



Elastix & Network Security Guide

Elastix[®] Network & Security Guide

First revision – January 2011

Bob Fryer

Elastix & Network

Security Guide

Author	Bob Fryer
Organisation	Blue Packets (ACT, Australia)
Date	09/01/2011
Revision	1.0
Level	Beginner/Intermediate/Advanced
Date for Review	30/03/2011 or Elastix 2.04 Release
Relates to	Elastix 2.0 – and some earlier versions
Licence	GNU/FDL
Contributors	

Introduction

Security is a very broad subject and rightly so. It is a very subjective topic as well, and to a certain degree it is a subject that will never have a definitive end. That's why I generally dismiss any book that claims to be a Definitive Guide to Security. It is a constant living subject, with improvements, changes, retractions and even changes in thinking and direction every year.

There is also no subject like security that stirs up the emotions, especially when statements are made as everyone has their own views and ideas.

Security is also not going to be fixed by one device that fixes everything. It is a set of tools, backed up by procedures, and ultimately backed up by diligent review and monitoring.

Security is only as good as the weakest link in the chain, which is why this guide covers Network as well, but no matter how well you follow this guide, implement its measures, follow procedures, it will always come undone by something very basic, something you have never considered to be an issue or a link in the chain of security.

This document is not a definitive guide on Security. I won't even promise you that by following this guide that your Elastix system will not be hacked. Only you can continue to work on this side of things, learning more about security, implementing new measures as you feel are needed.

Likewise, Security is as much as you want to make it. You may be able to secure your system to cover 70% of your system using tools/products that you have and no further hardware, you may be able to cover 95% of your system with a few hundred dollars, but to get to that 99%+, it could cost you thousands of dollars, and you may still have that 1% chance that someone gets through.

This guide will provide you with an introduction on tools and techniques you can implement to cover that 70% to 95%. This guide will describe some of the common techniques that these intruders use as well as tips and tricks to lessen the possibility of an intruder will make a successful intrusion into your system.

When implementing security I personally working the basis of four layers which generally come down to the basics

- **Firewall**

Most people know what I mean here, and the prime security measure needs to exist on the perimeter of your network. I have no issue with it existing on your Elastix system, especially as a properly implemented Linux IPTables is probably regarded as one the best firewall implementations, in fact many Firewall Appliance products both commercial and Open Source are based on a Linux Kernel with IPTables. However as a general rule it is always best if the primary firewall is separate from the product you are protecting. Have a think about it, wouldn't you prefer that you have some distance on the product that someone is trying to hack (in this case the firewall) and the product you are protecting. It adds that extra layer.

- **Authentication**

Very simple, but often very little thought goes into it. The number of systems I see in place where the implementer has used simple passwords, in many cases, the password is the same as the extension number. The use of a numerical password is also just as bad. It doesn't take long for a "Script Kiddie" to run a "password guesser" on your system, especially where the password is all numbers.

Authentication also comes down to the encryption of the password if it is possible.

- **Obfuscation**

Big term for nothing much, but it aptly describes what we are trying to do. Basically it means "to make things not clear".

This has a very successful application in the security world. Why make it clear to possible intruders by giving them a roadmap. If a possible intruder finds everything is laid out before them, they will use it, and it's the same as these "Script Kiddies", they expect everything to be as they expected, which is why these scripts have reasonable success.

Changing the system from defaults won't stop the good intrusion attacks, but it will definitely make it harder for all the others to attack your system. Unless they have a vested interest in the intended target, they will normally turn away from your system and look for an easier target, especially if the system is an automated script.

- **Monitoring**

No Firewall is 100% foolproof, no network is 100% static (never changed), and attackers are trying new measures every day. You cannot setup a firewall and forget. Constant monitoring of your Firewall or Intrusion logs is necessary.

In this guide you will find that many of the techniques and ideas implement these either all together or across a range of implementations.

