

# Internship report

Arnaud Desmons

Epitech 2006

March to August 2006



*Internet Service Provider*

This report is also available at :  
[http://arnaud.desmons.free.fr/desmon\\_a-roguelink.pdf](http://arnaud.desmons.free.fr/desmon_a-roguelink.pdf)



*Grants Pass, Oregon*

## **Introduction**

This report is about my internship at RogueLink LLC covering the period of 6 months from May to August 2006.

RogueLink LLC is a US nationwide Internet Service Provider based in Grants Pass Oregon which mainly provides dialup connections. Oregon is on the west coast of the United States.

Although broadband technologies (DSL and Cable) are all the rage right now, in reality, a lot of people still only have access to analog phone lines. Dial-up modems will remain a good way to get on the Internet for several years, especially in rural areas in the United States.

When looking for an internship I had three criteria :

1. Language and culture : I wanted to do my internship in a foreign country to improve my English and to discover a different way of life and business culture.
2. Technical diversity : a main point was to learn new things and especially not in software engineering, like I did in my former internship.
3. Responsibilities : I looked for an experience that will prove I am able to manage things by myself.

My job was to make the whole infrastructure stable, up and running. As described in this report I was focused on three main projects :



- 
- The mail server migration from Windows to Linux.
  - The web server migration because of an hardware failure
  - The billing system migration

In parallel, I had to maintain, monitor and improve the infrastructure. Each project was a huge challenge regarding both the financial and technical situation of the company and the important number of end-users. As you will see each project was managed with three main things in mind :

- **Simplicity** : I was the single administrator and a too complex infrastructure would be both difficult to explain and difficult to use for non technicians (account creation, passwords, dependencies between services). I keep it simple as much as possible.
- **Stability** : because nobody would be there after me for administration I had to make it automatized as much as possible (scheduled monitoring emails, self-learning anti-spam). Keeping simple the system also contributed to keep it stable.
- **Security** : the stability of the system also depends of its security. As I worked in this domain before I tried to keep everything secure (unfortunately in the limit of the first point).



## Contents

<b>1</b>	<b>Organizational aspects</b>	<b>6</b>
1.1	RogueLink LLC . . . . .	6
1.2	Internet Access in southern Oregon . . . . .	7
1.3	The internship . . . . .	7
1.4	Timeline . . . . .	8
<b>2</b>	<b>Technical aspects</b>	<b>9</b>
2.1	Mail server migration . . . . .	11
2.1.1	The old system . . . . .	11
2.1.2	Accounts and password recovery from the old system . . . . .	12
2.1.3	Mailboxes synchronisation . . . . .	13
2.1.4	Address books . . . . .	14
2.1.5	Mailing lists . . . . .	14
2.1.6	Mail aliases (forwarding) . . . . .	14
2.1.7	Firewall and DNS preparation . . . . .	14
2.1.8	The new system . . . . .	15
2.1.9	Database and backups . . . . .	19
2.1.10	Conclusion . . . . .	19
2.2	Web server migration (IIS, Frontpage, Webtrends and FTP) . . . . .	20
2.2.1	Why . . . . .	20
2.2.2	What . . . . .	20
2.2.3	How . . . . .	21
2.2.4	Conclusion . . . . .	21
2.3	Billing system migration . . . . .	22
2.3.1	Freeside . . . . .	22
2.3.2	Data extractions . . . . .	22
2.3.3	PayPal extension . . . . .	23
2.3.4	Conclusion . . . . .	24



<b>3 Conclusion</b>	<b>25</b>
3.1 Technical conclusion . . . . .	25
3.2 General conclusion . . . . .	26



## 1 Organizational aspects

### 1.1 RogueLink LLC

RogueLink LLC, formerly EchoWeb, was created about 10 years ago. It was a very successful company until the year 2000. The main activities of the company are :

- Internet connections : they provide dial up and adsl connections to the internet, with associated services such as email, personals web pages, hotline support.
- Web hosting and domain registration. Website design is subcontracted. Shopping carts to sell products online are also provided.
- Computer repair for customers who have problems connecting to the internet.

The dial-up technical infrastructure used to be owned by RogueLink but is actually now subcontracted by a nation wide operator named GlobalPOP. RogueLink manage their customer using a web interface from them but still have to bill customers.

While the main product sold by RogueLink is still Dial Up connections, some people request faster connections, and RogueLink has to provide ADSL connections. The company started to provide ADSL connections last year, but it was stopped as it was found to be profitable only for very important numbers of customers. Instead, ADSL connections are subcontracted to an other company in Grants Pass, whose name is VISP.

They already suffered a fire in the server room which destroyed many computers and caused the loss of many customers. At the moment, the company has about 800 customers using Dial Up connection, about 200 using DSL, and about 200 websites hosted.



## 1.2 Internet Access in southern Oregon

Most of the people in Grants Pass and around are connecting to the internet using 56k DialUp. Three reasons for that :

- Local phone calls are free, which makes it quite cheap to connect any amount of time.
- DSL or Cable connections are a lot more expensive, and are not worth the price for what most people want to do: sending/receiving emails, visiting a few web pages...
- A lot of people are living outside of the city, in the forest, therefore they don't have access to DSL or cable : Even if the number of people moving to DSL and cable is important, there remain an important number of dial up users, which is probably the reason why RogueLink is still profitable.

