International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Prototype of Computing Device That Aims to Secure User Data on a Compromised OS

Pavan Kulkarni[1], Aditi Halgekar[2], Avantika Dhavale[3], Mehak Daftari[4], Snehal Wayse[5]

[1]Assistant Professor, Department of Computer Engineering, TCOER, Pune University, Pune, Maharashtra, India

[2,3, 4, 5]Student, Department of Computer Engineering, TCOER, Pune University, Pune, Maharashtra, India

Abstract: ***Our objective is to build an environment which provides a secured workspace for handling data, such as copying and editing important user files, for secure transactions over the internet and much more. It could be viewed as a Secured Machine equipped with locked-down Operating System with services and applications that support the user to access data securely. This environment will also decide which applications should run and which ones should not. It will also act as a logger of User Actions, in order to record and prevent an unauthorized action. Hence, an unauthorized person will not be able to access your personal information, either through the machine or by snooping on the internet, and will maintain integrity. In particular, we propose a mechanism which will focus on who is having access to data and how, as well as it will secure end-to-end data sharing and will have thinnest chances of intrusion.***

Keywords: SSL, Secure Environment, REL.ID™, Locked Down OS.

1. Introduction

In a world like this which gets updated every second, various hacking techniques break the existing security systems. REL.ID is a product developed by Uniken India Pvt Ltd. It provides a secure service for internet transactions. Currently, the most secure way to transfer data is through SSL (Secure Socket Layer) which secures the tunnel. But it does not authenticate the end points which can lead to misuse of the data.

We don't know if the client is a thief or a hacker or a real verified identity. Similarly one is never sure if the server is a fake bank or a real verified identity i.e. a trusted bank. Hence a Terminal - Authentication Mechanism is required, which is provided by REL.ID.

The main aim, here, is handling all the transactions on any secure application in a compromised OS by developing a secure environment which will provide a secure workspace for handling user data.

The current system mainly consist of REL.ID, operating system (which may be compromised OS). The REL.ID executes on this OS which may be prone to vulnerabilities. Based on our knowledge, following might be the various chances of intrusions in this current system:

1. Through the Ethernet, one can enter the system through driver space available in OS.
2. Eavesdropping on system.
3. System may have a backdoor.

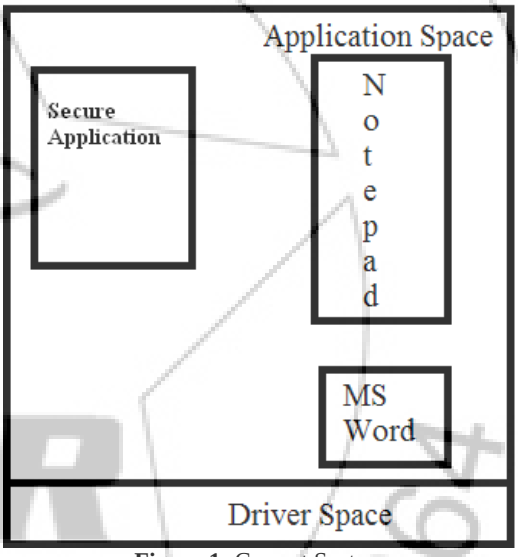

Figure 1: Current System

Hence we propose a system which is a peculiar environment on Host OS and to run the applications that provide security to user data and actions, inside it. Hence, providing a secure environment for a compromised OS.

2. Literature Survey

Through a prolonged and rigorous research and implementation of topics such as OpenPGP, PKI, OpenVPN, RSA Algorithm, we have found out that there are many ways to secure a transaction over the internet. Mainly, the SSL (Secure Socket Layer) helps in securing the tunnel between the client and the server (say, in a banking system).

But, there isn't any significant way to secure the end systems or the host machines. One such solution working towards this is the REL.ID infrastructure. In this, a REL.ID application is installed on a machine and only an authenticated user can perform all the desired transactions.

This infrastructure and a corresponding application is being used for banking systems at present. In this, the client gets authenticated by the server and vice versa. However, such a secured system may come across a threat viz. the end system itself, whether a client or a server, has been compromised, even before installation of REL.ID like infrastructure, with some type of malicious virus/backdoor to the O.S. In that case an authorized user of the system might still be thinking that the data transactions are secured, while the compromised OS is perhaps leaking out the data.

The description in this document is an attempt to propose a system that will allow safe transactions even on a compromised O.S. From our survey and experience, we acknowledge the necessity to have a secure environment over a compromised OS.

