The Spaminator

Mathew Hazel

Ira Nicks

John Healy

BSA 375

Professor Kaufman

September 22, 2003

Table of Contents

The Spaminator

Unwanted solicitations via email have become an extremely large problem in today's society. No one wants to be interrupted during dinner time with an unwanted phone call solicitation or a knock on the door by a door to door salesman. There has even been a law passed by the government stating that telemarketers can not call someone if they are on the "No Call List". So how do you feel about unwanted email solicitations? Millions of unwanted emails are sent to electronic mailboxes worldwide each day costing recipients valuable time sorting through the inbox to find needed information. This common occurrence slows down business resources and costs companies valuable man hours. Our system solution to this problem is a database called "The Spaminator".

This database will contain all known email advertisements and unwanted solicitations. The database will then be connected to subscribing companies email server and all incoming email traffic will be compared to this database before being delivered to a client's inbox. All email matching the known spam email will be filtered out of the incoming mail. We are currently considering whether or not the incoming spam email has to match 100%, or if we will be looking for catch phrases and keywords.

Once we have to database information in place and online, we then plan to test the database by setting up an email server and connecting it to our database and start beta testing the system until it is perfected. When we have a success rate of about 80% success rate, we then plan to offer our service to a company for a limited time free of charge. This will allow our system exposure to the business world as well as a live test

run. From this live test run we hope to gain valuable feed back and word of mouth advertising.

The feasibility outlook of the spam filtering project, codenamed “The Spaminator” looks very promising. Even though we are in the beginning phases of this project, there are no foreseeable problems, operational, technical or economic that could possibly keep this project from being completed.

Operational analysis has determined that there should be a minimal affect on our organization from our proposed spam elimination solution. Personnel within the organization will be able to adjust with out any type of training being required. This is due to the way that “The Spaminator” will be implemented. There will be no client side software and practices the computer users will have to learn to benefit from our proposed project. All spam filtering is done server side, in which case, personnel should only notice the benefits.

At this phase there is only one major technical feasibility issues. Since we will be filtering e-mails there are the possibilities of false positives. That means legit e-mails being detected as spam. There are certain blocked words that can appear within words that are not blocked. There also is the issue of blocking spam in different languages. Because users would rather receive spam then have legit e-mails being blocked, if any mail is blocked, users also have the option to notify us whether or not the messages are spam.

Economic feasibility should be considered a non-issue. As stated above, all processes are done server side so that eliminates having to develop, and or buy client side software to assist in blocking spam.

Our proposed system also eliminates the following:

- The Purchase of Hardware
- Training Cost
- Software Cost
- Installation Cost
- Conversion and Changeover Cost
- Redundancy Cost
- Operational Cost

This will save money that would otherwise be spent on software development or software licenses per computer system.

The first individual requirement is the ability for the team to create a piece of software capable of determining if two pieces of text e-mail are at or above an 80% body content match. What this means is we will take an unknown e-mail and compare the text body of the e-mail to a database of known SPAM e-mail. When we produce a match of 80% or more of the known SPAM and the unknown e-mail we will then flag the unknown e-mail as SPAM. Once we have flagged the e-mail as SPAM we will be able to several things with this piece of e-mail. We will be able to produce reports for the people who subscribe to our service and say things similar to we stopped X amount of SPAM from getting to you.

The second individual requirement is to create a piece of software with the capability of reading into the header of each e-mail and determine the true sender of the e-mail. This may tend to be one of the more challenging pieces of the puzzle. Even though it is a simple thing to read into the header of an e-mail it is a challenge to create a program that will understand what is their. There a few ways we can create this type of knowledge in a program. One possible method is to have what is called a decision tree this is a form of Artificial Intelligence (AI). Now we could make the AI as simple as we need to or we can make the AI as complex as we need to. The problem is knowing ahead of time if the AI will be simple or complex. When it comes to programming AI it is an entirely different beast then programming for a simple application. Each type of AI style is more appropriate for each particular type of problem. In our type of problem we will probably use a type of weighted tree structure. With AI one must do the proper planning and research before any code is written to maximize the time spent writing the code.

