The following paper was originally published
in the Proceedings of the
Tenth USENIX System Administration Conference (LISA X)
Chicago, IL, USA, Sept. 29 - Oct. 4, 1996

The PGP Moose - Implementation and Experience

Greg Rose
Qualcomm Australia

The PGP Moose – Implementation and Experience

Greg Rose – Qualcomm Australia

ABSTRACT

USENET news traffic is growing exponentially, at a rate only barely less than the World Wide Web. Many observers believe that moderated newsgroups are the way of the future, to keep content high quality and avoid advertising. However there is essentially no security enforced on moderated newsgroups, and only a very small chance that decent security could be integrated into the framework itself.

The PGP Moose is free software, which attempts to address this situation. The aim of this software is to monitor the news postings of moderators of Usenet newsgroups, and to automatically cancel forged messages purporting to be approved. This software and protocol is designed around cryptographic signatures. The protocol is designed to allow the use of different signature techniques. The current implementation assumes the use of PGP [1] (Pretty Good Privacy) signatures, but can be easily modified to use others, such as the Digital Signature Standard or MD5. PGP was chosen for its widespread availability around the world.

Since first being made available a few months ago, the PGP Moose has been placed in control of a number of newsgroups in the USA, Germany and Australia. It is also used by a few individuals, particularly in Germany. Other solutions to the same problem are also in limited use.

In this paper I examine the history and implementation of the PGP Moose, and some of the problems with its deployment and debugging.

How Does It Work?

This document is written from the point of view of a newsgroup moderator, but individual users could also use the facility provided by the PGP Moose in analagous ways.

Usenet News works simply by flooding the network. Cooperating news server machines negotiate to decide which articles they have not yet received, and then get them before passing them on in turn. This protocol is called *NNTP*, Network News Transport Protocol [2]. This is surprisingly efficient in use of bandwidth, which is why there is so much news being written. It is also a wonderful way to distribute advertising material, an act called *spamming* after the menu offered in a Monty Python sketch.

Moderated newsgroups function differently. The news system control files mark particular newsgroups as being *moderated* and when the normal mechanisms attempt to post news to such a group, the message is intercepted and instead is mailed to the moderator. The moderator marks the message as approved and resubmits it, and this time the news software recognises this and enters it into the system.

The problem is that there is nothing, really, preventing any old user marking the message in the same way. The special mark is simply a header line Approved:, usually with the moderator's name and email address on it. As far as the NNTP software is concerned, though, the existence of the header line is a seal of approval.

When a moderator wants to protect their group from forged/unapproved postings, they should register their interest with one or more of the sites running PGP Moose, and pick up the submission script. As part of this process, the moderator would specify one or more PGP public keys that are allowed to approve postings.

When a post comes in, and the moderator wishes to approve it, they do whatever they would normally

```
X-Auth: PGPMoose V1.0 PGP sci.crypt.research
        iQBVAgUBL1/Kg2zWcw3p062JAQEYIgH/Xyrz6LaGG+fHaSxoexMECovzkIoADrQx
        173IX1UQEIoF15jnDBBdHVvqTMEPS0118ytYVQZoQrdStuXB9Oc9gQ==
        =azqs
```

Figure 1: PGP header

do before actually using inews (or whatever) to post the message. As their last action, they run the PGP Moose Approval program ''pmapp''. This inserts another special header which looks Figure 1. If there are multiple moderated newsgroups, there might be multiple X-Auth: headers, one for each group that has requested assistance from the PGP Moose daemon. In this example you can see that the authentication carries the name of the authenticating program, a protocol version number, an identifier of the type of digital signature (currently only PGP) and the name of the newsgroup in question. These, as well as the From:, Subject: and Message-Id: lines, the list of newsgroups, and the non-blank lines of the message itself, are used as input to the PGP program to generate a signature.

The lines of the message body are preprocessed in a way that is meant to render harmless any mangling that a typical news system might do to the article. The article itself is not changed, only the input to the signature generation. If a news system subsequently mangles the article in a ''normal'' way, for instance by inserting a ''>'' in front of a line starting with ''From'', it will still pass the signature check.

The list of newsgroups must be handled specially, so that an article posted to multiple moderated controlled newsgroups can be appropriately handled. See below for a more detailed treatment of the issues of posting to multiple moderated newsgroups.

The PGP signature is then inserted into the X-Auth: header, mostly so that it won't interfere with, or be confused with, any signature in the body of the message.