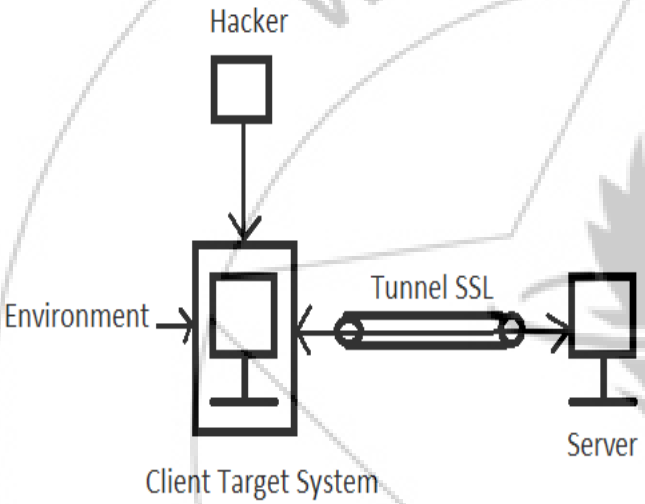

Figure 2: Secure Tunnel but Insecure End Point

3. Proposed System

The study of how data is transferred over the tunnel gives an idea about how the tunnel works. Sending data (both ciphered and non-ciphered) over a network is not only risky, but also is a serious vulnerability of the network. The said data may be any confidential data such as bank PIN etc., which users obviously would not desire to disclose.

The SSL (Secure Socket Layer) protocol proved to be a useful tool to prevent theft of data from the network. It is a protocol which provides security to the network. It authenticates the user at the other end of the network, who is communicating back with it. Thus data remains confidential and message integrity is maintained. The applications of SSL could be;

- Web browsing
- Instant messaging
- Email, Internet faxing, etc.

It uses a Public Key Infrastructure (PKI) and Certificate Authorities (CA) to authenticate the user, as well as to generate keys. But according to recent survey, CAs can be vulnerable to man-in-the-middle attacks (the 2013 mass surveillance disclosures have provided evidence for this). Thus, a question remains that if the spoofing is done not on the tunnel, but its end points then the data still remains vulnerable to outside threats. There was a need of securing both the users of the network. Supposing that one of those users is a server and the other is a client that is using the services, there is a need of assuring the client that it is talking to the server and also vice versa

RELID™ by Uniken India Pvt ltd is a protocol that is useful for the same purpose. RELID is an infrastructure that is designed not only for secure data transfer but also for securing the end points of the tunnel, that is, the end users, say client and server. Let us consider an example of a banking system containing a main server. Clients of the bank communicate with this server over the internet which is a vulnerable connection (for both ciphered and non-ciphered data). SSL helps secure this connection by providing a tunnel to the entire part of the network used by the client to communicate with the server. This makes it difficult to listen to the network and spoof into it.

RELID provides a way of ensuring that the data exchanged between client and server is safe, authentic and consistent. It uses the RMAK (REL.ID Mutual Authentication and Key exchange) protocol to assure the end users that they are communicating with an authenticated server and the data that they share is safe. The server verifies the client and vice versa. REL.ID acts as a gateway to sensitive data and applications.

As per our research, if the operating system on which a secure client application runs is compromised in some manner, then the user data may be vulnerable to leaks. To go further, if we have a hacker/sniffer monitoring the operating system activities, such as key logger etc., then such an operating system is found to be compromised.

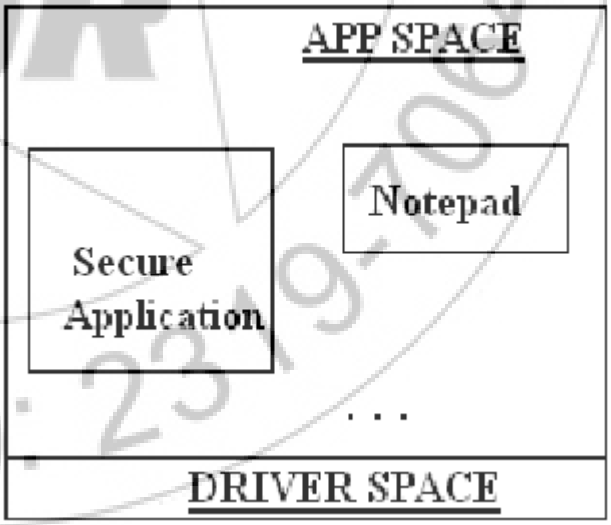

Figure 3: Applications running on a compromised OS

Hence we propose forth the prototype of a secure environment, which contains these attacks. As shown in the figure, the environment in question resides on the operating system itself, but its vulnerability to attacks is contained compared to the host the operating system.

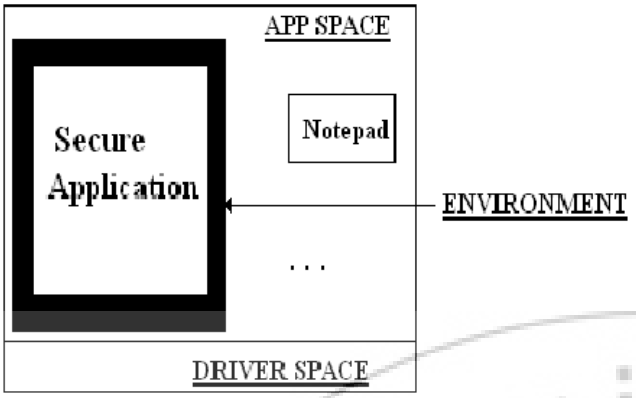

Figure 4: Environment in Application Space

The environment in question will secure the user's data from outside threat, as well as attacks on the system itself. To start with, we create authentication levels on the user's system by having two different users - general and special. The general user may be any user that can access all ordinary applications such as notepad or paint. The special user has access to these, and the secure applications which provide security to private data. On a higher level, we might try to block unwanted applications that would probably run in the background and watch the applications that are currently running in the system, in which case the system is still compromised. For blocking such applications, we prepare a whitelist of applications and all applications that are not in this list are blocked from running and stopped if already running.

