Moira and AFS in Theory and Practice: A Survey of Security Awareness in Actual Practices of the MIT Community

Ray Hua Wu

Abstract— **The Massachusetts Institute of Technology (MIT) is infamous for the lack of security in its electronic infrastructure, often patching vulnerabilities at a glacial pace and frequently relying on the delicate cover of security by obscurity and wishful thinking that a system will happen to not have holes. By focusing on Moira and AFS in particular, as well as their interaction with each other, we note how thin its veil of security really is, and furthermore examine to what extent its usage by members of the MIT community demonstrates general community awareness of proper usage of MIT's services.**

I. INTRODUCTION

Moira is a database management system for the retrieval and modification of information regarding mailing lists, printers, and several other electronic services in a Kerberos-authenticated system. Its majority use case among members of the MIT community is the management of mailing lists, for which multiple in-browser and command line interfaces exist.

Moira mailing lists could have several properties set. They could be private, indicating that only owners of the list can add members to the list, or public, indicating that anyone with credentials to WebMoira could add themselves (but not others, unless they also own the list). They could be visible, indicating that anyone can view the properties and membership of the list, or hidden, indicating that only the owners can see the list's properties and membership (in particular, not even members of the list that aren't owners can see the list's properties and membership). They could also be set to be groups upon which ACLs can be set for directories of AFS. There are also some other properties that this paper will not touch on. In the cases of all Moira lists, any user with Kerberos credentials can *remove* themselves from a list they're on, regardless of whether it's public or private or whether it's visible or hidden. This is a good idea: everyone should have the right to unsubscribe themselves from a mailing list when they see fit, and not have to deal with the traffic of a mailing list just because someone else added them. We'll later investigate how available this option really is in practicality.

Another feature of Moira lists is that one could add not only users but 'strings' to the lists. This allows people to place e-mail addresses outside of the MIT domain onto a Moira list. Note, however, that in these cases such people will be unable to remove themselves from the Moira list. Particularly problematic is that 'strings' without a specified domain are assumed to be @mit.edu, which we will touch on later.

One means of managing Moira lists is WebMoira[1]. The WebMoira interface allows a user, given provided Kerberos credentials, to make mailing lists, to view the membership and properties of mailing lists they own and for which they're on (in the latter case, as long as the list is not hidden), and to add and remove themselves or others from lists provided they have the permissions to do so, as well as to alter the properties of lists that they own. To facilitate ease of use, WebMoira implements several user-friendly features, like providing autocompletion suggestions for queries made for list lookups and for list or user additions. In previous work, Scott Robinson demonstrated that the autocompletion feature for adding to a list membership was insufficiently cleansed, and showed an XSS vulnerability[2]. Despite this revelation occurring over a year ago, the vulnerability is still not patched.

[1] http://groups.mit.edu/webmoira/

[2] http://css.csail.mit.edu/6.858/2017/projects/srobin.pdf

The command line utility blanche provides many of the list management capabilities that WebMoira provides, plus a few more. One of these is the specification of a MemACL for a list: to give a user or list permissions to alter the membership of a list but not to manage its properties. As we will see later, the blanche utility both does not have certain problems WebMoira has and also exhibits some problems WebMoira does not.

Curiously, neither WebMoira nor blanche gives the general MIT community member a functional means to delete a mailing list. In order to do so, one must resort to other Moira interfaces, like qy and moira. The former, qy, is infamous for being deplorably badly documented and to seem like utter incantational magic, and has already been known to contain vulnerabilities[3].

AFS (the Andrew File System) is MIT's file system of choice, on which files in Athena, MIT's computing environment, are stored. A key distinction of AFS from the typical UNIX filesystem is that read and write permission bits of individual files are ignored and overridden with directory access privileges. AFS's fs command suite includes the command fs la, which can be used to display the permissions of a directory. The various permissions that can be granted are l (lookup permissions: practically permissions to list and to see permissions), r (reading permissions), i (inserting permissions), d (deleting permissions), w (writing permissions), k (locking permissions), and a (administrative permissions). These permissions can be granted for individual users or many users treated as a group. As hinted above, a key interaction between Moira lists and AFS is that Moira lists can be made to also be AFS groups, and thus be on an ACL for a particular directory in AFS. The fs command suite also includes fs sa, which allows for changing the ACL on a directory for a particular group or individual user. AFS, interestingly, also has a web interface[4].

We will investigate how these systems are practically used by the MIT community, and note how common it is for them to be used in clearly insecure ways. Afterwards, we will discuss means of addressing the levels and forms of misuse of these systems in practice.

II. MOIRA SYSTEMS

A. Practical Assumptions of List Management Capability in the General MIT Community

Although there exists an entire palette of different tools with which to interface with Moira lists, from WebMoira, to blanche, to qy, to the moira command itself, the vast majority of members of the MIT community are unfamiliar with most of the capabilities of these tools. It is perennially evident in several public mailing lists of the MIT community that a significant number of people are not even acquainted with WebMoira, given the frequency of e-mails requesting addition to or removal from the list. As this is the only tool that is not a command line interface, any user unfamiliar with the command line cannot be assumed to be able to use the other options, and most that are familiar with the command line can only be reasonably assumed to be familiar with blanche.

B. Availability Attacks

The set of characters a Moira list's name is permitted to contain is not the same set as characters permitted by the listmaker tool. Beyond the listmaker permitted character set, $ and ' can be used in Moira list names, although they will have to be arrived at via renaming an already-made list (an operation not supported by WebMoira). WebMoira fails upon a request to resolve a name containing either of these characters, redirecting the user back to the WebMoira home page. This means that a community member could add someone with only WebMoira capabilities to a list they can't remove themselves from by simply including a $ in the list's name.

In fact, because the blanche utility is written with suboptimal argument parsing, and thus attempts to read any first argument to blanche starting with a hyphen (-) as an option, one could craft a list name beginning with - and containing a $ somewhere in it, and create a list that can't be operated on with either WebMoira or blanche, excluding the vast majority of the MIT population from effective list management.

