



# Case Study: User authentication and management in WWW

Juha Ylitalo, Nokia IM  
([juha.o.ylitalo@ntc.nokia.com](mailto:juha.o.ylitalo@ntc.nokia.com))



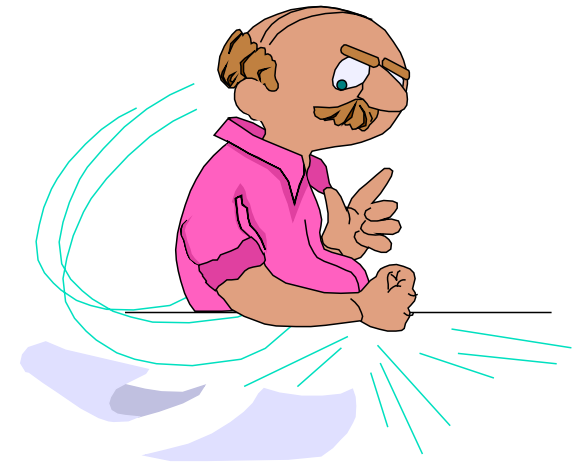
# Starting point...



- Based on Apache's basic authentication
- Psswd information was collected from NIS (with ypmatch)
- Problems:
  - All users were either trusted or untrusted and nothing between them
  - We started to have lot of users, who had NIS accounts only for WWW use  
they often forgot their passwords  
they didn't know how to change them
  - Lot of information had tighter access limitations than what they would have needed.
- In summary: we were at point, where either our web site wouldn't be able to expand its target group or we would have to find better way to handle user authentication and management issues.

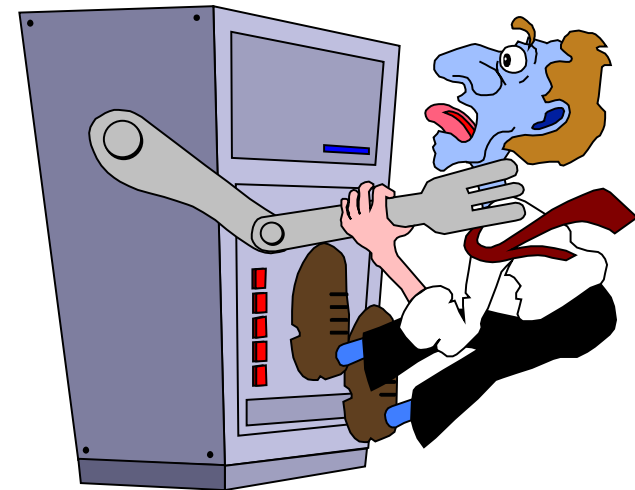
# Goal

- User management done by 3rd party
- No new passwords for users to remember
- Users have to be able to change their password
- Grouping has to be enough detailed, that we can do all authentication based on groups instead of individuals
- Comparable to any commercial solution in market
- Three sub-targets were found
  - user identification and validation
  - grouping
  - security issues
- *Current situation is that we have:*
  - *152 groups*
  - *2594 users in those groups*



