Freedom 2.2 Abuse Issues and Analysis

David Bratzer, Andrew Elkin
Zero-Knowledge Systems, Inc.
davidb@zeroknowledge.com

June 14, 2001

Abstract

Anonymous and pseudonymous systems change the dynamics of abuse management. Some problems become easier to solve, while others become more difficult. In this paper, we show that abuse management need not be dependent on knowing the true identity of the people using the system. We examine design features of the Freedom Network that limit the effects of common abuse problems such as hacking, denial-of-service attacks, and spam. Abuse incident statistics covering a ten-month period are also included.

1 Introduction

1.1 The Problem

Can a privacy-enhancing system protect the personal information of its users, and yet still prove resistant to abuse? At first glance, these two goals seem at odds with each other. After all, the first step most ISP's take to determine the validity of an abuse complaint is to examine their system logs. These logs typically show which sites a user has visited, when and to whom emails were sent, and can include a record of IRC and telnet sessions. In a privacy-enhancing system the amount of data logged must be sharply reduced or completely eliminated because this information can be used to compromise the true identity of a user. If log files are not maintained, how can abuse be minimized? The solution is to approach the problem differently: abuse resistance needs to be designed in a manner that is not dependent on knowing the true identity of the users.

1.2 Background

Distributed network infrastructure can be designed and shaped to create systems that add privacy to a user's Internet experience. In an anonymous system, the true identity of a user is hidden, and each transaction made by that individual is perceived as an independent act (i.e. two actions by the same user can't be linked with each other). A pseudonymous system allows for the creation of persistent identities that

cannot be linked to the true identity of user. In this way, a user can protect their privacy and still develop online relationships with others. Hybrid systems are also possible, with some components providing anonymity and some components providing pseudonymity. These three different categories of systems can be collectively referred to as privacy-enhancing systems. The principles of abuse management outlined in this paper are intended to be applicable to all of the above systems.

1.3 Definition of Abuse

Abuse can be defined as unwanted network activity. This is a system-specific definition; activity that is considered abuse on one particular network may not be considered abuse on a different network. This definition avoids assigning an absolute value to an incident. Instead, we suggest that abuse resolution is essentially a negotiated settlement between an aggrieved party and the service provider. There is no single right or wrong solution, but rather the common ground of an acceptable resolution which must be sought out between the two parties.

The range of permissible activity is usually explicitly stated in a legal document, typically referred to as an Acceptable Use Policy (AUP). The problem with most AUPs is that they are legal instruments, and as such do not necessarily indicate which types of incidents the organisation is actually capable of resolving. One of the aims of this paper is to fill that gap by providing a technical explanation of those capabilities with regards to the Freedom Network. This will help set expectations, which will lead to faster and more satisfying negotiated settlements in future abuse incidents.

1.4 Principles of Abuse Management

In a privacy-enhancing system the true identity of a user is not known and can't be relied upon to help resolve an abuse incident. The effectiveness of punishment as a deterrent is limited, shifting emphasis to more proactive approaches. With this in mind, the following principles of abuse management are suggested for consideration in the design, construction and operation of any privacy-enhancing system:

Containment: Implement proactive measures that will limit the damage caused by an abuse incident.

Reaction: This is the approach used by most ISPs for abuse management. Reaction (eg. shutting down an account) can complement containment measures in pseudonymous systems. The nature of anonymous systems precludes the use of most forms of reactive measures.

Alternatives: If neither containment nor reaction is possible, develop alternative resolutions outside the context of the system in question.

Multiple factors play a role in enabling certain forms of abuse. This means that a holistic approach, involving actions by parties other than a particular privacy-enhancing system, may sometimes be necessary to resolve an incident.

