Windows update management using Samba (among other things)

SambaXP 2012 May 9th

Matthieu Patou Samba Team

mat@samba.org

Agenda

- Disclaimer
- Why do we need an update management system
- Existing alternatives
- Yet another SUS, why ?
- Introducing Shuss
- How it works: for the client
- How it works: for the server
- Demo
- Issues
- Future

Disclaimer

The views and opinions in this presentation are my own
and do not necessarily reflect the views and opinions of my present,
past and future employers.

Why do we need an update management system ?

Because :

- applying all the updates proposed by a vendor is not always desirable
- going on each computer to select updates doesn't scale at all
- because not applying security updates is not an option

Existing alternatives

- WSUS: Windows Server Update Services
- LSUS: Linux Server Update Services (defunct) from samba-edu
- WSUSoffline: seems more offline oriented

(Yet) another SUS, why ?

- Was running a full IT infrastructure on top of opensource software.
  So why introduce non free software for just 1 thing ?
- WSUSoffline didn't exists at that time
- Didn't like the idea of LSUS trying to interprete Wsusscn2.cab content
  Microsoft didn't provide a documentation on this file and has already
  once changed the organisation of the archive
- There is an API to control update service on workstation
- Ideal compagnon of Samba 4 DCs

Introducing shuss 1/2

- Used to be called yasus or L4SUS but domains wheren't available
- Why shuss ...?
- well grep -E "s.u.s" /usr/share/myspell/dicts/ didn't yield interesting enougth words
- but shush looks promissing, but wasn't available or I was too tired ...
- Available at http://shuss.org/, contributors <u>very</u> welcome

Introducing shuss 2/2

- Fairly simple (I hope)
- Update are served out of a samba share
  tested only with samba4 DC (and ntvfs fileserver)
- Should work though with samba 3.x or s3fs if server encrypt is used and vfs_xattr
- Signed packets provide a proof that updates and update list hasn't been tampered by a man in the middle
- Each computer has its own folder with ACLs granting write only to this computer and the domain admins

How it works: for the client 1/2

- At scheduled period getupdates.vbs runs
- The script create an object manager of class *Microsoft.Update.ServiceManager*
- From the manager we create a searcher

```
Set updateSearcher = updateSession.CreateupdateSearcher()
```

- And then we do the search, default search criteria is:
  "IsInstalled=0 and Type='Software'"
- Once the search is complete, iterate on the results
- And write a file in the "computer" folder on the update server with entries like that:

```
Update: Security Update for Windows XP (KB956803)
        Update ID:      33a7edf1-2350-4102-8082-9540eff65704
        Name:           Binary en
        Url:            http://download.windowsupdate.com/msdownload/update/
                        software/secu/2008/09/windowsxp-kb956803-x86-enu_d075d359a2
                        8ab8b058a35a2e7b466bd0bca8e9ef.exe
```

How it works: for the client 2/2

- At scheduled period doupdates.vbs runs
- It checks for the existance of a file called *updatelist* with entries like this:

```
microsoft\33a7edf1-2350-4102-8082-9540eff65704.exe 1
```

- Entries indicate the relative path from the updates share on the update server for the file
  and a file tag, which basically indicate which flags should be used while running the update
- After each update, a line indicating return code of the update program is added for debug

How it works: for the client 2/2

- At scheduled period doupdates.vbs runs
- It checks for the existance of a file called *updatelist* with entries like this:

```
microsoft\33a7edf1-2350-4102-8082-9540eff65704.exe 1
```

- Entries indicate the relative path from the updates share on the update server for the file
  and a file tag, which basically indicate which flags should be used while running the update
- After each update, a line indicating return code of the update program is added for debug

How it works: for the server 1/2

- *updatewatcher.pl* is started as a background task with the update folder as first parameter
- For each folder a separate watcher is created
- If in the parent folder a new directory is create a new watcher is added
  It allows new workstation detection without restarting the watcher
- In other folders if a event concerns a file containing the string *proposedupdate.log*
  then the analyzer script is started with the "discovered" file

How it works: for the server 2/2

- This script will ... well "analyze" the list of updates the clients wants and do the following things:
  - If the update is new, add a notify entry
  - If the update is not new, check if last notify date is > 24H if so add a new notify entry

    - If the update is validated, add it to the list of update to download
    - Send an email if there notify entries
    - Start the download of updates
- *download_winupdate* will:
    - obviously download updates if the file didn't exists
    - add entry in the update list with the guessed tag
- Update are validated with validate_update script

Demo

Demo

Issues

- Shuss works great with Windows XP/2003
- But starting from Vista installation of update isn't working
  because most of the updates are .cab files and I don't know how to install on Vista
- But in Windows 7 there is pkgmgr or DISM and shuss is able to use it
- The update tag (which control the switch of command line) is guessed from the output of *file*
  results are descent but not perfect.

Future

- Update API has a way to copy a file to WUA cache
- The idea is to use this to populate the cache of WUA and then use the API to trigger the installation of selected updates
- Obviously emails reports for updates is not sufficient and so is the validation of updates only through command line
- This calls for a web interfaces to be able to lists which updates are needed,
  which host hasn't applied an update since xx days,
  to validate updates,
  ... <*add here your own whish*>

Questions

Questions ?

Thank you

Thanks for listenning

The discussion continue at blog.shuss.org