For the determination of requirements, we plan to use the traditional methods for collecting system requirements. This includes interviewing, questionnaires, and observations to determine requirements.

We will need to know what the current system in use is and what users like and want to change about the system. We will also need to know what forms of e-mails users would like to receive and which ones they would not like to receive. Observations can also help use determine what forms of e-mail are appropriate for personnel to receive and is it needed for them to complete their daily jobs.

The one piece of material that is critical to the project is having a program with the capability of comparing two e-mails. The first e-mail we will call "A" and the second e-mail

we will call “B” and “B” is the known SPAM. We must have a program that compares “A” to “B” and is capable of determining that "A" is 80% or more of a match to "B" when it comes to the body of the e-mail. The second program the one program with the ability to determine the sender’s true IP address is optional because it is not a key feature in stopping the e-mail from getting to the client. But it is a key feature is possibly stopping most of the SPAM e-mail because the information used from this piece of software will be used to flood the SPAM senders e-mail server with e-mail’s from the system.

The hardware needed to make the project work is a T-5 line to a server farm, and extra large database farm to store the known SPAM e-mail. We also need to be plugged into the backbone of the Internet.

The designing of the Spaminator brought about major questions of how it should function. The potential size of the database presented problems that had to be solved by trade offs. The size of the system did not allow for a simple software solution that can be installed on a client’s server. Due to the fact that a large amount of data had to be gathered and then compared to incoming email meant that a large amount of disk space needed to be allocated to store samples of unwanted email. With this fact in mind, the decision was made to have a designated network. This network initially has three servers that will have an abundant amount of disk space. The estimation of the database will be in the terabytes.

Another consideration that was made is whether or not to have a program similar to Norton on the server to route incoming mail on keywords. This will cut down on the amount of traffic the Spaminator network would have handle and minimize the need for a faster internet connection. The determination was made that this would not achieve that

accuracy of filtration that we are looking for. In order to deliver the accuracy of 80% all traffic from the client's server would need to be routed to the Spmainator network for processing via a Virtual Private Network.

The design process for the Spaminator is straightforward. The main goal for our Spaminator project is to provide a fast and easy solution to eliminating spam from our clients with out being intrusive into everyday client activities. The only interaction we might have with clients would be in the analysis phase in the form of questionnaires to determine user requirements, but we have designed this to not be noticed by users during the implementation phase. The only thing users should notice is the lack of spam in their inbox.

To achieve this we designed the Spaminator to be a server side solution. That means minimal on site work and our servers can be located miles away from our client's facilities.

For our software design we chose to build our own proprietary software to eliminate spam. As stated before, the server side solution allows us to not have to install software on every user's pc and we will not have to worry about licenses because we will own the technology. The Spaminator software has the capability to compare the e-mail's subject and body for known spam words or phrases to a database. If there is a match the e-mail will not be allowed to go through. The Spaminator is customizable so words or phrases can be added and removed as needed.

Once a message is determined to be spam, it also adds the server from where the spam e-mail is being sent from and we can decide on whether or not the server is needs to

be added to our blacklist for spam servers. A message is also sent to the user notifying them a spam message was blocked with detailed information so the user can decide if the message was indeed spam and not a weekly newsletter that might get mistakenly flagged as spam.

There will be no hardware requirements for any of our clients except for an existing computer and network infrastructure that we can access. All hardware requirements will be on our side. We plan to run three servers that will be expandable depending on how many clients we have. Each server will run dual Xeon processors with 1 GB of ram, 40 GB hard drives, running Windows Server 2003. We will require each client to setup a virtual private network so that we can remotely connect to there network to filter out their spam. This will require sufficient bandwidth on their part depending on how many users they have sending and receiving e-mail. Our server locations will be running a FULL T3 (45Mbps) connection to accommodate all our potential clients.