[3] Specifically, the 'geml' operation can be used to bypass the hidden property and view the contents of a hidden sub-list.

[4] https://stuff.mit.edu/afs/

On an even more fundamental level, though, causing a user or group of users on a list to receive mail they don't desire and have a hard time unsubscribing themselves is as easy as adding them to the list as a string rather than a user, because of the @mit.edu default assumption.

A malicious user could also not just target a particular user for mailing list inconvenience, but also everyone using Moira lists. Since mutating operations on the membership of Moira lists require the updating of all addresses anywhere recursively within the sublists of a list, the time to complete an operation in Moira scales with the depth of a Moira list; these operations can cause such a slowdown that Moira lists with 4096 sub-lists are actually hard-prevented from being added to another list. Attempting to add a Moira list with too many sublists, like list-of-lists-of-lists@mit.edu, to another list gets the user a `Moira internal consistency error`. Nevertheless, lists with nearly this number of sublists can still cause hours of Moira operation backup, creating significantly noticeable inconvenience for anyone else trying to update their list at the time. This has, in fact, occurred inadvertently in the past.

Ironically, the fact that a hard cap to the sub-list capacity of a Moira list presents an extremely theoretical and nearly completely impractical means to prevent another user from adding a list one owns to a different list, which comes with the effect of thwarting hidden list content attacks based on unchecked recursive list query. In fact, one could even still modify the list sub-structure of sub-lists, as the check for number of sub-lists only happens upon a list being added to another list, not changes to its internal structure. Hence, a list can actually have more than 4096 sub-lists, as long as one performs the list additions in the correct order. This is a property of Moira lists just waiting to go wrong.

It is also possible to cause a mailing list to effectively send mail to more than 4096 different lists without strategic ordering of list additions, because once again, lists can be added as strings rather than as the lists themselves. In fact, it is completely possible to create a mailing list loop simply by adding a list A to list B, and then adding list B itself as a string to list A, bypassing the check that prevents a list from being added to itself or any of its sub-lists.

C. Permission Sets on Lists

Moira performs no checking to reason if the set of properties a user desires a list to have makes any sense from a security perspective whatsoever. One actually fairly well-known problematic combination is for a list to be both public and owned by itself. We will discuss this combination here and being both public and a group in the next section, on AFS.

When a user specifies that a list is both public and owned by itself (often simply referred to as ”self-owned”), the user claims they want anyone to be able to add themselves to the list, but also that anyone on the list has full permissions to modifications to the list; in other words, anyone has full permissions to do what they want to a list, and they obtain this by simply adding themselves. This allows arbitrary hijacking of a list from someone who manages to discover it.

Certain manifestations of social forces make it easy to unwittingly create public self-owned lists. When a list was originally owned by one user, but that user has graduated and is about to lose their Athena account, it takes only a moment of not wanting to deal with finding specifically who to will a list to and to just make the list self-owned for its ongoing users to figure out themselves, while not remembering that the list has been made public, for this result to manifest. A simple alert upon detecting this combination of properties could prevent several cases of this; Moira does already perform some level of sanity checking, for instance to automatically populate a list upon creation with the list's creator if the list was created to be self-owned and private; the idea of sanity checking could simply be expanded.

Finding public self-owned lists is not difficult; for those that do not wish to write a script to recursively make a combination of `blanche`, `qy geml`, `qy glom`, and `qy gaus` queries, there's always Zixiao Wang's subscribe service[5], which will friendlily cater a randomized platter of public lists for sampling to you.

[5] https://garywang.scripts.mit.edu/subscribe/

D. When is a MemACL a good idea?

In giving a user or a list MemACL permissions to a Moira list, one provides them rights to add or remove everyone. Although it is quite reasonable to provide the rights to add anyone to a list while not providing rights to change the list's properties, it is awkward that there does not exist an option to provide just the add and not the remove permissions. In particular, in the event of a public list MemACLed to itself, although this situation is far better than a public list owned by itself, anyone could time removing a member of a list such that they would not receive a crucial e-mail when the expected time of arrival of the e-mail can be well predicted in advance.

III. AFS IN PRACTICE AT MIT

A. Permissions in User Lockers

Across Athena's users, there are several different permission sets of home directories represented. One notable distinction is between users with `system:anyuser` permissions on their home directory set to `l` (listing permissions only) and users with `system:anyuser` permissions on their home directory set to `rl` (reading and listing permissions).

It is very possible that users that chose the latter are in fact aware of the ramifications of their choice, and have taken the necessary precautions to make such a choice a reasonable idea, that is, either automatically cleaning up sensitive dotfiles in the home directory or setting up symbolic links to direct data that would be written in the home directory elsewhere. In a quick survey of users with `system:anyuser rl` home directories, though, this does not appear to be the norm. Numerous users with such home directories do not redirect their `.bash_history`, and in some even worse cases, do not redirect their `.my.cnf`, which stores their sql.mit.edu password in plaintext. Some users that use zephyr do not redirect their `.zephyr.subs`, which means that secret classes they're subscribed to are not secret at all if the only source of secrecy is a communal agreement to not share the class name, and encrypted secret classes can still be discovered, allowing an adversary to subscribe there and analyze traffic patterns at the encrypted class. If `.crypt-table` is also in the user's home directory, then an adversary can even know the location where keys for the crypt classes are stored. And because new directories that are made inherit the permissions of the directory it was made in, if such a user did not remember to edit permissions of new directories made to store crypt keys, the new directories would take the default value of `system:anyuser rl` and all crypt keys they store are accessible to anybody. Because of the existence of the `pts ex` command and its capability to lookup by ID number, systematically finding all users with lenient permissions in their lockers is a matter of finding the highest pts id before the massive gap for root instances and iterating lookups for ID numbers up to this determined number.

