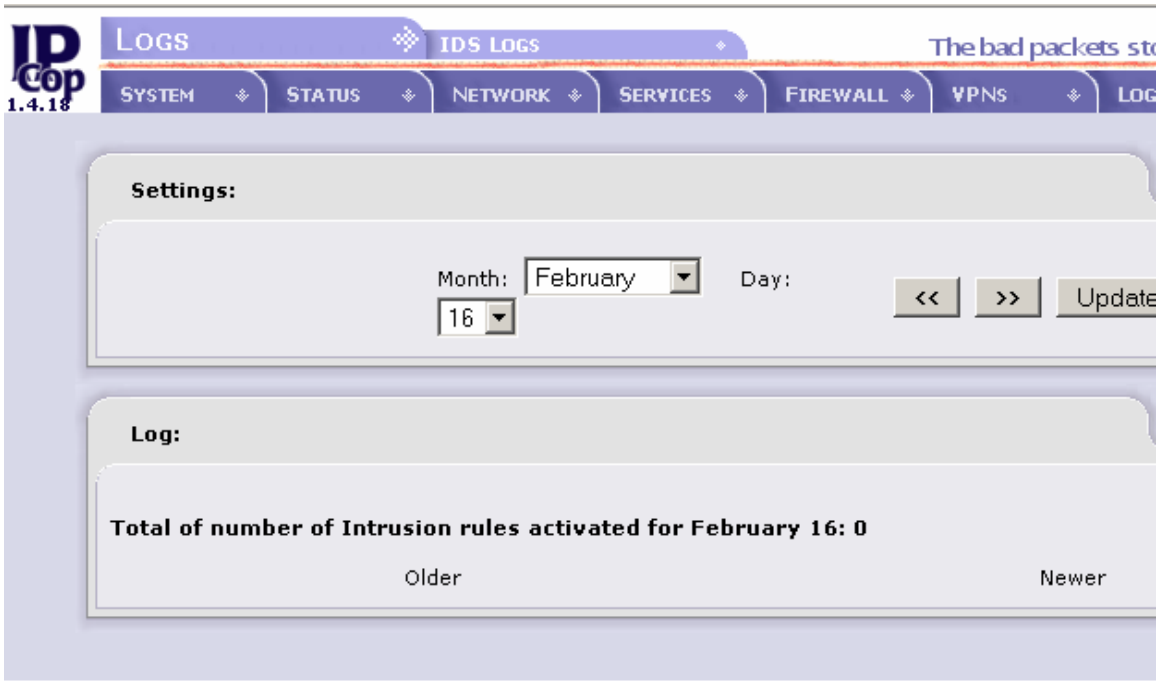


Figure 23. IDS Snort Logs Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

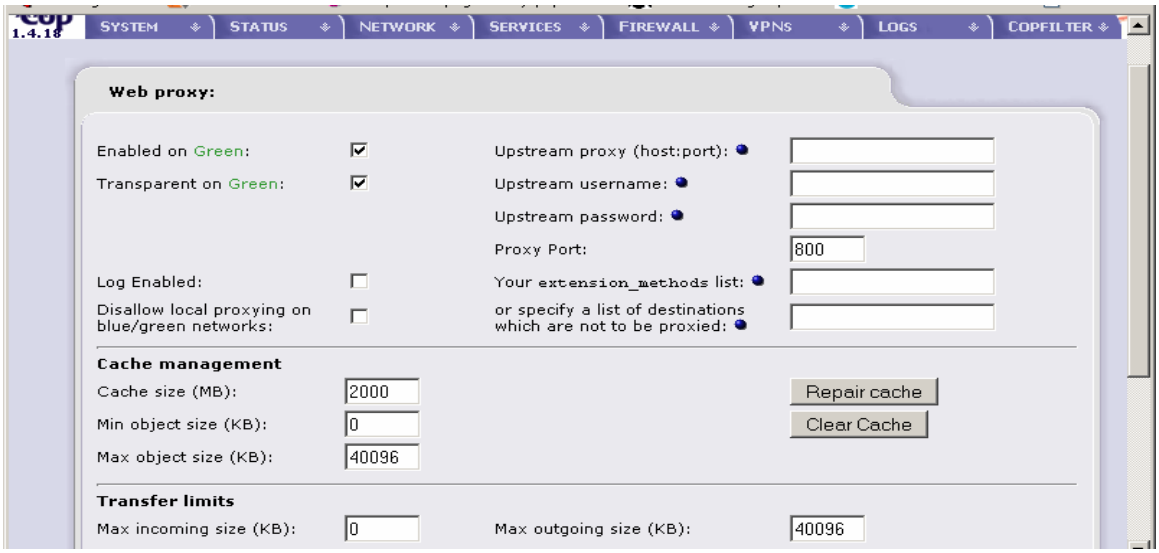


Figure 24. Proxy service administrative page (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

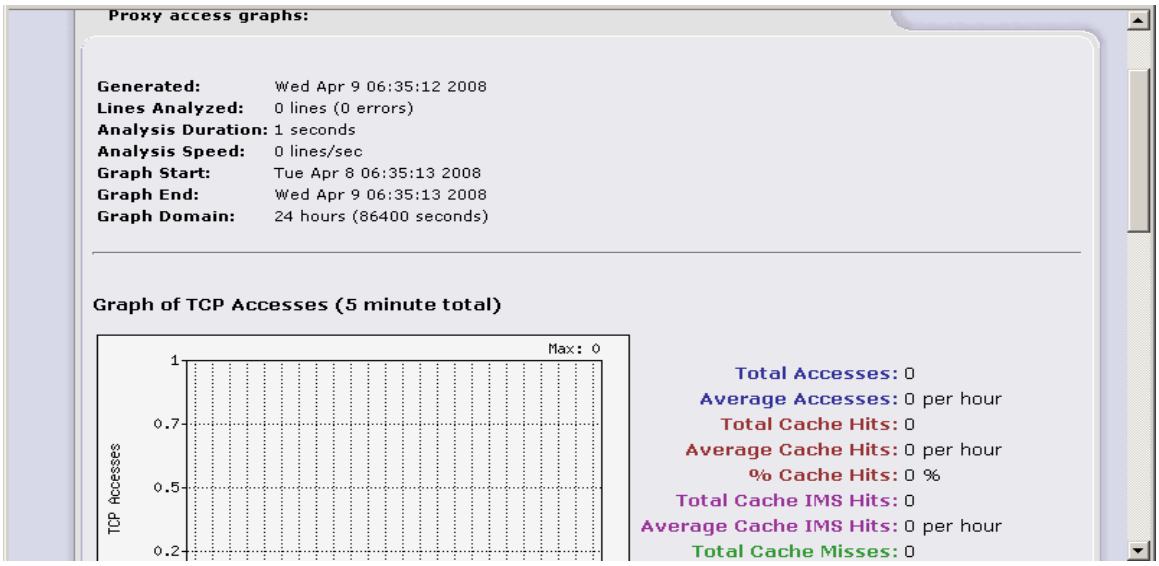


Figure 25. Squid Proxy Access Graphs and statistics (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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.

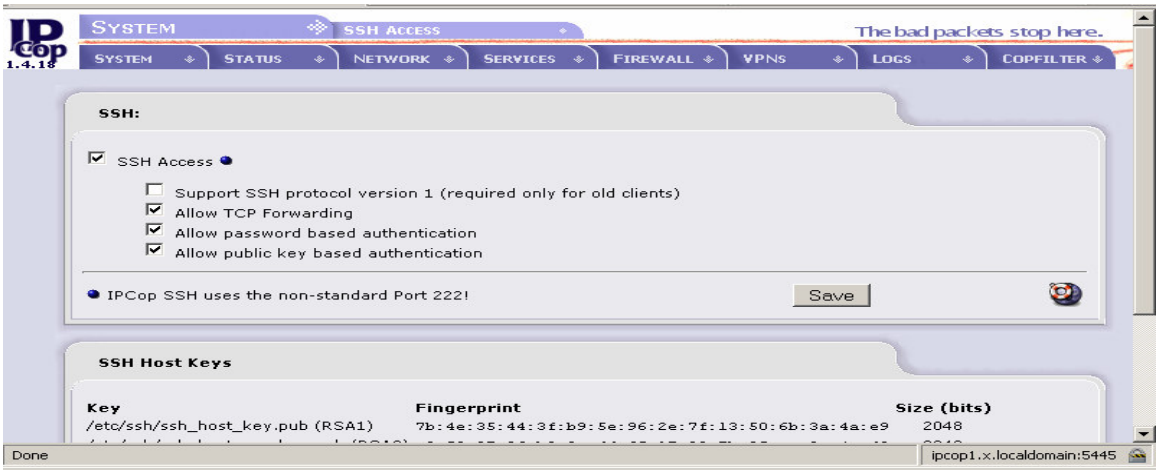


Figure 26. SSH Server administrative age (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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.

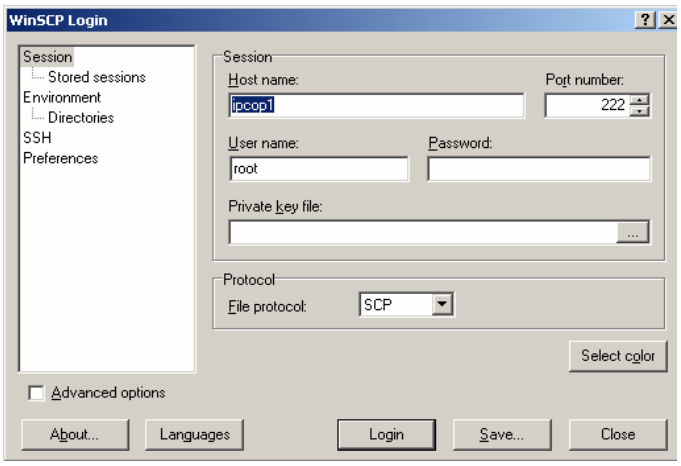


Figure 27. WinSCP connecting to IPCop using SSH (Prikryl, 2000).

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.

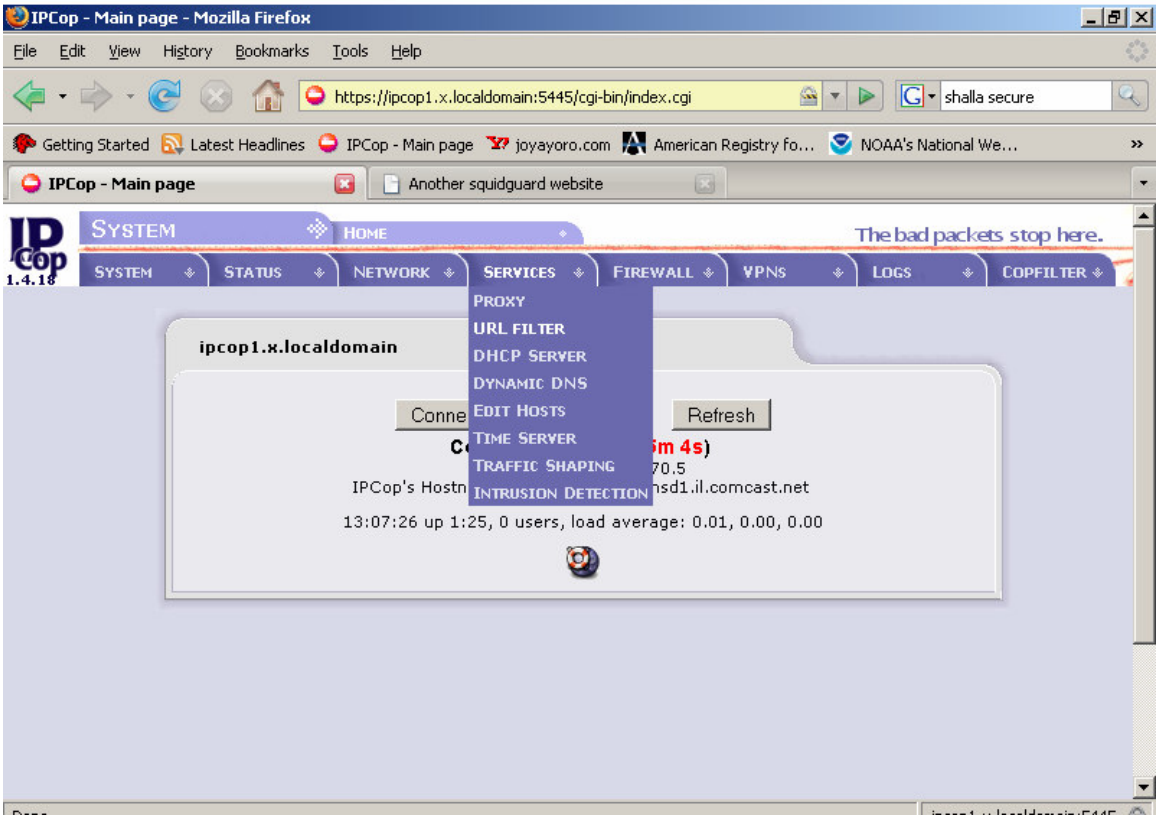


Figure 28. Services List tab within IPCop (Clancey, Goldshmitt, Kastner, Oberlander, 2004).