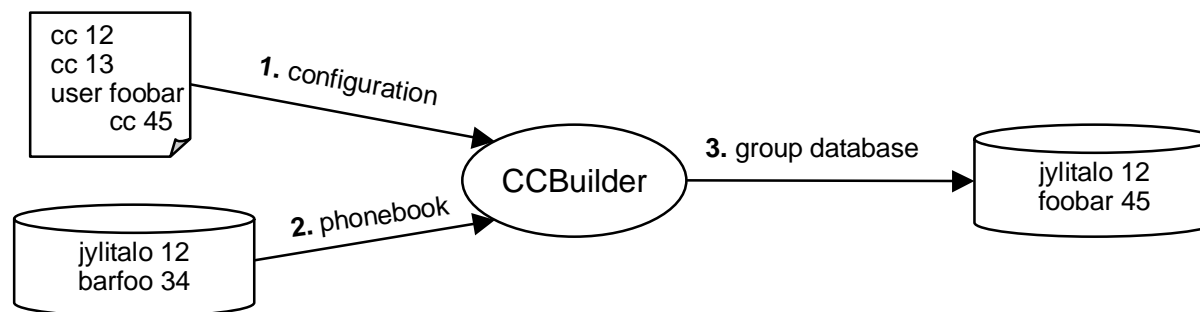
# User authentication

- NIS didn't work, because we don't have company wide NIS system
- Lotus Notes was studied, but its impossible to use Notes normal passwords and http passwords were not widely used (and we didn't like the concept...).
- Windows seemed to be last obvious choice.
  - There was company wide user database
  - Publicly available C API existed
  - First versions were based on smblib
  - Later we replaced smblib with pam\_smb to get
    - encryption to network traffic
    - use of domain names
    - better maintenance (latest version of smblib dates back to September '96)



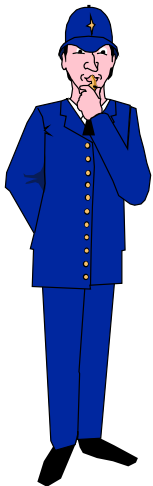
# Grouping

- Grouping in our company is important question, because company encourages to work rotation in 2-5 year intervals and one small R&D department doesn't have resource to keep track of people's transfers, so we had to find some way to put people into groups and only use those groups in authentication.
- Lotus Notes was once again studied, because it has good addressbook. However we didn't find acceptable API for our purposes.
- Windows was also studied, but all the groups in our domains seemed to be too randomly formed.
- At the end, solution was found from electronic copy of company phonebook, which has quite bit information including people's usernames and cost centers.
- While phonebook was good base, we had to create our own system, because there are always exceptions to rules, I.e. consultants