There are not that many big companies providing dial up internet access at low prices everywhere in the United States. However, there are still a lot of users connected to local ISP, even if it is a little more expensive. The main reason is not technical, it is the support. For experienced users it does not always make a big difference to connect to a local ISP, however for other users, they know that if they have problems connecting they can have free hotline support, or they can bring their computer and have it repaired, they can come and have their questions answered. This is what makes the difference with big ISPs, and probably why there still is a lot of small ISPs in the united states.

The wireless market tend to show up more now with company like Clearwire. RogueLink is starting to get partenariat with this company to counter the dial-up death.

## 1.3 The internship

RogueLink is used to hire trainee from both Epita and Epitech and others french school and university. I found this internship thanks to a student from Epita who made an announce on school newsgroups. It was not easy to obtain on time the J-1 visa required for this kind of internship. Indeed, it was properly the time France was not able to provide biometric passports. So the United States immigration administration was flooded of visa request for simple tourists (instead of passport).



I was hosted by and near the house of my managers, Kate and Jacques LeCompte, owners of the business. The company is split in two different places located at about 20 minutes by car from each other :

- The server room, where all the servers are is a few miles away from the downtown where the office is located.
- The office is where customers can come and pay their bill, bring their computer to have it repaired, ask questions about their dialup connection or web hosting, etc ... This is also where the hotline support is located.

I was working most of the time at the office. This allowed me to be in contact with customers and to be aware of issues as soon as possible in order to give advice to hotliners. The hotline support was really knowledgeable and Linux enthusiasts so I had no difficulties to explain things. I was working on the servers remotely, using terminal services<sup>1</sup> and ssh from my linux laptop. Sometime, I had to take my car and go to the server room.

### 1.4 Timeline

The first month was more to acknowledge the system and what would be my projects. I had also to start organising those projects to see how long it will take for each one.

The mail server migration took about two month (including the canceled outsourcing project).

The web server migration took one month.

The billing system set up took also one month.

---

<sup>1</sup>Terminal Services or Terminal Server Edition (TSE) is a component of Microsoft Windows NT operating systems (both client and server versions) that allows a user to access applications or data stored on a remote computer over a network connection. Terminal Services is Microsoft's take on server centric computing, which allows individual users to access network resources easily.

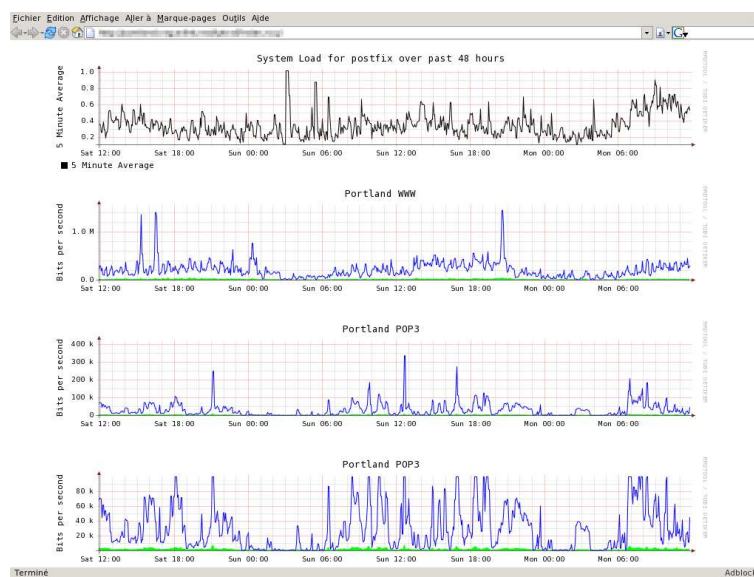




## 2 Technical aspects

This section is split into three parts. It is related to main projects I worked on, but as a recurrent background task I had to monitor services and to fix some little issues, like buying new SSL<sup>2</sup> certificates for online carts, adding new domain names in our DNS, setting up new web sites, fixing front page extensions<sup>3</sup>. For instance, I also had to delete a shopping cart customer's account directly into the SQL database.

The first week I was here, it was mainly to take notes on how the infrastructure is running. Thanks to a wiki<sup>4</sup> filled by former trainees I was able to learn more faster but some of those informations was also obsoletes. I also set up some scripts for graphing the network activity on the DMZ<sup>5</sup> :



<sup>2</sup>Secure Sockets Layer (SSL) and Transport Layer Security (TLS), its successor, are cryptographic protocols which provide secure communications on the Internet for such things as e-mail, internet faxing, and other data transfers.

<sup>3</sup>Software technology that allows Frontpage clients editor to communicate with web servers, and provide additional functionality intended for websites. Frequent security problems have marked the history of this Microsoft proprietary technology.

<sup>4</sup>A wiki is a type of website that allows users to easily add, remove, or otherwise edit and change some available content, sometimes without the need for registration. This ease of interaction and operation makes a wiki an effective tool for collaborative authoring.

<sup>5</sup>DMZ stands for DeMilitarized Zone. In computer security terminology, a DMZ is a network area that sits between an organization's internal network and an external network, usually the Internet.