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).

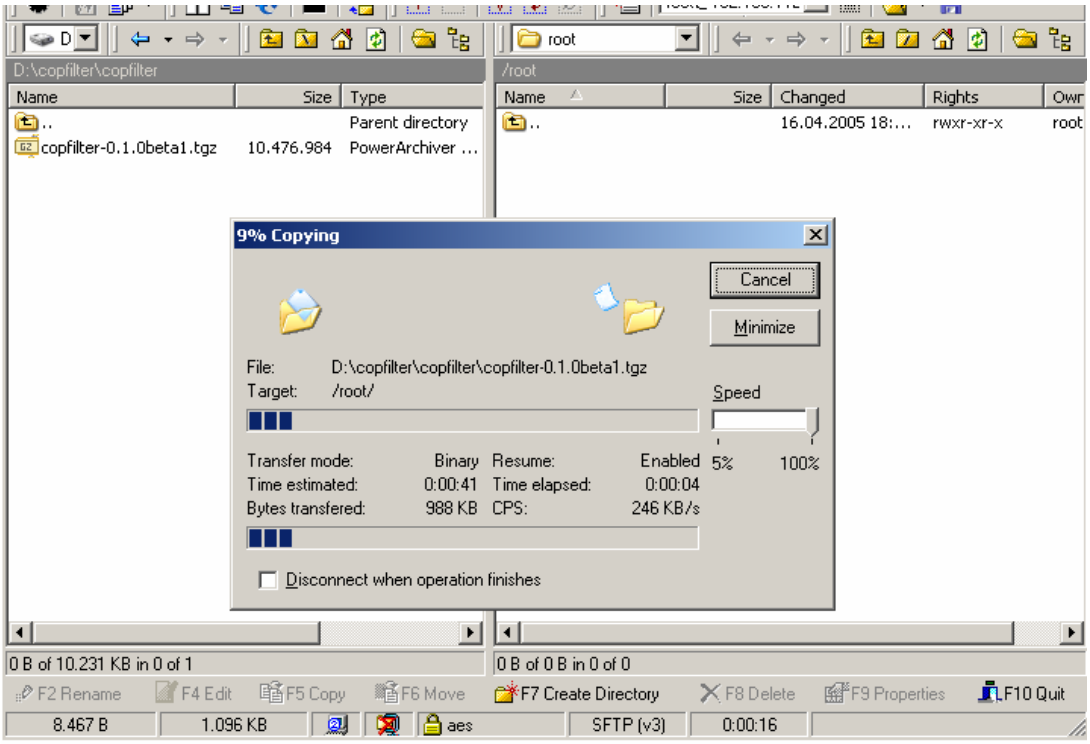


Figure 29. Copying Copfilter into IPCop using WinSCP (Madlener, 2002).

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@ipcop:~/copfilter-0.1.0beta10a # ./install
md5check                done
extracting ...  done
now executing /var/log/copfilter/0.1.0beta10a/setup_util -i

============================================================
Copfilter installation --  Version 0.1.0beta10a
============================================================


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 package adds firewall rules, proxies, filters, virus scanners
and precompiled binaries to your ipcop machine, which is a
big security risk ! There are probably several ways of breaking
or intruding copfilter (pls do report if you find any), so if
security is an issue, do NOT install this package.

