A lightweight remote NMS for private networks

Jeann C. Hernández Franco, and José Ortiz-Ubarri
Department of Computer Science, University of Puerto Rico, Rio Piedras Campus

Abstract

The goal of this research project is to create lightweight remote network monitoring systems (NMS) that gathers network data and securely transfers it to an external monitoring platform in order to allow remote secure monitoring of private networks. In this work we present an automated remote monitoring system developed with the use of VPNs, autorecon, and python scripts. This project's research aims to design and develop methods to provide remote secure monitoring of private networks by developing a lightweight network monitoring system (NMS) that collects network data and transfers it safely to an external monitoring platform.

Background

The ever-increasing demand for security in the cyberspace environment is always on the lookout for the safest way to protect its users against cyber attacks [1]. As a result, restrictions might be imposed to prevent malicious agents from gaining access to computer networks. Some of the measures taken to provide security in the perimeter of computer networks are the installation of a firewall [2] or the configuration of a private network. This approach whilst providing network protection from outsiders, lets its owners without the capacity of remote monitoring for insider threats, and network breaches without opening a remote access window. In many cases the owners of such networks do not count with the budget or the expertise to monitor their own network. This project's research aims to design and develop methods to provide remote secure monitoring of private networks by developing a lightweight network monitoring system (NMS) that collects network data and transfers it safely to an external monitoring platform.

Methodology

The project presented was separated into three phases:

1. Establishing a safe connection.
2. Automatization of data collection and transmission.
3. Automated detection of network changes.

Establishing a safe connection: Authorized users connect their sensors to the NMS's Virtual private network (VPN)[3]. Sensors are provided with RSA keys to connect to the VPN, and must also have valid SSH keys to perform the data transfers. This ensures that even if a perpetrator manages to connect to the NMS's VPN the added security measure won't allow said perpetrator to do much more harm

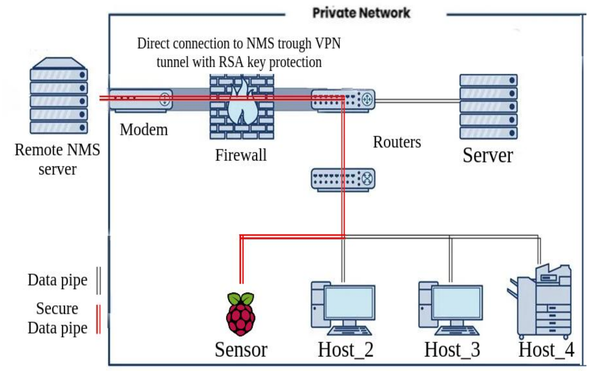

Figure 1 : Installed raspberry pi on a private network creating a secure data pipe connection.

Automatization of data collection and transfer: Currently the project's data collection is performed using autorecon [4]. We chose autorecon because it is a small project that leverages industry standard software that are easy to install, and are included in most unix-like operating systems package managers. To transfer the collected data we use another industry standard rsync, which efficiently synchronizes files between the sensor and the NMS server over the SSH protocol.

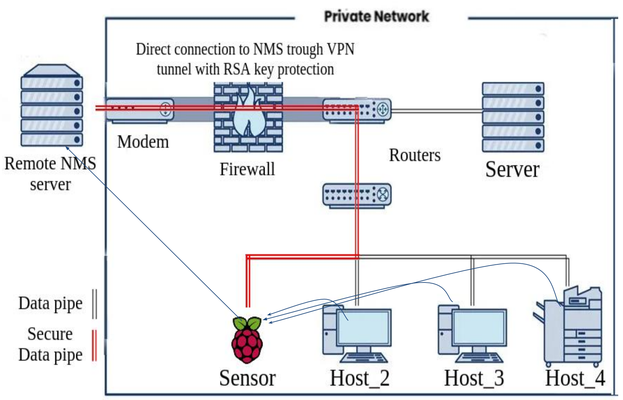

Figure 2 : Installed raspberry pi collects data such as services from hosts and transfers it directly to the NMS.

After a sensor's local network is successfully scanned, the results are transferred to the NMS server through rsync, to be analyzed and stored into a database. See Figure 2 for a detailed example.

To provide data integrity and confidentiality to the NMS from the sensors a script was developed to restrict sensor access as follows:

a) Gives access to the NMS only to authorized sensors by restricting access to rsync from users with no valid RSA keys.
b) Gives each sensor its own data folder with restricted permissions.
c) Sets each sensor an assigned a hard drive quota.
d) Disables No-agent-forwarding, no-port-forwarding and No-pty from rsync.

Figure 3 : Example schematic for phase 3.

Automated detection of network changes: Once the data has been transmitted by the sensor to the NMS server it is then analyzed to identify changes and possible intrusion on the sensor's local network. Currently the analysis consists in comparison of the current state of the network and services with the last state recorded previously. It looks for:

a) The addition or removal of a host in the network,
b) The addition or removal of host's services.
c) Hardware address changes.

Current State

The project currently has working scripts that make it simple to install our software on computers running the Debian operating system and on Raspberry Pi; It also has a collection of scripts that collaborate to build an framework that scans networks, looks for vulnerabilities, saves all gathered data, and sends it to our server. The framework can do this on a regular basis without human interaction. Furthermore, the database operates in conjunction with an in-development website to assist clients in seeing the recorded data. This is enhanced further by scanner's user authentication, so clients can only access their particular local network.

Future Work

In the future, better integration with autorecon's retrieved data and the aforementioned framework is currently planned, this is in hopes of being able to analyze it and potentially locate threat sources to be able warn users via the web site without the need of user interaction. Other plans such as the customer being able to access the website and view information such as the next scan time or the amount of resources being consumed on a particular computer, and what particular programs could cause security concerns are also currently also in the works.

References

[1] Yuchong Li and Qinghui Liu. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. Energy Reports, 7:8176–8186, 2021.
[2] Habtamu Abie. An overview of firewall technologies. 12 2000.
[3] Kuwar Kuldeep Veer Vikram Singh and Himanshu Gupta. A new approach for the security of vpn. pages 1–5, 03 2016.
[4] Tib3rius. Tib3rius/autorecon, 11 2020.