What is arguably particularly problematic about this situation is that one's default Athena setup provides some incentives for users to make their home directory universally readable, in that one could set a `~/.plan` file that other users could read via running `finger` on them, in a sense providing an option to personalize their Athena presentation. (In addition, due to the high prevalence of `system:anyuser rl` home directories in older accounts, it seems it may be the case that users may have had their home directories universally readable by default when they received their Athena accounts. If this has indeed been the case, it is a good step that the decision was made to discontinue this default, but it also means many old users are likely even less aware of the implications of this permissions set.) In fact, the `finger` command, among the other means via which it can be argued to be multitudinously privacy-insensitive, is an excellent litmus test for whether someone's home directory likely exhibits exposure of sensitive information.

B. Permissions in Activity and Class Lockers

Many student organizations and classes at MIT choose to utilize an Athena locker, and several add ACLs to their locker for various lists of people, so that they could directly contribute to the contents of the locker. As Moira lists support also being an AFS group, many organizations and classes provide permissions to people on certain Moira lists. Unfortunately, in several instances, the organization and class both decide to make the associated Moira list public and provide the list generous permissions to the locker. In particular, 15 classes, 2 of which are in the Department of Electrical Engineering and Computer Science (course 6), provide at least `rl` permissions to a public Moira list, and in the vast majority of these cases (including both course 6 classes), the permissions given are `rlidwka`, the entire complement of capabilities. That is, any member of the MIT community who uncovers this could change the contents of the activity of class locker to their will; there are zero electronically-enforced barriers stopping them.

It tends to be the situation in these lockers that the `OldFiles` directory, intended for use as a backup (though even still only processed once a night), also has `rlidwka` permissions as well, thus allowing any attacker to modify data in the backup as easily as they could modify the original file. Though IS&T also offers a ”backup from tape” option in the event `OldFiles` fails, they guarantee no less than a week to process such a request, so effects could be drastic for well-timed attacks.

Finding Moira lists that are also groups with bad permissions is even easier than finding users with lenient permissions on their home directory: as any MIT community member can create an AFS group at any time, and these group IDs are assigned sequentially, one can simply make an AFS group, note its ID, and iterate through the group IDs less than this maximum.

Among such Moira lists for activities, there exist multiple that are in fact all three of public, self-owned, and an AFS group: any arbitrary member of the MIT community has the full power to immediately upon discovery hijack the access control list and obtain full permissions to the activity including removing control from the current owners, and the only thing in their way is obscurity, which is taken away as easily as the above-mentioned iteration.

IV. PROPOSED SOLUTIONS

Several of these issues are greatly due to the obscurity of information about systems involved. Many of them could be greatly palliated by warnings that make users more aware of the implications of what they have just tried to do. If Moira's systems are unwilling to plainly disallow making a public list self-owned or a group, it could at least provide a warning to the user that such a change is about to be effected, as these are states of mailing lists one could easily bring about unintentionally.

Athena home directories should simply not come with a `~/.plan`. Very few people use it for a productive purpose anymore, and it tempts people to make their home directories universally readable so that witticisms can be viewed upon `finger`.

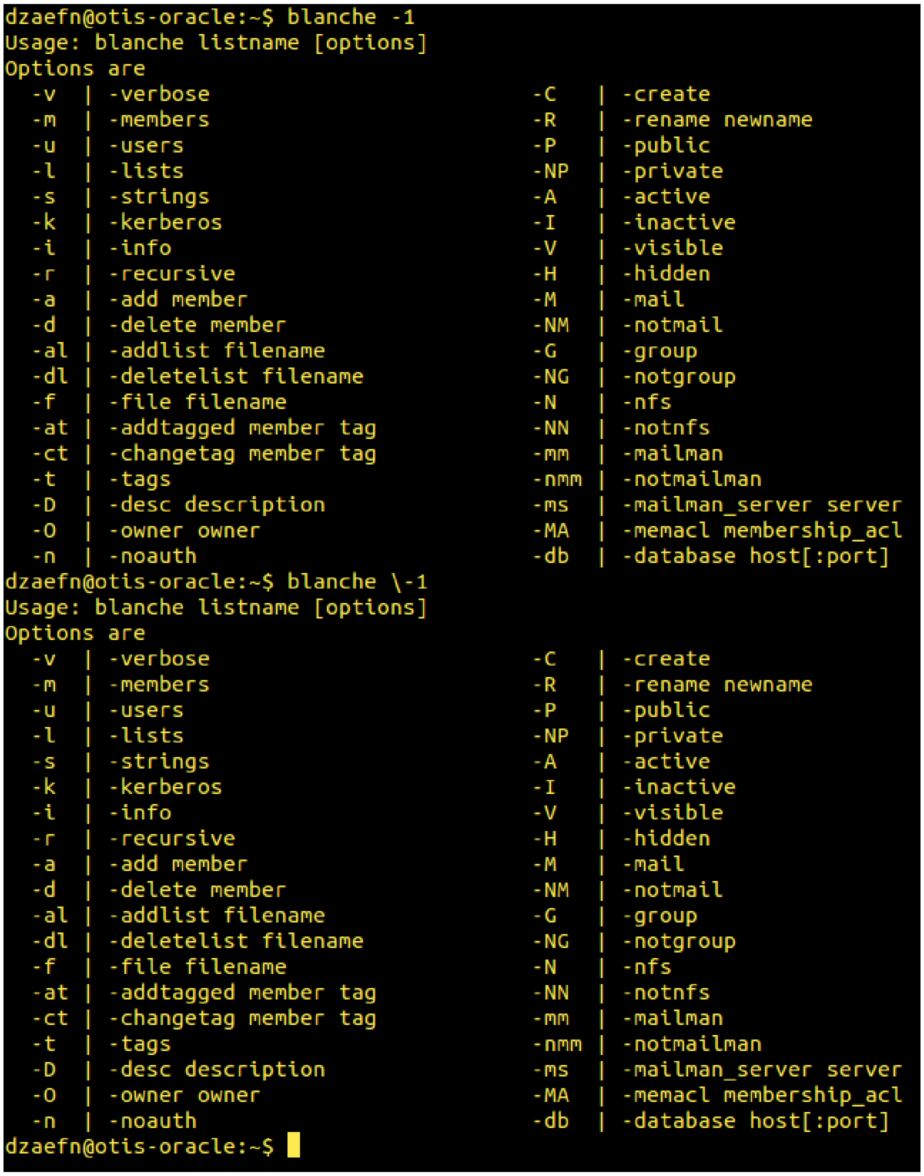
In some of these cases, there is a feature that really just does nothing except for causing issues. Strings added to a Moira list without a domain just simply shouldn't be defaulted to @mit.edu. There is no good use case for this.

