A Secure Computing Environment

Avantika Dhavale[1], Aditi Halgekar[2], Snehal Wayse[3], Pavan Kulkarni[4]
[1 2 3] Students, Computer Department, Trinity College of Engineering, Pune.
[4] Professor, Department of Computer Engineering, Trinity College of Engineering, Pune.
Contact Email- avantika.dhavale@gmail.com Contact number- 7387709896

Abstract

With the advent of various attack vectors on various computing devices, it is vital that we design a secure computing environment which would be resilient to such said attacks. However, there is always a tradeoff between securing a device and incorporating various functionalities into the device to make it more versatile. The more we secure a device, the tougher it becomes to add ad-hoc features to it. Moreover, this tradeoff is very subjective to the needs at hand. It is incumbent on the administrator of the said device to manage the tradeoff between securing the device and providing diverse functionality. In this paper, we describe our approach to securing a computing environment and explain our rationale. This would be of course one of the multiple layers of security with which we would secure the device.

Keywords— White-list, Hardened OS, environment, locked down OS, REL-ID,.

INTRODUCTION

A security system identifies and mitigates the system vulnerabilities, by either removing them, or restricting access to them, to a very small group. The competition between inventing new security measures to protect data and inventing hacking techniques in conjunction with discovering and leveraging pre existing vulnerabilities is infinite. Therefore, securing data and resources is becoming more and more challenging day by day.

Nevertheless, there exist several different techniques to secure the data being transferred over a network and also that on a user machine. Uniken India Pvt. Ltd. specializes in securing data in motion through the use of the patented REL-ID based mutual authentication scheme.

SSL is one such tool to secure data sent over a network, using cipher text. Using SSL data is kept confidential and message integrity is maintained. However, recently there have been network security breaches, including the famous “HEARTBLEED” bug.

But, the question that remains is “what if the user machine itself is hacked?” REL-ID by Uniken India Pvt. Ltd. can be used to ensure that the end user is secured as well as the tunnel. It also uses techniques of authentication to assure to each end user that it is communicating with an authorized user and not a fake one.

Such security measures are used to secure data in motion, meaning data that has been shared between computers. They may prove to be of minimum value, if the operating system on which it resides is compromised. It is therefore crucial to understand and remove the security flaws in the operating system itself. We, on the other hand, are trying to secure data at rest, by coming up with various approaches, one of which is application white listing.

In this paper, we will discuss ways to do this in the Linux Ubuntu operating system. Firstly, we try to harden the operating system. Hardening is a technique to reduce vulnerabilities of the existing operating system. It aims to eliminate security risks in an operating system. This is done by turning off all those services of the operating system which are not used or are risky and allowing only those which are secure for users data. Thus, this environment becomes a kind of locked down or reduce version of a fully fledged operating system.

While the services which are "turned off" in hardening may be useful or beneficial in some or other way, if through their use there use there exist back-doors to the system they must be shut down. Operating system hardening is a technique which allows us a security on the machine level. A hardened operating system can be considered as a smaller version of an otherwise compromised operating system.

Secondly, we implement a technique called as application white-listing. It is the technique of preparing a list of all applications that are safe to execute. All applications that excluded from this list are disallowed to spawn.

RELATED WORK

In our research related to data security we have discovered many ways to secure a transaction over a network. This research has led to an understanding of topics like PGP,PKI, various encryption algorithms like RSA,SSL. We also studied about significant ways to provide security to the end system.

Rel-Id, developed by Uniken India Pvt Ltd is one such infrastructure to secure the end system. This system along with an application is currently being used for various banking systems. In this in client server architecture both the client as well as the server are assured of the identity of the end system they are talking to. However, no matter how secure the application is, if the operating system is compromised in some or the other manner, there exist a constant threat of the data being watched or stolen or hacked.

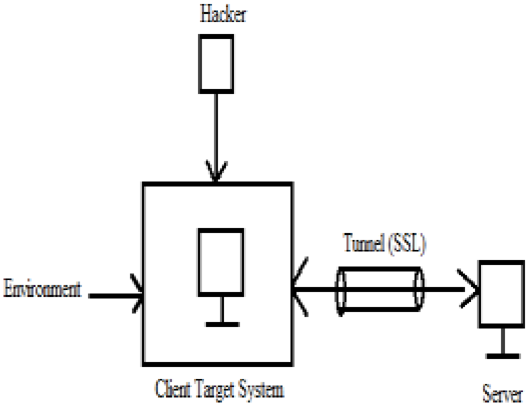

Figure 1: Secure Tunnel but Insecure End Point

METHODS

Hardening

For operating system hardening we may try to reformat the operating system and install only those parts of the operating system which are required for the users program to run. Hence hardening is depend upon the particular application for which it was done in the first place.

We may also consider disabling guest login as an added security measure. This is done because even if the guest user is not authorized, some kinds of penetration are always possible.

Also we may consider turning off services such as resource sharing, file sharing, printer sharing.

Whitelisting

In white-listing all applications that are found to be suspicious, or might be containing possible back-doors are denied permission to execute. This minimizes the threats to operating system.