Continue ? [y/N]
```

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

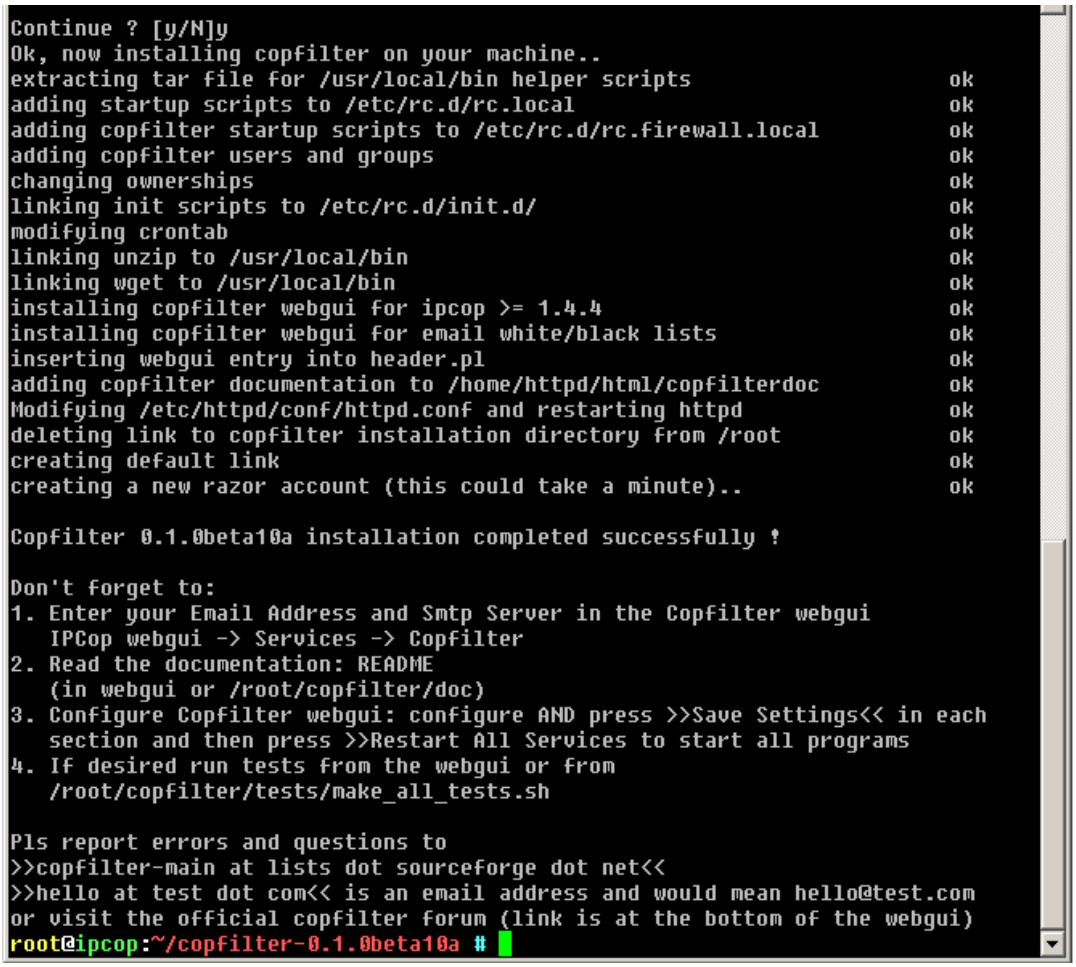


Figure 31. Running the Copfilter installer (Madlener, 2002).

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.

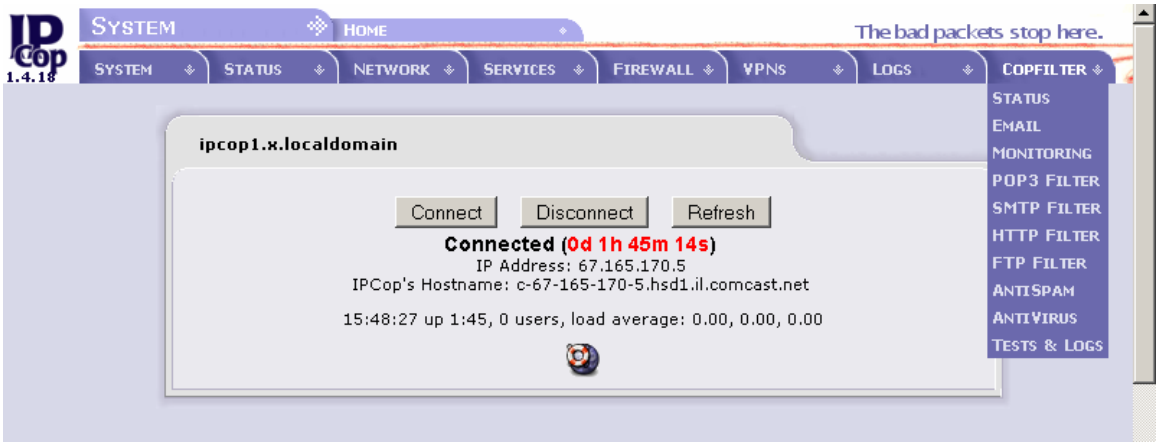


Figure 32. Copfilter available from IPCop GUI (Madlener, 2002).

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).

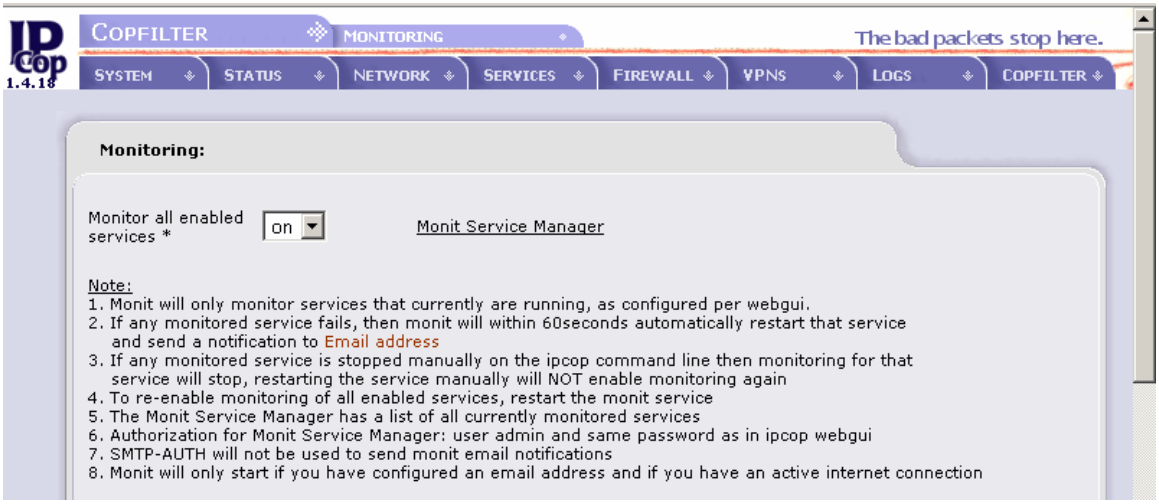


Figure 33. Monit management console (Madlener, 2002).

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)

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).

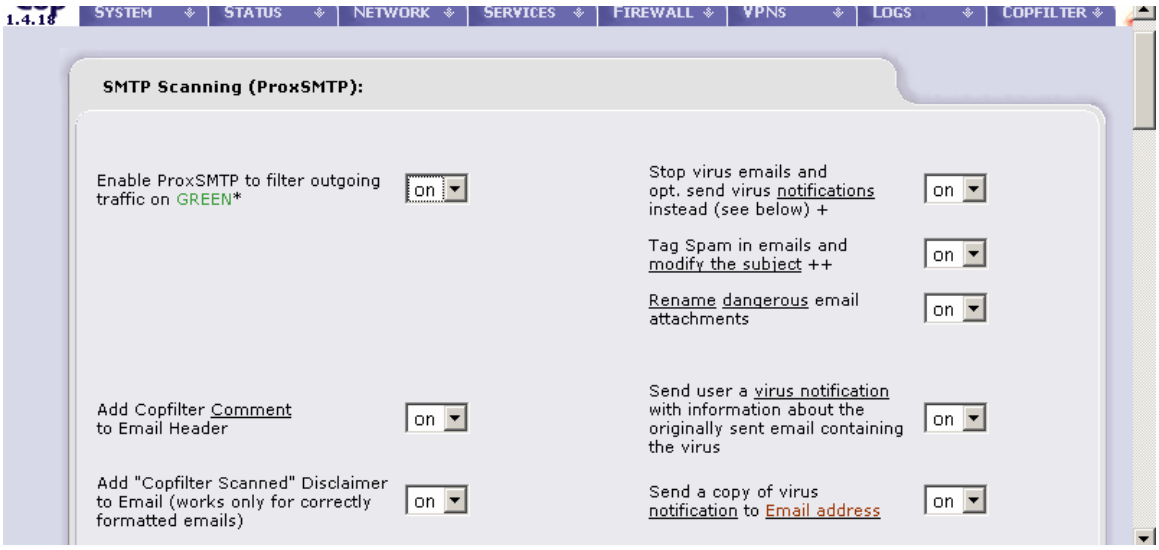


Figure 35. Configuration for ProxSMTP SMTP proxy filter (Madlener, 2002).

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.

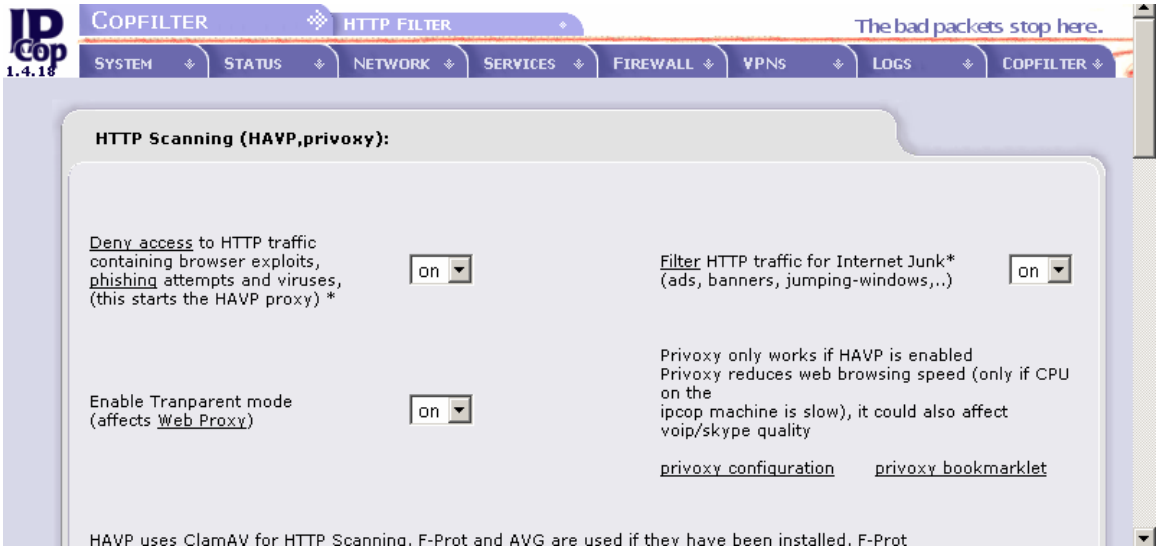


Figure 36. HAVP http proxy filter configuration (Madlener, 2002).

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.

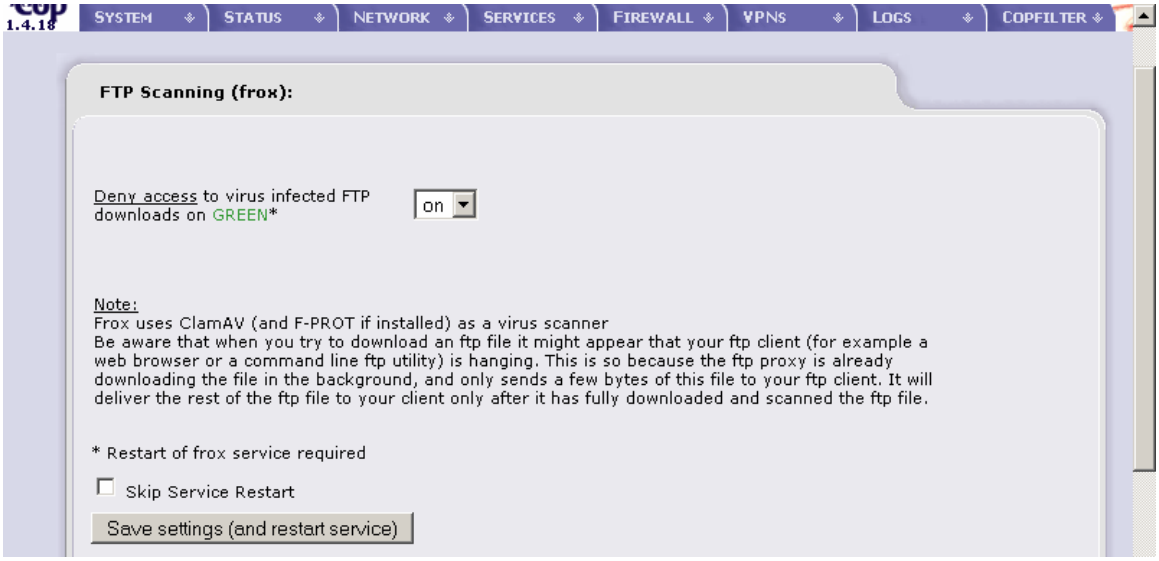


Figure 37. Frox ftp scanning proxy configuration page (Madlener, 2002).

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.

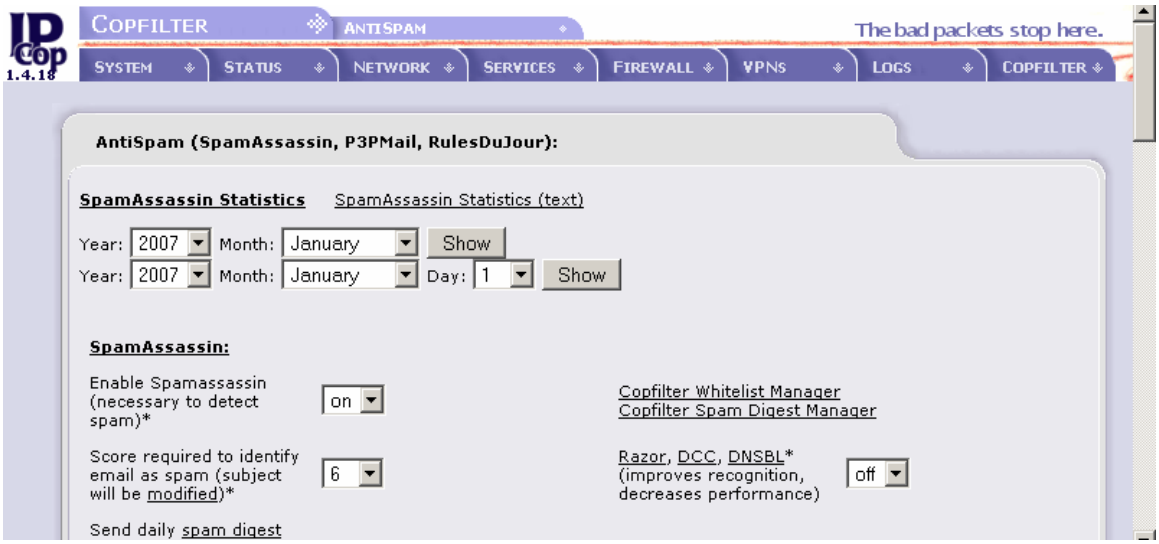


Figure 38. Setting up SpamAssassin filter (Madlener, 2002).

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).

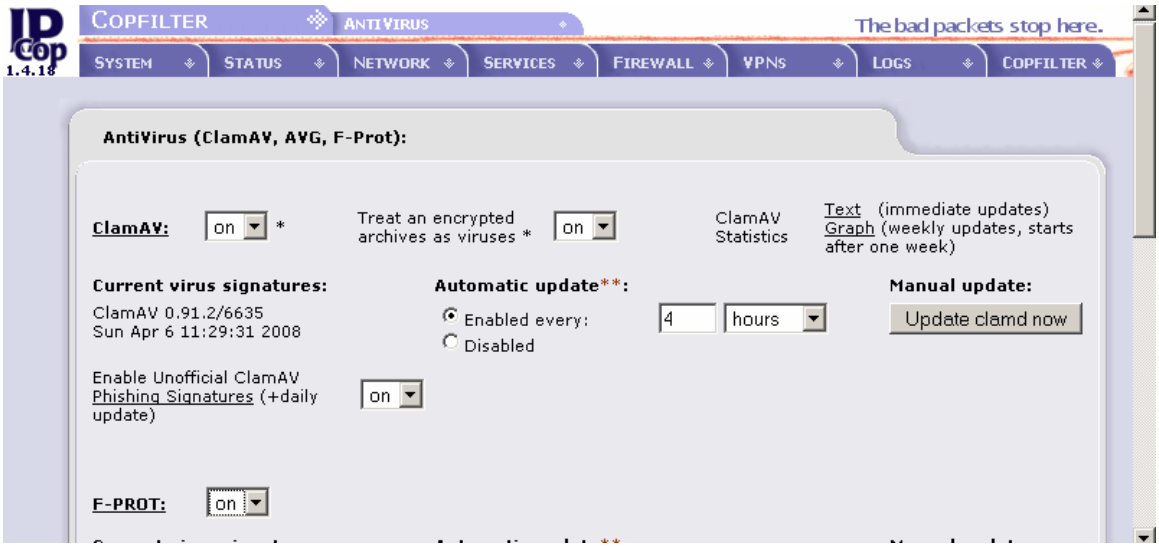


Figure 39. ClamAV configuration page (Madlener, 2002).

