Monitor Linux Security Using Scripts

Ashvini T. Deshmukh, Parikshit. N. Mahalle

Department of Computer Engg, Smt. Kashibai Navale College of Engg, Pune, India

ABSTRACT

This paper will show how to use basic Linux scripting to create a reusable Linux security monitor such as workstation, network and server security monitor that is simple to use and simple to maintain. Linux commands are discussed along with techniques to automate them and interpret their results. This paper gives enough information for security professionals to start creating their own generic reusable Linux script within their own collection of personal tool.

Keywords: Linux Scripting, Automating Linux command, Security aspects.

INTRODUCTION

Purpose

Many security professionals find themselves in a position where they do not have a complete set of commercial security tools. This happens often because of tools are lacking, in a low budget situation where purchasing tool is not possible or in limited environment where commercial products cannot be implemented. Even in limited situations, it is unacceptable to try to implement security without proper tools. There should be each security requirement must be addressed in some way even where extensive commercial tools are not available. Where extensive tools are not available simpler tools of some sort must be implemented to enable basic security.

Proposed Works

To achieve our purpose Linux scripting can help. Linux has many more powerful commands which are used for checking various security aspects. These commands can be harnessed and automated into a generic reusable tool that will provide desired results. So commands such as ping, netstat, nmap are automated. Using script workstation vulnerability report, disk utilization report, FTP server vulnerability report, network status report generated and it is in simplified form because of that simplification user who is not much aware about Linux security is easily understand report and give attention towards Linux security. Also alert messages are generated by seeing this alert messages user easily understand what are the vulnerability and what should be the resolution. Using script log analysis report is generated by seeing this report user will always keep watching what is going on inside the network if server may crash he will easily figure out the reason that is what happened prior to crash. Also using script network security monitored. Linux scripting provide basic functionality, consistency, ease of use, ease of maintenance.

RELATED WORKS

In [3] we address vulnerabilities in different areas such as workstation, network and server security. In this paper given vulnerabilities and its countermeasures.

*Address for correspondence:
ashu13.cse@gmail.com
In [4] we propose a new system implementation in order to harden the Linux. Following is proposed system implementation.

**Figure 2.1. System Implementation [4].**

The Linux Hardening model consists of three modules which makes the Linux more secure from the attackers which are:

1. Vulnerability check module
2. Log Analysis Module
3. Security Module

**Vulnerability Check Module**

This module will check such configuration files and scan for attribute which are important from security perspective. This module check current attribute value with best security value required for that attribute. If current configured value is not a best security value then it will consider it as vulnerability and generates the vulnerability report. Generated report is given to the security module.

**Log Analysis Module**

Linux system consists of very strong logging mechanism maintains the log for kernel, servers, users, system processes etc. These entire logs by default placed at different location. This module collects the log from these various places and generates report. This generated report is useful for finding the vulnerability. Generated report is given to the security module.

**Security Module**

This module collects the vulnerability report and log analysis report and applies security. By looking vulnerability report this module get the vulnerable configuration files and modify them with best security practice. Similarly by looking log analysis report this module apply the security attributes accordingly. This model is actually responsible for modifying the configuration files and making the Linux more secure.

**RESULTS AND DISCUSSION**

**Linux Commands**

The ping command will show whether a machine is available on network or not. It can also show timing delays and whether packets are being lost on the network.
The netstat command will show that what connections are currently active between the local machine and other network machine. On a server this would show who is connected to the server or communicating with it.

Automating Linux Command

Basic script can be created to automate these commands. This allows a single script to perform frequent test. Script can contain hard coded command so that command syntax does not have to be remember and so helpful options are not forgotten.

Result Interpretation

We have tested commands in network having three nodes. Machines having IP addresses are localhost (127.0.0.1), 192.168.10.10, 192.168.10.12. These commands are tested by interactively typing the ping command with count of three and results are tested.

On Existing System:

On linux prompt type following command but this command will have to type for three times.

```
# ping -c3 localhost
```

But you should have repeat this procedure to check another machine in network. This shows that 192.168.10.10 machine is up on the network. It shows that is not losing packets and also round-trip time is good. However it would be time consuming to frequently ping every machine and observe the results and results are also not in simplified form.

On Proposed System:

Consider the script called “network status report”. This script will execute the three ping commands for when the command name of the script is typed at the Linux prompt. It will shows following report.

```
-------NETWORK STATUS REPORT--------
HOST        STATUS
localhost respondig normally 0% packet loss, 0 ms
192.168.10.10 responding normally 0% packet loss, 16 ms
192.168.10.12 Not responding at all
```

So script will perform ping command, test the results and gives final conclusion in simplified form also generate the alert message so that administrator understand network status. The command do not have executed separately.
Discussion

We have taken ping command as an example for result interpretation. We have developed many more scripts for different security aspects. Following are the scripts:

1) vulnerability report in that script four type of report generated which are workstation vulnerability report, Disk utilization report, FTP server vulnerability report, Network vulnerability report.

2) Script for user security, in that script we provide different options for user management such as manage users without password. We can delete user or add password to user who doesn’t have password. Another functionality is apply age policy and single user mode password.

3) Script for package management, in that script three options are there. We can list out installed packages, verify installed package, install packages.