The biggest overarching theme of all, though, is that MIT's electronic systems should stop being so dependent on security by obscurity. Many of its systems are very complex and for which there are convoluted paths of least resistance around security safeguards, in which case the lack of documentation contributes to lack of clarity and issues and does not stop the actually determined adversary. Systems should be explicit with warning upon dangerous states, as it may very likely be the case a user that effected such a state had actually no intention of doing so.

V. CONCLUSIONS

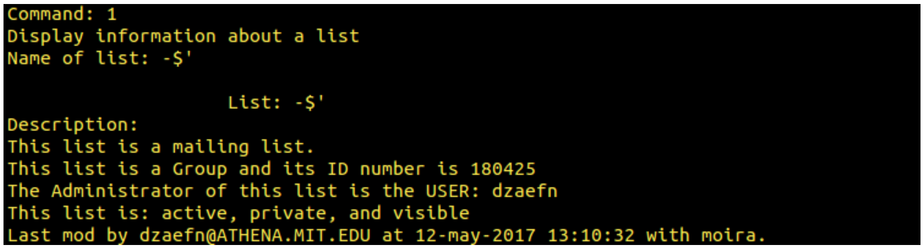
MIT's electronic infrastructure is full of security issues, but a possibly even greater problem is a lack of understanding in the general MIT community of proper usage of Athena's tools. Users, organizations, and classes alike configure their permissions poorly enough that in the worst cases any member of the MIT community could hijack control of their entire set of files on Athena.

Unfortunately, MIT's services and the documentation provided with them often use terms or present ideas in such a way that most users would not suspect dangerous security issues with casual usage. What's more, such documentation often specifically tries to avoid mentioning the security-sensitive cases, in order to effect a sense of security by obscurity. This ends up both not being effective and not warning users about potential pitfalls. Certain utilities available in Athena in fact guide users towards making security mistakes. In order for the state of security in MIT's computer systems to become less embarrassing, an important step is to equip utilities with warnings to users that an action they are taking is potentially dangerous, and to not include ”features” that incentivize dangerous usage, particularly those that add absolutely no value in the modern day.



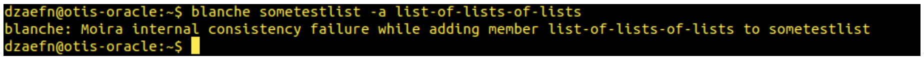
```
dzaefn@otis-oracle:~$ blanche -1
Usage: blanche listname [options]
Options are
  -v  | -verbose                         -C   | -create
  -m  | -members                         -R   | -rename newname
  -u  | -users                           -P   | -public
  -l  | -lists                           -NP  | -private
  -s  | -strings                         -A   | -active
  -k  | -kerberos                        -I   | -inactive
  -i  | -info                            -V   | -visible
  -r  | -recursive                       -H   | -hidden
  -a  | -add member                      -M   | -mail
  -d  | -delete member                   -NM  | -notmail
  -al | -addlist filename                -G   | -group
  -dl | -deletelist filename             -NG  | -notgroup
  -f  | -file filename                   -N   | -nfs
  -at | -addtagged member tag            -NN  | -notnfs
  -ct | -changetag member tag            -mm  | -mailman
  -t  | -tags                            -nmm | -notmailman
  -D  | -desc description                -ms  | -mailman_server server
  -O  | -owner owner                     -MA  | -memacl membership_acl
  -n  | -noauth                          -db  | -database host[:port]
dzaefn@otis-oracle:~$ blanche \-1
Usage: blanche listname [options]
Options are
  -v  | -verbose                         -C   | -create
  -m  | -members                         -R   | -rename newname
  -u  | -users                           -P   | -public
  -l  | -lists                           -NP  | -private
  -s  | -strings                         -A   | -active
  -k  | -kerberos                        -I   | -inactive
  -i  | -info                            -V   | -visible
  -r  | -recursive                       -H   | -hidden
  -a  | -add member                      -M   | -mail
  -d  | -delete member                   -NM  | -notmail
  -al | -addlist filename                -G   | -group
  -dl | -deletelist filename             -NG  | -notgroup
  -f  | -file filename                   -N   | -nfs
  -at | -addtagged member tag            -NN  | -notnfs
  -ct | -changetag member tag            -mm  | -mailman
  -t  | -tags                            -nmm | -notmailman
  -D  | -desc description                -ms  | -mailman_server server
  -O  | -owner owner                     -MA  | -memacl membership_acl
  -n  | -noauth                          -db  | -database host[:port]
dzaefn@otis-oracle:~$
```

Figure 1: The Athena `blanche` command fails to correctly parse Moira lists whose names start with a hyphen, even when the hyphen is escaped.



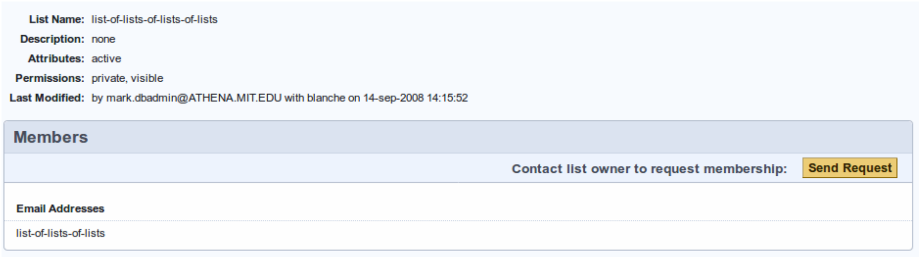
```
Command: 1
Display information about a list
Name of list: -$'

                    List: -$'
Description:
This list is a mailing list.
This list is a Group and its ID number is 180425
The Administrator of this list is the USER: dzaefn
This list is: active, private, and visible
Last mod by dzaefn@ATHENA.MIT.EDU at 12-may-2017 13:10:32 with moira.
```

Figure 2: This list, -$'@mit.edu, is a valid Moira list, but because it averts usage with blanche by starting with a hyphen and averts usage with WebMoira by containing a dollar sign and an apostrophe, operating with it (for instance removing oneself from it) is outside the capability of most of the MIT community.



```
dzaefn@otis-oracle:~$ blanche sometestlist -a list-of-lists-of-lists
blanche: Moira internal consistency failure while adding member list-of-lists-of-lists to sometestlist
dzaefn@otis-oracle:~$ █
```

Figure 3: Attempting to add list-of-lists-of-lists@mit.edu to a list results in a moira internal consistency error.



