WebNMS

A Division of ZOHO Corporation

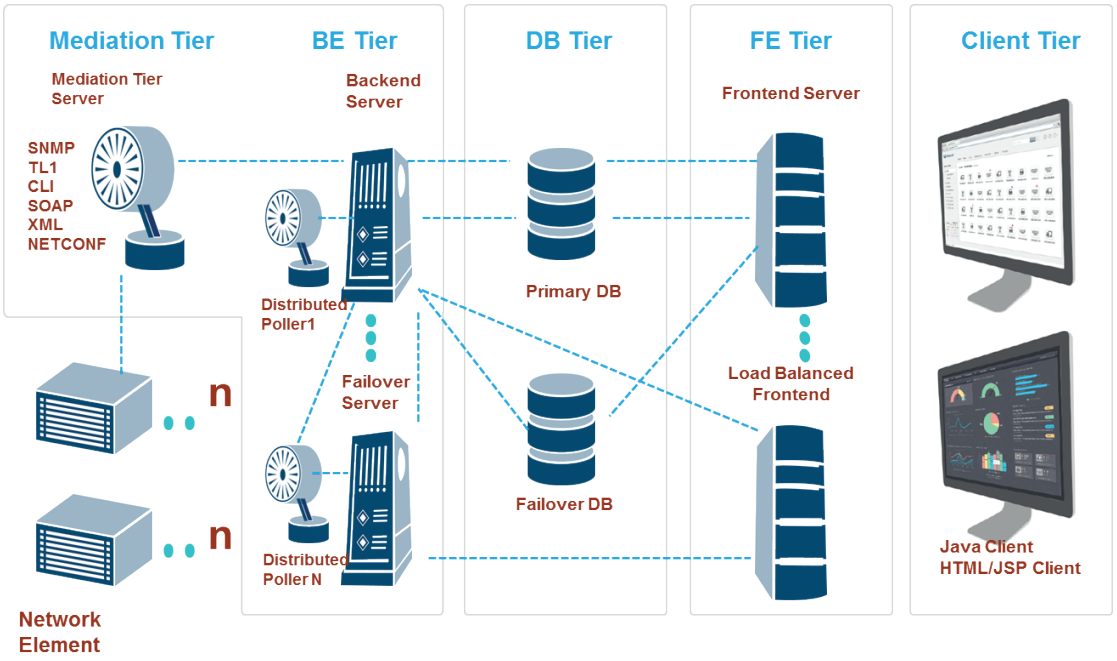
SECURING WebNMS

Table of Contents

Introduction

Security is concerned with ensuring legitimate use, maintaining confidentiality, data integrity, and auditing in the network. Security Management involves identifying the assets, threats, vulnerabilities, and taking protective measures, which if not done might lead to unintended use of computing systems. In this document, we will see the various respects in which WebNMS deployments can be secured to the highest extent possible.

The following image gives an overview of the different components in WebNMS Framework and the interactions between them:



The following sections capture the measures to be undertaken to safeguard each of the layer/component.

Securing the Backend server

This is the machine in which the primary backend server runs. Securing this machine / JVM involves the following steps:

Folder level privileges

The directory level restrictions and privileges to the BE installation can be configured such that only the concerned user (in Linux/Windows) is able to access the WebNMS server installation and the other under privileged users will not be able to view the directory at all. This will ensure that only the concerned user will see the WebNMS installation.

BE - FE server communication

WebNMS Front End Server (FE) provides a link between the Back End Server (BE) and the Clients. FE server accesses the database either directly or through BE server based on the type of client's request (read/write operation). FE server communicates with BE in TCP & RMI protocol. FE server can be configured to communicate in secure mode, which will ensure that in the unlikely event of the packets (between BE & FE) being sniffed, the intruder will not be able to decrypt the communication.

Securing the database connections

Both BE & FE servers store & retrieve data in a relational database (like Pgsql/ MySQL/ Oracle) through jdbc drivers. This communication can be secured by use of SSL between the JDBC driver and the database server. The SSL configuration is typically done at the database server as mentioned in the above links.

Securing the channel between server & devices

When the communication protocol is SNMP, the BE server communicate with the managed devices in UDP protocol. This channel can be secured by configuring the server and the devices to communicate in SNMPV3 protocol

Securing Client-Server communication

WebNMS client (basically the web browser) is the gateway for any end user to connect to the server and is the most exposed element in the communication. It is very likely that the servers (BE & FE), database & the devices exists in the same LAN/WAN (or in the same VPN) and the client is connected from anywhere in the outside world (even from a mobile / tablet). Hence it's very much essential to safeguard the client-server communication.

Following are the steps needed to ensure a safe & secure client-server communication

Firewall between clients & the server

Any critical network needs firewall to be protected. The LAN/WAN in which the servers & database exist need to be safeguarded by a firewall. The firewall can be restricted to:

1. Connect clients only from certain IP addresses or a set of IP addresses to connect to the BE/FE server.
2. Ensure that the client is allowed to communicate with the server only in the specified ports.

SSL protocol for client-server communication

WebNMS BE server can be configured to server clients only in https mode with appropriate settings as discussed below.

SSL Protocol version

Since the recent POODLE vulnerability in SSLv3 protocol, TLS is the most secure protocol and the TLSv1.2 scheme is the strictest version of securing https.

Ciphersuites for securing SSL

A ciphersuite is a named combination of authentication, encryption, message authentication code (MAC) and key exchange algorithms used to negotiate the security settings for a network connection using the Transport Layer Security (TLS) / Secure Sockets Layer (SSL) network protocol. Client-server SSL communication can be configured to use the strictest Ciphersuite combination without affecting the user experience.

Authenticate all http(s) requests

Except the basic files required to login (images/.css/html), all other files need be served only if there's a valid client-server session exists for that request.

HttpOnly cookies

When you tag a cookie with the HttpOnly flag, it tells the browser that this particular cookie should only be accessed by the server. Any attempt to access the cookie from client script is strictly forbidden. This prevents spoofing of the WebNMS Client's cookie object.

Other security practices

Since WebNMS is a development platform, the applications developed with WebNMS need to be concerned with security with regards to the customization done on WebNMS. Here are some of the security practices to be followed with regards to customizing or developing & deploying applications using WebNMS:

1. One-way authentication of passwords needs to be enabled. This will ensure that the user passwords cannot be decrypted.

2. Passwords configured in any of the text files need to be encrypted and stored.

3. Ensuring all third party components (JRE / Tomcat/ database/ jars) are free from all exposed vulnerabilities.

4. Change all the default passwords for database / keystore / snmp.

5. Sign WebNMS jars - This will ensure that none of the jars could be manipulated.

6. Enable SSL over RMI, which ensures that the traffic through RMI protocol is also encrypted.

7. SSL Renegotiation needs to be prevented.

8. Ensuring passwords & no sensitive information are transmitted in plain text between server & client.

9. SQL Injection is sealed in all possible ways by using prepared statements in places wherever necessary.

10. Security parameters are tightened in terms of the following processes:

    * Password complexity

    * Maximum attempts allowed for wrong passwords

    * Change passwords when the client is logged in for the first time

    * Periodically expiring passwords and ensuring that any password is not permanent.

    * Authenticating all menus & links before they are served in client