SNAPWebD: a web desktop for Standalone desktop or Nfs server using APache (SNAP) web server

S S Krishna Vutukuri(11MCMT35)

Advisers

Anupama P

Dhurga Bhavani S

Sobha Rani T

School of Computer and Information Sciences
University of Hyderabad
July 8, 2013

Outline

Abstract

Web Desktop is a free and open source software, will allow users to access their home directories and remote applications through the web browser.

SNAPWebD is a web desktop for NFS mounted logical volumes of home directories of users, where the users are authenticated using Light Weight Directory Access Protocol(LDAP).

The advantage of this tool is:

- It requires no additional software on client side.
- No need to open any ports across the firewall, so the chances of getting attacked are mitigate.

Motivation

There are several products existing in the market. But They have some lacuna's.

Exixting Products Problems

- Limited to access the special accounts created in a server.
- Proprietary and commercial.
- Require additional software on client systems.
- Uses the protocols such as RDP and ICA to achieve remote access, so that additional ports need to be opened.

Our product is designed to addresses these lacunae in remote desktop products.

My Contribution

We have proposed a web desktop, **SNAPWebD**, that overcomes most of these limitations.
To build the **SNAPWebD** product, we used

Lightweight Directory Access Protocol (LDAP): It is to provide centralized authentication for users to login from anywhere in the network.

Network File System (NFS) Protocol: It is to access the users' home accounts(files and directories) from remote system.

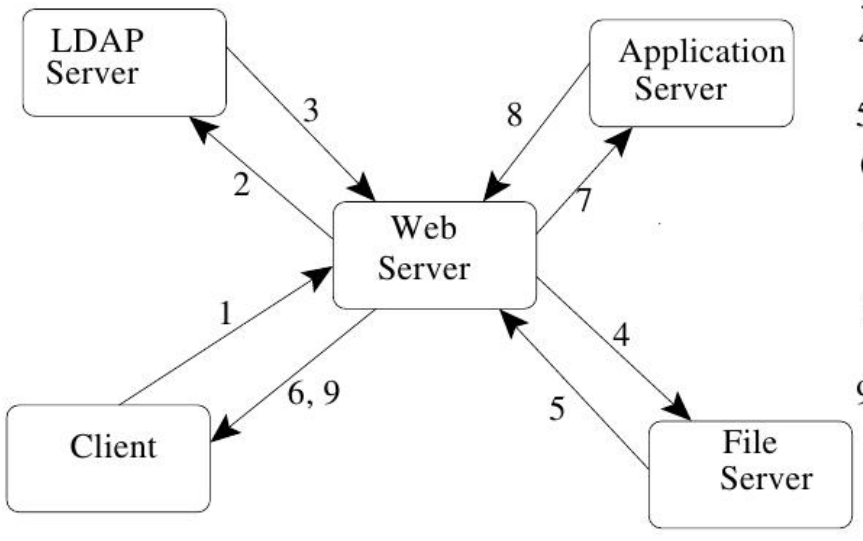
Secure Shell (SSH) Protocol: It is to access the applications from remote system.

Apache Tomcat: It is the place where **SNAPWebD** software has to be installed.

The **SNAPWebD** architecture consists of the following components

1. Lightweight Directory Access Protocol (LDAP) Server
2. Network File System (NFS) Server(User home accounts)
3. Web Server
4. Client
5. Application Server

Proposed Architecture



1. User logs in
2. User details sent to LDAP server
3. LDAP server sends validity of user
4. Web server sends request for mount to NFS server.
5. Web server mounts the user's home
6. Web server sends home directory details to the client.
7. Application request from client is forwarded to the application server.
8. Web server mounts the appln. server directory to launch the application.
9. Web server launches the application

Figure: **SNAPWebD** Architecture

Main Tasks

1. **Access to files and directories**
2. **Access to applications**
3. **Search**
4. **Access to the specified remote computer**

Access to files and directories I

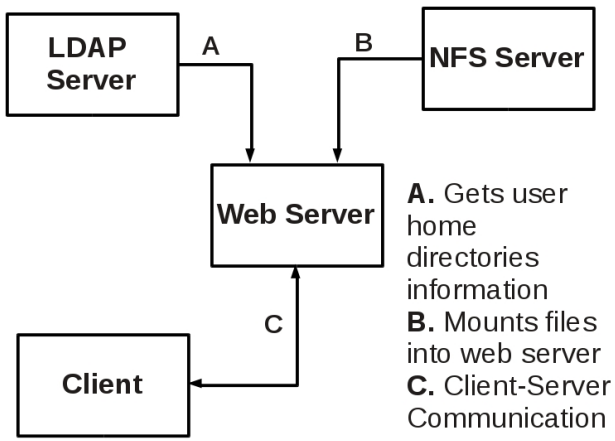
We are mounting the user home directories from his remote system (NFS server) into the Web Server, so that user can access his files and directories from a web browser.

- Exporting the home directories

  User-home-directories-path **Web-Server-IP(rw)**
- Mounting the home directories

  Web-Server-Path **NFS-Server-IP:User-home-directories-path**
- LDAP server gives the user home directory information by the command,

  ldapsearch



Access to files and directories II

File Exploration

- We are displaying all the files and directories as a hyper-links on browser by using File object.
- When ever user click on a directory, based on path of the directory we list its files and directories on browser where user feels like it is a file explorer

Upload and Download Features

- As the files and directories are in web-server user can able to download the files by right click options.
- When user upload files, they will be upload in mount point of user home directories of web server, so that it also available in remote server(NFS Server).

Access to Applications I

How We Done

To access the remote applications we used SSH protocol.