Anybody can check whether the message has been modified in any significant way, simply by running the PGP Moose Approval Checking program ''pmcheck''. More importantly, though, the sites running the PGP Moose Checking Daemon will be doing this automatically for every posting to the registered newsgroups, or from the registered users. And, if a posting fails the checks, it may be automatically cancelled, and a notification sent to the moderator.

The important things to notice about this process are that the actual news transport software isn't changed in any way, and that it provides a sort of ''morning after'' protection. A knowledgeable person can *still* send spams to moderated newsgroups, but they will be automatically cancelled soon thereafter.

The Bits

The approval and checking part of the PGP Moose consists of a number of Bourne Shell scripts calling standard Unix utilities and PGP. I could have used perl more elegantly, but this stuff is marginally more widely available. If there are Unix version dependencies, they should be considered to be bugs and I'll happily attempt to remove them.

pmapp usage: pmapp [newsgroup|user] [file]

> This script takes the not-yet-posted article, specified either by filename or from standard input, and creates a signature for it, which is then inserted in the X-Auth: header. The article, ready for posting, appears on the standard output.
>
> In the configuration section at the top of the script, the moderator may build in the default name of the newsgroup or user, PGP User Id to be used for the signature, and the corresponding password. This is simply for convenience, since spammers are not so likely to go cracking the computer to get the password, and it is a relatively simple matter to generate a new user if it is, indeed, compromised. For the paranoid, like myself, if the password is not configured into the script it is read from the terminal instead.

pmcheck usage: pmcheck [newsgroup|user] [article]

> This script takes the article, specified either by filename or from standard input, and checks that the X-Auth: line is something it considers to be correct and that the article has not been tampered with. Pmcheck returns successfully if everything checks out. Otherwise it will return failure and issue one of a number of error messages, see Figure 2.
>
> Anybody can run pmcheck. It behaves slightly differently depending on the existence of a file called (by default) PGP_Moose_accept, and the presence or absence of a newsgroup or user argument. This file, if it exists, should contain lines with a newsgroup name or email address, some whitespace, and the PGP User Id

```
Posting for $NEWSGROUP not approved with PGP Moose.
Invalid designated signature from $GROUP
No public key for signature $GROUP
Signature doesn't match $FILE for $GROUP
```

Figure 2: Pmcheck error message

```
sci.crypt.research        moderator <ggr@sydney.sterling.com>
sci.crypt.research        moderator <pgut01@cs.auckland.ac.nz>
ggr@sydney.sterling.com   Greg's News <ggr@sydney.sterling.com>
```

Figure 3: Multiple moderators

approved by the moderator or user (usually made up specifically for this purpose). Multiple lines for the same newsgroup/user are allowed. For example, see Figure 3. If such a file exists, and a specific newsgroup or user is specified, pmcheck is silent if all is well, and issues the last of the error messages above if everything else was all right but the signature was from the wrong person. There must, in this case, be a signature applying to the designated newsgroup or user.

Without such a file, or if no specific newsgroup or user is given, all the signatures in the article are checked. In this case it is not considered an error if the signature cannot be checked due to a missing public key. If each signature is otherwise valid you will get a message like:

```
Valid signature from '$SIG'.
```

In any case, if there is a problem with a signature mentioned in the PGP_Moose_accept file, it will be reported and an error status will be returned.

pmcanon
pmnewsgroups
These two scripts are used by pmapp and pmcheck to recreate the exact input for the signature, and to extract the list of newsgroups in the header, respectively. More documentation is in their manual pages.

The PGP Moose checking daemon is packaged separately, as there would not seem to be a lot of value in having too many people running it. Accordingly, I was less precise in making it run absolutely everywhere. It requires the Korn shell or equivalent, and perl, and currently only interfaces to INN. I expect it would be easy to interface it to CNews, but I don't have one.

pmdaemon
Runs pmcheck to check the X-Auth: header for each controlled newsgroup for each article that arrives in an appropriate newsgroup. Mail is sent about any errant articles, and automatic cancellation may be enabled.

pmcancel
prepares a cancellation message based on the headers of another message.

When (if) I get a chance, I will create mail server scripts that allow moderators who are not using Unix to use these facilities. The first allows a moderator to mail a PGP signed copy of the article to be posted. The server will then verify that the moderator sent it, and post it with a (different but corresponding) approval. The second will accept an article and return something that you can check the signature on. Either way, any moderator will still need PGP.