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

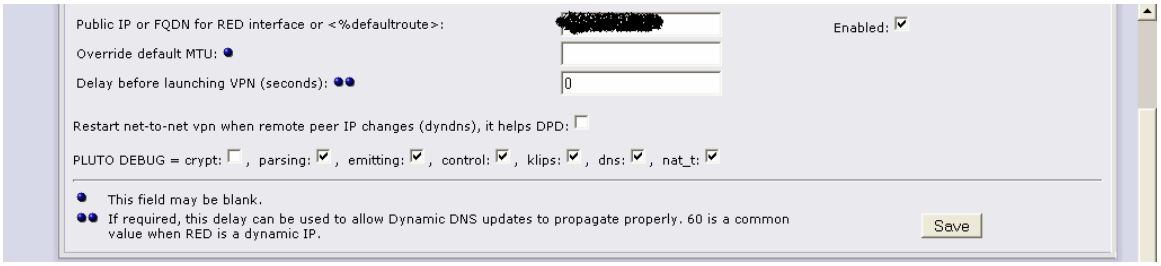


Figure 40. VPN configuration page (Support, 2007).

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.

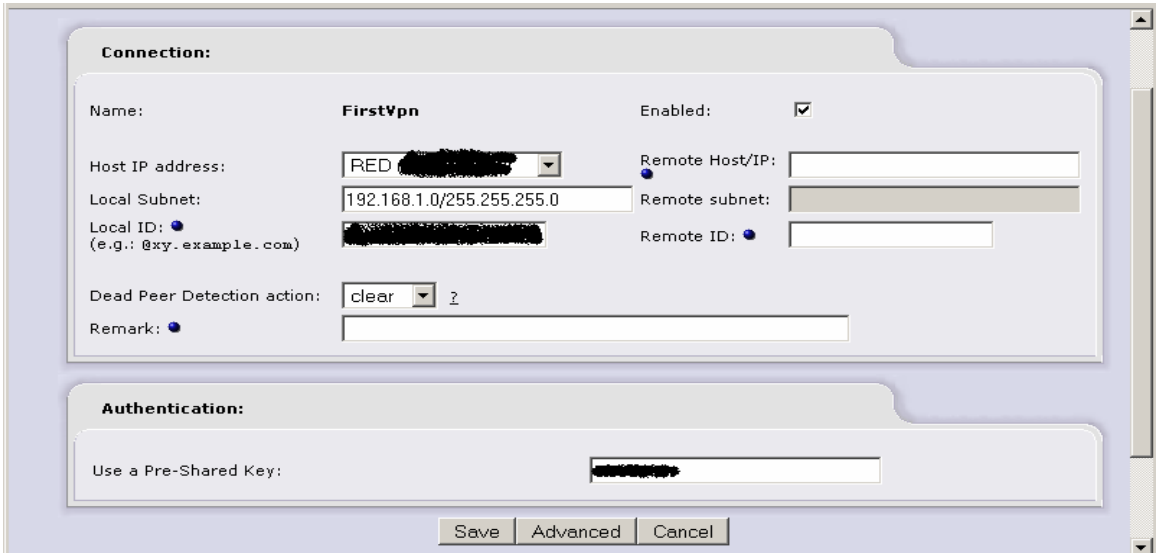


Figure 41. Setting up a RoadWarrior connection (Support, 2007).

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.

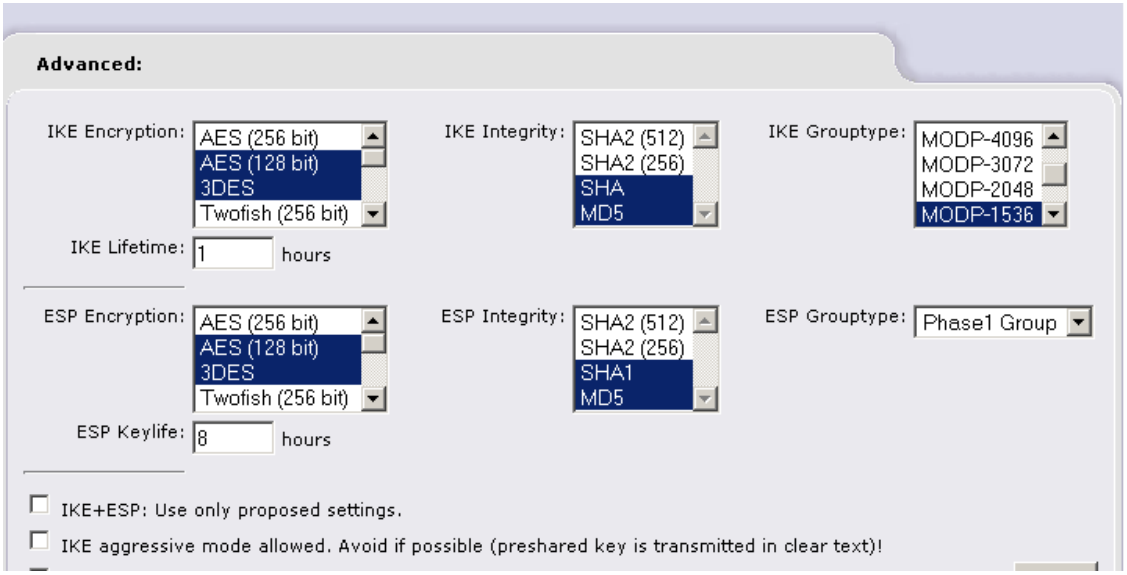


Figure 42. Advanced settings for the RoadWarrior connection (Support, 2007).

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".

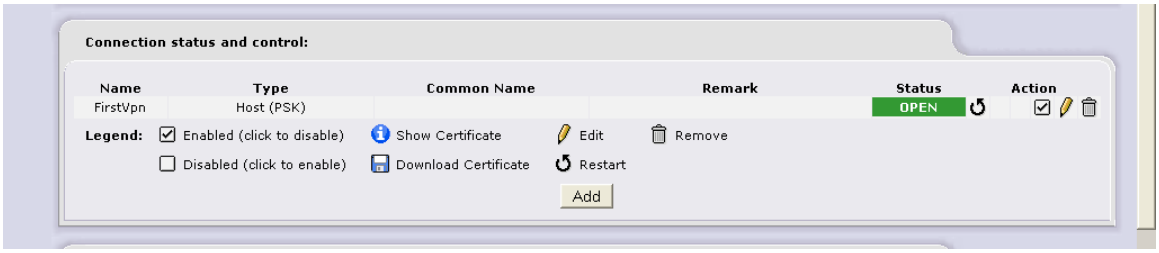


Figure 43. Connection status of the RoadWarrior connection (Support, 2007).

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.

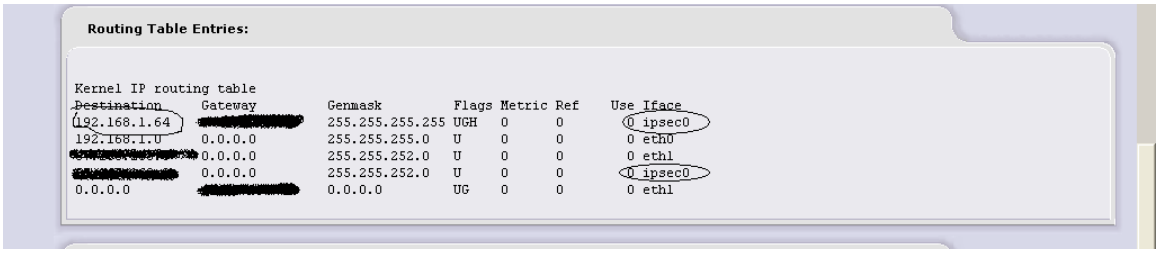


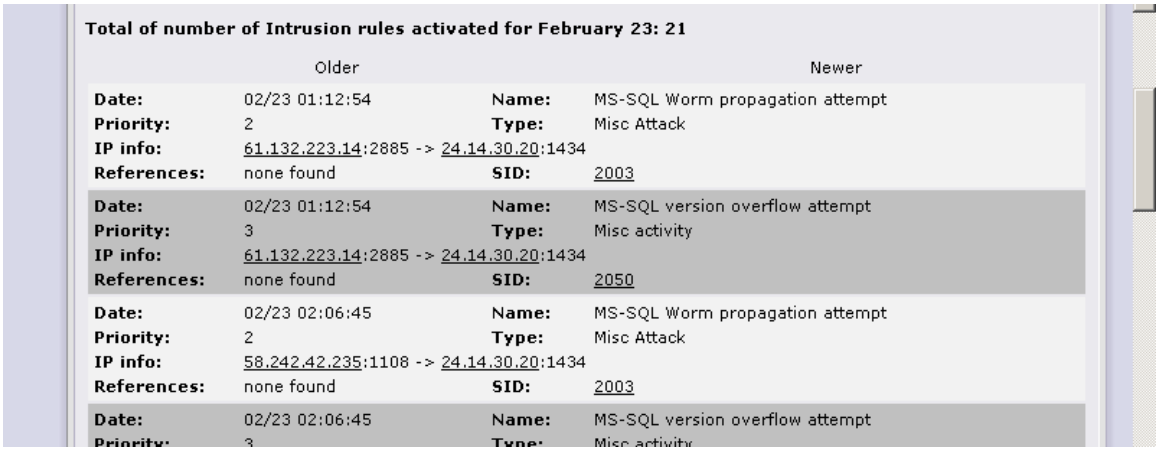
Figure 44. VPN session routing shown in the Network Status (Support, 2007).

UTM implementation Tests

For about three weeks, from February 17th to March 4th 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 23rd 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 23rd, 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 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:	61.132.223.14:2885 -> 24.14.30.20:1434		
References:	none found	**SID:**	2003
Date:	02/23 01:12:54	**Name:**	MS-SQL version overflow attempt
Priority:	3	**Type:**	Misc activity
IP info:	61.132.223.14:2885 -> 24.14.30.20:1434		