2 The Freedom Network

2.1 The Freedom Network

The intent of this section is to provide basic technical information about the Freedom Network that will help the reader understand Freedom-specific abuse issues discussed throughout the remainder of this paper. However, this section only scratches the surface of a very complex system. A more detailed description of the network can be found in a series of white papers have been published about the Freedom Network. These papers include a network architecture paper by Back, Goldberg and Shostack [1], a security analysis by Back, Goldberg and Shostack [2], a description of the nym creation process by Samuels and Hawco[3], and an overview of the mail system by McFarlane et al.[4].[1]

The Freedom system is based on the concept of a unique pseudonym, known simply as the 'Freedom nym'. Freedom is designed to afford the end user a range of anonymous and pseudonymous Internet services and is composed of client and server components. The Freedom Network is operated by Zero-Knowledge Systems in conjunction with independent server operators worldwide. The network has three main components:

1. The Freedom client, installed at the user end to filter HTTP/SSL, Internet Relay Chat, Telnet/SSH, POP, SMTP and outgoing NNTP;

2. A 'cloud' of Freedom Network nodes called Anonymous Internet Proxies (AIPs), enabling the creation of anonymous IP tunnels to carry TCP, UDP, and ICMP traffic;

3. Clusters of Core Servers providing PKI (keys), network information and mail services.

A list of available Freedom servers and their public encryption keys is cached by the client on the user's local machine. This enables the user to manually select the route their data will take through the Freedom Network. Once a route has been negotiated, the client encrypts the data up to three times, depending on the length of the route. The data is first signed with the nym's private key, and then encrypted with the key(s) of the server(s) in the chosen route: exit AIP, middle AIP, and entry AIP in that order. Not only is the data encrypted, but the IP address of each hop in the route is hidden.

Because the route the traffic will take is selected and then encrypted on the client side, no Freedom server ever knows more than the IP address of the previous and the next hop in the route. As the traffic moves from hop to hop, each Freedom server removes its layer of encryption - revealing the address of the next hop - and forwards the traffic.

Thus, the destination host sees the IP address of the exit AIP, not that of the user's computer. Unless the entry AIP is also the exit AIP (which is the case in a one-hop private route), it will never know the final destination of a packet received from any Freedom client.

The exit AIP decrypts the last layer of encryption and delivers the data to the real destination. The exit AIP uses IP Masquerading (similar to Network Address Translation) to handle simultaneous client connections. When encrypted data arrives at the exit AIP, it is decrypted, mapped to one of a pool of available source ports and then

[1] http://www.freedom.net/info/whitepapers/index.html

forwarded to the final destination. The source port awaits the return connection from the host, facilitating re-encryption and routing the packets back to the original Freedom client.

2.2 Untraceable nym creation

Nyms are created through a step-by-step process initiated with the Freedom client software. Nym creation is an untraceable, user-controlled process that begins with the exchange of a purchased activation code for nym tokens.[2] Tokens act as a layer of indirection which prevent a link being created between an activation code and a nym.

The public parameters of Freedom's 2048 bit DSA Master key are hard-coded into the Freedom client, enabling the creation of a private/public encryption key pair unique to each nym. This key exists only in a passphrase-protected 'nym file' on the user's local machine.

The user then selects a nym name, and the Freedom client establishes a route through the network to a Nym Server, again using the Nym Server's public key to encrypt the data. The Nym server checks the token and the nym name to ensure that they have not yet been used, and then stores the public parameters of the nym's encryption key. The Nym Server closes the transaction by confirming to the client that the nym has been created and the token has been spent.

2.3 The Freedom Client

Freedom is not a 'trust me' system - rather, the user is in total control. Not even Zero-Knowledge Systems has the ability to reveal the true identity behind a nym; system design has made it very difficult.

The user activates the Freedom client filters by selecting a nym in the Freedom GUI. The filters monitor the supported protocols and strip IP addresses and other identifying characteristics from outgoing packets. The packets are then encrypted and sent on into the Freedom Network cloud.

Because the servers that make up the Freedom Network are operated by many different organizations and located in several countries, trust is distributed between many persons and jurisdictions. While it is possible that several server operators could work together to attack the privacy of a nym, this risk has been reduced by placing control over private route selection in the hands of the end user.

2.4 The Freedom Mail System

The Freedom Mail system offers untraceable, encrypted email and is essentially a separate system built on top of the Freedom Network. Local mail commands are filtered by the Freedom client, which sanitizes the message of compromising information, and modifies the message headers so that the sender is the nym selected by the user. Messages are then sent via an unauthenticated route to a Nym Mail Transfer Agent, and

[2] When purchased with a credit card, an activation code is associable to a transaction record at http://store.freedom.net.

passed across a firewall to a backend server for processing and delivery. While the route the client uses to connect to the mail system is anonymous, the authentication at the server level is not: only connections signed by a valid nym's Message Authentication Code (MAC) key are accepted. The same applies at the mail2news proxy handling pseudonymous NNTP posting.

