Decentralized User Authentication

CSC2231, Fall 2007

Amin Tootoonchian

Problem

- How do you share files with others? Web? Email? How easy is that?
- Is moving between machines easy?
- How do you collaborate with others?
  - How do you delegate permissions?
  - Creating an account for each one?
- How is authentication done in the Internet?

User Authentication has Limitations

- Centralized approaches (Kerberos)
  - Doesn't scale + can't work across admin domains + privacy
- Certificate-based (SPKI/SDSI)
  - Decentralized verification is challenging
  - Certificate management + revocation is hard
- Public key exposure (SSH)
  - Needs secure channel + key management

Desired Properties

- Support user/group access across admin. realms
  - Have a unique identity across all systems
  - No centralized authority
  - Make delegation easy
- Ease of use and transparency for user
  - Delays are not acceptable
    - Caching of remote principals

How is this system different than SFS?

- Can you give others access to your files using SFS? YES...
  - But they will get your privileges!
- Remote principal delegation made possible
  - Naming remote users/groups (self cert. paths)
- Making SFS ACL-aware
  - ACLs Defined by users
  - Group namespace: username.group

Limitations of this paper's scheme

- Must use SFS client to access files
  - Metadata is stored in the file itself
    - ACL is stored in the first 512 bytes of the file
    - Design choice
  - Easy to implement differently
- Delegation chain can be long
  - Long chains increase chance of malicious link
  - Can't restrict length of authentication chain

Revocation

- How to revoke users keys?
  - Using named ids helps
    - amin@cs
  - Using finger prints for ids makes it hard
    - m9ssn9i5s9j99wdwpe896dqkjn2ptp7a
- How to revoke server keys!
  - Hard

Important Questions

- How do public sites provide access to you?
  - Based on an id you can prove ownership of
    - Email Address
- What is your id in this paper?
  - How do others know you=what do they auth you against?
  - You public key/fingerprint
    - It is global...

Important Questions – cont'd

- Why is single/global identity useful?
  - You are A everywhere
    - Others should be able to verify you being A
  - Any problem?
    - Forging identities is easy
      - Creating thousands of key pairs/URLs
    - Revocation is hard for server keys
      - Is SRP useful here?
  - Do people like keypairs?
    - OpenID, a URL is your global identity

Conclusion

- 'Centralized access control' is dead
  - Does not scale
- 'Decentralized access control' rules
  - No authority
  - Doesn't need an infrastucture for key mgmt.
  - Multiple identities frustrate users :-(
  - Use global identity instead :D
    - Revocation becomes a harder problem

Personal Content Sharing

- We only care about friends
- Social Access Control
  - Attestations (RE)
    - Verification/Granting is done interactively
  - Maintaining public keys of friends
    - Non-interactive
      - Users should be able to extract data using their credentials
- Challenges for personal content sharing?
  - Access control is not personalized in OSNs...

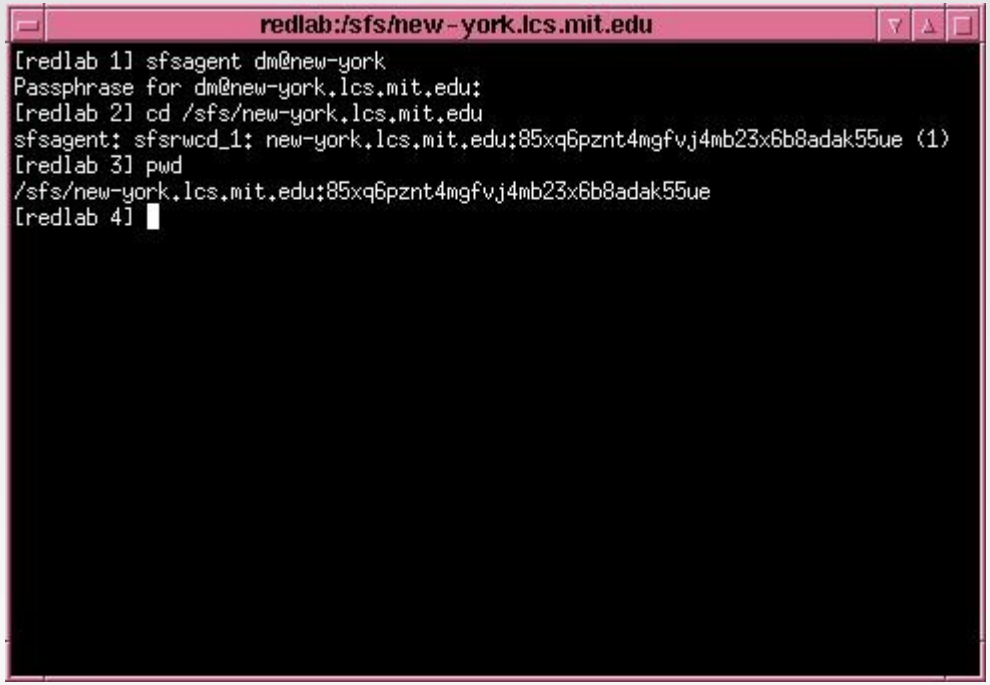
Fun: SFS is a killer app :-)

- Retrieving a self-certifying pathname with a password:

```
redlab:/sfs/new-york.lcs.mit.edu
[redlab 1] sfsagent dm@new-york
Passphrase for dm@new-york.lcs.mit.edu:
[redlab 2] cd /sfs/new-york.lcs.mit.edu
sfsagent: sfsrwcd_1: new-york.lcs.mit.edu:85xq6pznt4mgfvj4mb23x6b8adak55ue (1)
[redlab 3] pwd
/sfs/new-york.lcs.mit.edu:85xq6pznt4mgfvj4mb23x6b8adak55ue
[redlab 4]
```

Fun: SFS is a killer app :-)

- Even cooler when the xterm has reverse video!!!



References

- Decentralized User Authentication in a Global File System, Michael Kaminsky et al., SOSP 2003
- User Authentication and Remote Execution Across Administrative Domains, PhD Thesis of Michael Kaminsky @ MIT, 2004
- A Social Networking Based Acess Control Scheme for Personal Content, Kiran K. Gollu, Stefan Saroiu, Alec Wolman, SOSP07