

## Centralized Tool for Software and USB

Prof. Bhanu Tekwani<sup>1</sup>, Shubhangi Mishra<sup>2</sup>, Mythri Ravindra<sup>2</sup>

<sup>1</sup>Professor, <sup>2</sup>Student

<sup>1,2</sup>Vidyalankar Institute of Technology, Mumbai, India

**How to cite this paper:** Prof. Bhanu Tekwani | Shubhangi Mishra | Mythri Ravindra "Centralized Tool for Software and USB" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-3, April 2019, pp.1197-1200, URL: <https://www.ijtsrd.com/papers/ijtsrd23187.pdf>



### ABSTRACT

We propose a standalone application which facilitates a user to send request for USB access to the server, and after an approval from the administrator the user can use the USB. Along with that, the application also generates report for the administrator regarding hardware configuration and list of software in a computer, in a given network. Another additional feature includes, automated and silent software installation in the client computers over a network.

**KEYWORDS:** Centralized, Remote, Automated, Customized, USB, blocking, Software, Installation.

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



### I. INTRODUCTION

This paper proposes an application for centralized management of USB access. Currently, the institute needs a tool that can help with centralized USB blocking, and if someone requests permission for USB access then it would be granted by the server room after authenticating the requester. This would ensure security because only authorized users are allowed to access the USB port. Also, this tool would facilitate remote software installation in the host computers. If any software is requested from a host computer, it would be automatically and silently installed from the client computer to all the computers in the network. This automated, remote installation for an institute would save a lot of time, energy and cost. This tool can also generate a report of all the hardware and software configurations of an individual computer when requested from the server. The proposed project aims to focus at developing a centralized network by automated software installation from a peer to peer machines residing in the network, silently. This tool will also generate a report of all the host machines; regarding their hardware and software list. Thus, this tool helps at integrating many networking features and managing devices from a remote center with cost and time efficiency.

### LITERATURE SURVEY

[1] In this paper they present the implementation of access and identity management for endpoint protection and data

security from USB devices to maintain information security and data theft prevention in a corporate environment. The vision of this project is to track record and limits the use of USB devices in a secured environment (network) thus maintains confidentiality and integrity to meet information security standards. We are proposing to keep a centralized repository of allowed devices such as USB key board, mouse, and printer etc. based on organization's security standards.[1]

[2] In this paper, we proposed a model that can remotely perform various software maintenance related actions. These actions can be software release upgrade, run-time patching and status monitoring etc. All these actions can be performed in four different node configuration-standalone, active-passive, combo and parallel. Proposed model is a very simple and scalable in nature which makes it extremely suitable choice for software maintenance in telecommunication network.[2]

[3] Ansible is open source software that automates software provisioning, configuration management, and application deployment. It is designed for multi-tier deployments since day one, Ansible models your IT infrastructure by describing how all of your systems inter-relate, rather than just managing one system at a time.[3]

**II. EXISTING SYSTEM**

Currently, the institute has blocked all the USB ports in all the computer for the sake of security. If some legitimate user wants to access the USB devices then the system administrator is called upon. Similarly, if we want to install any software on any computer the administrator has to download and install it on every computer personally and individually. The administrator has no way of knowing the hardware and software configurations of any computer in the network unless he/she goes up to the computer.

**III. LIMITATION OF EXISTING SYSTEM**

In the current system, a lot of time and effort is wasted on installing software on multiple computers. The USB access is blocked permanently whereas its access can be managed efficiently. The logging of reports of individual computers' configuration is done nowhere.

**IV. PROPOSED SYSTEM**

The proposed system is to create a tool for the college which will give the following features in the following way:

1. Centralized USB blocking and unblocking.  
The default setting for all the systems would be to block the USB device plugged in. If any genuine user wants the access, then the person can send a request to the server room via a user interface with authentication using username and password and the request will be responded with accepting or rejecting the request for the access.
2. Remote software installation.  
Through a portal, any teacher can request for a software installation in a particular lab and after the authentication of the requester, the software will be installed in that lab automatically.
3. Report of all software installed.  
Whenever the administrator wants to know about the software and hardware installed in any one of the host machines, a report will be generated of that terminal and sent to the server.