References:	none found	**SID:**	2050
Date:	02/23 02:06:45	**Name:**	MS-SQL Worm propagation attempt
Priority:	2	**Type:**	Misc Attack
IP info:	58.242.42.235:1108 -> 24.14.30.20:1434		
References:	none found	**SID:**	2003
Date:	02/23 02:06:45	**Name:**	MS-SQL version overflow attempt
Priority:	3	**Type:**	Misc 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).

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).

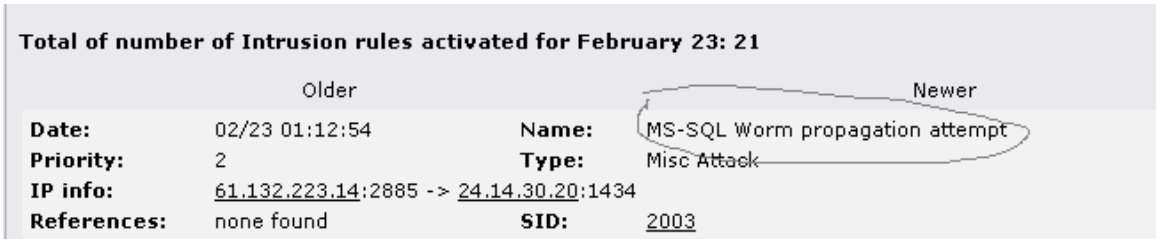


Figure 47. IDS Name of Attack (Snort, 2007).

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.

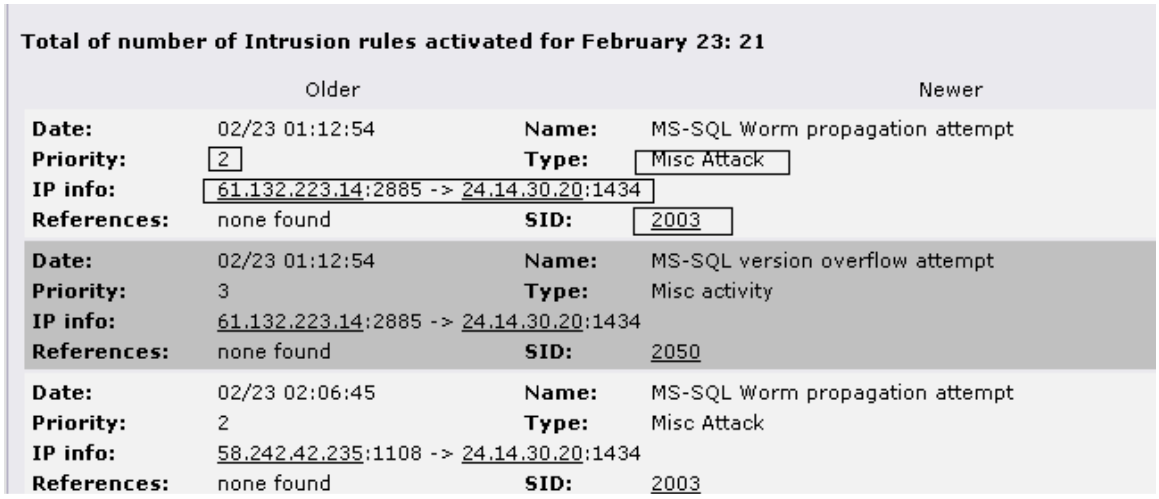


Figure 48. IDS Attack detail (Snort, 2007).

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).

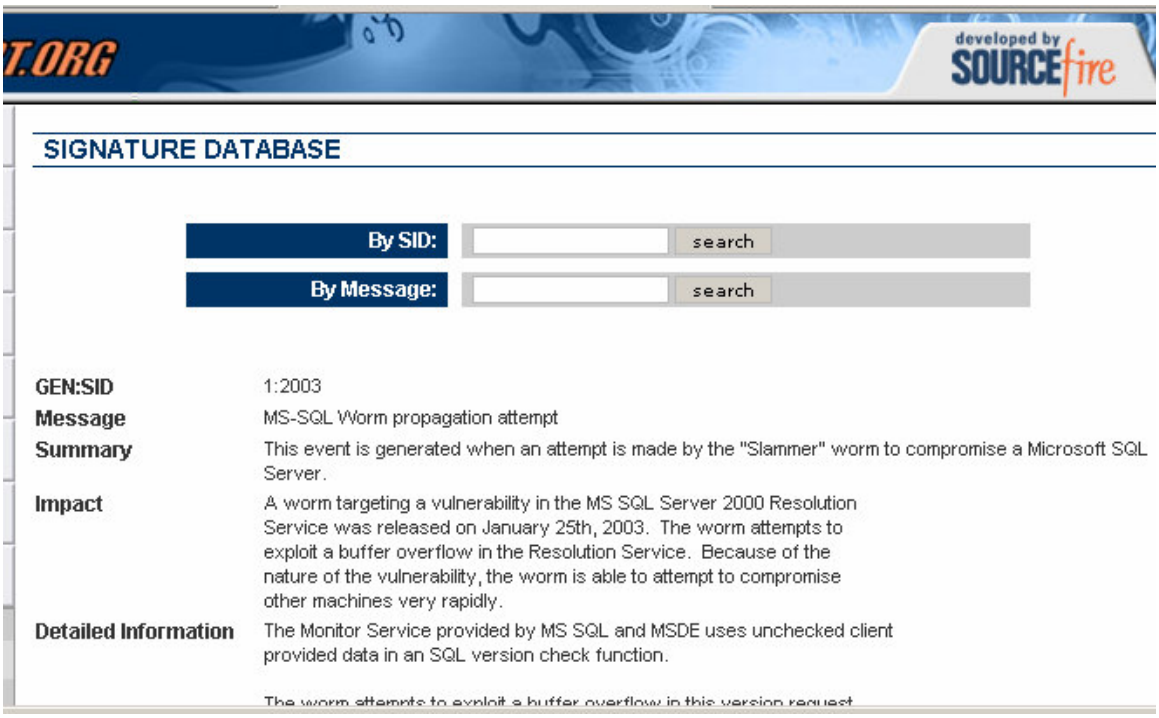


Figure 49. VRT attack signature database detail (Snort, 2007).

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).

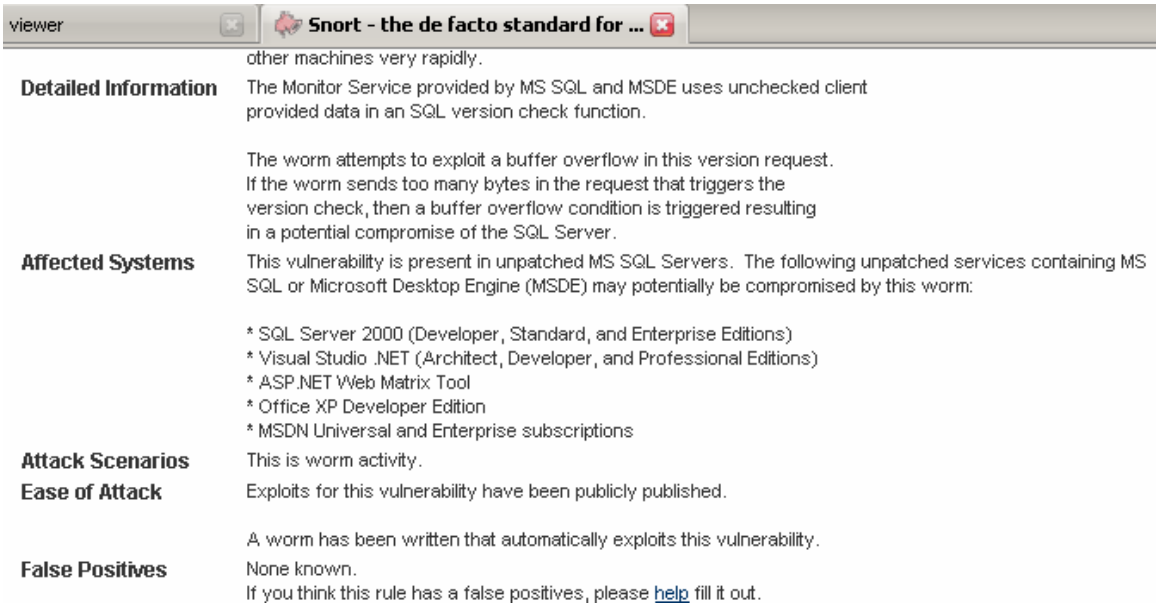


Figure 50. VRT SID attack signature entry other important details (Snort, 2007).

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.

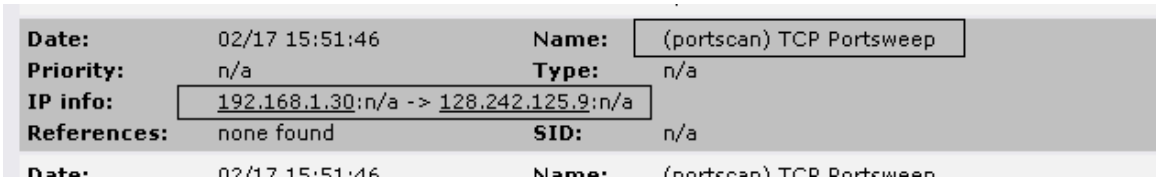


Figure 51. Possible Spyware trying to communicate with external address (Snort, 2007).

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 17th and 18th 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 Ad-aware 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.

