

Network Security Management for a National ISP

R. Deepak, Timothy A. Gonsalves, Hema A. Murthy, and N. Usha Rani

TeNeT Group, IIT Madras

Chennai – 600 036

Email: jrdeepak@tenet.res.in, tag@tenet.res.in, hema@tenet.res.in, usha@nmsworks.co.in }

Abstract

A network is one of the most important basic resources a large institution (educational or commercial) needs to have. Today, networks play a very important role in every organization. With widespread distributed deployment, managing the security of a network becomes very complex especially from the viewpoint of an ISP. ISP's are by inherently more vulnerable because they have to offer a multitude of public services. Both for their customers and on behalf of them.

Effective network and security management needs to be implemented taking into consideration the lack of bandwidth and availability of computing resources at the nodes. Security management now plays a larger role as all communication is over the insecure Internet.

In this paper, the various issues involved in developing a Security Management System for a low bandwidth network is discussed. Then, a solution implemented based on open standards for n-Logue (a national ISP focusing on rural areas) is presented.

1. Introduction

1.1 Security Management :

Security Management involves protecting a network from all kinds of unauthorized access. This includes many sub functions like collecting and reporting security related information, pro-actively detecting and preventing intrusions, etc.

This assumes even greater significance with the rapid growth of the Internet

1.2 n-Logue :

n-Logue is an unusual operator in that its focus is on providing affordable voice plus Internet access in villages and small towns throughout India. As such, it has a far-flung network and must keep costs to a minimum. Network security management is essential and the bandwidth consumed by management traffic must be kept very low.

1.3 Challenges :

There is very little bandwidth available to us. There is no backbone management network and all communication is over the insecure public Internet. All nodes are already running services and the management overhead should not be too high.

Any system developed should easily integrate with the Network Management System already being used .

1.4 Overview :

In section 2, we'll take an in-depth look into how the n-Logue network is organized and what kind of security management we actually need. We'll follow this up with Architecture, Design and Implementation in sections 3 and 4. And conclude with a discussion on some performance parameters in section 5.

2. Background

2.1 n-Logue network :

The network consists of a national data center and Local Service Providers (LSP) distributed all over India. The national data center is connected to the Internet with a 256kbps link and the LSP's have either a 64kbps or a 128kbps link based on the number of subscribers.

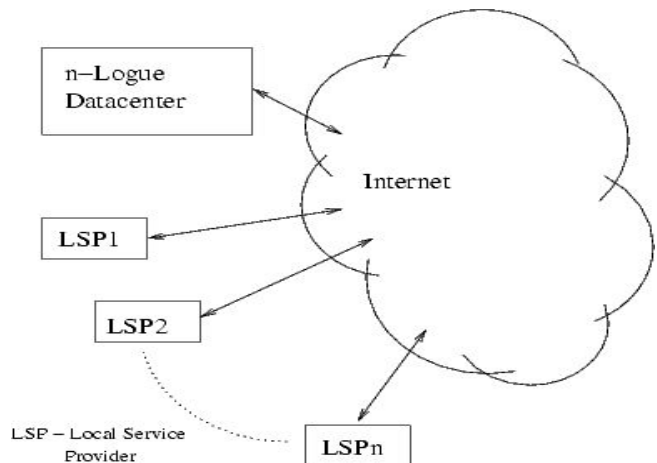


Figure 1. Topology of n-Logue network

Currently, n-Logue has around 25 LSPs and this is expected to grow to 100's shortly. Each LSP in n-Logue has the following elements to provide voice and Internet services; corDECT WiLL system, Minnow servers, router and a leased line. Distributed management is needed to keep costs low and because the network is spread very wide. It is not economical to send someone over to the LSP to fix problems regularly.

2.2 Security Management :

Security Management covers the following aspects:

- a. Intrusion Detection (Network and Host)
- b. Configuration Management of remote nodes
- c. Analysis of data collected at the remote nodes
- d. Taking action based on analysed data (delayed)
- e. Real-time response to certain types of intrusions

A lot of research is currently going on in the area of Intrusion Detection. But most available products both open-source and commercial do not handle Distributed Intrusion Detection very well. The type and amount of data to be shared between multiple sensors has never been clear. In this regard, there are IETF drafts that try to set a standard on what types of messages need to be exchanged and the format of those messages.