- Run application remotely by using the *ssh* command.

  sshpass -p "password" ssh user-name@NFS Server-IP "application-command"

- Before the application starts, the application display (xserver display) should be export to the client system by using the DISPLAY environment variable as

  DISPLAY=Client-IP:0

Finally, to run a application remotely,

sshpass -p "password" ssh -o StrictHostKeyChecking=no user-name@NFS Server-IP

"DISPLAY=Client-IP:0;application-command"

Access to Applications II

A click on a file will open the file with associated application, based on the extension of file.
An example command for *ex.pdf* file is

sshpass -p "krishna.48" ssh -o StrictHostKeyChecking=no Krishna@10.5.0.95 "DISPLAY=10.5.1.0:0;okular ex.pdf"

Access to Applications III

Terminal

We implemented 2 kinds of terminals

Complete Terminal(XTerm)

XTerm is also an application which runs with the command "xterm".

Lightweight Terminal

- Because the software is written in Java Server Pages (JSP), it is a lightweight process.
- When the user gives a command, then only it connects to the Remote NFS Server, execute and gets the result to client; otherwise, it will be idle.
- We cannot execute interactive commands.

Search I

File or Directory Search:

Name based Search

Input: Home_Directory_Name, Searching File Name
Output: Files or Directories with Matched Name
List = new ArrayList();
`Search(`*Home_Directory_Name, Searching_File_Name*`);`
File := Directory.listfiles();
foreach *File* **do**
    if *(File is Directory)* **then**
        `Search(`*Directory_Name, Searching_File_Name*`);`
    else
        if *(File.name == SearchingFileName)* **then**
            List.add(*File*);
    end
end
Print the *List* with neat GUI;

Search II

Content Search

Input: Home_Directory_Name, Searching_Content
Output: Files or Directories with Matched Name
List = new ArrayList();
`Search(`*Directory_Name, Searching_Content*`);`
File := Directory.listfiles();
foreach *File* **do**
    if *(File is Directory)* **then**
        `Search(`*Directory_Name, Searching_Content*`);`
    else
        result := execute "grep -ic *SearchContent File*";
        if *(result* != 0*)* **then**
            List.add(*File*);
    end
end
Print the *List* with neat GUI;

Search III

Application Search

We can search particular application in the NFS Server(users' home accounts system) by executing the related command in remote system using SSH protocol.

If application exist, it opens otherwise remote system returns an error to web-server and web-server sends same error to client.

Access to the Specified Remote Computer

- A user who is not registered in LDAP Server is also able to connect his personal computer by using this SNAPWebD. user personal computer should able to export the home directories with NFS protocol
- The Remote system must run the NFS Server to export the home directories.
- We achieve this by SSH protocol as,

  sshpass -p "password" ssh -o StrictHostKeyChecking=no user-name@Remote-computer-IP

  - valid user → returns null.
  - Invalid user → returns an error.

Features of SNAPWebD

- **User with His Personal Desktop**
- **File Exploration**
- **Upload and Download the Files**
- **Search**
  - **Application search**
  - **File search**
- **Opening the Terminal**
  - **Lightweight Terminal**
  - **Complete Terminal(Xterm)**
- **History**
- **Access the Specified Remote Computer.**

DEMO

Summary of Features

A table comparing the features supported by various commercial and free software web desktop products and **SNAPWebD** is given below.

Features	**Products**			
	VNC	**eyeOS**	**OVD**	**SNAPWebD**
Display files and directories in a browser	NO	YES	YES	**YES**
Can a user get his personal desktop	YES	NO	NO	**YES**
Opening a file when the associated application does not exist in the remote system hosting the file	NO	NO	NO	**YES**
Ability to upload a file to the remote host from local system	NO	YES	YES	**YES**
Ability to download a file to the local host from the remote system	NO	YES	YES	**YES**
Lightweight terminal (execute all types of non-interactive commands)	NO	NO	NO	**YES**
Complete terminal (execute all types of commands)	YES	NO	NO	**YES**
Search for an application in user's remote system	YES	NO	NO	**YES**
Search for a file based on name	NO	NO	YES	**YES**
Search for a file based on content	NO	NO	NO	**YES**
Store the history of users' actions since login time	NO	NO	NO	**YES**
Accessing a specified standalone system (using its IP address)	NO	NO	NO	**YES**

Conclusions

- We have developed a web desktop, SNAPWebD, that allows users to access their NFS logical volumes on a browser with no need of any special software installation at client side.
- SNAPWebD is designed to search various files and applications in NFS servers.
- The SNAPWebD can be used to also work with command line.
- The users can also download and upload files to and from the remote file servers.
- SNAPWebD have the ability to access any standalone remote system given its name or IP address in the login page.

Future Work

- Currently the **SNAPWebD** is supported by linux systems only. In future it can be extended to support all operating systems.
- The **SNAPWebD** can be extended to connect to Virtual Machines(VMs) and allow the users to access their VMs using the web browser.
- Security issues related to this product need to be explored and fixed.
- The feature to upload and download the files will be enhanced to a drag-and-drop operation.

References

EyeOS.
http://www.eyeos.com/.

Lively Kernel.
http://www.lively-kernel.org/.

Network File System(NFS).
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.473,
http://tools.ietf.org/html/rfc5661.

Open LDAP.
http://www.openldap.org/.

Dhaval Manvar, Mayank Mishra, Anirudha Sahoo.
Low Cost Computing Using Virtualization For Remote Desktop.
In *Fourth International Conference on Communication Systems and Networks, COMSNETS 2012, Bangalore, India.*, 2012.

Jan Miettinen, Tommi Mikkonen and Antero Taivalsaari.
The Lively Dock: Web Applications as Live Thumbnails.
In *International Symposium on Web Systems Evolution, WSE-2010, Timisoara, Romania*, 2010.

THANK YOU