Return messages are handled by the Internet Mail Encryption Proxy (IMEP). The IMEP in turn passes mail to a firewalled machine for encryption and delivery to the appropriate nym's mailbox.

These backend mail entities communicate with other Core Services in order to verify the operating parameters of the service to which the nym is entitled and access the nym's public encryption and signature MAC keys.

While Zero-Knowledge does currently log message sizes and times in order to track volume and aid debugging, this information is stripped of email recipient and pseudonym names.

3 Abuse Issues and Countermeasures

Through an agreement with the independent server operators, Zero-Knowledge Systems, Inc. is responsible for investigating and resolving all abuse incidents on the Freedom Network. The company staffs an Abuse Team which operates by the principles described in section 1.4.

3.1 Abuse at the IP and Transport Layers

3.1.1 Penetration

An unsecure host can be penetrated by an attacker using Freedom, provided two basic caveats are met.

First, the exploit must be executed over a port to which the exit AIP permits an outbound connection. This requirement is easy to meet, as all ports are presently open except port 25.[3] Port restrictions can be configured remotely by the Network Operations Center responsible for running the Freedom Network. These restrictions represent corporate policy decisions, and have changed over time in response to an evolving production environment. When Freedom 1.0 was released in December 1999, outbound connections on the exit AIP were only permitted to reserved ports of supported protocols. HTTP, for example, was permitted on port 80 but not port 6346.[4] This approach created significant customer service issues. Users found themselves denied access to legitimate services, not because Freedom didn't support the protocol used by the service, but because the services could only be accessed through non-standard ports. Given that the majority of exploits take place on standard rather than obscure ports, it became clear the security benefits gained from restricting ports were largely an illusion. Freedom 2.0 initiated a controlled expansion of the range of open ports.

[3] Port 25 is blocked for spam-control purposes.

[4] 6346 is the default port for Gnutella.

A second basic requirement is that the high-level protocol associated with the exploit must be supported by the architecture of the AIP. This may seem counterintuitive, because it is the proxies in the client which anonymize the high-level protocols such as HTTP and SMTP, but the AIP does affect which high-level protocols the client can support. Active FTP is a good example. A client requesting a file transfer tells the FTP server which port to send the file. In the presence of IP masquerading this can't work, because the client doesn't know to which port the exit AIP has renamed its traffic.

3.1.2 Denial-of-Service Attacks (DoS)

The analysis below focuses on determining which known denial-of-service attacks may be launched through the Freedom Network against an Internet host. Attacks against components of the Freedom Network (servers, clients, and core servers) are not discussed here as they are likely of little interest to the general incident response community. Also, note that the list of attacks examined below is not meant to be exhaustive; they are presented in order to give a general idea of what can get through the Freedom Network.

Certain design features of both the Freedom client and server limit the ability of an attacker to launch anonymous DoS attacks. However, it's unwise to rely on limits imposed at the client level. The user can change any client-based rules for controlling traffic, either by modifying the source code (available for the Linux client) or through reverse-engineering. The conclusions drawn below are based on the architecture of the AIP. However, these would not apply if an attacker managed to compromise an actual AIP, instead of using it as an exit point for anonymized traffic. In that scenario, the attacker could exploit the services of the AIP at will.

The ability to forge the source address of an IP packet plays an important role in executing a DoS attack. Changing a source address - more commonly known as 'spoofing' - protects the attacker by making it difficult to discover the true origin of a packet. The anonymity implications are largely irrelevant here, because the entire function of the Freedom network is to provide anonymity. Of more concern is the use of IP spoofing to target a host. In this technique, the attacker places the IP address of the victim in the source address field of an IP packet. The packet is sent to a host, which acts as an intermediary. This intermediate host sends a response - often disproportionately large compared to the original request - to the forged source address. Multiply this process ad nauseam and the desired effect can be achieved: the bandwidth capabilities of the target machine are overwhelmed, thereby denying service to legitimate users.

DoS attacks launched through Freedom will not be able to target hosts in this manner because the exit AIP becomes the source address for all packets leaving the network. The attacks themselves can still be executed anonymously, but they become less interesting because they will be directed at a host responsible for its own security.