As I mentioned before, just implementing these ideas will not all of a sudden make things secure. It is also a case of monitoring your system on a regular basis, investigating what you see. Many systems that are "hacked" are generally not monitored, and if they had monitored on a constant basis, they would have caught the issue before it moved to a full blown attack.

Script Kiddies (I refer to them as Kiddie Scripts)

Silly name – in fact the name belies the costly possibility of what can occur to your system. Generally these are wannabe hackers, but it extends far more than this. Quite often their information they gain is distributed around their groups, sometimes using the IRC channel, so that others can either use the information or use it to gain further access to your system. But these scripts are generally generic in nature, running over hundreds of systems looking for a way in.

Security – A big beat up??

You might be wondering if this is all a big beat up, not worth the time. You might be having troubles wondering if these scripts and hacks are real. Have a look at this video on <http://enablesecurity.com/products/enablesecurity-voippack-sipautohack-demo/>.

This is a GUI tool they are using but it clearly shows how simple these tools are and how quickly they can determine a simple password setup.

Now you are probably thinking, fine, I can trace the address it has come from and prosecute them and get my money back.

Whilst we are looking at a GUI tool in the video from a company that makes tools that you can use to secure your system, the rest of the hacking world has written their own tools. These tools are faster and probably even more cunning than what this company has written, which include Random word generators, number generators, using common defaults, and include looking for known exploits.

These guys are even smarter than this, they don't run these tools from their own systems, they use other hacked systems to perform the scans, in other words they have found systems that are less secure than yours, implemented their Hacking toolkit on it, and let it run. The same when they use your accounts to make calls, they use other systems to route the calls through your system.

You might be thinking that they can't make that many calls on your system, but what is actually occurring is that they are selling your calls (basically known as Toll Fraud). This can be done by calling cards that people legitimately buy (particular in countries that are not effective to removing this type of issue), and when they ring the special number before making a call, it is ringing a hacked VoIP PBX that might connect to up to 40 other hacked PBX systems, and it tries each one until it has a trunk that works (for instance one or two might be no longer available as their owners had found them attacked). To the calling card customer, they just notice a longer than normal delay, which is not uncommon with international calls.

Still don't believe Elastix systems are on the radar, look at the list of features in the VoIPPack on this page <http://enablesecurity.com/products/voippack/>

Pointing this out is not to purposely take a swipe at Enable Security, they just produce products that allow you as a system owner to check how good your security is. However it does show that this is a real threat.

Just for the record, I have no association with Enable Security or their affiliates, nor do I own any of their products. I use them as an example as they have a very good video on how these sort of attacks can occur.

The Basics

Before we move any further forward...lets cover a few basic mistakes that many users make (I've made them as well). These are items that can immediately improve the security of your system, and should be the first items that you tackle.

Passwords or Secrets

Passwords or secrets as they are called in the VoIP world, is one of the biggest improvements you can make. One of the biggest issues is that many users make is putting simple passwords on the system while they are testing, but the system ends up moving from testing to production without a security review. It's partly due to the nature of the product, where testing needs to be done on the real carrier lines, and once this testing is complete, many would rather move forward due to pressure from the business to get it in.

The first thing to put into place, and this can even done if the system is in production, is reasonable passwords/secrets on each SIP extension you have implemented. At a guess, more than 70% of systems implemented need SIP communication with the outside world, so it is absolutely necessary to implement this measure as it is one of the simplest ways that intruders can hack your system.

These passwords need to be of a reasonable quality, as a guide the following should be used

<u>Passwords</u>	<u>Suitability</u>
201	< basically useless – one of the first passwords that they try
94932	< still poor – Script Kiddies will use a rolling number generator and try again
holiday2	< poor – Script Kiddies use a database of common words and add numbers
H883ksd3	<good - a mixture of upper and lower characters and numbers
h17kdi2993FDI29p23e2	<great - probably this and the one before would be suitable

It might be painful using these passwords, especially the last one, but use a spreadsheet or there are a few random password applications out there which are also a database holding these passwords for you and allow you to add a comment on what they were for.