The plan for testing the Spaminator will consist of two parts; testing the proprietary software designed to filter the incoming email, and testing the load capacity of the network. The software designed for the system will be a form of artificial intelligence that will compare known undesirable email to incoming email. The initial software test will consist of gathering known undesirable email and using the Spaminator software to filter the email past through the server. The software itself will go through code and syntax checks to make sure it is functioning properly. Documentation will also be kept throughout not only the test phase, but the entire project. Once 80% accuracy is achieved, the software will be considered a success.

In order to test the software in a real live environment, an email server will be set up with the proprietary software on it and connected to a secondary email server which will act like a client server. The project team will access the internet and give out the email address of the dummy client server to as many public websites as possible. This will allow the Spaminator server to start gathering undesirable email to put into its database. It will recognize undesirable emails by keywords and the presence of .jpeg and .bitmap files. The emails gathering in the database will also be followed closely by the project team to make sure the emails we are retrieving are considered undesirable. The project team will also be looking to see that emails not gathered and let through to the dummy email server are desirable email. The team will then make adjustments to the email gathering process as necessary.

Once the Spaminator software is performing to the 80% goal, the project team will then create a network of three servers, a router, and a T1 line. The system will then be load tested to make sure the capacity of the system can stand up to a real life environment. The goal of this phase of testing is for the system to be able to handle 60000 emails per minute. Once this goal is achieved the system will go through a live test. The Spaminator's services will be advertised free of charge for a limited time to select companies interested in filtering their email. This part of the process will not only give the system a real live test, it will also give the system credibility in the business world as the system of choice for filtering out undesirable email.

The process that will be used in developing the Spaminator follows closely to the System Development Life Cycle. The project will start with the determination of the

scope of the project and objects the project team hopes to achieve. This phase of the project will last approximately 90 days. In the phase, the project team will visit prospective companies to gather preliminary data and statistics on how big of a problem companies have with unwanted email solicitations. The statistics the project team is looking for is how much unwanted email solicitation traffic they receive, the number of email accounts they possess in the company, and how many man hours are lost in accordance with these emails. Documenting data on the amount of traffic a company receives will help determine the amount of bandwidth and the type of hardware the prospective system will require. The number of email accounts will give a percentage of the amount of email solicitations per email account when compared to the amount of traffic on the email server. The amount of man hours lost will allow for the project team to determine a reasonable price to make the system appear feasible to prospective clients.

The system analysis and design phase of the project is scheduled for 45 days. In this phase several logical designs for the system will be put into flowcharts and relationship modeling will begin to take place. Once one flowchart is determined to be the logical design of choice, the system will begin to be designed around it. Hardware requirements will be determined and the feasibility aspect of the system will begin to become a factor. Within the analysis and design processes trade-offs will begin to take place. Whether or not to buy software or design proprietary software will be a decision that needs to be made. Since the system that will be designed here will be a form of

artificial intelligence, the project team will need to design the software themselves in order for the system to perform the functions needed.

The next phase of the process will be the detailed design. In the detailed design the type of hardware and the coding of the software will be determined. Gathering the hardware required will be the least time consuming in the stage. The development of the proprietary software will be the main objective and most time consuming. The phase is scheduled for 120 days. The development of the software will also consist of the hierarchical input process output and database structure. When this part of the detailed design is complete, the project team plans to have a full functioning database, proprietary software, and network complete with a connection to the internet. At this point testing will take place and gathering email solicitations will begin. This will allow the project team to begin building a database of emails that will be used for comparison and filtering functions. All syntax, code, and load testing will be preformed at the end of this phase. Once the system reaches the 80% accuracy level, we will proceed to the next phase.

Implementation will be the next phase of this project. In this phase the project team plans to begin training system technicians and help desk personnel. The project team will also determine the security measures that will need to take place to protect the system from viruses and hackers. Standard operating procedures and "how to" manuals will also be documented to help the personnel that will be maintaining this system around the clock. This phase of the project is scheduled for 110 days.