Fig 6.2 Work flow diagram for Software installation

**V. METHODOLOGY**

**Server side:**

Server side application has a login page and after successful login in database connection, a dashboard frame is opened. The dashboard contains two tabs; one to keep track of USB requests and other gives the option to generate reports of the client computers in the network.

**The USB tab:**

This tab contains a table which extracts the requests entered in the database from the client side. Each row displays the detail of the request such as the IP address, hostname, MAC address, and two buttons labelled "Enable" and "Disable".

The administrator can allow or deny the request. The response is updated in the database and fetched from the client side application. As soon as any response is provided for a row, the row is removed from the table of the tab. The database will have the log of USB requests received by the server.

**The Report tab:**

This tab contains a table with details of the computers present in the network: IP address, hostname and two buttons available: one for software report request and another for hardware configuration report request to the client machine.

The communication between the server and the client is done using TCP/IP connection. The client is listening continuously on the port 5000. As soon as the request is generated, it connects to the client port and file transfer takes place from the client machine to the server machine.

These files are stored in the "reports" folder with the hostname and timestamp as the name of the file and can be accessed at any point of time. This file is read and the content is displayed on the text frame present on the UI.

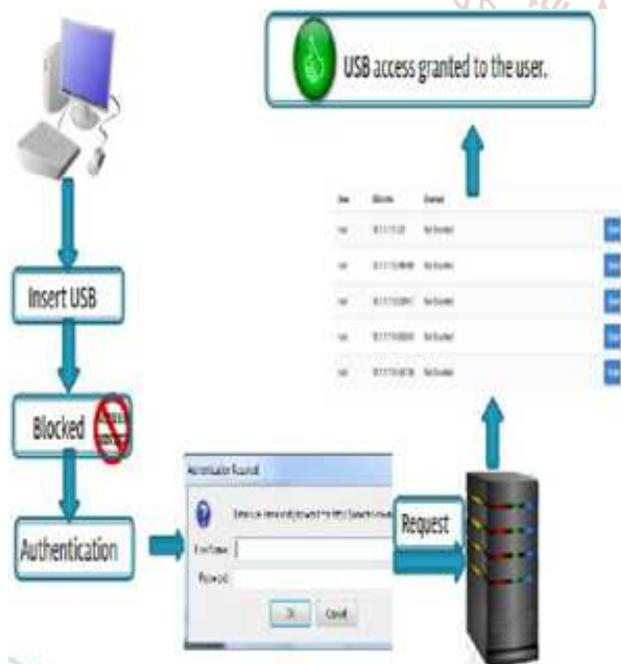


Fig 6.1 Work flow diagram for USB module

**Client side:**

Server side application has a login page and after successful login in database connection, a dashboard frame is opened. The dashboard contains two tabs; one to keep track of USB requests and other gives the list of software in the database.

**The USB tab:**

The user can send request for USB access to the server. Only after the access permission from server, the USB will be enabled unless it will be disabled. Even after the USB access given, the USB will be disabled after a timeout of few minutes say, 20 minutes, predefined by the administrator.

**Software tab:**

There will be a list of software from the database. After clicking on a software, the software will be installed in the all the computers in a given range of ip address in the network. This will save time and effort of installing software in a multiple computers in a single click.

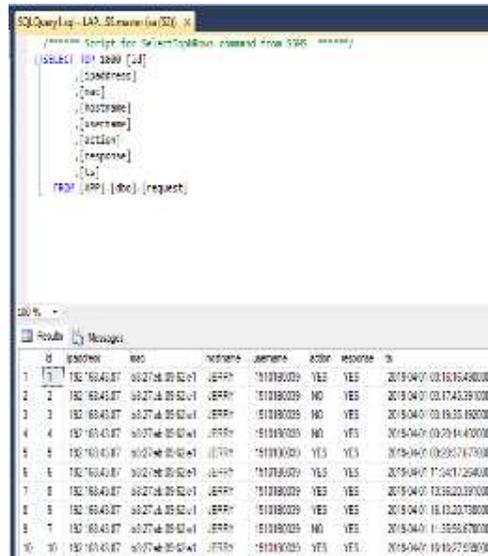


Fig 4.1.3 Database entry for requests

**VI. IMPLEMENTATION SCREENSHOTS:**

4.1: Client side :