Believe me, it is far better than having to explain to the boss why \$5000 worth of calls have been made by your system over the weekend. And this is not just a possible, it does happen, I have witnessed it on several occasions when asked to look at other peoples systems.

I talk about these Script Kiddies and probing your system for an opening. It is possible for them to send to your system over 40 authentication requests per second, possibly a lot more with multiple machines, so just a simple number or common word is not that hard to crack and the majority of the time, you are totally unaware, so the attack can last for days before you might get wind of it. If you take time to think about it, over a week, they can try 400,000 combinations. It's not impossible for them to make a correct guess, especially if you have used weak passwords or passwords based on English words.

It is not just the possibility of being hacked, but also the possible impact on your Elastix system and your network. I have seen routers that have not been able to handle the voraciousness of the attack and to the Network manager, it looks like either his/her Internet or Router has failed. He reboots his/her router, it works for 20 minutes and it happens all again. Likewise, it almost makes VoIP trunks useless as the communications is so broken up, or the packet loss makes it sound like static is on the line.

So far have been speaking about passwords, and in most cases they apply to the extension passwords/secrets, however, this same rule needs to apply to your Trunks either to your Voice Provider (VSP) or even other SIP devices such as ATA's or GSM Gateways. Whilst in Elastix they appear to be treated as different devices, but to Asterisk they are just another SIP device, and likewise to the intruder, it is another SIP channel in which they may gain access to your system.

Your VSP may only provide you with a number as a login name, and a number as a password, and to be honest, this is a little more secure than a simple three digit easily guessable extension number and a short number as a password, but like anything, see if your VSP has a way of changing the password to something a little more substantial. Don't panic if you can't, we have other measures we can put in place, but as I mentioned before, the more layers, the harder it is.

Turn off Allow Anonymous SIP

Turn it off!!!!

Unless you need ENUM functionality, which most do not, you do not need ALLOW ANONYMOUS SIP turned on. Most businesses have never heard of ENUM (which generally means that there is no demand for it), at least until it becomes simple to register and even then DUNDI appears to have a better way of implementing this sort of service, lessening the security implications.

Anonymous SIP is not a huge hole in your system, but if you can think of it, it is basically a Sand Pit that someone can plug away looking for or trying out vulnerabilities. Why would you want to do that??? In fact for any client, I ask a client to sign a document stating that they understand the security implications of this option, before I turn it on.

What's worse is that users turn this on believing it's a magical fix for their VSP connection which was failing up until they did. It didn't fix anything except open up the front door to allow anyone to walk in which also included their VSP. It's especially annoying when you see users (not just Elastix users) proclaiming it as the fix for many issues, especially connecting to either VSP's or devices that they wish to connect to their system.

To be fair, some the device manuals for products that users wish to connect, have very simple configurations as examples. The issue is that many people follow these examples (which generally are pure Asterisk configuration file examples), and find that turning on ALLOW ANONYMOUS SIP is the only way to get it working. The goes for some of the VSP Setups as well. They don't employ any authentication (or even simple host IP authentication), and as such, the only way to get the

connection working is again turning it on. A quick rule is that not all VSP's are created equal. I personally refuse to use any VSP that does not support a basic layer of authentication.

The same goes for devices, if it doesn't support a strong authentication method I will not use it. However that said, most VSP's and devices can employ a strong authentication method, but you may have to learn how to utilise this method and write a suitable SIP configuration. It may mean that you have to perform some SIP debugging to see what the device is sending so that you can provide the correct responses, especially as you implement the authentication. This may sound like hard work, and for the beginner it is, but truly understanding the security implications of your system will hold you in good stead.

Now having said this, be careful, especially on production systems with just turning this option back off, especially if you were not the person who implemented this Elastix system in the first place. Many have just turned this option off and found that several hours later, they couldn't make mobile calls or worse still weren't getting phone calls coming in because they didn't realise a VSP or device was not authenticating properly and was using this ALLOW ANNONYMOUS SIP to work around the issue. It is best to make this change after normal hours, reboot the system and perform a full range of tests, including restarting other SIP devices.