## 2 TECHNICAL ASPECTS

---

We have a 5Mb bandwidth on both way thru fiber optic provided by Charter. We have 30 IP addresses (/27 block) with this connection.



## 2.1 Mail server migration

The mail migration was for sure the most usefull, difficult, sensitive task I had to do during this internship. I will explain here :

- Why this server migration was necessary and urgent.
- What was part of the migration (password recoveries, mailbox synchronisations, address books, webmail, mailing lists, aliases, administrators and administration interface)
- How I did the migration (DNS, firewall, choose of components, scripts).

### 2.1.1 The old system

When I arrived, the mail server that was used was IMail v8.05 from Ipswitch on a Windows 2000 server, which is probably one of the worst email server program available. The main problems with this server were :

1. Ipswitch Imail is RFC ignorant (RFC 1893 or 3463). For instance, when a server answer with an Error code beginning with the number 4, it means that it is a Persistent Transient Failure, not a Permanent Failure, and sending again in the the future the same message may be successful. Imail doesn't and bounce the message with a « user unknow ».

Knowing the fact that most spammers won't take the time to send again their message when they receive a temporary some people had the idea to use a method called Greylisting<sup>6</sup> to block a significant amounts of spam by first rejecting an email from an unknown server with a temporary error (with an error message begining with a 4), then accepting and adding the sending server to a whitelist when it try to send again the same email after a certain amount of time. The spams are then rejected while the emails sent using a regular email server are received. The problem is that the IMail server doesn't handle this correctly, and it was not possible for our customers to send emails to servers implementing GreyListing because of the « unknown user » problem.

---

<sup>6</sup>Greylisting is a simple method of defending electronic mail users against e-mail spam. In short, a mail transfer agent which uses greylisting will "temporarily reject" any email from a sender it does not recognize. If the mail is legitimate, the originating server will try again to send it later, at which time the destination will accept it. If the mail is from a spammer, it will probably not be retried



2. The server used to crash for no reasons and needed to be reboot very often to « solve » problems.
3. We had hackers, directly logged on windows (moving the mouse and everything... very impressive to see ! ) using our system to send spam.
4. The configuration is saved in the Windows Registry which is horrible.
5. All the configuration and management of accounts is done using a Graphical User Interface (which is not convenient for scripting tasks). The GUI is program that allow you to crash the whole mail server in a few clicks. It happened to me once but fortunately I had a backup of the Windows Registry.
6. The program lacks many options, and is not modular. For example it is not possible to add external spam filters, add special routing instructions, add greylisting.

For all thoses reasons it was more than urgent to switch this server to a quality open-sourced one.

### **2.1.2 Accounts and password recovery from the old system**

As explained the IMail windows server stored accounts into the windows registry. I installed ActivePerl, wich is a bring of the Unix Perl to the Windows platform and the Win32::TieRegistry Perl module. Then, I wrote a scripts to extract user attributs, including the encrypted password.

The password algorithm used on the old system was not a common one (like crypt, md5 or sha1) which made impossible to copy encrypted passwords directly because the new system would not be able to use them. However, thanks to a very weak and sadly amusing encryption algorithm on the old system it was possible to extract, decrypt and re-encrypt them with a usable algorithm (crypt) for the new system.



The encrypted password in hexadecimal was in fact the addition of the email address and the clear password. This was done with a few efficient lines of Perl :

```
# my $mail = "test";
# my $password = "BDD4EAE2EDD4E8";
my @hex_mail = unpack("C*", $mail);
my ($i, @decrypted_password, @hex_password);
while ($password) {
    push @hex_password, hex(substr($password, 0, 2, ""));
}
foreach (@hex_password) {
    push @decrypted_password, ($_ - $hex_mail[$i++ % length($mail)]);
}
print pack("C*", @decrypted_password). "\n";
```

### 2.1.3 Mailboxes synchronisation

Another difficulty was that the migration had to be transparent for end-users. I had to plan synchronization of their mailboxes so it will take a minimum amount of time during the migration.

I used a recurrent approach for that. The first time it was really long to synchronize mailboxes using the IMAP protocol and during this long time new messages came. So I did it again and it take a little less time. So, during this inferior amount of time a little less messages came, etc... At the end it took only about 6 hours so it was enough for doing that the night of the migration.

I also prevent a problem that would be catastrophic if not noticed. Indeed, POP3 uses special IDs, wich are named UID, to know which messages were already downloaded by the end-users. If thoses UID are not synchronized between the old and the new system, then all users would have to download all their emails again. Those would be very inconvenient for end-users that use slow dial-up connections for instance and would result in a lot of phone calls. Besides that, everybody downloading all their mails would made a huge load on our server. Thanks to the fact that the old mail server kept those id directly in messages I was able to configure the new system to use them.



### 2.1.4 Address books

In the old webmail there were address books. I had to extract those, using Perl scripts, to put them in the new webmail database which uses a different format. Old address books were stored in different files and directories for each domain and we host about 200 domains... The new database of address books was really useful for the new anti-spam system as I will explain there after.