Reliable Network Intrusion Detection Systems do exist which can operate at the gateway. The best open-source system is Snort. Extensive documentation exists for this tool and therefore, it can be used as a base for developing the Intrusion Detection part of the distributed system.

A lot of research has been going on in the field of remote configuration management of computer systems. One of the major systems being developed is cfengine (a system configuration engine for UNIX systems).

3. Architecture

Each LSP has two servers already being used to provide all the services to the subscribers. The Master Internet Server (MIS) and the Redundant Internet Server (RIS).

Our Network Intrusion Detection System (NIDS)

and the configuration management system will be running on these servers. The various services provided by RIS and MIS are

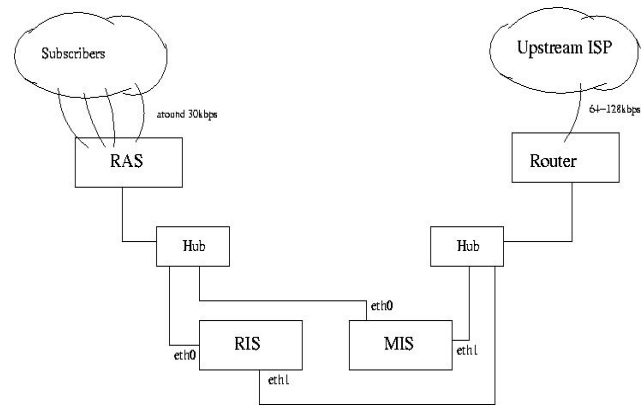


Figure 2. LSP organization

- a. Proxy for WWW access
- b. email
- c. DNS
- d. Web hosting
- e. All other connections to the Internet are provided through Network Address Translation (NAT).

We need to protect these servers against attacks from the Internet and from the subscriber network.

4. Design and Implementation

4.1 Network Intrusion Detection System :

The NIDS chosen is snort. Snort is a high performance, light weight, highly customizable open source NIDS. It supports a wide range of reporting formats, which will be really useful in our case.

4.1.1 Customizing the NIDS :

Snort normally has thousands of rules. Having everything enabled will drastically increase resource requirements on the RIS/MIS servers. Therefore, the rules will have to be tuned to only include what we actually need. Tuning the rules also helps in bringing down the number of false positives.

4.1.2 Reporting format :

For standards compliance and easy integration with any upper level NMS, the default format has to be

the Intrusion Detection Messages Exchange Format (IDMEF) which is an IETF draft. For real-time reporting however, SNMP traps are much better.

4.1.3 Attack Classification :

We should also look for only certain types of attacks and classify them according to severity. And then, based on this severity, we can decide whether we want to take any immediate automated action.

Attacks are classified as:

- a. Denial of Service (DoS) attacks either directed at our servers or directed at some server on the Internet from within our network.
- b. Worm traffic. This is a major problem faced by every ISP. Clogs up all available bandwidth.
- c. Policy Violations.
- d. Targeted attacks on LSP servers.

4.2 Configuration Management System :

A lot of research has been going on in the field of remote configuration management of computer systems. One of the major systems being developed is cfengine(a system configuration engine for UNIX systems).

The system consists of one central server running a cfengine daemon(cfservd) and all the managed nodes running a cfengine agent(cfagent). The cfagent on each remote machine has minimal configuration done just to enable communication with the cfservd on the central server. All host specific configuration is done at the centralized location and then the agents import the proper configuration information from the server.

Extensions can be easily written for cfengine and an extension has been written that will enable cfengine to monitor snort logs and take any automated action if necessary.

Cfengine also has host intrusion detection. It can monitor files for changes and restore any changed files from backup copies. The basic cfengine based system that has been implemented is shown in Figure 3.

4.3 Integration with a higher level Network Management System

A standardised mechanism is needed for this. We need to be easily use/integrate the security management system with any higher level NMS. We need to be able to use a standard NMS to monitor and control the security management system.

To monitor the security management system, we need to use standard reporting formats for both statistical data and real-time updates. IDMEF is a XML based reporting mechanism recommended by the Internet Engineering Task Force (IETF). And this suits us best for transferring statistical information.

For real-time alerts, we've decided to use SNMP traps. This is supported by all NMSs being used today.

For secure control of the system, cfengine by itself provides a remote command execution mechanism that uses SSL. An interface has been implemented to allow a higher level NMS to use this.