Don't install additional products on your Elastix system unless you really have to

One of the biggest issues I see with Elastix systems is the disregard for security by implementing products on the Elastix system that compromise security, or even if it is not the product itself, then the product allows the user to compromise security without knowing it.

I have come across an Elastix system where the system owner had decided to use the Elastix Server as an FTP Server as well. Someone installed FTP and configured it. What they did was someone used a well known exploit of that FTP Server to give themselves root access to the entire system. Not only did they get root access, but they installed a hackers toolkit onto the system. Naturally this spent many months searching for other PBX systems to hack, reporting back to an IRC channel with any system that it hacked with all the details. In the meantime, the reliability of their PBX suffered and as did their internet connection (and they got charged for excess Internet). Their IT Providers response was to disconnect the IP PBX system for 5 minutes and reconnect. It gave them a reprieve for a few hours, maybe a day or so, but off it went again. These toolkits are not stupid they also have a sleep function, which activates on a regular basis, which can also include when it recognises the environment changes (e.g. reboot or cable disconnection)

Installing Webmin is another issue. It is such a powerful package that requires rights to the system. The Internet is covered with issues relating to hacks taking place on Webmin, and to Webmin's credit many of them do not appear clearly successful, however the product itself allows the inexperienced user to make changes to the system without considering the ongoing security implications or understanding them. Some say they install it to get the mail working, or to implement IPTable firewalling. If you need this product to perform these tasks, then I strongly recommend that you take the time to learn how to perform this at the configuration level.

Use the Permit/Deny options in FreePBX/Unembedded FreePBX.

Device Options

This device uses sip technology.

secret	d882ueUee92e2
dtmfmode	rfc2833
canreinvite	no
context	from-internal
host	dynamic
type	friend
nat	yes
port	5060
qualify	yes
callgroup	
pickupgroup	
disallow	
allow	
dial	SIP/201
accountcode	
mailbox	201@device
deny	0.0.0.0/0.0.0.0
permit	172.22.22.0/255.255.255.0

As you can see, you have the ability to add the range of IP addresses that you will allow the phones to connect from. The DENY line as it is set above disallows all addresses (classic deny all then allow specific). The permit line is set to allow only a SIP device the local network to connect to this extension.

The same goes for access to the Asterisk Management Interface (AMI) which is set by the Asterisk API manager in FreePBX (tools). Only allow the addresses that you need to access the AMI interface, using the same Permit/Deny options.

That covers some of the basics that you can implement almost immediately and will provide a greater level of security than you had previously. They don't involve a high level of skill to implement, they don't involve you having to install any additional software, and realistically they should take very little time to implement.

We will now move onto some of the more advance ways of increasing your security

Advanced Security Measures

Perimeter Firewall / Router

First of all, let me be very clear, not all routers/firewalls are created equal. The number of people who have good intentions of setting up a Firewalled Elastix system, but ending up opening their router/firewall to get their Elastix system working reliably. And it happens this way because their “tested” system went into production, and a day or so later (sometimes the same day), started dropping calls or worse still not receiving calls at certain times of the day. This is generally attributed to being forced to setup of Port Forwarding (Static NAT). If you are not sure what I mean, can I recommend you read the section at the end of this document titled “A simple guide to SIP and interaction with Firewall/Routers”

What this means however, to make your system reliable, is that you are opening up the ports for basically anyone to try and connect to your system. Especially as SIP is generally defined as a single port number (5060), all these Script Kiddies need to do is probe away at your Router, find the port 5060 open, and they probe for the next few days, generally without your knowledge.