Attacks based on the domain name system illustrate this concept well. The Freedom client will proxy DNS requests. Normally, they are resolved by a name server running on the exit AIP, but it is also possible to direct anonymized DNS requests to external name servers. In this scenario, using Freedom, the use of IP spoofing to redirect responses from a DNS server to a target machine will fail. This does not mean that it is impossible to launch an anonymous DNS-based attack. There is still a group

of DoS attacks based on vulnerabilities in BIND, and these attacks can be launched through the Freedom Network directly against an unsecure host.

Another example is the Smurf attack. An ICMP echo request, or 'ping', can be used to measure the availability of a server. These requests can be sent to a single machine or a group of machines. In a Smurf attack, an ICMP echo request is sent to a group of machines, causing all of those machines to respond at once. This flood of responses can be directed to a victim by forging the source address of the original ICMP echo request. The restriction on IP spoofing prevents this from happening on the Freedom Network. A user can still send out ICMP echo requests, but any response generated will be returned to the client.

Not all attacks rely on IP spoofing to target a victim. Simple floods, such as a TCP Syn flood, target a victim directly. In a Syn flood, a host is sent many SYN packets. The server responds by sending SYN-ACK packets, and also by reserving memory for the anticipated connection. Instead of following protocol by completing the handshake with an ACK, the attacker continues to generate half-open connections by sending more SYN packets. The amount of memory that can be reserved is finite, and soon legitimate users are being denied access to the host. Although this attack can easily be executed through Freedom, its scope is attenuated because outgoing connections are limited to a single source address. If IP spoofing was possible then a very large number of connections could be initiated. In reality, the exit AIP only has one IP address, and it can only initiate one connection per port. The source port field in a TCP packet is a 16-bit field, which means the exit AIP can only generate a maximum of 65536 TCP connections at any one time.

Simple UDP port floods, which send a barrage of UDP packets to a host, are also trivial to launch through Freedom. More complex floods, designed to amplify the bandwidth consumed, are another matter. One example of an amplified flood occurs when two UDP services, echo and chargen, are played against each other. Echo is a UDP service which returns any input it receives, while chargen is a UDP service which responds to input by outputting a string of ASCII characters. In this attack, spoofed packets are sent to one of the UDP services. The source address of the spoofed packet is the address of the host running the second service. By connecting the two services - the output of chargen with the input of echo - an endless loop of packets can be created. With Freedom, the IP spoofing restriction prevents this feedback loop from occurring, and instead turns the attack into a simple packet flood.

Malformed IP packets form the basis for a third group of attacks. The Ping of Death, for example, is launched by sending an ICMP echo request packet that exceeds the maximum size normally permitted in IP (i.e. greater than 64K). Teardrop is an attack tool which uses overlapping IP fragments to exploit TCP/IP stack vulnerabilities. Both of these attacks fail because the exit AIP attempts to refragment outgoing traffic after decrypting it. Another attack, Land, sends packets formed in a manner designed to cause target machines to hang. Typically, this involves sending a packet with the same source address as the destination address, which crashes some TCP/IP stacks. Once again, the restriction on IP spoofing makes this attack harmless.

In sum, most DoS attacks either can't be launched through Freedom, or the damage they can cause is limited.

3.1.3 Distributed Denial-of-Service (DDoS)

A DDoS network consists of a client, handlers, and agents. The client is controlled by an intruder, and is used to relay instructions to the handlers. The handlers, installed on compromised machines, contain a list of known agents. On receiving instructions from a client, the handlers can distribute information about the attack (such as the target list), to the agents. Once the agents receive these instructions, they launch the specified DoS attack. The structure of a DDoS network results in a huge increase in the amount bandwidth that can be directed at a target, while simultaneously adding a layer of obfuscation to any post-incident forensic investigation. Tools such as Trinoo, Tribe Flood Net (TFN), and TFN2K have served to popularize this form of attack.

It's highly impractical to create a DDoS network in which the agents are running on machines with the Freedom client installed. As described above, Freedom is a poor platform for launching DoS attacks. More interesting is the use of Freedom to set up and control a network of DDoS agents. This is a feasible endeavour, but one that is subject to the host penetration restrictions outlined in section 3.1.1.