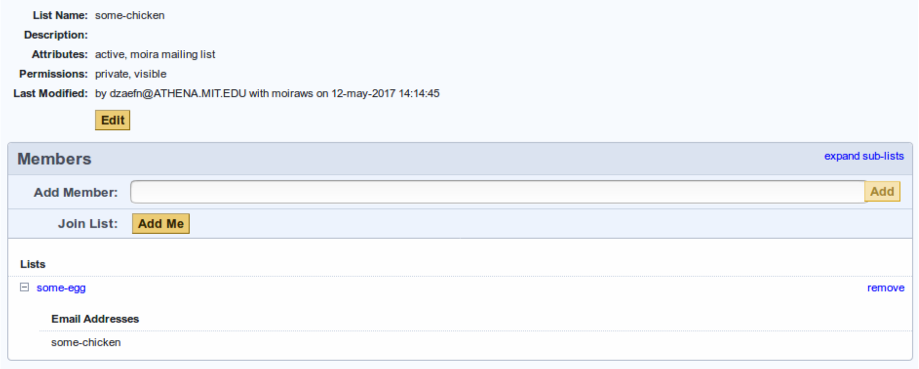
List Name: list-of-lists-of-lists-of-lists
Description: none
Attributes: active
Permissions: private, visible
Last Modified: by mark.dbadmin@ATHENA.MIT.EDU with blanche on 14-sep-2008 14:15:52

Members

Contact list owner to request membership: Send Request

Email Addresses
list-of-lists-of-lists

Figure 4: The fact that list-of-lists-of-lists@mit.edu is saturated with lists does not prevent it from being added to list-of-lists-of-lists-of-lists@mit.edu as a STRING.



List Name: some-chicken
Description:
Attributes: active, moira mailing list
Permissions: private, visible
Last Modified: by dzaefn@ATHENA.MIT.EDU with moiraws on 12-may-2017 14:14:45

Members

Add Member:

Join List:

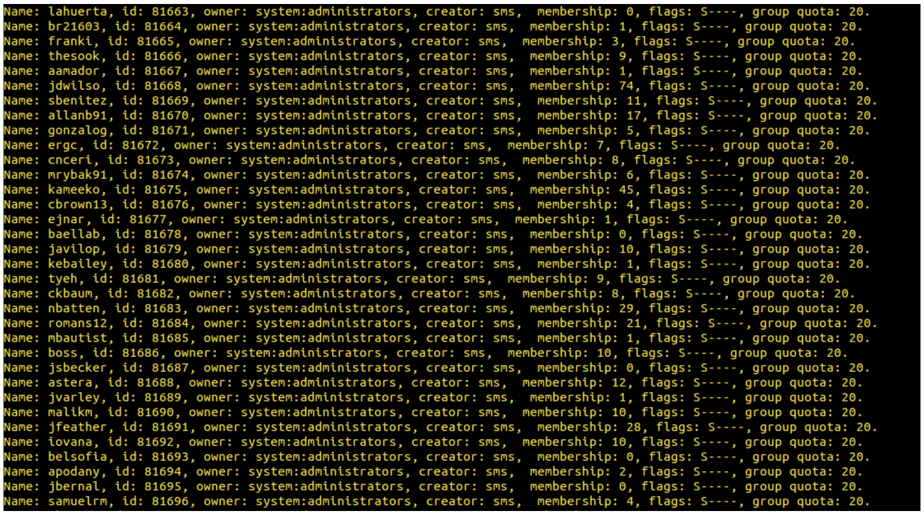
Lists

some-egg

Email Addresses

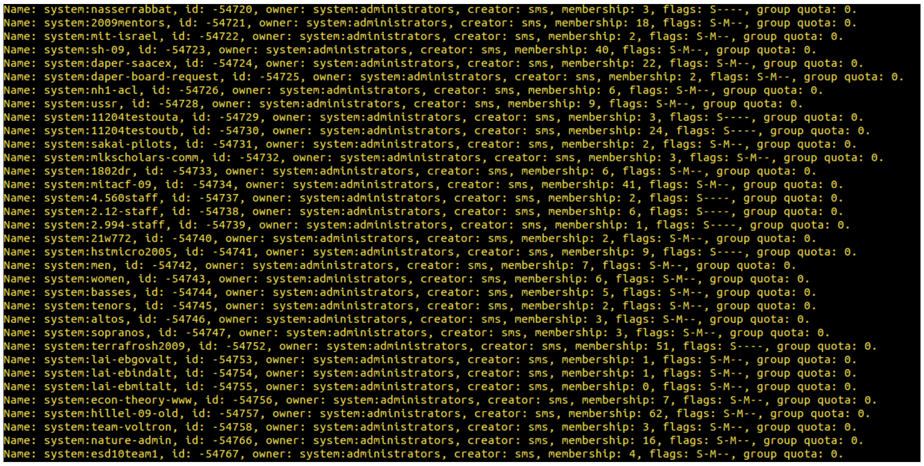
some-chicken

Figure 5: WebMoira is perfectly happy to allow a user to add a list somewhere in its sub-list structure, so long as it is added as a STRING. Without a domain, the STRING some-chicken will default to resolving @mit.edu.