5. Performance Parameters

The performance factors that will affect this system the most are:

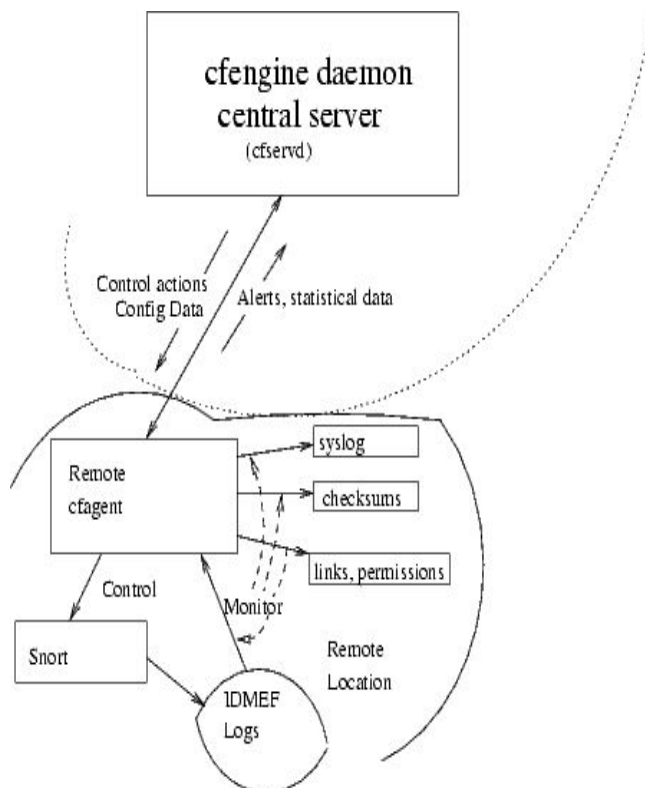


Figure 3. cfengine integration

5.1 Bandwidth used :

We have only around 64kbps available at each node and 256kbps at the Data Center (DC). At the DC, we will have to manage 100's of nodes.

Most of the bandwidth savings are obtained by writing cfengine extensions to take decisions for automated actions at the nodes instead of sending real-time reports to the DC.

5.2 Latency :

We need to minimize the time taken for any control action to take place. It could either be automated or operator assisted. In case of operator assisted actions, we need to provide an interface to make the operators job easier.

5.3 Load on the servers at the LSP :

The servers on the LSP are already providing various services to the customers. Any management system is an add-on that must not consume too much of the available resources.

The way host based intrusion detection is performed, the rules on the NIDS, the reporting mechanism, etc. play a role in how good the developed system is.

6. Performance Evaluation

Performance was evaluated with a 100 node simulator. Bandwidth and CPU utilization at the DataCenter were measured using actual traffic collected in the n-Logue network.

Some results showing bandwidth and CPU requirements at the DataCenter are presented along with a comparison between the proposed architecture and the normal centralized approach.

<i>Number of Nodes</i>	<i>CPU(%)</i>		<i>Bandwidth(bps)</i>	
	<i>Central</i>	<i>Distributed</i>	<i>Central</i>	<i>Distributed</i>
20	8	3.8	2735.8	550.4
30	13	7.5	4210.3	840.6
40	25	12	5503.2	1100.2
50	35	16.3	6950.9	1387.9
100	85	40.5	14200.8	2850.7

Table 1. Comparison of Centralized and Distributed Security Management

7. Summary and Conclusion

We've shown the basic design and architecture of a Security Management System that can work on low bandwidth networks. Most of the challenges have been met. As shown in Table 1, considerable improvements in both CPU utilization and Bandwidth utilization are obtained.

The system is based completely on open standards and uses open source components wherever possible. This helps us to drive down costs and makes it easy to customize the various components involved.

References

- [1] cfengine – GNU project, [http:// www.gnu.org /software/cfengine/ cfengine.h tml](http://www.gnu.org/software/cfengine/cfengine.html)
- [2] Mark Burgess, Cfengine: A site configuration engine, USENIX Computing systems, Vol8, No. 3 1995
- [3] T. A. Gonsalves, Ashok Jhunjhunwala, and Hema A. Murthy et al., "CygNet: Integrated Network Management for voice+Internet," NCC 2000
- [4] A.G.K. Vanchynathan, N.Usha Rani, C.Charitha, and T. A. Gonsalves, "Distributed NMS for Affordable Communications", NCC 2004, January 2004
- [5] Snort - NIDS, [http:// www.snort.org](http://www.snort.org)