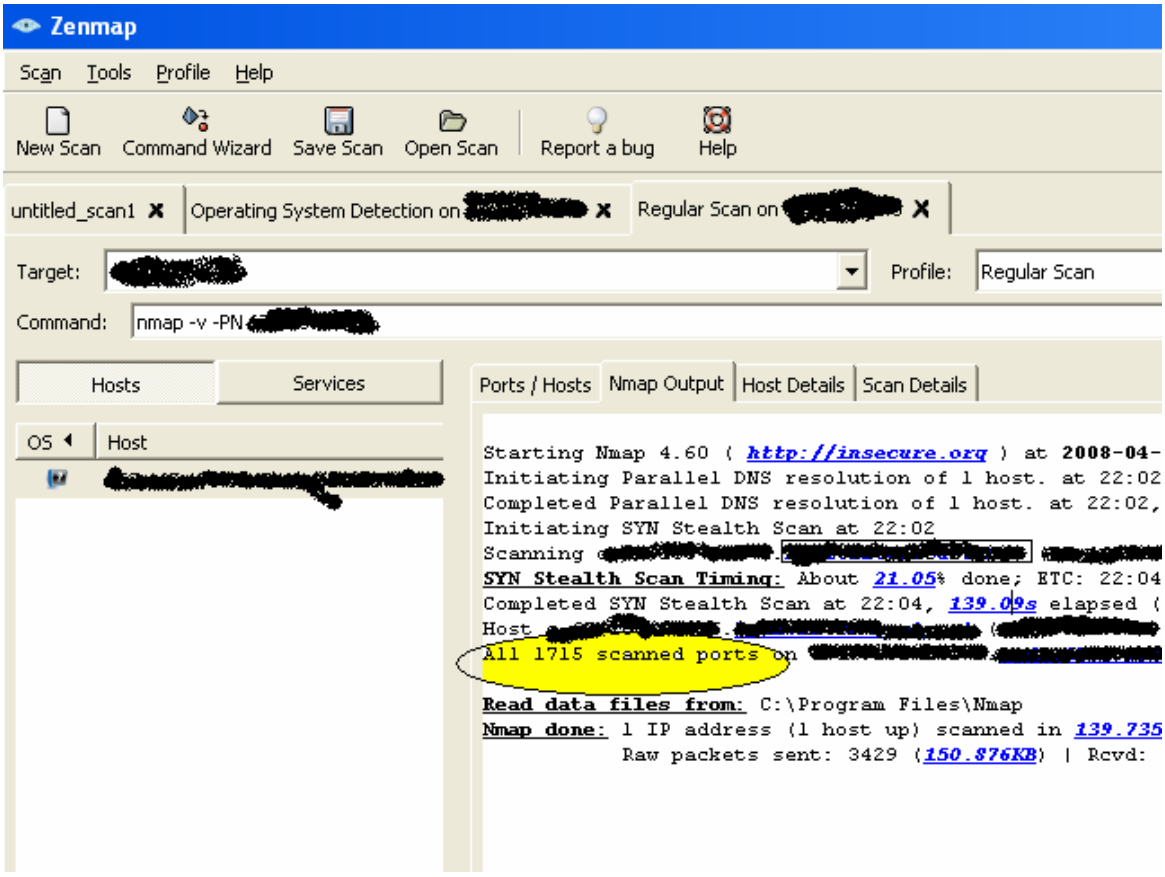


Figure 52. Nmap Output screen details (Insecure.org, 2000).

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.

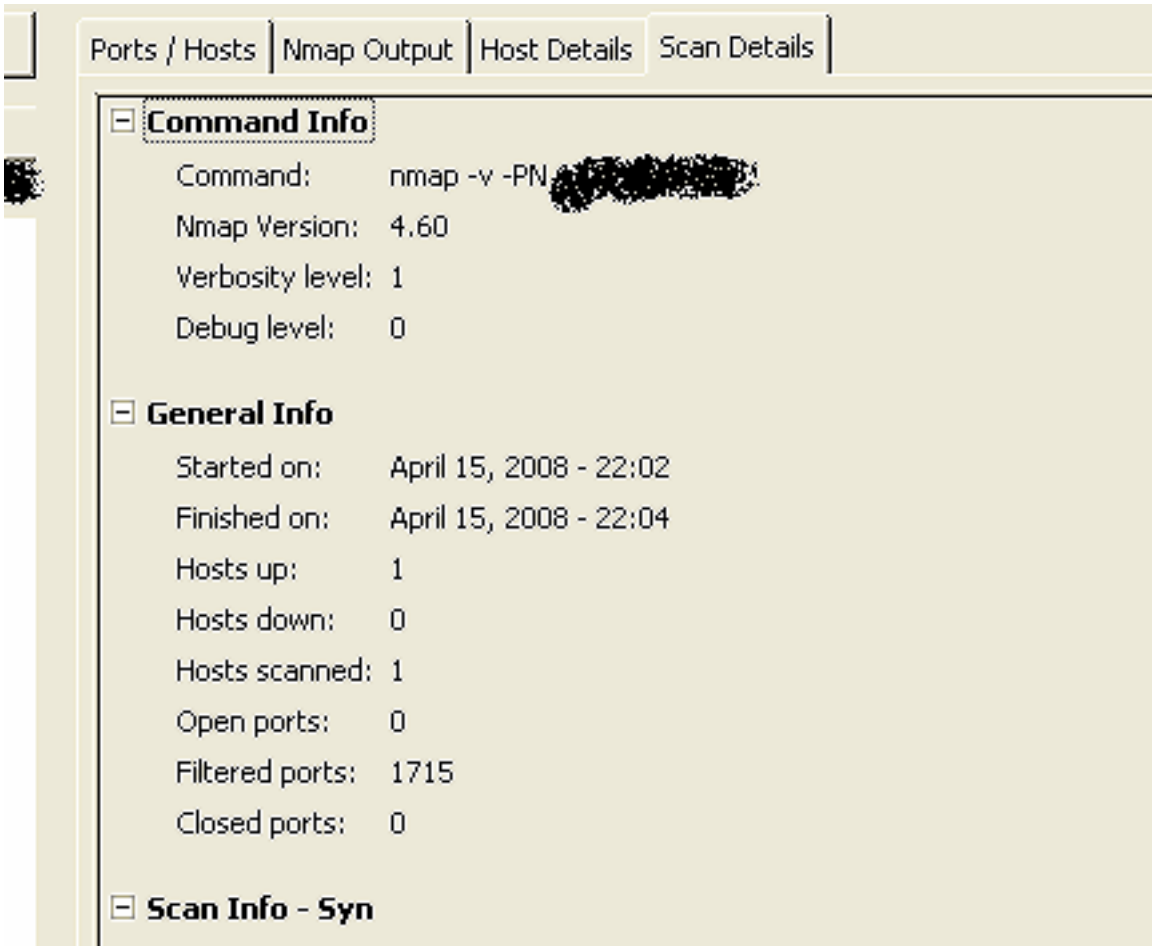


Figure 53. Zenmap regular scan details from our UTM configuration (Insecure.org, 2000).

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).

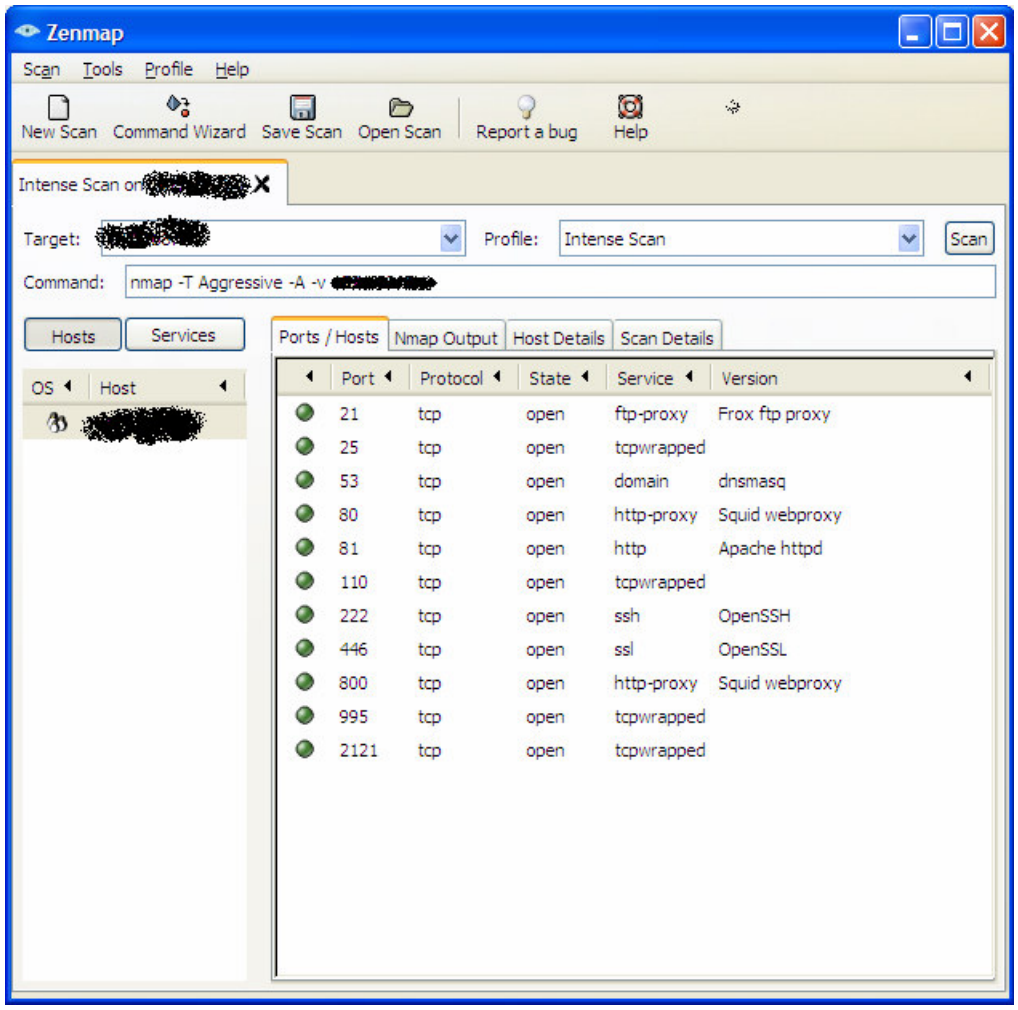


Figure 54. Detected ports from within Green zone by Zenmap (Insecure.org, 2000).

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.

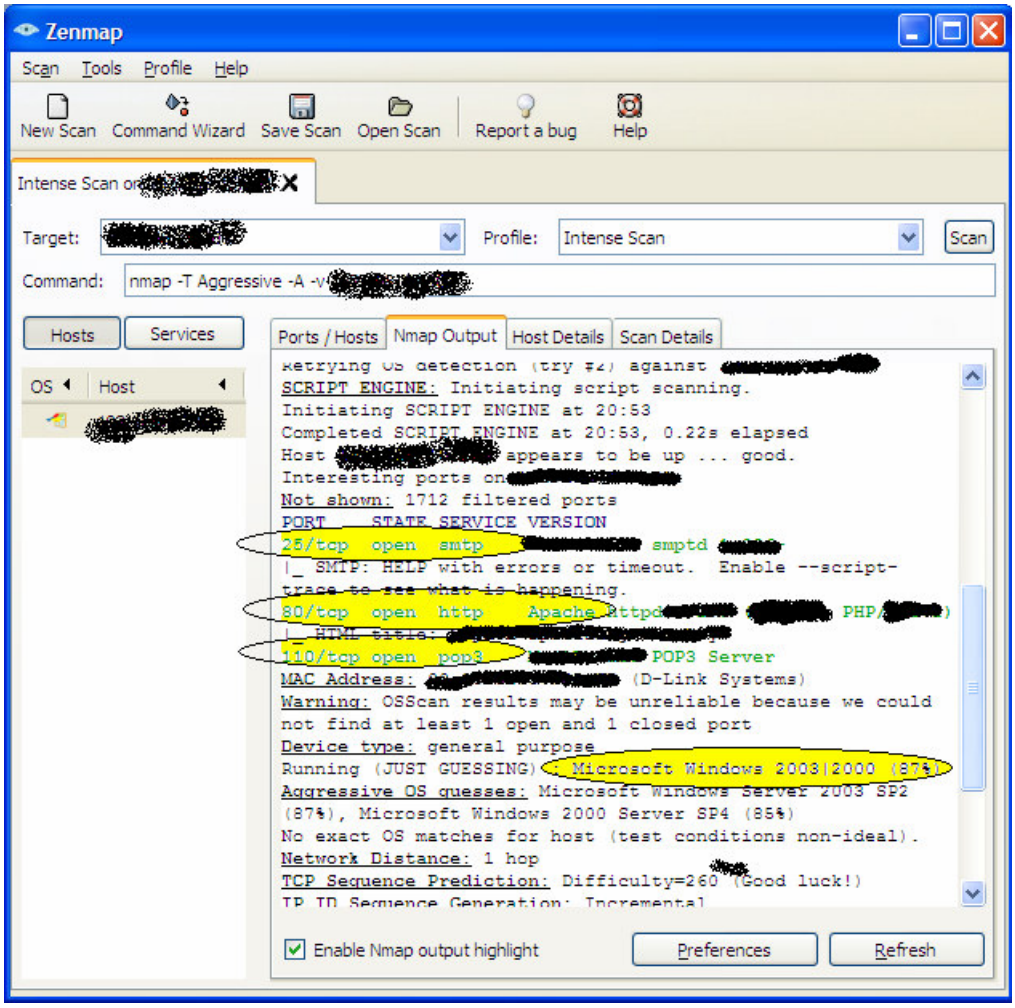


Figure 55. Host scan details from Zenmap on the Red zone with ports open (Insecure.org, 2000).

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).

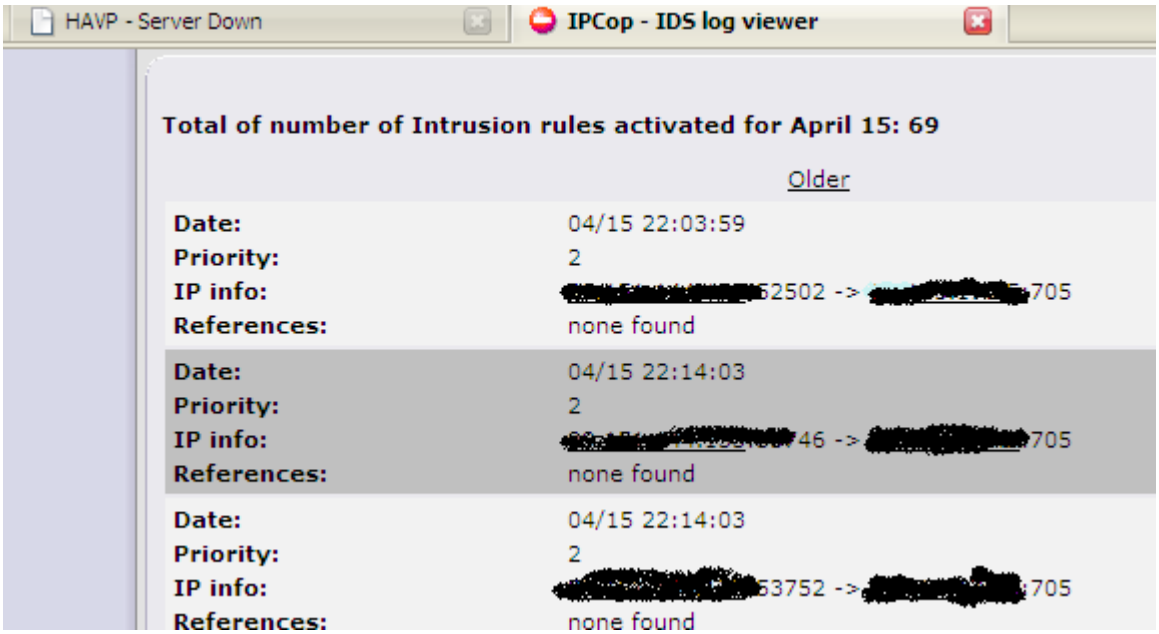


Figure 56. Snort IDS intrusion entry detail (Snort, 2007).

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).

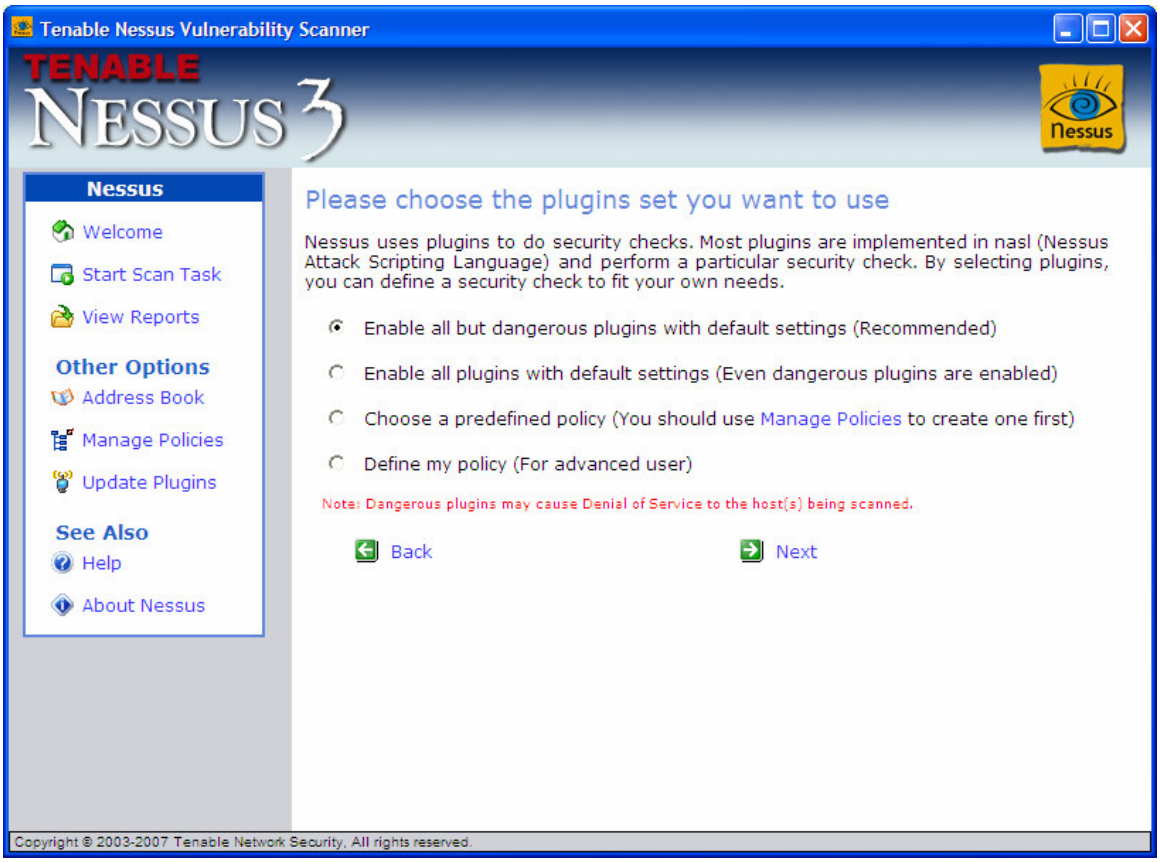


Figure 57. Scanning our UTM machine for vulnerabilities using Nessus (Deraison, 2002).

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).

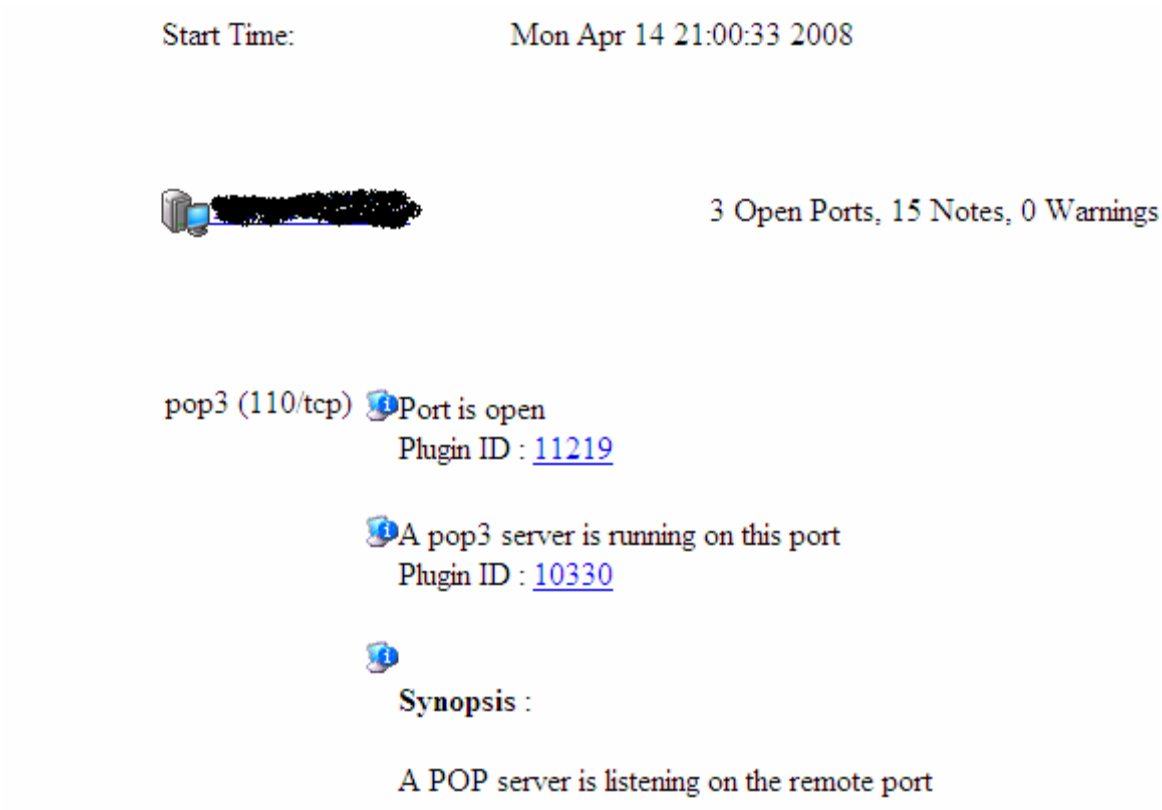


Figure 58. Nessus report of vulnerabilities found on the Red zone port forwarding open (Deraison, 2002).

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).

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).

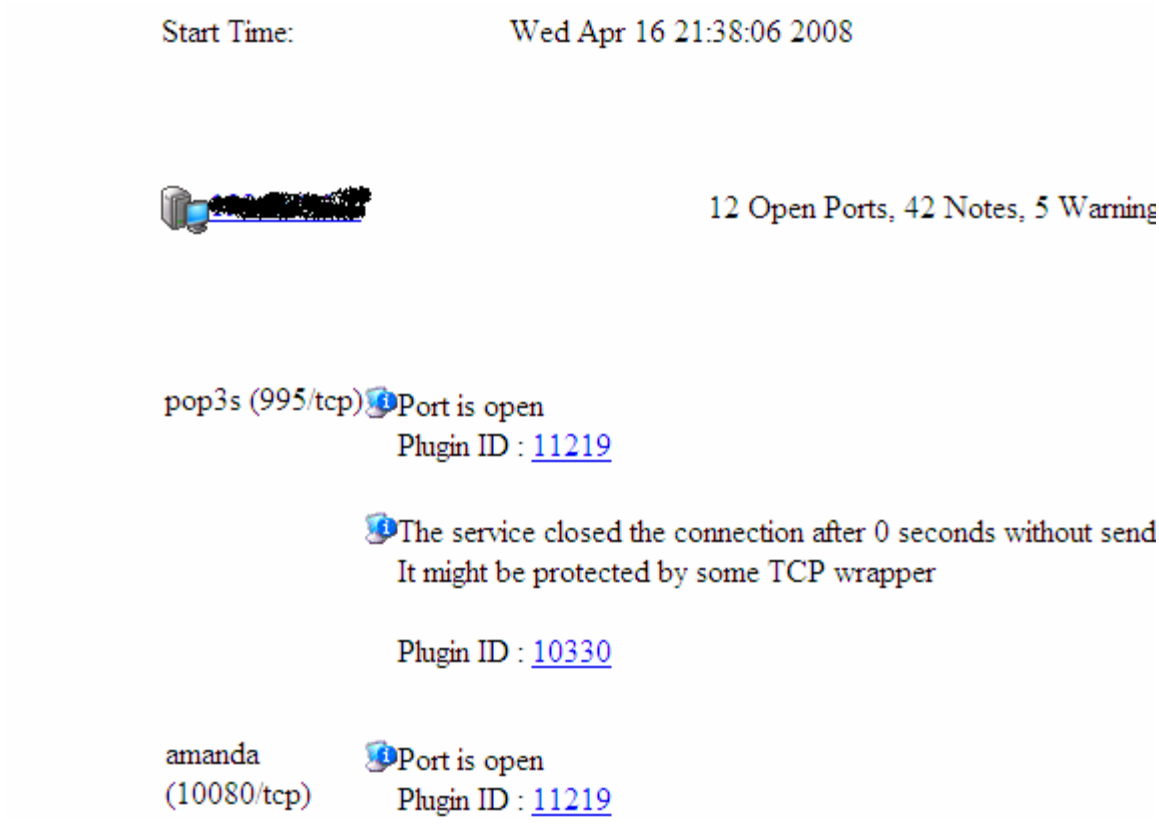