Generally for your Elastix system, when you are having problems with NAT, on the forums, they recommend to forward 5060-5084(SIP - UDP) and 10000-20000(RTP - UDP). However, you only need to forward 5060 for SIP if you are using an Elastix system (even for multiple conversations). The RTP ports 10000-20000 are needed as these are used as standard by Asterisk based distributions. These can be limited based on how many concurrent calls you expect, and you can make changes to one of the Asterisk configuration files for RTP, but seriously, you have to open at least some, so why not leave it at the standard expected by Asterisk, and concentrate on other security measures.

These other security measures are by utilising your Firewall function of your Router/Firewall. NAT by itself is not a true firewall but it can end up performing many of the same functions. Generally NAT by itself has no packet filtering, it checks the header to see what port it requires, and if a NAT rule is in place, it forwards it to whatever address you provided in the rule. Most good router/firewalls provide port forwarding (NAT) and also a Firewall as well. A good firewall will allow you to block or pass packets based on port number, destination address, source address and what IP type (TCP/UDP), and generally both ways (in and out)

With a Firewall/Router that provides proper firewall capability, means that even though we have opened up the ports needed for SIP and RTP, we can restrict them to what addresses they communicate with. Now this requires a bit of research on your VSP’s website, or possibly an email to your VSP to find out the public addresses that your VSP uses for their SIP Servers. Don’t just assume that the one you are connected to is the SIP Server address that you need. Some good VSP’s use multiple servers for redundancy, and may be using a round robin DNS to balance out the load.

With these addresses, you can now use your firewall to only allow SIP and RTP to come from those addresses on those particular ports. To anyone else trying to access these ports, they will see the ports closed, which is exactly what we want.

So take the time with your router/firewall and learn how to use the firewall function. If your router/firewall does not have this firewall ability, then look to invest in a better router/firewall.

I thoroughly recommend using a perimeter firewall. It makes plain sense not to even allow the attacker inside your network to give them any opportunities. Furthermore, with good change practices in place, you can do what you like inside your network knowing that the perimeter firewall is in place and unchanged, protecting you at all times.

Alternatively if you cannot implement a perimeter firewall, wait for Elastix 2.04 which has a GUI Based firewall based on IPTABLES built in, which is covered briefly in the next section.

Elastix Firewall (This is in Elastix 2.04 which is currently Beta at the time of writing)


As of Elastix 2.04, Elastix has a GUI Based IP Tables Firewall. Many commercial Firewall products rely on IP Tables as the basis of their firewall, and the developers have done a great job of implementing an IP Tables firewall with a very nice looking Web Based GUI.

Personally, I prefer to use a perimeter based dedicated Firewall as the first line of defence, but if you haven't got this capability, then the built in Elastix firewall in 2.04 is the next best thing. Or if you want, use a perimeter based firewall as well as the Elastix firewall, it adds that extra layer of security which I always recommend.

Below is a screenshot of the Firewall GUI that is in Elastix 2.04. The Elastix developers have made it easy by implementing a default set of rules which activate when you activate the firewall. It covers all of the communications rules needed for the products currently included in Elastix. Just be aware that it has a few GUI bugs and there is a rule missing to allow YUM and Freepbx updates, but that's why it is in Beta. But when it is released as stable, it is a fantastic tool to use.

Delete	Order	Traffic	Target	Interface	IP Source	IP Destiny	Protocol	Details
	1			IN: lo	0.0.0.0/0	0.0.0.0/0	ALL	
	2			IN: ANY	0.0.0.0/0	0.0.0.0/0	ICMP	Type: ANY
	3			IN: ANY	0.0.0.0/0	0.0.0.0/0	UDP	Source Port: ANY Destiny Port: 5004:5082
	4			IN: ANY	0.0.0.0/0	0.0.0.0/0	UDP	Source Port: ANY Destiny Port: 4569
	5			IN: ANY	0.0.0.0/0	0.0.0.0/0	UDP	Source Port: ANY Destiny Port: 5036
	6			IN: ANY	0.0.0.0/0	0.0.0.0/0	UDP	Source Port: ANY Destiny Port: 10000:20000
	7			IN: ANY	0.0.0.0/0	0.0.0.0/0	UDP	Source Port: ANY Destiny Port: 2727
	8			IN: ANY	0.0.0.0/0	0.0.0.0/0	UDP	Source Port: 53 Destiny Port: ANY
	9			IN: ANY	0.0.0.0/0	0.0.0.0/0	TCP	Source Port: ANY Destiny Port: 22
	10			IN: ANY	0.0.0.0/0	0.0.0.0/0	TCP	Source Port: ANY Destiny Port: 25
	11			IN: ANY	0.0.0.0/0	0.0.0.0/0	TCP	Source Port: ANY Destiny Port: 80
	12			IN: ANY	0.0.0.0/0	0.0.0.0/0	TCP	Source Port: ANY Destiny Port: 443