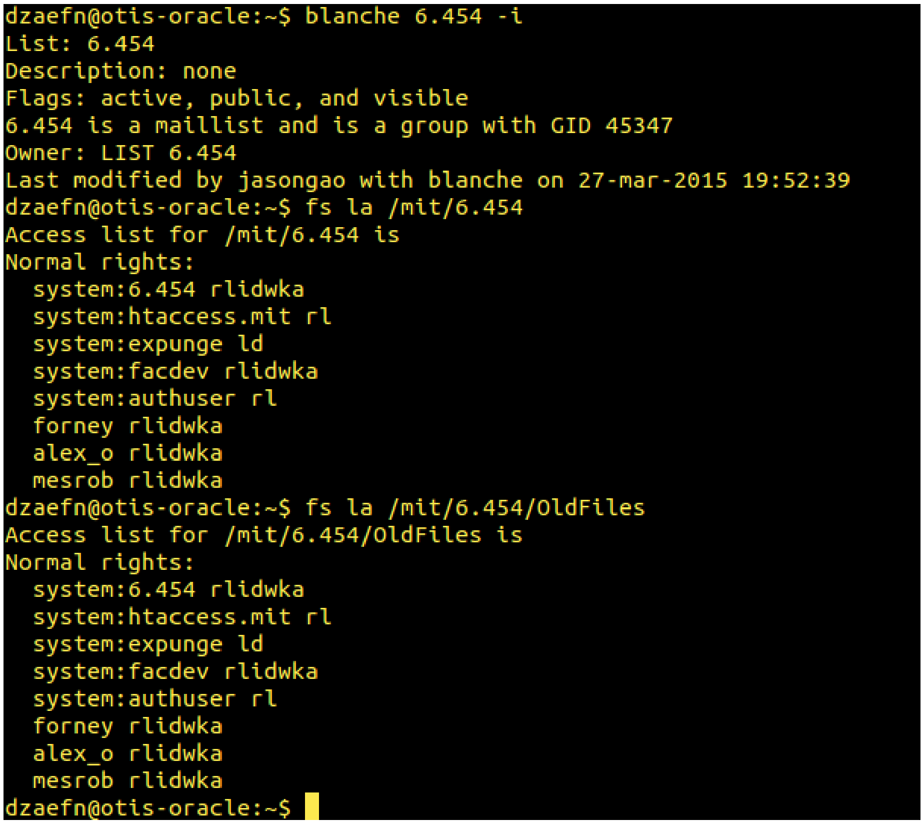
```
Name: lahuerta, id: 81663, owner: system:administrators, creator: sms,  membership: 0, flags: S----, group quota: 20.
Name: br21603, id: 81664, owner: system:administrators, creator: sms,  membership: 1, flags: S----, group quota: 20.
Name: franki, id: 81665, owner: system:administrators, creator: sms,  membership: 3, flags: S----, group quota: 20.
Name: thesook, id: 81666, owner: system:administrators, creator: sms,  membership: 9, flags: S----, group quota: 20.
Name: aamador, id: 81667, owner: system:administrators, creator: sms,  membership: 1, flags: S----, group quota: 20.
Name: jdwilso, id: 81668, owner: system:administrators, creator: sms,  membership: 74, flags: S----, group quota: 20.
Name: sbenitez, id: 81669, owner: system:administrators, creator: sms,  membership: 11, flags: S----, group quota: 20.
Name: allanb91, id: 81670, owner: system:administrators, creator: sms,  membership: 17, flags: S----, group quota: 20.
Name: gonzalog, id: 81671, owner: system:administrators, creator: sms,  membership: 5, flags: S----, group quota: 20.
Name: ergc, id: 81672, owner: system:administrators, creator: sms,  membership: 7, flags: S----, group quota: 20.
Name: cnceri, id: 81673, owner: system:administrators, creator: sms,  membership: 8, flags: S----, group quota: 20.
Name: mrybak91, id: 81674, owner: system:administrators, creator: sms,  membership: 6, flags: S----, group quota: 20.
Name: kameeko, id: 81675, owner: system:administrators, creator: sms,  membership: 45, flags: S----, group quota: 20.
Name: cbrown13, id: 81676, owner: system:administrators, creator: sms,  membership: 4, flags: S----, group quota: 20.
Name: ejnar, id: 81677, owner: system:administrators, creator: sms,  membership: 1, flags: S----, group quota: 20.
Name: baellab, id: 81678, owner: system:administrators, creator: sms,  membership: 0, flags: S----, group quota: 20.
Name: javilop, id: 81679, owner: system:administrators, creator: sms,  membership: 10, flags: S----, group quota: 20.
Name: kebailey, id: 81680, owner: system:administrators, creator: sms,  membership: 1, flags: S----, group quota: 20.
Name: tyeh, id: 81681, owner: system:administrators, creator: sms,  membership: 9, flags: S----, group quota: 20.
Name: ckbaum, id: 81682, owner: system:administrators, creator: sms,  membership: 8, flags: S----, group quota: 20.
Name: nbatten, id: 81683, owner: system:administrators, creator: sms,  membership: 29, flags: S----, group quota: 20.
Name: romans12, id: 81684, owner: system:administrators, creator: sms,  membership: 21, flags: S----, group quota: 20.
Name: mbautist, id: 81685, owner: system:administrators, creator: sms,  membership: 1, flags: S----, group quota: 20.
Name: boss, id: 81686, owner: system:administrators, creator: sms,  membership: 10, flags: S----, group quota: 20.
Name: jsbecker, id: 81687, owner: system:administrators, creator: sms,  membership: 0, flags: S----, group quota: 20.
Name: astera, id: 81688, owner: system:administrators, creator: sms,  membership: 12, flags: S----, group quota: 20.
Name: jvarley, id: 81689, owner: system:administrators, creator: sms,  membership: 1, flags: S----, group quota: 20.
Name: malikm, id: 81690, owner: system:administrators, creator: sms,  membership: 10, flags: S----, group quota: 20.
Name: jfeather, id: 81691, owner: system:administrators, creator: sms,  membership: 28, flags: S----, group quota: 20.
Name: iovana, id: 81692, owner: system:administrators, creator: sms,  membership: 10, flags: S----, group quota: 20.
Name: belsofia, id: 81693, owner: system:administrators, creator: sms,  membership: 0, flags: S----, group quota: 20.
Name: apodany, id: 81694, owner: system:administrators, creator: sms,  membership: 2, flags: S----, group quota: 20.
Name: jbernal, id: 81695, owner: system:administrators, creator: sms,  membership: 0, flags: S----, group quota: 20.
Name: samuelrm, id: 81696, owner: system:administrators, creator: sms,  membership: 4, flags: S----, group quota: 20.
```

Figure 6: A segment of the output of an exhaustive iterative query of users, using the lookup-by-id capabilities of `pts ex`.