3.2 Abuse at the Application Layer

The vast majority of abuse occurs here, likely because most communication initiated by the end user makes use of the high-level protocols found at the application layer.

3.2.1 Web-Browsing

For privacy reasons, the exit AIP does not maintain detailed HTTP logs. Errors generated by a nym's activities are logged, but even these logs are kept for a very limited duration. This is considered a high-risk area because the information contained in typical HTTP logs could be used to create a detailed profile of which web sites a particular nym visited, and this profile could in turn be used to discover the real-world identity of a nym.

Without HTTP logs, the Abuse Team is not able to link a web-based abuse incident to a particular nym. Nor are there any proactive measures that limit HTTP abuse, so authentication processes need to be in place at the destination web server if there is a need to validate or censor content received from visitors. These authentication processes can be as simple as requiring a valid email address for a guest book entry or as complex as the Address Verification Systems used for approving credit card transactions.

3.2.2 Telnet and Internet Relay Chat

As with HTTP, there are no logs containing detailed information for telnet and IRC sessions, so it is not possible to link an abuse incident to an individual nym. Some potential problems with telnet and IRC, such as channel bot wars and flooding attacks, could be avoided through the use of rate limiting. Although this would be an interesting endeavor, as of this date not a single incident has occurred in which this countermeasure would have been useful.

3.2.3 Mail

There are several strategies in place to limit the amount of spam originating from the Freedom Network. First, daily limits are imposed on nym mail. This limit - currently set at 250 messages per day - is orders of magnitude below the minimum required to make spamming a worthwhile endeavour. Second, a zero-tolerance policy is in effect for spam incidents. Unlike HTTP, telnet and IRC, the clearly pseudonymous nature of the mail system makes it easy to link an incident to a nym. Although it is not possible to discover the actual identity of the person using the system, the nym itself can be identified, and if necessary shut down. After a spam complaint is received and its validity confirmed, the nym is sent a takedown notice. The confirmation process is slightly different than that which a regular mail provider might use, because the Abuse Team is unable to refer to logs in an investigation. To work around this problem in a privacy-friendly manner, a digital signature is included in the x-headers of every outgoing email:

Received: from unknown (207.107.115.229)
by 0 with SMTP; 7 Mar 2001 22:15:47 -0000
From: spamgirl@freedom.net
Message-Id: <200103072215.RAA06346@cc.zeroknowledge.com>
X-Freedom-Envelope-Sig: davidb@zeroknowledge.com **AQF5WW1QY**
YQxBXaRdi7hwmvsNcejOZITeGvTlkmyiCt0wSUskvYRfEb3
Old-From: spamgirl@freedom.net
To: <davidb@zeroknowledge.com>

When a complaint is received that includes the full mail headers, this signature can be used to verify the date, time, recipient and nym sender.

Use of a nym as a dropbox is prohibited. In this scenario, spam is sent from a third party that contains a nym address either in the reply-to field or in the message content. By using a third party to send the spam, and a nym to collect responses, the mail limit restrictions imposed by the Freedom Network are circumvented. There is no easy way to confirm whether a nym is truly being used as a dropbox, as it's always possible an attacker could be trying to frame a nym with a fictitious complaint. To resolve this issue we 'go undercover' and send an email from a throwaway account requesting more information. If we receive a reply that indicates confusion, the nym is likely an innocent victim. If the nym responds with information on how to 'make money fast', the Abuse Team sends a takedown notice. This technique is not without drawbacks. The confirmation process causes a delay between the incident and nym termination, which gives the spammer time to collect responses. Also, it is possible a clever spammer could recognize the Abuse Team's 'response' as a fake and ignore it, thereby making confirmation impossible. This appears to be rare, however, and overall we find this technique effective, especially when combined with an aggressive pursuit of the party responsible for sending the spam in the first place.

Complainants who no longer wish to receive email from a particular nym are re-

ferred to a web-based email blocking form.[5] Subsequent email sent from the nym to the address is dropped without notification. (A bounce notice would alert the nym to the existence of a block, which the Freedom user could then avoid by sending from another nym.) It is also possible to prevent an individual email address from receiving any email from the entire freedom.net domain.