The next phase of the system will be the acceptance testing / changeover. In this phase we will offer the system to prospective companies for a period of 60 days free of

charge. This will allow the system to prove itself in a real world environment. At the end of the 60 day trial period and evaluation of the system will be preformed to make sure the 80% accuracy level is being met. The project team will also check the software and hardware to determine if the scope of the network needs to be expanded or if the initial network will be able to handle the workload for the time being. A secondary database will also be created to log trouble tickets to further the uptime of the system. Every 15 days the system load will be evaluated. If the load of the system reaches 80%, the project team will expand the existing network and or bandwidth to accommodate growth of the system, the business, and profits.

The documentation plan for our proposed system is primarily focused on the system documentation compared to the user documentation. The reason for this is that there is less documentation that will be required for end-users to review for our system except for Information systems departments whom will need documentation to help install our system in there networks.

For system documentation we will cover the following phases:

- System requirements
- System development
    a. Architectural design
    b. Prototype design
    c. Detailed design & implementation
    d. Test specifications & implementations

The System requirement documentation will document how we need and expect the system to perform. Since are proposed system is to reduce spam we would document what it needs to accomplish to meet our users needs. For example, filter of 80% incoming unsolicited e-mail, or require minimal maintenance. It will be mandatory for our proposed system to meet all goal requirements stated in the documentation or it will be considered a failure.

System development documentation will cover the four parts of the development phase listed above. This is the main documentation that will cover the developmental process of our system proposal.

Architectural design documentation will document the initial designs and explain how we want our system to work and how it will meet our proposed requirements.

Prototype design documentation will cover the first rudimentary working model to determine whether or not our architectural design will work. This will include software code documentation, data flow documentation and hardware documentation. In relation to our proposal, this would document how the data will be routed to our system, reviewed, changed and sent back to our clients.

Detailed design and implementation documentation will cover exactly how our proposal will function and be installed on our client's networks.

Test specifications & implementation will document what we tested our system for and what results we were expecting and what we achieved. This will include numbers, problems encountered. For our project, we would document all results of the testing

process summary which are accuracy and network load test and how those test were setup and run.

User documentation will be less of an importance and not as extensive as the system documentation. Our proposal is to eliminate spam from our client's user base but we are not providing software that users will have to learn but a service that users will benefit from. That eliminates the need to write an extensive documentation explaining how to install, use and troubleshoot our software. We will have to have documentation that will show those in charge of running the networks how to help us setup and install the necessary software on their networks so that we will have access to their incoming e-mail. This documentation will be the basis for providing technical support to the Information System Departments of our clients.

The support plan for the system is a simple but complex system. The daily maintenance is normal monitoring of the servers and data-farm to determine how well the system is performing. With the data-farm we will keep time stamps on each record and when a record has not had a match is 60 days we will delete those records. This method will allow us to have a data-farm of a manageable size and the smaller database is the faster we can check incoming e-mail ageist the database of known SPAM.

On the servers we will have a daily, monthly, and yearly maintenance plan. The daily maintenance plan will be just checking the system log files and looking for any errors. The monthly maintenance plan will include a cycling of the power on each server. A look at the overall performance and messages in the log files over the past month which includes looking for any specific error messages and a look for a performance

trend. The installation of any new server patches for the servers which have been released over the past month will be done at this time. The yearly maintenance will include cleaning out the dust in the server and a performance check of the log files over the past year. At this point you are looking for trends in performance rather then a specific error message.

The network speed at the start of the project will only be a 1Gig network. As the system gets more of a load and the network traffic gets grater then 80% it will be time to upgrade. One idea for the final upgrade is to connect all the servers and switches with fiber optic wire.

When we first start this project we will have a cluster of small servers and a T1 line. As the project gets known and we get more companies requiring our services well will move to bigger and better things. The mark now is when the T1 line gets to about 80% capacity we will install a T3 line and when the servers are running at 80% usages over a set time we will move to more servers. The long term idea is to have a T5 line and a Cray X1 computer system.