How Do You Register For The Service?

Ahhh, this is the hard part. After all, it would be pretty undesirable if someone, meaning well, took any old body's word for it that some important moderated group should start working this way, before the moderator was able to start approving postings. A great way to hijack a newsgroup. Similarly for hijacking some other user's postings (tempting though it might be :-).

Another possibility is that someone, having seen what the valid signature looks like, simply creates a whole new PGP key that happens to have the same PGP User ID. Then they can sign and post stuff too.

The solution to both of these problems is the classical one for public key systems. You need either a certifying authority or the PGP Web of Trust. We're using the Web of Trust. This means that the moderator of the newsgroup must establish a path through only other people who are trusted, to the administrator of one of the PGP Moose checking Daemon sites.

For each newsgroup that wants to utilise this program, the moderator will have to create a special PGP key pair (preferably 512 bits to keep the X-Auth: lines short), and sign it. They must then establish a path of trust to someone who is running the PGP Moose server. It will be up to the administrator of that server to make sure that only trusted moderators' keys ever get into the server's keyring. In one case I achieved this by telephoning the University switchboard and asking for the moderator, then verifying the key's signature over the phone.

There can be no shortcuts to this procedure. Otherwise we are all back where we started.

In the case of an individual user, again you should establish this verification path to one of the administrators of the PGP Moose service. Contact me <ggr@usenix.org> for the time being to mutually figure out how to do this.

Handling Multiple Moderated Groups:

When I first proposed this tool, I was under the impression that posting to multiple moderated groups was an aberration that should be stamped out. This turns out not to be the case, and revisions to support this have been the cause of some delay in the deployment of this tool.

When the news system sees that an article has been posted to one or more moderated groups, it checks for an Approved: header. If the header exists, the article is accepted and processed normally, otherwise it is mailed to the moderator of the first moderated newsgroup mentioned in the Newsgroups: header. There seem to be three cases of interest.

The trivial case, and the most normal one, is that there is only one moderated newsgroup mentioned. The moderator approves the posting, and it is done.

The next, and probably most important, case, is when a moderator wants to cross-post a FAQ to their own group, as well as news.answers (for example). In this case their approval counts for both groups, so they can insert the Approved: header and post away. Presumably the other groups are not under the control of the PGP Moose Daemon. In this case the moderator can just go ahead and put in the Approved: header, and save themself and pmapp a lot of time. It will be passed right through.

The other case is harder to get right. This is when the article really is meant to be posted to two (or more) unrelated moderated newsgroups. Currently, if the first moderated group's moderator approves the posting, the other ones never hear about the article, at all. If this second group is controlled by the PGP Moose an automatic cancel will be generated. So it becomes very important for the moderators to do what they should have been doing already, namely forward the article to the next moderator. This tool can't help people who don't use it, but it provides some support for those who do.

The approval script checks whether there are any moderated newsgroups left that don't have X-Auth: headers for them. If there are none left, an Approved: header is inserted and the article gets posted. Otherwise, it issues a warning, and re-orders the newsgroups header with a newsgroup which is moderated but has no X-Auth: line at the start. When the article is posted, the news system will forward it to the moderator of the (new) first moderated group. If all moderators are sensible, and check for moderated newsgroups in this fashion, the mess should sort itself out and the last moderator will go ahead and post it. A warning message to the subsequent moderator NOT to change the article is also inserted, since such a modification would invalidate the previous signatures.

To ease this process, a second type of X-Auth: header is supported. this has the form:

```
X-Auth: None ... Newsgroup
```

The important fact about this is that the newsgroup appears last on the line, allowing a sort of partial approval, from moderators who don't use the PGP Moose.

The Newsgroups: line is split into a sorted list of newsgroups in order to generate the digital signature. Note that this means that once an article has been approved and authenticated by one moderator, it cannot be altered in any way by a moderator of a subsequent group, including altering the set of newsgroups mentioned in the Newsgroups: header, the body of the posting, or the other headers mentioned above.

Possible Problems

If an article is truly mangled, e.g., by truncation, it will fail the authentication and be cancelled. Until it is demonstrated otherwise, this is assumed to be a rare and minor problem. When a cancel is issued, mail is sent to the moderator of the group telling them, and they can tell us if it becomes a problem.

Currently the signature produced is assumed to be a PGP version 2.6 compatible one.