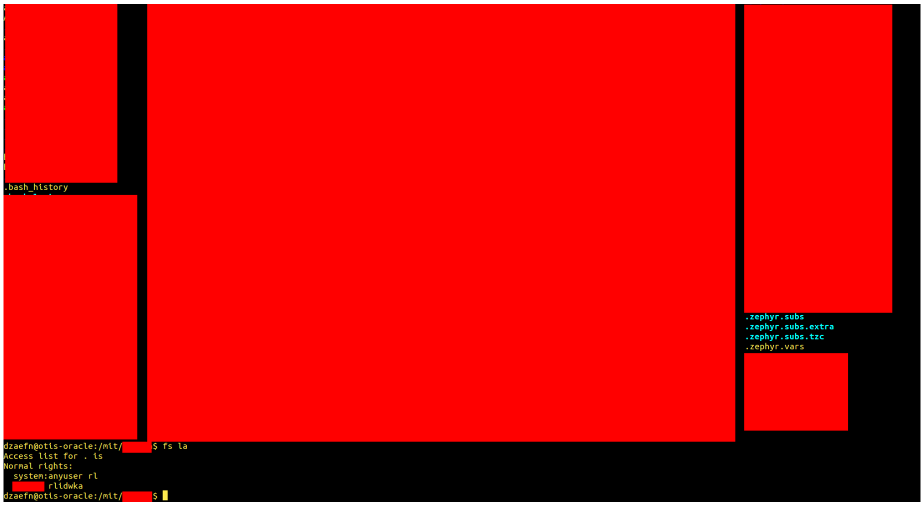
```
Name: system:nasserrabbat, id: -54720, owner: system:administrators, creator: sms, membership: 3, flags: S----, group quota: 0.
Name: system:2009mentors, id: -54721, owner: system:administrators, creator: sms, membership: 18, flags: S-M--, group quota: 0.
Name: system:mit-israel, id: -54722, owner: system:administrators, creator: sms, membership: 2, flags: S-M--, group quota: 0.
Name: system:sh-09, id: -54723, owner: system:administrators, creator: sms, membership: 40, flags: S-M--, group quota: 0.
Name: system:daper-saacex, id: -54724, owner: system:administrators, creator: sms, membership: 22, flags: S-M--, group quota: 0.
Name: system:daper-board-request, id: -54725, owner: system:administrators, creator: sms, membership: 2, flags: S-M--, group quota: 0.
Name: system:nh1-acl, id: -54726, owner: system:administrators, creator: sms, membership: 6, flags: S-M--, group quota: 0.
Name: system:ussr, id: -54728, owner: system:administrators, creator: sms, membership: 9, flags: S-M--, group quota: 0.
Name: system:11204testouta, id: -54729, owner: system:administrators, creator: sms, membership: 3, flags: S----, group quota: 0.
Name: system:11204testoutb, id: -54730, owner: system:administrators, creator: sms, membership: 24, flags: S----, group quota: 0.
Name: system:sakai-pilots, id: -54731, owner: system:administrators, creator: sms, membership: 2, flags: S-M--, group quota: 0.
Name: system:mlkscholars-comm, id: -54732, owner: system:administrators, creator: sms, membership: 3, flags: S-M--, group quota: 0.
Name: system:1802dr, id: -54733, owner: system:administrators, creator: sms, membership: 6, flags: S-M--, group quota: 0.
Name: system:mitacf-09, id: -54734, owner: system:administrators, creator: sms, membership: 41, flags: S-M--, group quota: 0.
Name: system:4.560staff, id: -54737, owner: system:administrators, creator: sms, membership: 2, flags: S----, group quota: 0.
Name: system:2.12-staff, id: -54738, owner: system:administrators, creator: sms, membership: 6, flags: S----, group quota: 0.
Name: system:2.994-staff, id: -54739, owner: system:administrators, creator: sms, membership: 1, flags: S----, group quota: 0.
Name: system:21w772, id: -54740, owner: system:administrators, creator: sms, membership: 2, flags: S-M--, group quota: 0.
Name: system:hstmicro2005, id: -54741, owner: system:administrators, creator: sms, membership: 9, flags: S----, group quota: 0.
Name: system:men, id: -54742, owner: system:administrators, creator: sms, membership: 7, flags: S-M--, group quota: 0.
Name: system:women, id: -54743, owner: system:administrators, creator: sms, membership: 6, flags: S-M--, group quota: 0.
Name: system:basses, id: -54744, owner: system:administrators, creator: sms, membership: 5, flags: S-M--, group quota: 0.
Name: system:tenors, id: -54745, owner: system:administrators, creator: sms, membership: 2, flags: S-M--, group quota: 0.
Name: system:altos, id: -54746, owner: system:administrators, creator: sms, membership: 3, flags: S-M--, group quota: 0.
Name: system:sopranos, id: -54747, owner: system:administrators, creator: sms, membership: 3, flags: S-M--, group quota: 0.
Name: system:terrafrosh2009, id: -54752, owner: system:administrators, creator: sms, membership: 51, flags: S----, group quota: 0.
Name: system:lai-ebgovalt, id: -54753, owner: system:administrators, creator: sms, membership: 1, flags: S-M--, group quota: 0.
Name: system:lai-ebindalt, id: -54754, owner: system:administrators, creator: sms, membership: 1, flags: S-M--, group quota: 0.
Name: system:lai-ebmitalt, id: -54755, owner: system:administrators, creator: sms, membership: 0, flags: S-M--, group quota: 0.
Name: system:econ-theory-www, id: -54756, owner: system:administrators, creator: sms, membership: 7, flags: S-M--, group quota: 0.
Name: system:hillel-09-old, id: -54757, owner: system:administrators, creator: sms, membership: 62, flags: S-M--, group quota: 0.
Name: system:team-voltron, id: -54758, owner: system:administrators, creator: sms, membership: 3, flags: S-M--, group quota: 0.
Name: system:nature-admin, id: -54766, owner: system:administrators, creator: sms, membership: 16, flags: S-M--, group quota: 0.
Name: system:esd10team1, id: -54767, owner: system:administrators, creator: sms, membership: 4, flags: S-M--, group quota: 0.
```