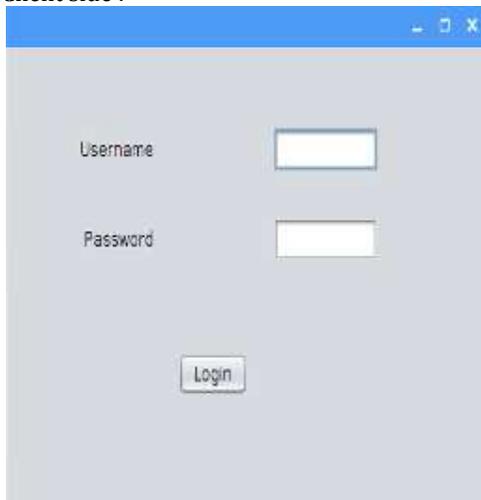


Fig 4.1.1 Login page

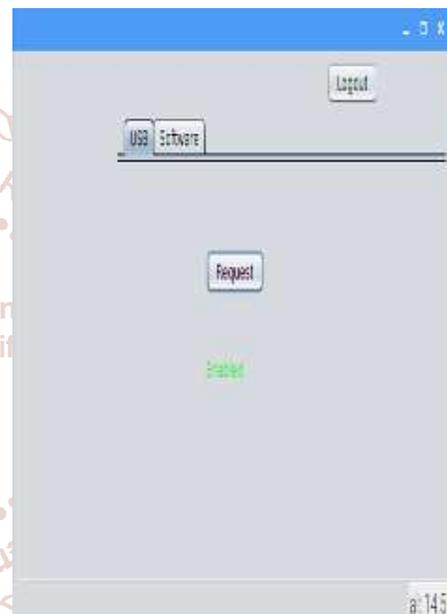


Fig 4.1.4 Request response for access

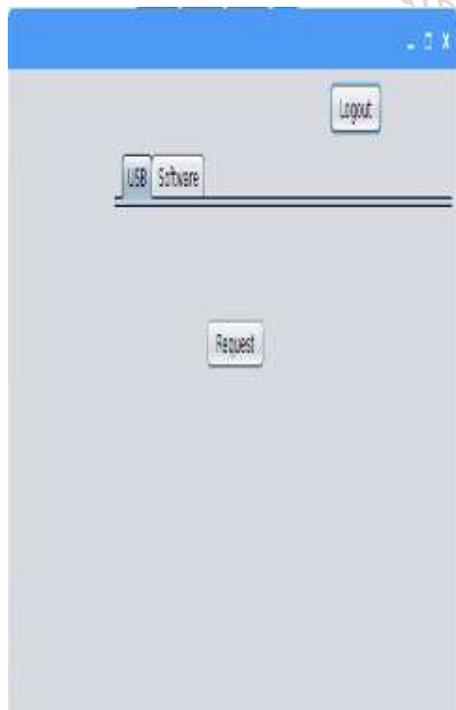


Fig 4.1.2 Request for USB access

4.1.2. Server side:



Fig 4.2.1 Login page

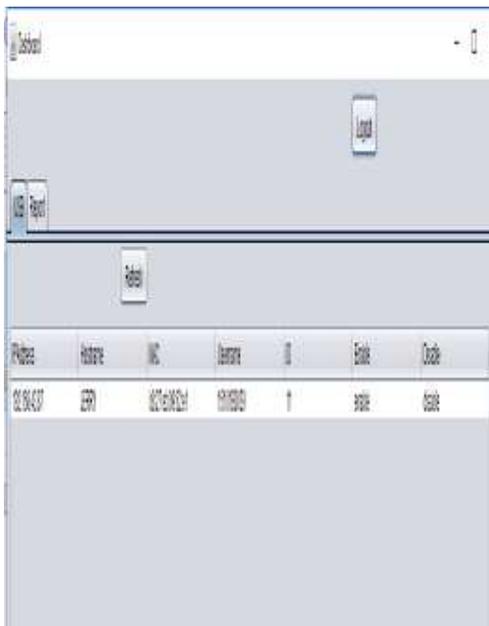


Fig 4.2.2 List of requests

Fig 4.2.3 Reports generated saved in a folder with timestamp.