### 2.1.5 Mailing lists<sup>7</sup>

They were also important mailing lists on the old mail server, used by hundreds of people. Because they were not so many and because it was pretty easy to set them up on the new Linux server. I did that manually and just cut and past members' email of each mailing list. The main part for this job was to explain to each administrator of each list how the new system works. It was pretty successful.

### 2.1.6 Mail aliases (forwarding)

I had to extract, using Perl scripts, each redirection rule in the 200 domains we host. That was not easy because the old system had different ways to store aliases. Most of them were stocked into the Windows registry and it was scriptable so. But for the other ones I had to extract them manually using the GUI.

### 2.1.7 Firewall and DNS<sup>8</sup> preparation

Because updating DNS for the new server can be pretty random, I preferred to use TCP redirections. So if there was a problem, it would be for everybody and a lot more easy to fix. Also it would be possible to switch back to the old system really quick. The rollback possibility was really important and I take care of that in every step of the migration.

---

<sup>7</sup>A mailing list is a collection of names and addresses used by an individual or an organization to send material to multiple recipients.

<sup>8</sup>The domain name system (DNS) stores and associates many types of information with domain names, but most importantly, it translates domain names (computer hostnames) to IP addresses. It also lists mail exchange servers accepting e-mail for each domain. In providing a worldwide keyword-based redirection service, DNS is an essential component of contemporary Internet use.



I first updated DNS for all domains we host to point to a firewall that will do the redirection. I had to use bash and perl scripting to update all domains. Then, I did the switch on the router a few days after the DNS so I was sure that everybody was using the router.

### 2.1.8 The new system

The new system improved a lot of things and I had a lot better control on what is going on with the system. It allows access to more detailed log files. I was also able to set up some cron<sup>9</sup> tasks to email the hotline support about biggest mailboxes every week for instance or to email me the mailing lists activity or the backup results.

### Anti-spam

Before, there was only greylisting thanks to a linux mail relay in front of the IMail windows server. Because the new system is a linux system we were able to do the greylisting directly on the server. There was also blacklist like relays.ordb.org, an Open Relay DataBase or bl.spamcop.net, A fast and automatic list of servers reported to send spam. But there was still spam so I setted up a statistical spam filter.

Statistical filtering, once set up, requires no maintenance per se: instead, users mark messages as spam or nonspam and the filtering software **learns** from these judgements. Thus, a statistical filter does not reflect the software author's or administrator's biases as to content, but it does reflect the user's biases as to content; a biochemist who is researching Viagra won't have messages containing the word "Viagra" flagged as spam, because "Viagra" will show up often in his or her legitimate messages. A statistical filter can also respond quickly to changes in spam content, without administrative intervention.

The task here was to provide a convenient way for customers to feed Bogofilter with spam and non-spam. It is really important to provide both spam and non-spam or the filtering would be catastrophic.

The webmail provides a convenient way for end users to report messages as spam via an HTTP link. They do it well after we explained to them that doing so will reduce their amount of spam.

---

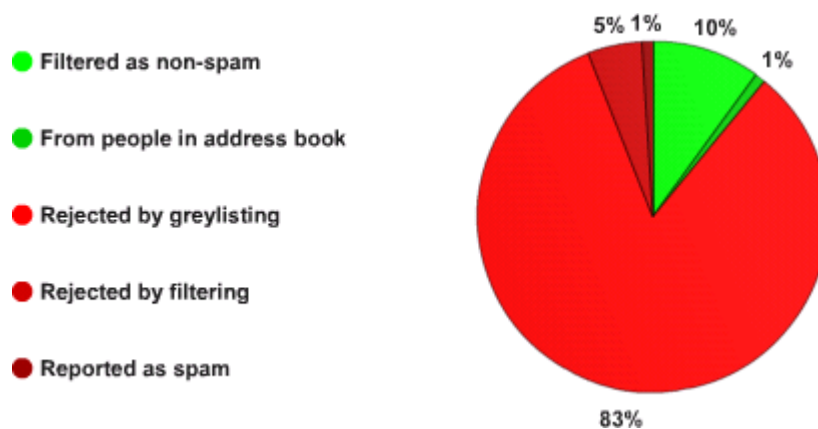
<sup>9</sup>A cron server, found in Unix and Unix-like operating systems, is used to schedule commands to be executed periodically.



## 2 TECHNICAL ASPECTS

The amount of spam reported decrease of 90% ! I assume that users receive 90% less spam than at the begining of the filter learning process. I had some congratulations from customers about that.

For non-spam, I find out that all the address books that customers have in the webmail would be a good way to mark as non-spam messages coming from people in thoses address books. I did a script on the mail server that look into the address books database for every message comming in. If the expeditor is into an address book then the message will be marked as non-spam.