3.2.4 Usenet

Incidents involving Usenet spam are handled much the same as email, with some small changes. A complaint can be confirmed first-hand by viewing messages in the newsgroup itself, and therefore signature verification is not needed. Also, a cross post limit of five news groups per message is imposed in addition to the regular limit of 250 messages per day.

Regarding other forms of abuse, Zero-Knowledge Systems Inc. reserves the right to shut down accounts that are being used for activities which are illegal or which contravene the Freedom Network Access Agreement and Policies. This does not mean the Abuse Team will take punitive action against a nym simply because the content of a message is offensive or in bad taste. Such a policy would form the basis for arbitrary denial of privacy attacks against Freedom users. Instead of endorsing censorship, an attempt is made to teach complainants other ways to avoid reading news posts from a particular nym. In practice, this translates to the use of kill files, documentation for which is available on the freedom.net web site.[6]

4 Incident Statistics and Analysis

4.1 Taxonomy

Statistics are meaningless if the context in which they are used is misunderstood. Understanding the taxonomy used to classify abuse will help to place the following statistics in context.

At the core of the taxonomy is the concept of an 'incident.' It represents unwanted network activity confirmed to have originated from the Freedom Network. A 'complaint' represents how people communicate about an incident. In the ticket tracking system we use, an incident and the first complaint received about that incident are represented by the same ticket. Public incidents, such as those involving Usenet, tend to generate more than one complaint. These are stored as additional tickets.

The initial complaint is only upgraded to the status of 'incident' after it has been confirmed by the Abuse Team. If a system administrator alleges that a Freedom user defaced a web site, the ticket remains categorized as a complaint until the Abuse Team receives evidence in the form of server logs. Complaints either represent much ado about nothing, as is the case of unconfirmed complaints, or they are a duplication of previously confirmed information. Complaints are still a useful metric because they can

[5] http://www.freedom.net/support/abuse/blockemail.html

[6] http://www.freedom.net/support/abuse/blocknewsposts.html

provide insight into the effect of response time on abuse levels and the effectiveness of countermeasures.

The classification of abuse occurs largely along technological lines. For example, if a nym uses a web-based service such as Deja to post spam to Usenet groups, that incident is categorized as a web incident.

4.2 Incident Distribution

Incident statistics have been maintained since May of 2000, and they are based on the following categorization scheme:

Web: alleged credit card fraud, site defacements, guestbook abuse, etc.

Telnet: alleged trolling, bots, flooding on MUDs, etc.

IRC: alleged trolling, bots, flooding on MUDs, etc.

Email: alleged spam, dropboxes, and harassment.

Usenet: alleged impersonation, inflammatory posts, copyright infringement, etc.

During this time period, no complaints of IP or transport layer abuse were reported. At the application layer, no complaints involving telnet were reported. The distribution of the actual abuse incidents is shown below.

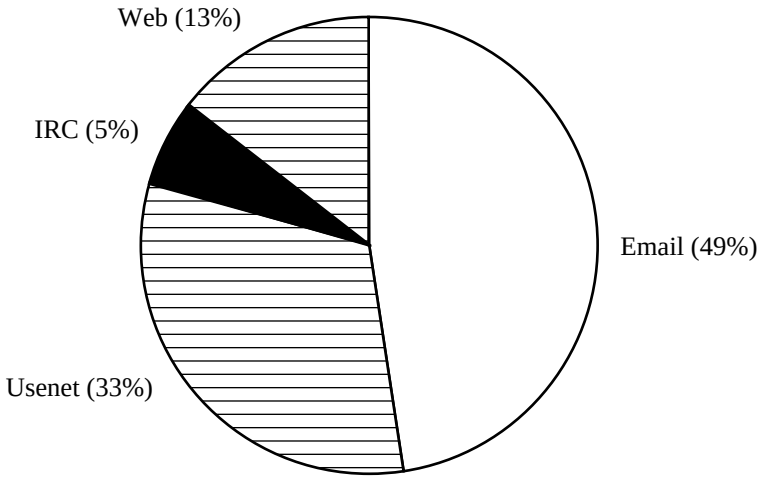

Figure 1: Incident distribution at the application layer.

