

NETDOG - CONFIGURATION MANAGEMENT AND MONITORING SYSTEM



Mini Project 2018

Done By

ASWIN BABU K

Guided By

Prof. Baby Sylva

Assistant Professor

Department of Computer Applications
College of Engineering
Trivandrum-695016

ABSTRACT

The NetDog Project started with the aim to develop a software system that can bring up and down computers on the network. It has now added a number of abilities to its feature list, including the ability to execute scripts or commands across machines on the network and the ability to copy files to machines on the network without relying on third party protocol. The system is developed to be fully extensible so that developers can extend the system as per their requirements through plugins.

ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgment, then let words play the heralding role of expressing our gratitude.

First of all I would like to thank God almighty for bestowing us with wisdom, courage and perseverance which had helped us to complete this project ***NETDOG - CONFIGURATION MANAGEMENT AND MONITORING SYSTEM***. This project has been a reality as a result of the help given by a large number of personalities.

I would like to remember with gratitude **Prof. Jose T Joseph**, Head Of Department Department of Computer Applications, College of Engineering, Trivandrum for the encouragement and guidance rendered.

I express our sincere thanks to **Prof. Baby Syla**, Assistant Professor, Department of Computer Applications, College of Engineering Trivandrum for his valuable guidance, support and advices that aided in the successful completion of my project.

Finally, We wish to express our sincere gratitude to all our friends, who directly or indirectly contributed in this venture.

ASWIN BABU K

Contents

List of Figures	iv
1 Introduction	1
2 Requirement Analysis	2
2.1 Purpose	2
2.2 Overall Description	2
2.2.1 Product Functions	3
2.2.2 Hardware Requirements	3
2.2.3 Software Requirements	3
2.3 Functional requirements	3

List of Figures

Chapter 1

Introduction

The netdog project started with the aim of designing a program which can easily bring computers up and down remotely. The idea came from the realization that, quite a few computers were left powered up when the college lab closes for the day. Thus the initial name "Project Green".

A number of features were added to the feature list, most importantly tracking computers even if their IP addresses changed, the ability to execute commands or scripts on a specified range of machines with a single command, and the ability to copy files to a range of machines without requiring protocols such as SFTP or FTP.

The abilities does not end there and netdog can provides many notable features such as early warning of HDD failure on the machines on the network. The exhaustive list of features is listed in later section. Once completed, netdog will be a completely extensible system to which features an be easily added through plugins.

Chapter 2

Requirement Analysis

2.1 Purpose

The purpose of this system is to build a configuration management and monitoring system that is dead simple to use. Highly efficient and feature rich configuration management systems such as Ansible exists. But the problem is that they have very steep learning curve. This is also the case with monitoring systems such as Nagios which require considerable amount of setup from the part of admins.

The learning curve makes these systems unsuitable to small office environments, schools or colleges where the systems are managed by regular employees. NetDog is dead simple, with zero learning curve. As long as a person is aware of the command he/she wishes to execute on the remote machine, the person is ready to use NetDog.

2.2 Overall Description

NetDog has a client server architecture. The server is responsible for issuing commands to the clients and is to be used by the administrator of the network. The client application is to be run on machines on the network to be administered.

After installing NetDog server or client on a machine, a unique public-private key pair for the machine is generated which is then used for uniquely identifying the machine and securing data transmission between the client and server.

Once the server program is up and running, it listens on the port 1337 for connections from clients. Once the client program is up, it starts listening on port 1994. These ports serve dual purpose of facilitating communication and allowing the identification of server and clients from the rest of the machines on the network. Both NetDog server and client are daemons. They are system

services which remain in memory and automatically starts during system boot.

When a client starts for the first time, it looks for active servers on the network. When it finds one, it starts the pairing procedure. During the pairing process, the client sends its hostname and public-key. The server in turn provides the client with its public key. These keys are then used for identification and encrypted communication between machines.

The client and server uses public key encryption to identify and secure the communication between them. The network administrator can issue commands from the server machine which will then be sent to all the clients on the network. A client can also contact the server occasionally, for example if the client detects undseriable network traffic or if the system is overheating.

NetDog is capable of shutting down all the clients on the network at once by remotely executing the shutdown command. It is also able to power up systems which support remote Wake-On-LAN feature, by sending magic packets.

2.2.1 Product Functions

- Bring all computers on the network up and down remotely
- Execute commands/scripts remotely on machines
- Copy files to remote machines without third party protocols
- Track and identify clients through IP changes
- Secure client-server communication using public key encryption
- List all machines on the network which are not clients (detect intruders)
- Centralized logging of all data regarding clients
- Web interface

2.2.2 Hardware Requirements

- Intel Pentium IV or equivalent CPU
- 512 MB or more RAM
- 100 mbps Network Interface Card

2.2.3 Software Requirements

- Linux
- Python 3
- pip

2.3 Functional Requirements

The system is should be designed to accept communication requests from many clients at once. For this, a multi threaded server is necessary. Also, network outages can occur during the operation. The server should be resilient enough so that, it checks the status of the connection every once in a while and restarts the communication process once the network is up and running again. Also, the functions should extensively log themselves so that in case of a system failure, the culprit can be easily found.

2.4 Performance Requirements

The system would need a gigabit ethernet controller for the server to make sure that it can properly handle connections from multiple clients on the network. The client can work satisfactorily well even on old hardware such as a 10 mbit network card. There are no specific requirements for the CPU or the rest of the hardware. The machine must be powerful enough to run a recent version of Linux, which means any usable computing hardware would do.

Bibliography