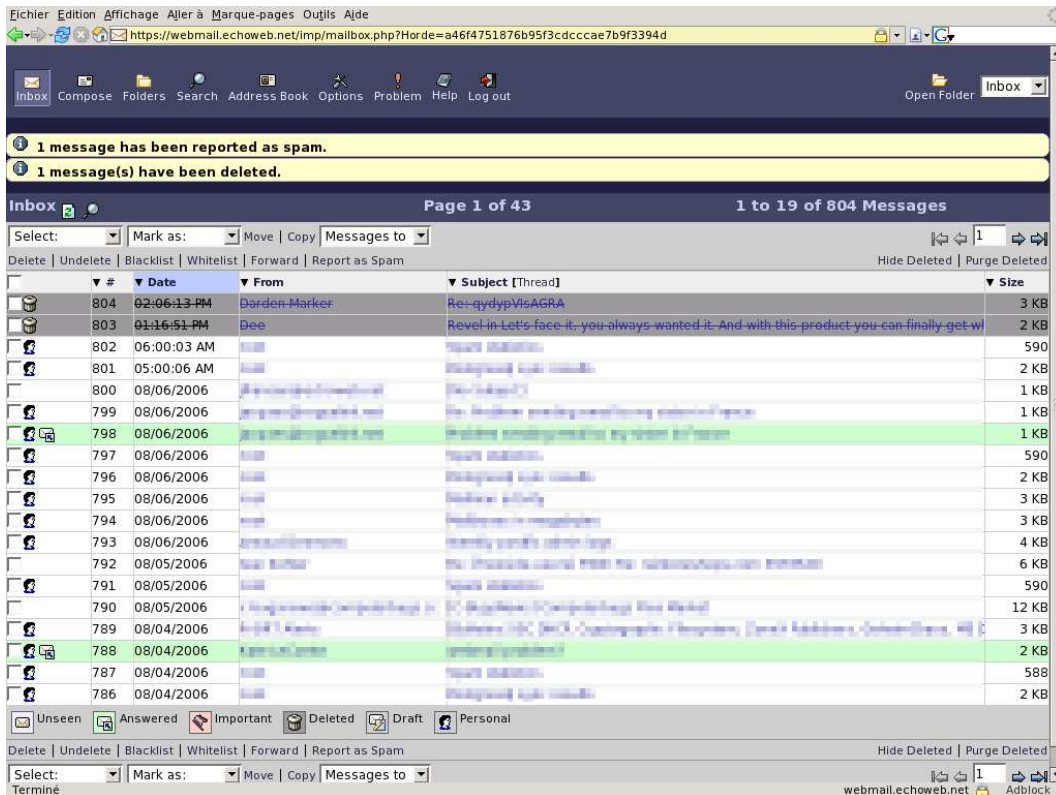
I used also what we call «spam traps». This is non-used mailboxes, with names choosen to be easily spammed like firstnames or «info», or «contact», etc... All messages received from thoses mailboxes are then automatically reported as spam to the statistical filter.





## Webmail

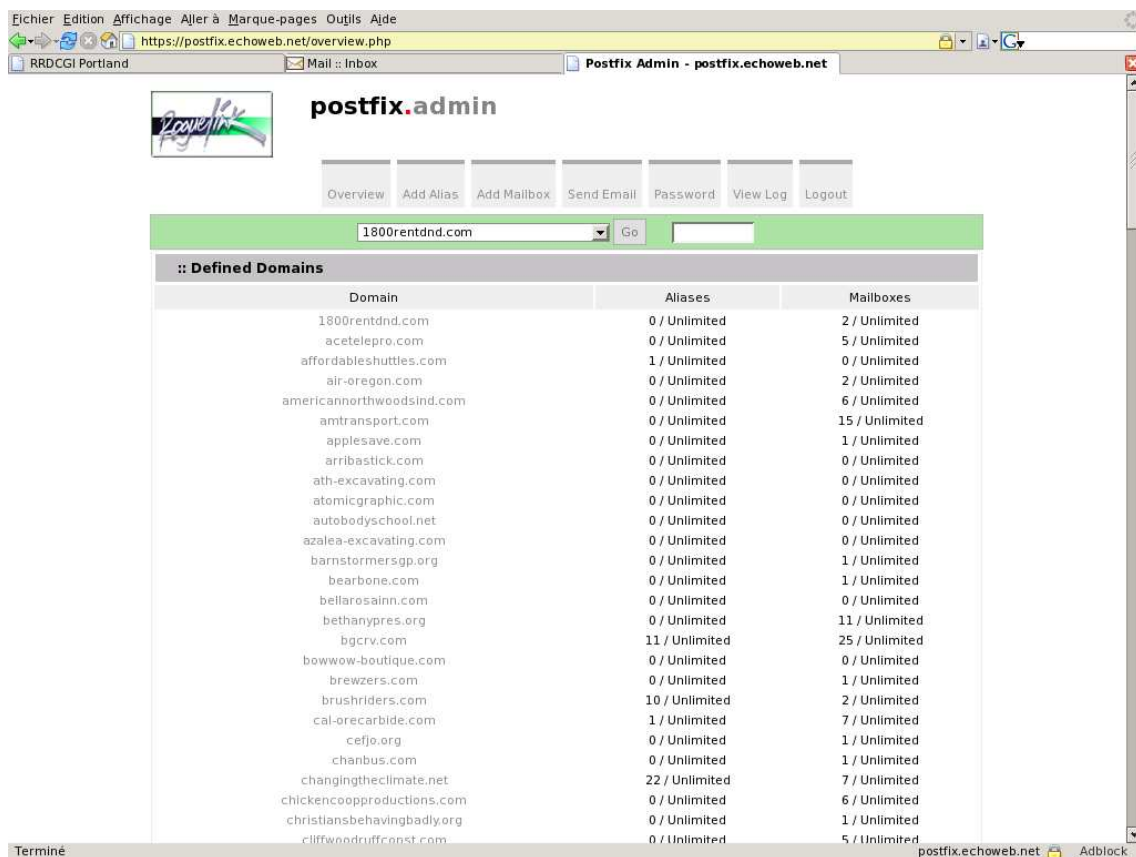
The new webmail is Horde IMP which is a famous open source webmail. It is based on IMAP and allows customers to report problems easily, to manage their address book and folders and to report messages as spam. They can also set up filtering using both withlist and blacklist but as far as the anti-spam is efficient enough they don't use it so much. There is also online help but the system is pretty intuitive. I customized a little bit default options after getting some customers feedback. I had to write scripts to update the configuration database for each customers.