The mediums which offer the most granular response capabilities - email and news posting - account for 82% of the incidents. Analysis suggests that the countermeasures in place for the mail system are effective in reducing the number of complaints received on a per incident basis. For example, the ratio of spam complaints to spam incidents originating from a nym is 1.6:1. This means that on average, 1.6 complaints are received for every spam incident originating from the Freedom Network. Compare this to the statistics involving dropbox spam, where the ratio of complaints to incidents is 3.2:1. On average, an incident of dropbox spam generates 200% more complaints than a spam incident originating from the Freedom Network.

Although the current mail limits reduce the level of spam complaints, they are not a perfect solution. One drawback, for example, is that they prevent the pseudonymous operation of large mailing lists. One workaround would be to introduce a sliding scale of mail limits, whereby the daily limit increased over time. In researching this option, a list was compiled of all the nyms which were shut down for spamming. Comparing the nym creation date with the date of the spam incident revealed that half of the nyms were shut down within thirty days of their creation, and the remaining half were shut down within the first six months. This suggests that spammers could be further discouraged by reducing the current daily limit to 100 messages per day for nyms less than one month old. At one month, the limit could be upped to 250 messages per day, and perhaps thereafter the limits could be upped again at six month or one year intervals. This approach would limit spam incidents during the high-risk period. After the high-risk period ended, it would reward nyms who were not abusive with increased mail limits. (Note that there are no plans to incorporate a sliding scale at this time.)

4.3 Overall Incident Rate

During the ten months between May of 2000 and February of 2001, the overall ratio of abuse incidents to nyms was below 0.2%. That is, one incident occurred for every 500 nyms created. Although it is difficult to know for sure, intuitively we feel this is an acceptable level.

Several factors influence abuse levels. It is understood that the nature and amount of abuse may change as the user base matures, but it is difficult to determine exactly how this will occur. The incident rate may increase dramatically. Or, the case may be that the incident rate will peak early in the history of the network, as first adopters test the limits of what is possible with the new system. So far, there is not enough evidence to support either theory.

Product usability also plays a role. Obviously, if it's not possible to use Freedom, it won't be possible to abuse it, either. With Freedom 1.0, the mail system was sometimes unreliable, there were performance problems with the network, and some compatibility issues with the client. Freedom 2.0 introduces an entirely new mail system, faster and more scalable AIPs, and resolves many of the compatibility issues. It remains to be seen whether the increase in usability will translate into an increased incident rate.

Finally, we note that outreach efforts by the Abuse Team - such as a clearly defined AUP and a willingness to openly discuss policy - may also have contributed to the low levels of abuse. In at least one instance, the team has been publicly referred to as the 'nym police.' We know that we are heard, if not seen, and this may act as a deterrent.

5 Conclusion

The principles and techniques outlined here have enabled us to maintain a low incident rate since the product launch over a year ago. The Freedom Network will continue to evolve, however. New services will be developed, and abuse resistance will need to be built into these new services. Zero-Knowledge Systems Inc. will continue to share

the impact of such changes with the incident response community. Feedback on these initiatives is welcomed.

It is also hoped the principles and techniques outlined here may be of real benefit to regular service providers and their customers. Clearly, if service providers begin to manage abuse proactively, the need for them to resolve abuse incidents in a privacy-invasive manner will be reduced.

6 Acknowledgements

While this paper was in development many staff at Zero-Knowledge Systems Inc. provided useful comments and feedback, for which we are grateful. We thank Adam Back, Zach Brown, Philippe Desaulniers, Gus Hosein, Adam Shostack, Jonathan Wilkins, and especially John Bashinski for contributing expert technical advice.

References

[1] Philippe Boucher and Ian Goldberg and Adam Shostack, “Freedom System 2.0 Architecture”, Zero-Knowledge Systems Inc. white paper, December 2000.

[2] Adam Back and Ian Goldberg and Adam Shostack, “Freedom 2.0 Security Issues and Analysis”, Zero-Knowledge Systems Inc. white paper, November 2000.

[3] Ed Hawco and Russell Samuels. “Untraceable Nym Creation on the 2.0 Freedom Network”, Zero-Knowledge Systems Inc. white paper, November 2000.

[4] Roger McFarlane et al, “Freedom 2.0 Mail System”, Zero-Knowledge Systems Inc. white paper, December 2000.