Adding a new rule is simple as per the screen shot below, and for anyone that knows how to implement IPTABLE rules, they will find this extremely easy to follow.



The screenshot shows the Elastix web interface with the 'Firewall' tab selected. A 'New Rule' dialog box is open, allowing configuration of a firewall rule. The dialog includes fields for Traffic (INPUT), Interface IN (ANY), Source IP (0.0.0.0 / 24), Destine IP (0.0.0.0 / 24), Protocol (ALL), and Target (ACCEPT). Buttons for 'Save' and 'Cancel' are also visible.

If you can't wait for the Elastix Firewall product, then you can implement this same security by using IPTABLES at the Linux command prompt and manually configuring the configuration files. To try and turn this guide into an IPTables implementation guide is beyond the scope of this document, but on the Elastix forums, you will find numerous posts on how to implement IPTables manually, as well as hundreds of guides on how to best implement IPTables rules.

Fail2ban

Fail2ban is a very unique tool in that it attempts to identify Brute Force and Script Kiddie break-in attempts and then implement rules automatically to block these attempts. This all occurs automatically and is without your intervention.

It actually fulfils a role of both active blocking and monitoring of these attack attempts.

With Fail2Ban, it monitors the number of failed authentications from a particular IP address. If failed authentication attempts matches the number set as part of the Fail2Ban configuration, then it blocks that IP address for a preset time. This generally causes these Script Kiddies to look for another system to attempt to break into. Likewise these dictionary attacks mean that they only manage to try three login/passwords every 10 minutes, instead of 40-50 per second on a system without Fail2ban. The chances of someone successfully guessing your login and passwords on your system are greatly diminished.

Fail2ban works in conjunction with IPTables, so you need to setup your IPTables first. Elastix is also introducing a Web Based GUI for Fail2Ban very shortly to complement their IPTables GUI, but as yet is not in beta, but I suspect it won't be too long. Again the forums have info on implementing this manually if you can't wait.

Other Security Measures

Changing the default port for SSH

One of the most common areas targeted for attack is Elastix systems with Port 22 exposed to the world. Some will sit back and ask why you would do such a thing, others feel that they can't do without it, for remote access. Personally I wouldn't leave it open, but if you really have to, then change the standard port number.

This is a relatively simple thing to do and in the example below I am changing the SSH port to 6222

Add a line in the file `/etc/ssh/sshd_config`:
Port 6222

Reload sshd by
service sshd reload

This will have now changed the port to 6222 for SSH, and while this is not foolproof, it implements one of the layers I spoke about called Obfuscation, by not providing the attackers with a roadmap. Change the landscape that they are expecting, and reduce the chances of being attacked.

Utilising VPN's where possible especially for remote Phones

More and more Firewall/Routers have the capability to act as a VPN Endpoints. What this means is that you can implement a secure VPN between your Network (where PBX is situated) and the Network at the other end (where the Remote Phone is located).

Alternatively, you could implement OPENVPN on your Elastix system, and select Yeastar T28 phones as your remote phones which have a OPENVPN Client built into them. What this means is that you can select any port on your Router/Firewall for the VPN to come through on. This has the benefit in that you have to worry little about the infrastructure at the remote end, knowing that you can pre-configure a phone and let them install it.