### Administration interface

The new administration interface is simple which is certainly the main quality of it. The old one was really slow. Now, this is more like a frontend on the backend database used by the mail system. Here you can create, delete, and change password of those more than 2,000 accounts split on 200 domains we host. This interface is accessible also for each domain administrators with a different level of permissions. Indeed, I had to write scripts to extract kind of «flags» into the windows registry of the old system to distinguish normal user and administrators. On the new system, end-users can also change their password using a dedicated part of the administration interface. This is a security improvement (negligeable regarding the improvement from Windows to Linux) because it was not possible on the old system.





### **2.1.9 Database and backups**

The new system use an SQL database to store encrypted passwords. The Postfix SMTP server and Dovecot POP3 server both use this database to check passwords.

Thanks to this database, a backup of the accounts is made every day on another server.

Because people don't leave their messages in their mailboxes too long a bacup of them would be both difficult and not so usefull.

### **2.1.10 Conclusion**

The migration was a great success. Although I took all precautions there was still risks but I didn't have to switch back to the old system. The system is actually really stable and end-users are already used to it. Every day, domain's administrators use the administration interface and end-users report spam via the webmail.

There was a project to outsource the mail server to a dedicated hosting service but the project was canceled by my managers. Also, the greylisting was at first on a separate outsourced server but the company that was hosting this server disconnect it after a delay in their payment. I had to install the greylisting service on the new server and everything was fine but it was a huge change.



## 2.2 Web server migration (IIS, Frontpage, Webtrends and FTP)

This task was less complex than the mail migration but the difficulty here was to find out a way to automatize the procedure to avoid errors and to get it done as soon as possible. For most of the tasks I found how to automatize the migration but for some others I had to do it manually. This project was also a good opportunity because I was not used to Windows servers.

### 2.2.1 Why

When I arrived, the IIS web server (Microsoft Internet Information Services) had only one hard drive left out of the six slots available. It was more than urgent to fix that because the single hard drive could die at any time, bringing in his death the hundreds of web sites we host. We host company's website and shopping cart. Some of them process thousands of dollars a day. No need to say that any interruption of service would not be appreciated.

Because it was too risky to put new hard drives and try a synchronisation with the one left, we set up a new server. I had to automatise as much as possible web domains creation. On top of domains themselves there was a lot of related services like FTP<sup>10</sup>, Webtrends<sup>11</sup> and frontpage extensions to set up. The migration was also a good way to clean up the GUI from a bunch of old websites. The security was also improved because there was kind of « sleeping » viruses on the old IIS server.

### 2.2.2 What

I had to move 200 web domains. Which means also more than 200 ftp accounts and about 100 frontpage extension enabled domains. The database used for online shops was on a separate server and was not part of the migration. Most of the websites use ASP<sup>12</sup> scripts but also PHP<sup>13</sup> scripts.

I spent a lot of time studying the possibility to switch to a Linux server. However, because of ASP and because a lot of customers use frontpage extensions which are both not available on Linux, it

---

<sup>10</sup>FTP or file transfer protocol is a commonly used protocol for exchanging files over any network that supports the TCP/IP protocol (such as the Internet or an intranet).

<sup>11</sup>Webtrends is a software that analyses IIS log files to generate statistical reports on web sites

<sup>12</sup>Active Server Pages (ASP) is Microsoft's server-side technology for dynamically-generated web pages that is marketed as an add-on to Internet Information Services (IIS).

<sup>13</sup>Php Hypertext Preprocessor (recursive acronym) is an open-source, reflective programming language. Originally designed as a high level scripting language for producing dynamic Web pages, PHP is used mainly in server-side application software



was decided to stay on Windows. We tried also Windows 2003 but it was too expensive for the company regarding its low improvement. The migration was finally from a Windows 2000 to a Windows 2000.

### 2.2.3 How

I first copied all web directories on the new server. Then, thanks to the « Automating Administration for IIS 5.0 » documentation on Microsoft TechNet I found a way to create a new web domain from the command line using *mkw3site* and *adsutil*. I did a perl script that generated a little script for each domain based on its directory name.

For FTP accounts, I didn't find a way to automatize the process but I was able to create an account using less than 10 keyboard shortcuts (including copy-past of the password from the wiki). So I did it manually...

I tried also to protect the web server via a reverse proxy<sup>14</sup>. It was fine for web browsing but because frontpage extensions use NTLM<sup>15</sup> it was not possible to break the flow with a reverse proxy for this part and I had to remove the reverse proxy.