This may start by preparing a list of names of applications which are safe to execute and checking the name of each spawned application against this list. If a match occurs we need not take any action. If the application name does not match with any of the supposed white-listed application, it is killed immediately. On a higher level, it should not be allowed to spawn in the first place.

RESULT AND DISCUSSION

As a result of the implementation of the above mentioned methods, we obtain a secure environment, which the user may trust for handling of confidential data. This environment can be depicted by the following figure

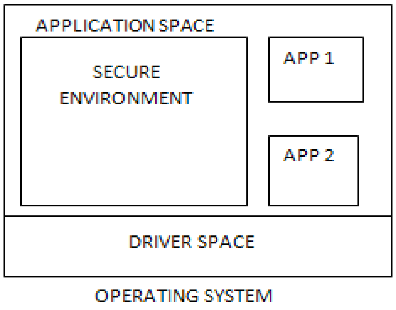

Figure 2: A secure environment

Examining the results of above mentioned methods, it became clear that unwanted applications and services of an operating system can indeed be stopped from exploiting the vulnerabilities of a system. If the operating system is in fact vulnerable to malicious applications, these methods can definitely provide a way to reduce these vulnerabilities.

CONCLUSION

Examining the results of above mentioned methods, it became clear that unwanted applications and services of an operating system can indeed be stopped from exploiting the vulnerabilities of a system. If the operating system is in fact vulnerable to malicious applications, these methods can definitely provide a way to reduce these vulnerabilities.

ACKNOWLEDGMENT

First of all, we would like to thank our guide, Prof. Pavan Kulkarni, for the motivation and guidance that he has provided us. Secondly, we thank our head, Mr. Vamshi Krishna Ramaka for his help and support. Finally, we thank our friends Miss Mehak Daftari, Miss Vrushali Deshpande, and Mt. Delwin John for their timely help and assistance.

.

REFERENCES:

[1] Chunxiao Li, Anand Raghunathan, and Niraj K. Jha, “A secure user interface for web applications running under untrusted oerating system” 10th IEEE International Conference on Computer and Information Technology, 2010.

[2] Uniken Systems Pvt Ltd, http://www.uniken.com/relid-platform”

[3] Bei Guan, Yanjun Wu, Yongji Wang, “A novel security scheme for online banking based on virtual machine”, IEEE Sixth International Conference on software security and reliability companion, 2012.

[4] Daojing He, Sammy Chan, Yan Zhang, Mohsen Guizani, Chun Chen, Jiajun Bu, “An enhanced public key infrastructure to secure smart grid wireless communication networks”, IEEE Network, January-February 2014.

[5]Xiongwei Xie, Weichao Wang, “Rootkit Detection on virtual machines through deep information extraction at hypervisor-level”, 4th International Workshop on Security and Privacy in Cloud Computing, 2013.

[6] Ngangbam Herojit Singh, A. Kayalvizhi, “Combining Cryptographic Primitives to Prevent Jamming Attacks in Wireless Networks”

[7] Khoa Dang Pham, Abhishek kumar Jain, Jin Cui, Suhaib A. Fahmy, Douglas L. Maskell, “Microkernel Hypervisor for a Hybrid ARM-FPGA Platform”, IEEE, 2013.

[8] Chunxiao Li, Anand Raghunathan, and Niraj K. Jha, “A Trusted Virtual Machine in an Untrusted Management Environment”, IEEE Transactions on services computing, Vol. 5, No. 4, October-December 2012.

[9] Sujit Sanjeev, Jatin Lodhia, Raghunathan Srinivasan, Partha Dasgupta, “Protecting cryptographic keys on client platforms using virtualization and raw disk image access”, IEEE International Conference on Privacy, Security, Risk and Trust, And IEEE International Conference on Social Computing, 2011.

[10] VMware Player, http://www.vmware.com/products/player, 2012.

[11] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D.

Boneh, “Terra: A Virtual Machine-Based Platform for

Trusted Computing,” Proc. ACM Symp. Operating

Systems Principles, pp. 193-206, Oct. 2003.

[12] J. Yang and K.G. Shin, “Using Hypervisor to Provide

Data Secrecy for User Applications on a Per-Page Basis,” Proc. ACM Int’l Conf. Virtual Execution Environments, pp. 71-80, Mar. 2008.

[13] Sujit Sanjeev, Jatin Lodhia, Raghunathan Srinivasan, Partha Dasgupta, ” Protecting cryptographic keys on client platforms using virtualization and raw disk image access” IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing, 2011.

[14] Jan Just Keijser,” OpenVPN 2 Cookbook: 100 simple

and incredibly effective recipes for harnessing the

power of the OpenVPN 2 networks”,Edition 1, published in 2011.

[15] Markus Feilner, Norbert Graf, “Beginning OpenVPN 2.0.9”,Edition 1,published: December 2009.

[16] Pavan Kulkarni, Aditi Halgekar, Avantika Dhavale, Mehak Daftari, Snehal Wayse, “Prototype of Computing Device That Aims TO Secure User Data on a Compromised O. S. ”, IJSR, Vol. 3, Issue 10, October 2014.