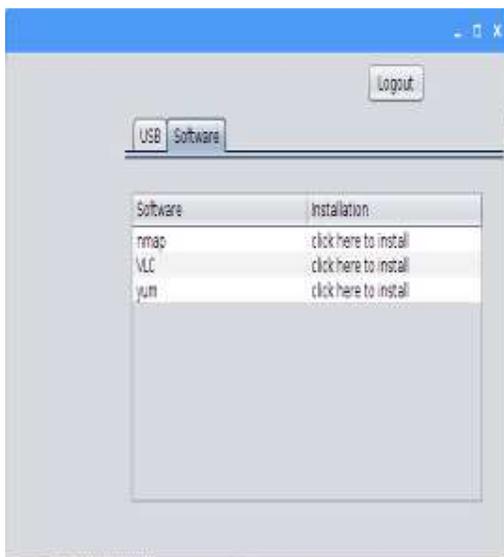
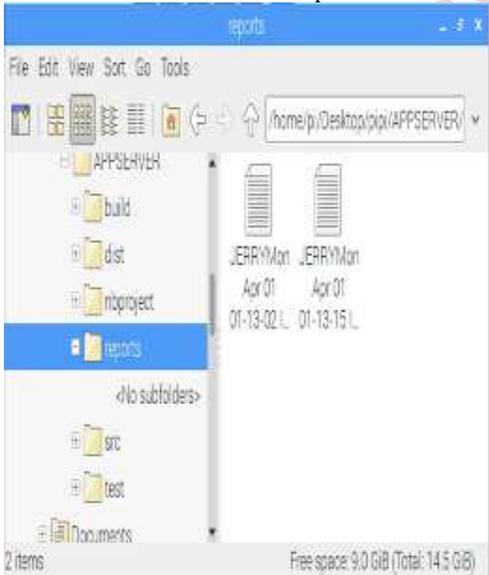


Fig 4.2.4. List of software

## VII. ADVANTAGES OF PROPOSED SYSTEM

1. Centralized system.
2. Automated system.
3. Saves time and effort.
4. Cost of the institute is saved.
5. No manual labor is required.
6. The system is more organized and efficient.
7. The system has become more organized.

## VIII. CONCLUSION

Data protection and data maintenance is one of the biggest administrative concern of the institute. This application aims at legitimate access and usage of the USB devices. This tool is centralized at its core, hence, the access permission can only be granted by the server administrator after authenticating the user. With the help of this project, we focus on reducing the time, cost and effort that goes into the logging of information of the individual computers in the network. The administrator can know about the hardware configuration as well as the software installed in each computer. Plus, the files of those reports can also be stored on the server side. The effort of installing a single software on to multiple computers is reduced down to one click on a computer and that software will be installed in all the computers in the given network.

## REFERENCES

- [1] Saurabh Verma, Abhishek Singh. Data theft prevention & endpoint protection from Unauthorized USB devices. IEEE- Fourth International Conference on Advanced Computing, ICoAC 2012 MIT, Anna University, Chennai. December 13-15, 2012.
- [2] Promila Jangra, Soma Das, Sandeep Kumar Khurana. Remote software maintenance system for telecom network. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).
- [3] Using Ansible for network automation. Cell Stream, Inc. [www.ansible.com](http://www.ansible.com)
- [4] <https://emcosoftware.com/remote-installer>
- [5] <https://www.remoteutilities.com/support/docs/remote-install-tool/>
- [6] <https://4sysops.com/archives/free-usb-blocker-centralized-access-control-for-usb-devices/>
- [7] <https://www.usb-lock-rp.com>