### 2.2.4 Conclusion

When everything was ready on the new server I just changed the DNS. It was successful, we just find out after the migration that some ASP scripts were broken because they needed some DLL<sup>16</sup> library not yet installed on the new server but it was a quick fix.

I didn't put back web statistics just after the migration and I did right because a really few people claimed them. So I put them back one by one and it was fine.

---

<sup>14</sup>A reverse proxy is a proxy server that is installed typically in front of a webserver. All connections coming from the Internet addressed to one of the webserver are routed through the proxy server, which may either deal with the request itself or pass the request wholly or partially to the main webserver. This is done mainly for security reasons.

<sup>15</sup>NTLM is a computer networking security protocol which operates in a variety of Microsoft Windows network protocols for authentication purposes.

<sup>16</sup>DLL stands for Dynamic-Link Library. It is Microsoft's implementation of the shared library concept in the Microsoft Windows operating systems. These libraries usually have the file extension DLL, OCX (for libraries containing ActiveX controls), or DRV (for legacy system drivers).



## 2.3 Billing system migration

The actual system is running for a while and start to be wasted by a irrelevant data. Setting a new system would help first to clean the database. On top of that, the actual billing system is completely disconnected from the technical infrastructure. Freeside was advised by the former student and looks to be an interesting solution. It would be possible, for instance, to create accounts automatically on the new mail system, or to disconnect someone that does not pay anymore. The ambitious idea is to run a completely standalone infrastructure where people can subscribe and pay online. Right now, RogueLink need an office and an employee to receive payments and also need someone to collect the postal payments, send invoices, going to the bank, etc. . .

### 2.3.1 Freeside

Freeside is an open-source billing, ticketing and automation system for ISPs and online businesses. It has a web based interface. It is written in Perl and it is easily customizable because full source code is available under the GNU GPL.

Freeside also provide a customer interface so they can pay, signup online and change their password. Bills sent by Freeside are also very neat and professional thanks to LaTeX<sup>17</sup> (this is also what is used for this report). One disadvantage of Freeside is that its doesn't look professional but because it is open source we should ask a web designer to make it more fancy. Also, I didn't like the way it is packaged. It put files everywhere and it is even hard to remove it cleanly. . .

### 2.3.2 Data extractions

The old billing system use a SQL Server database to store all its data so I was able to extract customers data to put them into the PostgreSQL database of Freeside using a Perl script.

There is three kinds of data :

- Customers data : contains name, address, phone number, etc. . .

---

<sup>17</sup>LaTeX is a document preparation system for the T<sub>E</sub>X typesetting program. It is used mainly by mathematicians, scientists, and engineers in academia. It is also widely used by people outside of these fields as a primary or intermediate format (e.g. translating DocBook and other XML-based formats to PDF) due to the quality of typesetting achieved by T<sub>E</sub>X



- Packages : contains the price and a billing frequency of a service provided by the company.
- Provisioning data : contains data to synchronize accounts with the technical infrastructure (mail, DNS, web sites).

I extracted only customers data as it was planned we will associate them to packages progressively. I also set up a provisioning source for the mail system and a test package to test the PayPal payments.

### 2.3.3 PayPal extension

The most popular way to get paid on internet today is certainly PayPal. PayPal, was used to allow PayPal members to transfert PayPal money to others PayPal members (merchants) thanks to their email address. But PayPal diversified then and now allow to use direct payments (with credit card).

They provide API<sup>18</sup> for credit card payment in a way that is possible to make it transparent for end users. The API is working with several languages as it uses an abstraction layer called SOAP<sup>19</sup> for sending and receiving data or for triggering RPC (Remote Procedure Call).

There was no PayPal extension provided with Freeside. But as far as the former provide a good and simple API for adding new extensions I wrote one for PayPal support into Freeside and put it under GPL on my website. This is a Perl module of about hundred lines that actually makes the interface between Freeside and PayPal using the SOAP protocol.

PayPal provide a really good way to test their API. The concept of sandbox paypal account is that you can send and receive money but it's virtual money.

---

<sup>18</sup>An application programming interface (API) is the interface that a computer system, library or application provides in order to allow requests for services to be made of it by other computer programs, and/or to allow data to be exchanged between them.

<sup>19</sup>SOAP is a protocol for exchanging XML-based messages over a computer network, normally using HTTP. SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework that more abstract layers can build on.



#### 2.3.4 Conclusion

The system was ready, but it was not really intuitive and stable and would require a lot of time to fix and learn it. As it was my last month and because there was many other projects it is not used right now.

Anyway, this project was a good way to see how a billing system works and especially for ISP <sup>20</sup>.

However, I had to move the actual billing server from the office to the server room because the server was not in the best environment for a so important service. In term of air-conditioning, power supply, monitoring, security, etc. . .

---

<sup>20</sup>Internet Service Provider





## 3 Conclusion

### 3.1 Technical conclusion

My main motivation since I arrived, after understanding the situation, was to be able to leave 6 months later with a stable, secure, up and running system that I would not have to worry about anymore.

I had no problems to achieve my goal with the good practical oriented training we got at Epitech. I was able to work on things I never used before because I already learned to learn. For making it short, this internship was a struggle against three things : spam, hardware failures, and... frontpage.