It might sound like a bit of work, but it comes down to how serious you are about protecting your Elastix system from attack.

Monitoring

For many Elastix users, once they have implemented their firewall or other security measures, they sit back believing everything is covered. They might have spent a day monitoring it to make sure no mistakes have been done, but after that, it does not get a thought until something goes wrong, or appears to be going wrong, or they had a lazy afternoon and decided to have a look.

Like backup systems, you can't just assume that the backup is working, only to find out when you need it most, that the backups haven't been working for several weeks. The monitoring of your Elastix PBX is just as important

You need to set a regular procedure to check the logs to monitor for these possible attacks. It won't take long, but you need to do it regularly.

One of the log files that you need to review on a regular basis is found at

`/var/log/secure`

```
Oct 13 09:01:24 asterisk1 sshd(pam_unix)[10106]: authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=64.186.160.100
Oct 13 09:01:28 asterisk1 sshd(pam_unix)[10108]: authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=64.186.160.100 user=mailnull
Oct 13 09:01:32 asterisk1 sshd(pam_unix)[10110]: authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=64.186.160.100 user=nfsnobody
Oct 13 09:01:37 asterisk1 sshd(pam_unix)[10112]: authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=64.186.160.100 user=rpcuser
Oct 13 09:01:41 asterisk1 sshd(pam_unix)[10118]: authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=64.186.160.100 user=rpc
Oct 13 09:01:45 asterisk1 sshd(pam_unix)[10131]: authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=64.186.160.100 user=gopher
```

The above example shows someone attempting to break into the system via SSH. You can see the users names that they are trying, and in this case they are trying common user names using on Unix/Linux systems.

You will find the archived logs named secure.1, secure.2 and so on. If your system is under heavy attack from one of these Script Kiddies, then you may find that these archived files may contain attack attempts just from the one day.

Another file to check regularly is

`/var/log/audit/audit.log`

This shows the login successes and failures. This is basically the Linux audit system. You mainly are looking for unusual login failures which will give you an idea that your system is under attack.

Appendix 1 - A simple guide to SIP and interaction with Firewall/Routers

For many users, their first need to fully understand Network Address Translation (NAT) on their Firewall/router has been forced by the use of SIP by their Elastix system. The reason why they need to understand it is because it is probably the first application where, in most cases, the standard dynamically created NAT doesn't always work. Dynamically created NAT is where your router understands that you have initiated a conversation from within your own network, and the router, for a predetermined time opens up the port to the outside, to allow the traffic back in from that address that you commenced the conversation with.

The reason why SIP via Network Address Translation (NAT) fails is due to the fact that IP telephony (utilising SIP) does not just use a single port like many applications. It is actually a two way negotiation, where within that negotiation (SIP Protocol), they decide what ports they are going to stream the audio conversation (RTP) on. For the vast majority of routers/firewalls, they do not inspect the SIP packets and therefore do not open up the ports dynamically.

The other major reason for failure of SIP utilising NAT is due to the NAT Sessions collapsing due to the lack of SIP Traffic when a conversation is taking place. This predetermined time can vary from router to router, it can vary whether your router is hard set with this timeout, or you have a setting on how long the NAT session lasts after the traffic has ceased (e.g. Cisco Routers) . How many forums have you read where someone says their call drops after 3 minutes, or their call drops after 10 minutes. This normally comes down to the router having collapsed the dynamically created NAT session. Why does it close this session?? Generally it's because no further traffic was received either way on the SIP Port, which can happen, as now all the traffic is on the RTP Port and only very occasionally a SIP packet might be generated from the either system (Elastix or your VSP). Most connections use a PING/PONG Keepalive, which generates SIP traffic which keeps the NAT Session alive, but not all.

It is generally (but not always) these combination of conditions that cause these dropouts. So for these modems, we need to implement Port Forwarding (Static NAT) to tell the router where to redirect these conversations to.