Figure 60. Nessus report of vulnerabilities found from scan within the Green zone (Deraison, 2002).

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).

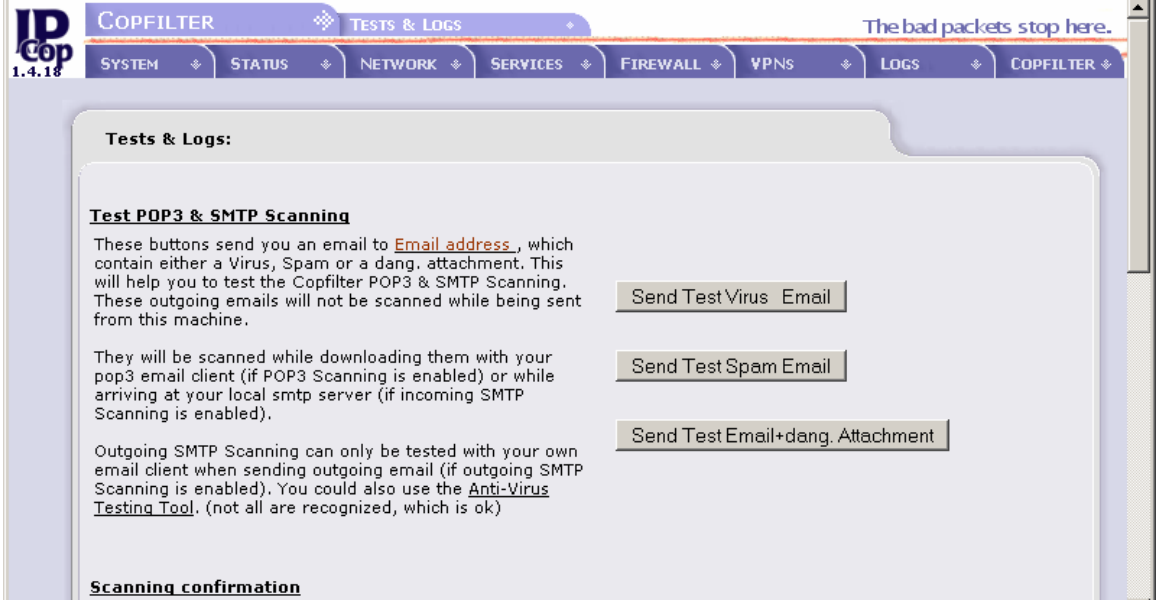


Figure 61. Copfilter virus, spam, and spam-attachment test emails (Madlener, 2002).

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).

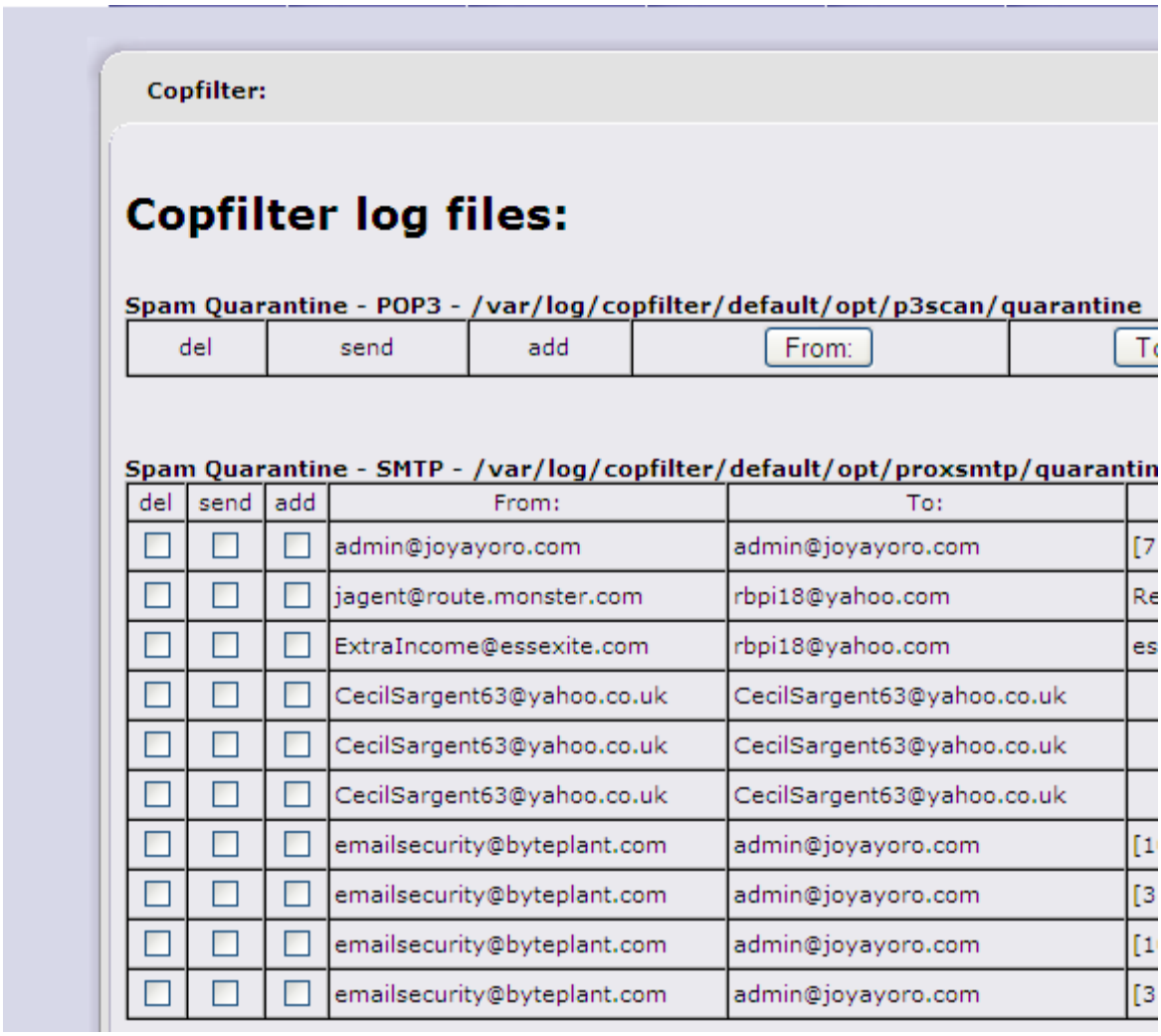


Figure 62. Quarantine virus and spam emails in the quarantine bin (Madlener, 2002).

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 virus-like 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.

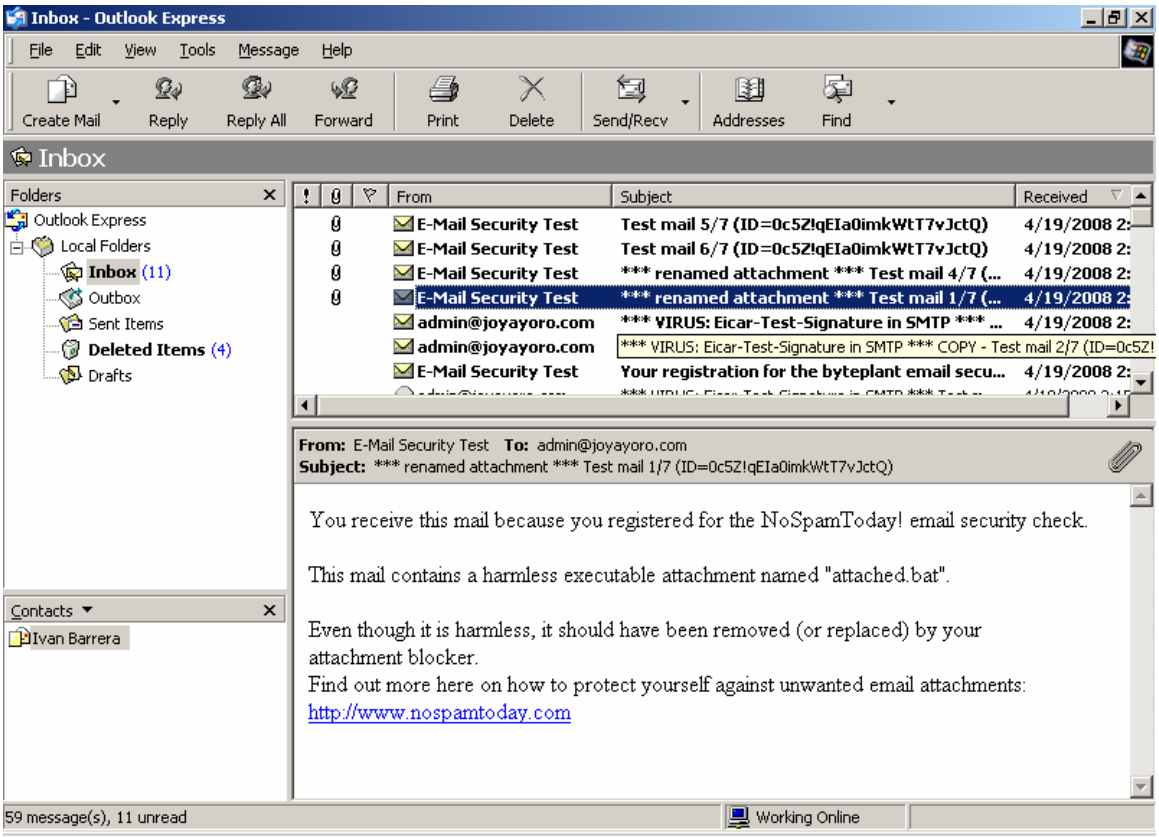


Figure 63. Seven emails from nospamtoday's automated system (No Spam Today, 2003).

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).

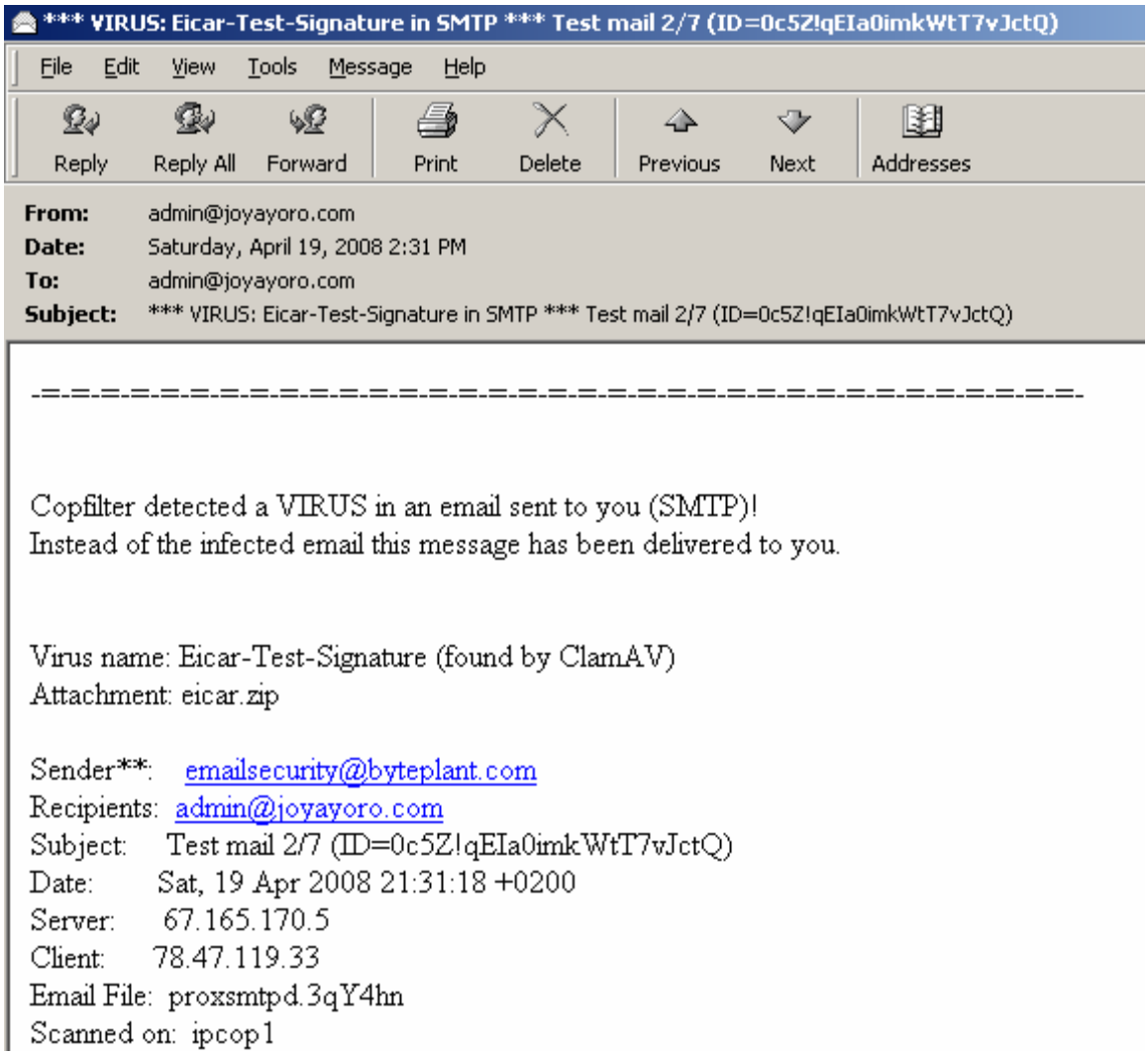


Figure 64. Copy of virus email test sent from nospamtoday and filtered by the SMTP filter (No Spam Today, 2003).

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.

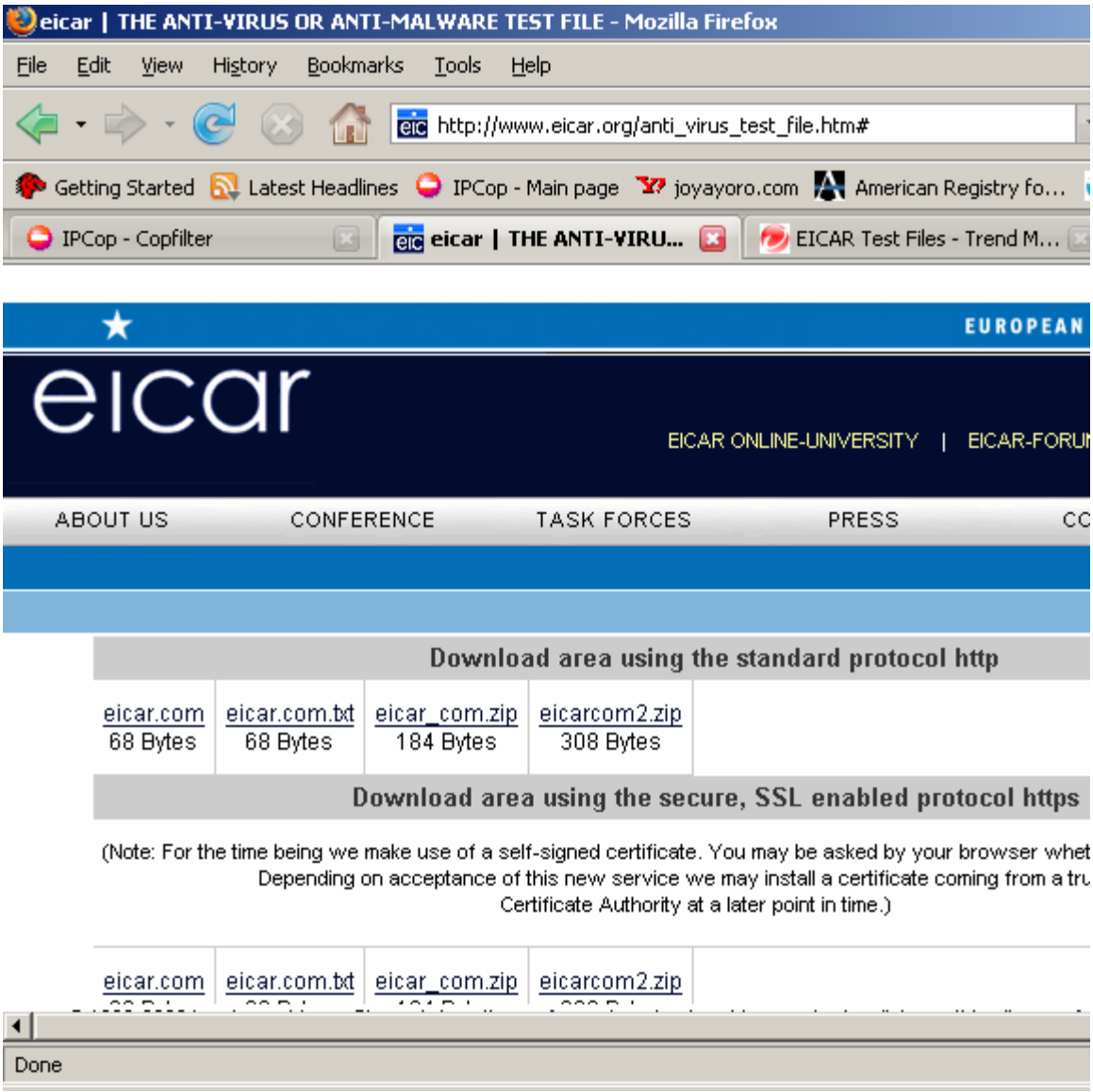


Figure 65. Eicar malware test files downloaded through HTTP and FTP method (Eicar, 2006).

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).

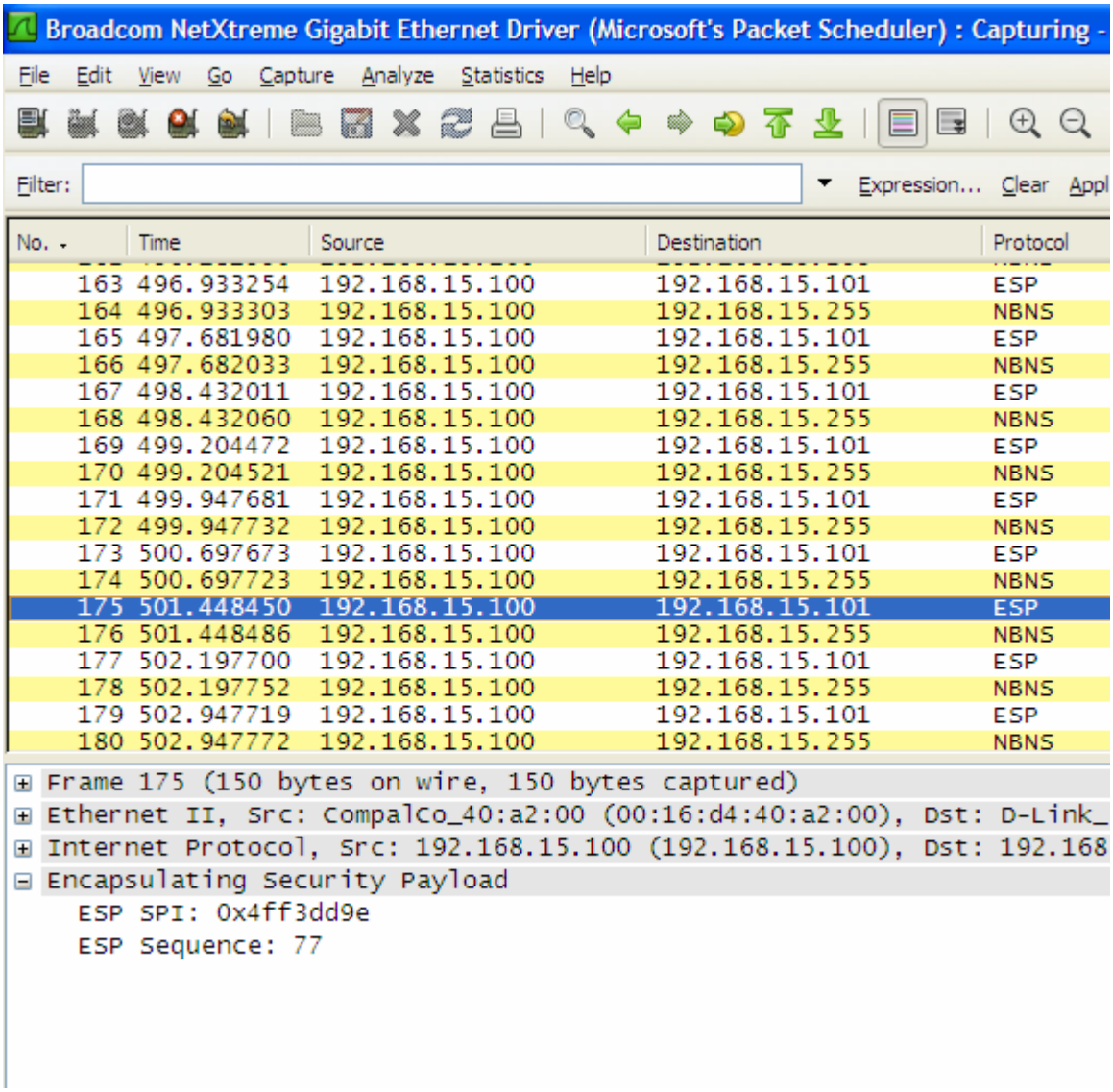


Figure 67. Encrypted ESP packets as seen by the network probe Wireshark (Combs, 1998).

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).

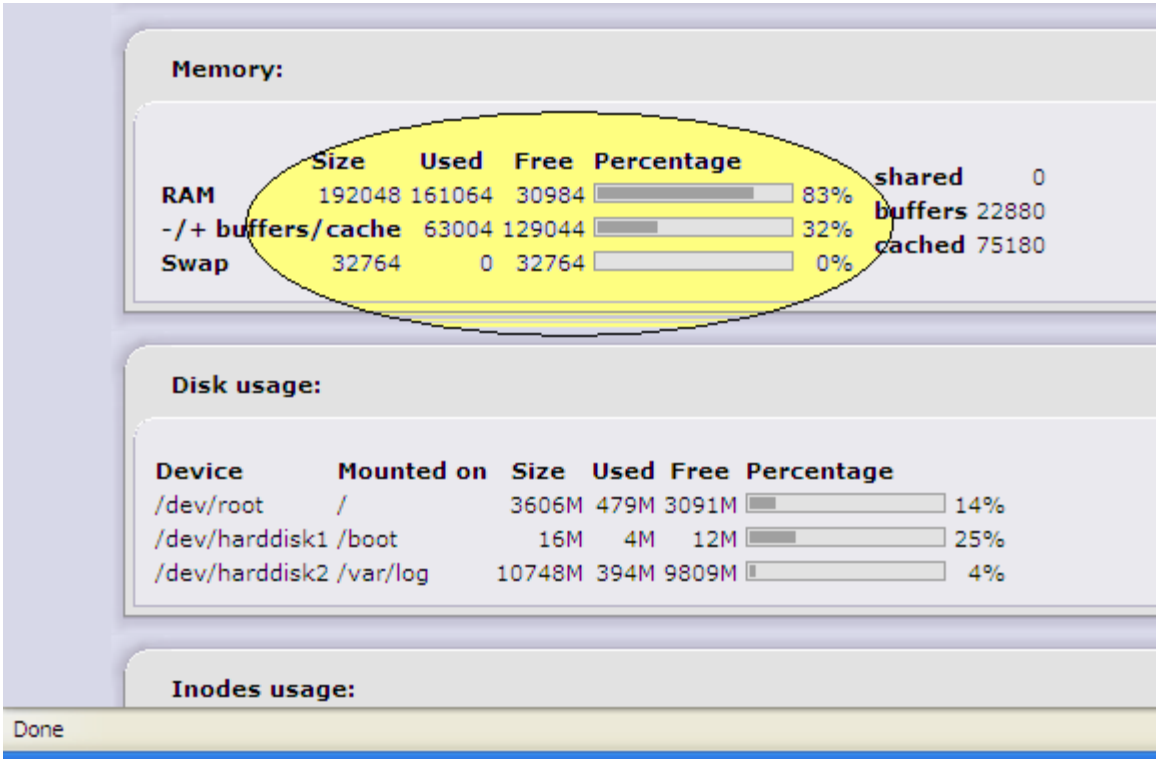


Figure 68. URL filtering eating up available memory with no other active services (Sondermann, 2004).

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: http://www.apnic.net/info/faq/abuse/index.html#2

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

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-threat-management.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: http://www.juniper.net/techpubs/software/erx/erx51x/swconfig-routing-vol1/html/ipsec-config5.html

Eicar, (2006). The Anti-Virus or Anti-Malware test file. Retrieved March 12, 2008, from Eicar Web site: http://www.eicar.org/anti_virus_test_file.htm

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
http://www.networkcomputing.com/172901783

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
http://www.crn.com/security/202404172;jsessionid=F4XSIENL0RZHSQSNDLPCKHSCJUNN2JVN?pgno=1

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=F4XSIENL0RZHSQSNDLPCKHSCJUNN2JVN?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: http://www.urlfilter.net/

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

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: http://www.nessus.org/nessus/

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: http://www.copfilter.org/

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 http://www.thegreenbow.com

(This page was left blank on purpose)