Technically, it was really diversified and mixed Windows and Unix administration, networking, scripts coding and developpements for the new billing system. System administration was also a new domain for me. Although I was used to the technical components, I had to think more with an end-user point of view in mind. That will help for any of my futur jobs because there is always a question of end-users.

To sum up, the security and stability were improved thanks to linux and, on windows servers, thanks to anti-virus checks and updates. Backups were also made. The monitoring was efficient and it never takes long to aknowledge problems.

I had a taste of system administration and I measured how oppressing it is. I was never totally. I was monitoring and even fixed things on the week-end. We had hard drive issues, serveral power outages (for the whole city) and our connexion provider had sometimes hours of blackout.

This internship was a good opportunity to take responsabilites. Indeed, one single error on the mail or web migration would be fatal for the company. So, I had to elaborate procedures and often had to be very prompt finding solutions in case of problems because hundreds of people (and sometimes whole business) would be blocked otherwise. I had to manage priorities because problems were often related between them.

I took initiatives and it was a good learning to measure each risk and to justify them. I establish how to do technically each mission, like choosing the components for new infrastructure (postfix, postfixadmin, dovecot, postgresql) or migrating tools (adsutils, imapsync). It was interesting to



plan each step and prevent possible problems. It was like designing a software as I did in my former internships.

### **3.2 General conclusion**

It was interesting to see how a small business works, how it is vital to spare money and to make strategic choices. I used to speak with the manager, Jacques, and he advised me a lot on management and strategy.

I was sometimes alone at the office and had to welcome customers, take care of their requests if possible or take messages. I had also to communicate with the hotline support to notice incoming problems or to advise on what to say or what to ask to customers. A good thing was to hear the hotline support. This contributed to improve my English and I learned a lot on how to speak to customers. However, one disadvantage of this internship was maybe not being able to work in a team but just trying to explain what I was doing.

Humanly, it was really enriching to meet American folks of any ages and with diversified personalities. Globally, it was surprising how nice they are and particularly in Grants Pass where is located RogueLink. I had for instance the opportunity to assist, as a French guest, to French classes and it was fun trying to explain my « French » point of view on the Middle East or on the American gas consuming and so forth. . .

Besides the valuable addition on my resume, the experience of the United States was a main point. It allows me to improve my English, to meet American people and visit San Francisco, the Silicon Valley, the Napa Valley, Seattle, Crater Lake, Las Vegas, Monument Valley, the Grand Canyon. Small cities all around Grants Pass were also beautiful. Great memories so !



## Thanks

I want to thanks Jacques and Kate for their welcoming. Dwayne and Gail for their collaboration. The Epitech school staff. The french class' students for having so much fun with them.



The Bay Bridge which connects San Francisco with Oakland and the East Bay.



## Glossary

- ASP** ASP stands Active Server Pages. It is Microsoft's server-side technology for dynamically-generated web pages that is marketed as an add-on to Internet Information Services (IIS).
- DMZ** DMZ stands for DeMilitarized Zone. In computer security terminology, a DMZ is a network area that sits between an organization's internal network and an external network, usually the Internet.
- DNS** The domain name system (DNS) stores and associates many types of information with domain names, but most importantly, it translates domain names (computer host-names) to IP addresses. It also lists mail exchange servers accepting e-mail for each domain. In providing a worldwide keyword-based redirection service, DNS is an essential component of contemporary Internet use.
- FTP** FTP or file transfer protocol is a commonly used protocol for exchanging files over any network that supports the TCP/IP protocol (such as the Internet or an intranet).
- Greylist** Greylisting is a simple method of defending electronic mail users against e-mail spam. In short, a mail transfer agent which uses greylisting will "temporarily reject" any email from a sender it does not recognize. If the mail is legitimate, the originating server will try again to send it later, at which time the destination will accept it. If the mail is from a spammer, it will probably not be retried
- IMAP** IMAP allows users to access new messages instantly on their computers, since the mail is stored on the network. With POP3, users either download the e-mail to their computer or access it via the web. Both methods take longer than IMAP, and the user must either download any new mail or "refresh" the page to see the new messages.
- PHP** Php Hypertext Preprocessor (recursive acronym) is an open-source, reflective programming language. Originally designed as a high level scripting language for producing dynamic Web pages, PHP is used mainly in server-side application software.
- SSL** Secure Sockets Layer (SSL) and Transport Layer Security (TLS), its successor, are cryptographic protocols which provide secure communications on the Internet for such things as e-mail, internet faxing, and other data transfers.



- TCP TCP stands for Transmission Control Protocol. This is one of the core protocols of the Internet protocol suite. Using TCP, applications on networked hosts can create connections to one another, over which they can exchange data in packets.
- TSE Terminal Services or Terminal Server Edition (TSE) is a component of Microsoft Windows NT operating systems (both client and server versions) that allows a user to access applications or data stored on a remote computer over a network connection. Terminal Services is Microsoft's take on server centric computing, which allows individual users to access network resources easily.
- Wiki a Wiki is a type of website that allows users to easily add, remove, or otherwise edit and change some available content, sometimes without the need for registration. This ease of interaction and operation makes a wiki an effective tool for collaborative authoring.