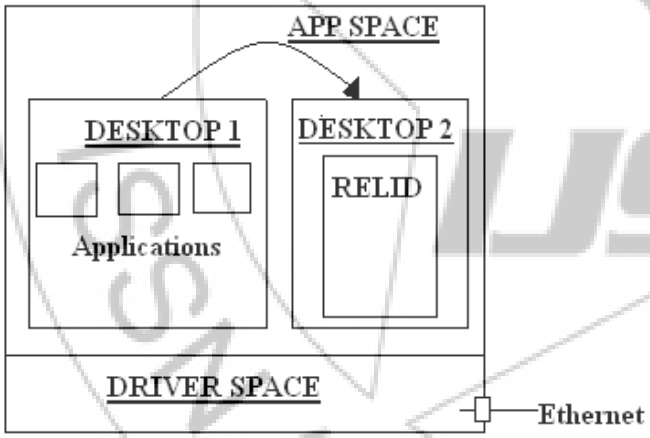

Figure 5: Creation of new Desktop

On a still higher level, we run the desired application on a second, new desktop which has its own separate memory space in the RAM. Let us consider an example. Suppose there is an application which is watching the internal structure of the RAM. It is running without the knowledge of the user and hence will retrieve all data that user enters. It is commonly known that when a new desktop is created by an application, it is assigned a different space on the RAM, thus, limiting access to other applications/services in another desktop environment. According to our research, this is the basic level of security that can be provided.

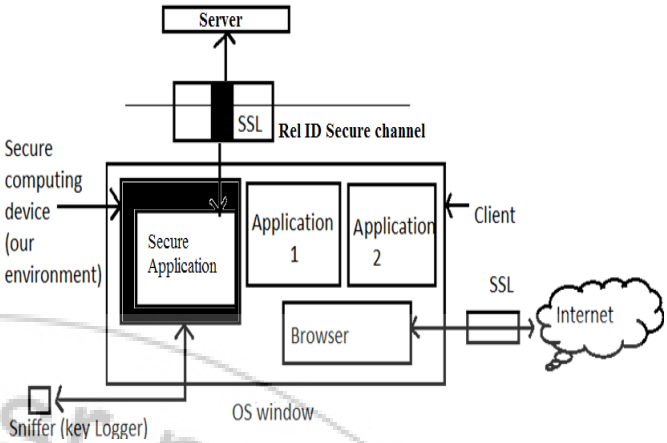

Figure 6: Block Diagram of Proposed System

Thus, as we move further, we will have obtained a secure environment that attempts to provide similar, but a much higher level of security to users data and also avoid maximum threats to the system itself. This paves path for further measures to decrease the vulnerability of user applications to external and internal threats.

4. Conclusion

Finally we conclude that, there is a necessity to have a secure environment over a compromised OS and hence this is the idea we are putting forth to build a secure environment in a compromised operating system, which provides a secured and trusted workspace for handling user data.

5. Future Scope

There are always people who keep trying to find new ways to hack into a system. So even if our system is secure for the time being it may not remain so forever. So we have to keep updating the proposed system.

References

[1] Chunxiao Li, Anand Raghunathan, and Niraj K. Jha, "A Trusted Virtual Machine in an Untrusted Management Environment," IEEE Transactions on services computing, October - December 2012, Vol. 5, No. 4.
[2] Uniken Systems Pvt Ltd, "http://www.uniken.com/rel-id-platform"
[3] C. Li, A. Raghunathan, and N.K. Jha, "Secure Virtual Machine Execution under an Untrusted Management OS," Proc. Int'l Conf. Cloud Computing, pp. 172-180, July 2010. 482 IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 4, OCTOBER-DECEMBER 2012.
[4] P. Barham et al., "Xen and the Art of Virtualization," Proc. ACM Symposium on Operating Systems, Bolton Landing, NY, October 19–22, 2003.
[5] Amit Vasudevan, Sagar Chaki, Limin Jia, Jonathan McCune, James Newsome and Anupam Datta, "Design, Implementation and Verification of an eXtensible and Modular Hypervisor Framework", 2013 IEEE Symposium on Security and Privacy.

[6] R. Figueiredo, P.A. Dinda, and J. Fortes, "Resource Virtualization Renaissance," IEEE Internet Computing, May 2005, Vol. 38, No. 5.
[7] T. Garfinkel and M. Rosenblum, "When Virtual Is Harder Than Real: Security Challenges in Virtual Machine Based Computing Environments," Proc. Conf. Hot Topics in Operating Systems, pp. 20-25, June 2005.
[8] R.P. Goldberg, "Survey of Virtual Machine Research," IEEE Computer, June 1974, pp.34–45.
[9] G.J. Popek and R.P. Goldberg, "Formal Requirements for Virtualizable Third-Generation Architectures," Comm. ACM, July 1974, pp. 412–421.
[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," 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing.
[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.