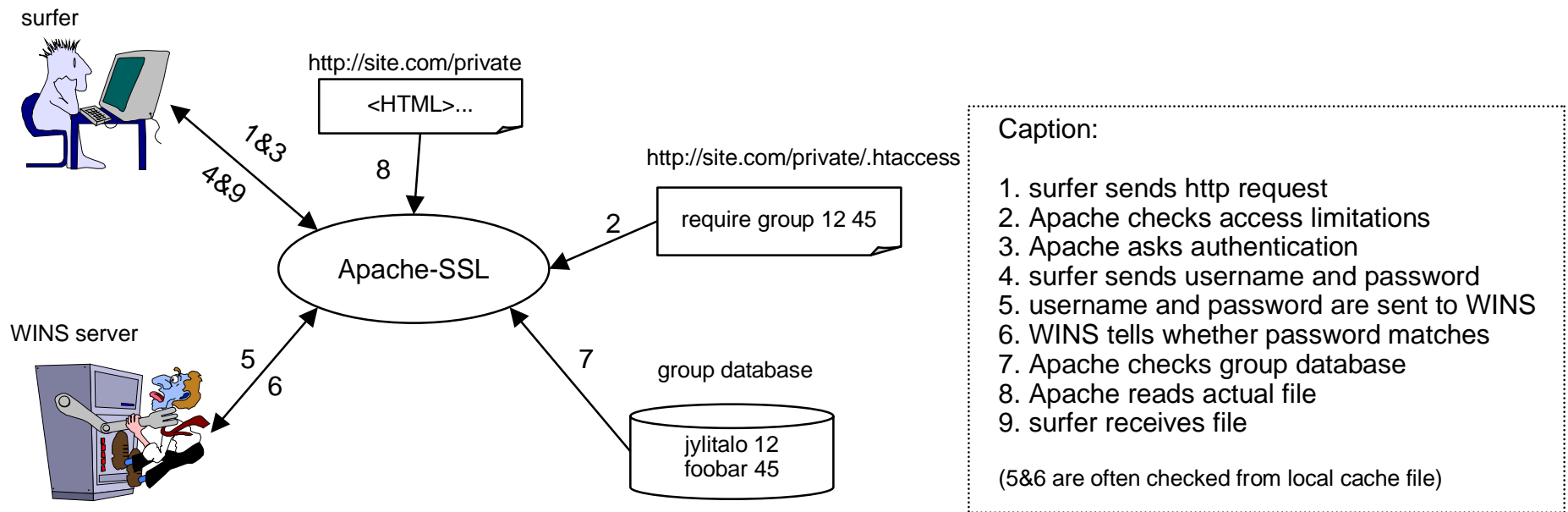
# Security

- Unless you use one time passwords, you need some sort of protection to network traffic, because:
  - HTTP/1.1 uses BASE64 encoding to encrypt user authentication information in its requests. :)
  - Your browser has to authenticate you to web server everytime, when you access page, which requires authentication.
- SSL is pretty obvious choice for encryption, since browsers support it, but should you take:
  - 40-bit international version, which is not banned by US Export Laws
  - 128-bit "domestic" version
    - can't be exported from US without license
    - has OpenSource implementation from it, which is free from export laws (SSLey)
  - Problem is that average IE and Netscape only support international version. Netscape can be "fortified" from international to domestic version.



# Implementation (1/2)

- Apache-SSL was selected as our new web server software, because
  - we had used Apache in past
  - its SSL implementation was free from US Export Laws limitations
  - its module structure allowed us to build our own authentication module
  - we were able to combine multiple authentication modules.
  - open source provided lot of examples on how to code module



## Implementation (2/2)

- We implemented small application that collects grouping information and insert it into ndbm database, which our authentication module uses as a source of information for group information. This was left out from actual module, because its NTC specific, its performance is significantly better this way than if it would always contact phonebook, etc.
- As experiment, we created our own CA and created SSL certificates for our websites, SMTP and NNTP services.
  - In future we are moving to 3rd party certificates, because current plan causes unnecessary dialog windows to users, when they enter our site for the first time (and in future events as well, unless they select accept until expires from menus).

# Indexing site that has protected web pages

- Problems:
  - Our site has lot of dynamically created pages, so we couldn't use filesystem based indexing.
  - We didn't want to hard code any username-password pairs as plain text into configuration files for search engine
  - We don't have any way to create dummy user accounts into NT domains
- Solution:
  - We changed our system so that `mod_auth` is primary authentication module and `mod_auth_samba` is only secondary authentication module
  - At the beginning of each indexing, we use random and file access times to generate dummy user account, which belongs to group that has access to everything, in to `mod_auth` passwd and group files.
- **NOTE:** If you place authentication modules in wrong order, NT administrators will probably send inquires to your direction.



# What's missing ? (aka Future plans)

- Access policies are usually done by managers and system administrators only enforce it, so there is need for tool that
  - creates graphical sitemap
  - finds out access rules for all pages and visualize it in the sitemap
- To improve mod\_auth\_samba so that it would send error code if someone using http instead of https tries to authenticate
  - work-around has been that we place redirects into configuration files and that way make sure that all connections go to https
- Authentication tokens seems to be coming more and more popular, so it would be nice to have authentication modules for those as well.



# References

- Apache WWW server, <http://www.apache.org>
- Apache-SSL, <http://www.apache-ssl.org>
- Fortify, <http://www.fortify.net/>
- mod\_auth\_samba, [http://www.iki.fi/~jylitalo/apache/mod\\_auth\\_samba/](http://www.iki.fi/~jylitalo/apache/mod_auth_samba/)
- mod\_ssl for Apache, [http://www.engelschall.com/sw/mod\\_ssl/](http://www.engelschall.com/sw/mod_ssl/)
- smblib, <ftp://samba.anu.edu.au/pub/samba/smblib/>
- pam\_smb, [http://www.csn.ul.ie/~airlied/pam\\_smb/](http://www.csn.ul.ie/~airlied/pam_smb/)

## Questions ?