As with most things, the true vulnerabilities don't necessarily show up straight away. The ones that I have thought of are pretty simple. (I'm thinking in terms of its primary purpose here, for moderators to sign approved postings, and auto-cancellation of anything not properly approved when it reaches another site.)

If you have access to modify news as it is being propagated, and you wanted to harass a poster or the moderator, you could intentionally alter the piece of news. Without the PGP Moose, your alteration would go undetected, so you could change the sentence "I love Newt" to "I hate Newt" (or vice versa) and damage the poster's reputation (and maybe get them sued for slander). With the PGP Moose, the change will be obvious and the article will be auto-cancelled. The attack here is that it is a very anonymous way to get articles cancelled – the PGP moose does it for you, and there is no way to trace down that you caused the corruption.

Suppose someone posts something badly thought out in a moderated group. You could keep a copy of the posting, and keep reposting it every few days. This would make the original author and the moderator of the group pretty upset. Note that the date of the approval is part of the signature, so there is no question about whether or not the originator really reposted it himself, but it would still be annoying.

The PGP Moose only includes certain headers in its signature. You could take an article and insert other headers (anything starting with "X-" is otherwise ignored) which offend people. However the really important ones can't be fiddled.

Status

These scripts are implemented already, except as noted above. The approval scripts are considered production quality, and can be obtained from my web page, They checking daemon is also in day-to-day use and is available via the World Wide Web interface mentioned. There is a mailing list (maintained using Majordomo) called `pgp-moose@ornl.gov` for people using and maintaining the software.

In the meantime, if you want to use the tools, or particularly if you want to run a PGP Moose checking daemon, contact me (ggr@usenix.org).

Obtaining, Installing, Configuring

I regret that I don't have a public ftp site, but I do have a web page where you can get a shar archive of the approval code. It is http://usenix.org/˜ggr/PGP-Moose.shar off my home page.

It is hard to talk in detail about installation and configuration, since many users are not in charge of their own news server configuration. In my case, I run all of the things out of a subdirectory of my home directory. The only thing outside this area which must be changed is the INN newsfeeds file, if you are running the checking daemon. So, get the distribution file as above and unpack it wherever you want it to live.

There are configuration sections at the top of pmcheck, pmapp and pmdaemon. I like to think that they are relatively self-explanatory. One of the harder decisions is whether to use a separate keyring for PGP Moose applications or not. It is very strongly recommended that you do, if you are going to run the PGP Moose checking daemon, as the keyring files will need to be readable by the userid which INN runs under (usually "news"). Most of these options can also be overridden by environment variables or command arguments, so it is possible to leave the scripts unmodified and simply put a wrapper around them (which is what I do).

In the case of pmapp, the newsgroup or user that the authentication applies to can be specified on the command itself; The PGP user id and password, and the Approved: header's contents, can be specified by environment variables PMUSER, PMPASSWORD and APP, respectively.

For pmcheck, the important one is the name of the configuration file specifying which signatures are valid for which newsgroups or users.

Pmdaemon runs from INN, and needs some special care to set it up. "news" needs access permission to the directory and files for PGP Moose, and also read permission on the public keyring. Note that PGP creates keyrings with only owner permissions. The search path is rarely correct, and should be set at the top of the pmdaemon script. There are also a number of file names and mail addresses, but the comments should be clear enough.

Lastly, you want to incorporate pmapp in your moderation script and possibly your posting script. In my case, the last line of my posting script basically said

```
/usr/local/news/inews -S -h <tempfile
```

but now it says

```
pmapp <tempfile | \
      /usr/local/news/inews -S -h
```

To authenticate postings as an individual (as opposed to a moderator) I had to take a copy of the installed Pnews script, make sure it came earlier on my search path than the normal one, and modify that. You have to be careful that no extra signature files get appended after the pmapp is executed. Again, immediately before the "inews" call is the right place. I'm not sure whether this will work for all versions of news, this is not really my field of competence.

History and Experience

A few years ago, the idea of starting to post advertising to Usenet in bulk quantities started to take hold. The law firm Canter & Siegal raised this to an art form and even published a book about how to do this. In retaliation, a person who chose to call themself "Cancelmoose" started to forge cancellation messages for such postings. At the time, the very thought of forging cancel messages was considered a sin. After a while Cancelmoose (the person) created Cancelmoose (the program) to do the job automatically.