4) Script for network security, this script is very worth because it checks network related security aspects. Following are the options over there - close open ports, remote live monitoring, remote port scan, remote live monitoring, login banner, block packet forwarding, block reply to ICMP broadcast, enable protection against bad ICMP messages, enable SYN flood protection, block source routed packets.

5) Script for log monitoring, in that script we collect log from different log files and separate it according classification and generate respective reports such as summary report, failed summary report, authentication report, login report, login from remote host report, account modification report, anomaly report. Also performs live log monitoring within network.

GRAPHS

Time Required For Report generation

Following graph shows time required for checking different vulnerability and report generation in existing Vs proposed system.

![Graph showing time required for report generation](image)

**Figure 4.1. Time required for report generation**

Graph shows time to check workstation vulnerability in existing system is 3 minute but for proposed system it is 1 minute. In similar way time to check disk utilization and FTP server vulnerability in existing system is more than proposed system. Also for network vulnerability time to check vulnerability is very high in existing system than proposed system.

Time Required For Log Analysis and Network Security

Following graph shows time required for log analysis and network security in existing system and proposed system.
Figure 2.2. Time required for log analysis and network security

In existing system, the time required for log analysis is 15 minutes but it is 4 to 5 minutes in the proposed system including live log monitoring. For network security such as listing open ports, closing unrequired open ports, killing unnecessary processes, and managing login banners takes more time in the existing system than the proposed system.

**COMPARISON PARAMETERS**

**System Characteristics**

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Existing System</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>General purpose</td>
<td>Security purpose</td>
</tr>
<tr>
<td>Acceptable processing time</td>
<td>Long time</td>
<td>Less time</td>
</tr>
<tr>
<td>Handling complexity</td>
<td>Hard to handle</td>
<td>Easy to handle</td>
</tr>
</tbody>
</table>

**Security Practices**

<table>
<thead>
<tr>
<th>Practice</th>
<th>Existing System</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security awareness</td>
<td>Usually low</td>
<td>High</td>
</tr>
<tr>
<td>Working with security parameters</td>
<td>Difficult</td>
<td>Easy</td>
</tr>
<tr>
<td>Need of security expertise</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**5.3. Impact Of Negative Event**

<table>
<thead>
<tr>
<th>Event</th>
<th>Existing System</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Losses</td>
<td>Information, data more chances</td>
<td>Less chances of loss</td>
</tr>
<tr>
<td>Cost of successful attack</td>
<td>More</td>
<td>Less</td>
</tr>
</tbody>
</table>

**CONCLUSIONS AND FUTURE SCOPE**

Linux is an open-source operating system so security professionals should take advantage of it and tune and customize the kernel to their specific needs and hardware. So harden the Linux according to requirement. Because of the characteristics and more powerful command it becomes popular. Linux is secure but when its security parameters are set to standard security values, and for that security awareness is very important. This paper gives security awareness so that a user who is new to Linux also understands security aspects. This is achieved through scripting. Basic Linux scripting can be used to develop tools in less amount of time and easily with little cost. It is important to always remember your scripts should be reusable and generic. Otherwise, they will have to be rebuilt from scratch every time. The more maintainable a script is, the more likely it is to be adapted to changing requirements and ported to new environments.

**REFERENCES**

Ashvini T. Deshmukh & Parikshit. N. Mahalle “Monitor Linux Security Using Scripts”


AUTHORS’ BIOGRAPHY

Ashvini Tanaji Deshmukh has obtained BE degree in Computer Science and Engineering from Shivaji University, Kolhapur, India and pursing M.E. degree in Computer Networks from Savitribai Phule Pune University, Pune, India.

Prof. Parikshit N. Mahalle has obtained his B.E degree in Computer Science and Engineering from Sant Gadge Baba Amravati University, Amravati, India and M.E. degree in Computer Engineering from Savitribai Phule Pune University, Pune, India. He completed his Ph. D in Computer Science and Engineering specialization in Wireless Communication from Aalborg University, Aalborg, Denmark. He has more than 14 years of teaching and research experience. He is a member board of studies in computer engineering, Savitribai Phule Pune University, Pune, India. He is IEEE member, ACM member, Life member CSI and Life member ISTE. He is paper reviewer for Springer journal of Wireless Personal Communications and Elsevier journal of Applied Computing and Informatics. He has also remained technical program committee member for International conferences and symposium like ICC – 2014, ICACCI 2013, IEEE ICC 2015 – SAC-Communication for Smart Grid, IEEE ICC 2015 – SAC-Social Networking, IEEE ICC 2014 – Selected Areas in Communication Symposium, IEEE INDICON 2014, CSI ACC 2014, IEEE GCWSN 2014. He has published 42 research publications at national and international journals and conferences. He has authored 5 books on subjects like Data Structures, Theory of Computations and Programming Languages. He is also the recipient of “Best Faculty Award” by STES and Cognizant Technologies Solutions. He has also delivered invited talk on “Identity Management in IoT” to Symantec Research Lab, Mountain View, California. From 2000 to 2005, he worked as Assistant Professor in Vishwakarma Institute of technology, Pune, India. Currently he is working as Professor and Head in Department of Computer Engineering at STES’S Smt. Kashibai Navale College of Engineering, Pune, India. He has guided more than 100 plus undergraduate students and 10 plus post-graduate students for projects. His recent research interests include Algorithms, Internet of Things, Identity Management and Security.