Figure 7: A segment of the output of an exhaustive iterative query of groups, using the lookup-by-id capabilities of `pts ex`.



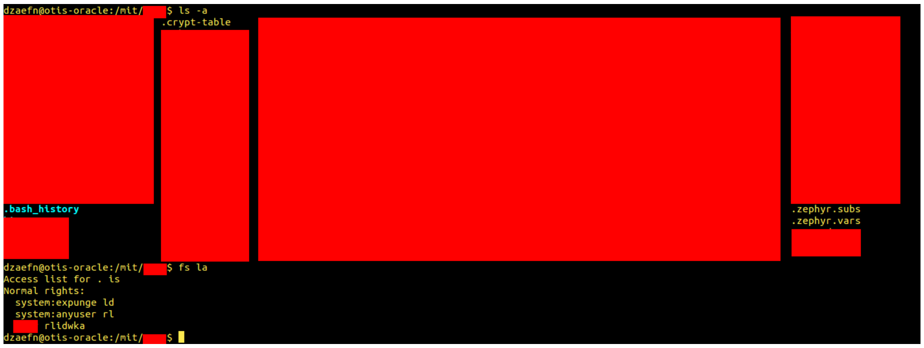
```
dzaefn@otis-oracle:~$ blanche 6.454 -i
List: 6.454
Description: none
Flags: active, public, and visible
6.454 is a maillist and is a group with GID 45347
Owner: LIST 6.454
Last modified by jasongao with blanche on 27-mar-2015 19:52:39
dzaefn@otis-oracle:~$ fs la /mit/6.454
Access list for /mit/6.454 is
Normal rights:
  system:6.454 rlidwka
  system:htaccess.mit rl
  system:expunge ld
  system:facdev rlidwka
  system:authuser rl
  forney rlidwka
  alex_o rlidwka
  mesrob rlidwka
dzaefn@otis-oracle:~$ fs la /mit/6.454/OldFiles
Access list for /mit/6.454/OldFiles is
Normal rights:
  system:6.454 rlidwka
  system:htaccess.mit rl
  system:expunge ld
  system:facdev rlidwka
  system:authuser rl
  forney rlidwka
  alex_o rlidwka
  mesrob rlidwka
dzaefn@otis-oracle:~$
```

Figure 8: A public self-owned list with rlidwka permissions to the Athena locker of a course 6 class, as well as its backup directory.



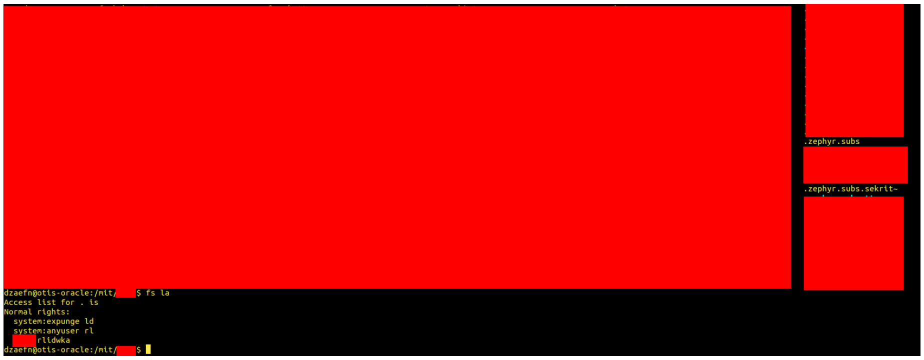
```
[redacted]              [redacted]              [redacted]
.bash_history                                   .zephyr.subs
[redacted]                                      .zephyr.subs.extra
                                                .zephyr.subs.tzc
                                                .zephyr.vars
                                                [redacted]
dzaefn@otis-oracle:/mit/[redacted]$ fs la
Access list for . is
Normal rights:
  system:anyuser rl
  [redacted] rlidwka
dzaefn@otis-oracle:/mit/[redacted]$
```

Figure 9: A user who has made sure to simlink out the .zephyr.subs file, but left their .bash_history viewable by anyone.



```
dzaefn@otis-oracle:/mit/[redacted]$ ls -a
[redacted]              .crypt-table
                        [redacted]      [redacted]      [redacted]
.bash_history                                           .zephyr.subs
[redacted]                                              .zephyr.vars
                                                        [redacted]
dzaefn@otis-oracle:/mit/[redacted]$ fs la
Access list for . is
Normal rights:
  system:expunge ld
  system:anyuser rl
  [redacted] rlidwka
dzaefn@otis-oracle:/mit/[redacted]$
```

Figure 10: A user who has made sure to simlink out the .bash_history file, but left their .crypt-table and their .zephyr.subs viewable by anyone.



```
[redacted]
[redacted]
.zephyr.subs
[redacted]
.zephyr.subs.sekrit~
[redacted]
dzaefn@otis-oracle:/mit/[redacted]$ fs la
Access list for . is
Normal rights:
  system:expunge ld
  system:anyuser rl
  [redacted] rlidwka
dzaefn@otis-oracle:/mit/[redacted]$
```

Figure 11: A user who clearly did not intend to leak a part of their zephyr subscriptions.