In early 1995, and again in April, there were a couple of spams which carried Approved: headers. I had recently become a co-moderator of `sci.crypt.research` and was at first amused, and later depressed, by the proposed solutions flying back and forth on the Moderators' mailing list. (Some moderators asked to be removed from the list due to the noise level!) I wrote the first version of the PGP Moose scripts, but without any news system hooks and without the capability to handle multiple moderated newsgroups. Things went quiet for a while, and I was waiting for a collaborator who knew more about NNTP and INN than I did to integrate it with news. In the meantime there was a reasonable amount of feedback regarding the need for support of multiple newsgroups in particular.

In September I gave up on my hypothetical collaborator and learned enough about INN to integrate the PGP Moose into posting and checking roles. I had spent a lot of time on the multiple moderators problem. I re-announced the software and got a number of interested parties, but it was now six months since the last bulk spamming of moderated newsgroups.

The PGP Moose fully controls postings to `comp.society.privacy`, `soc.russia.moderated`, `misc.news.bosnia` and `comp.std.c++`, as well as a number of German newsgroups which I don't see, and partially controls `sci.crypt.research`. By "partially controls", I mean that not all of the moderators use the tools, so the checking daemon verifies that, in the absence of a strong digital signature, at least the "approved" line is one that it is prepared to accept. Four or five individuals also use it to approve their personal postings. It is also incorporated in Igor Chudov's robo-moderation software.

Debugging the software has proven interesting. DEC's ULTRIX has an amazingly primitive shell as its `/bin/sh`, which took a week or two to track down.

The problem with a digital signature of any kind is that you get no indication of what went wrong. In the very rare cases when a digital signature failed to check out when an article arrived at my machine in Sydney, we had to attempt to recreate the input correctly for comparison. Since the moderator then sent that input to me by e-mail, there was a finite possibility that the same mangling happened along the way. PGP encryption and ASCII armour has proven invaluable for getting the *exact* input from one place to another. One bug involved trailing spaces on Newsgroup: or Subject: lines.

Not all of the problems which need to be handled come from the PGP Moose itself. The posting software used by one of the `comp.std.c++` moderators insists that there should be a space after every colon on the subject line, but it puts it there after the signature is generated. This is interesting given the use of "::" in C++.

Except for things which could be called deficiencies in the design, there has not yet been any identifiable corruption of an article in transport. Thus one of the bugbears of the original discussion has simply failed to materialise.

Len Levine, moderator of `comp.society.privacy`, has been a great help with testing the software. He posted two copies of a message describing what the PGP Moose was intended to do, one with authentication, and one without, and asked for replies from anyone who saw only one of the two messages. Out of an estimated readership of 10,000, there were two replies, and one of the people had replied to the message which did have the extra header! He also intentionally mangled one article to verify that the software noticed, which it did.

The big challenge at the moment is to get more sites to run the checking daemon, and to use the power of the digital signatures to automatically distribute configuration information. Currently `ornl.gov` and `qualcomm.com` are running well-connected Checking Daemons.

Other Efforts

The newsgroup `comp.os.linux.announce` uses PGP to sign the body of each valid message. Its moderator, Lars Wirzenius, is more concerned that his readers be able to verify the postings for themselves than with automatic cancellation.

The moderator of `comp.dcom.telecom` uses a very simple scheme with MD5 hashing of the body of the message with a secret password. He has trusted users running a copy of the script at other places to generate automatic cancellations. The only problems with his approach are the issues of sharing the secret key around, which prevents this solution from scaling well, and the fact that the headers of the message are unprotected.

David Lawrence, more commonly known as `tale@uunet.net`, has begun using a similar concept for validating newsgroup control messages. In his case there are patches to be incorporated into the news servers. It is a pity that these two solutions to similar problems are not better integrated.

Author Information

University of New South Wales B.Sc. Worked for Fawnray Pty Ltd (became Neology Ltd), Softway Pty Ltd, Visiting Scientist at IBM's Thomas J. Watson Research Center, the Australian Computing and Communications Institute, RoSecure Software Pty Ltd, Sterling Software, and currently with Qualcomm Australia. Interested in the issues of distributed system administration, including network management, and development of tools for more effective and cost-effective use of network environments. Also security and privacy considerations, including use of cryptography.

References

[1] *Protect Your Privacy*. Stallings, W. Prentice Hall PTR, 1995.

[2] Kantor, B., and Lapsley, P. *Network News Transfer Protocol*. RFC 997, February 1986.