ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

Kernel Based Process Level Authentication Framework for Secure Computing and High Level System Assurance

Pradnya Patil, Shubham Joshi

Research Scholar, Dept. of C.S.E., D.P.C.O.E, Savitribai Phule Pune University, Pune, Maharashtra, India.

Research Supervisor, Dept. of C.S.E., D.P.C.O.E, Savitribai Phule Pune University, Pune, Maharashtra, India.

ABSTRACTIn modern operating system kernels level security is not present and a well-known approach to protecting systems from malicious activity is through the deployment of Mandatory Access Control (MAC). Existing MAC solutions belongs to authorization mechanism however authorization mechanism along is not sufficient for achieving system assurance. Today's modern computing era operating system Kernel should have process level authentication mechanism, where process of user level application proves its identity to kernel. Current process authentication is done using information such as process names or an executable path that is conventionally used by OS to identify a process is not reliable. This may results as malware may impersonate to other processes thus violating of system assurance can occur. We propose a lightweight secure application authentication framework in which user-level applications are required to present proofs at runtime to be authenticated to kernel. In order to demonstrate the application of Process Authentication proposed System Call monitoring framework for preventing unauthorized use or access of system resources like HDD, RAM. It verified the identity of processes before completing the requested System calls.

KEYWORDS: Operating System Security, Process authentication, Secure Computing, System calls monitoring.

I. INTRODUCTION

Today's era we are heavily rely on mission critical high computing machine to get most of our day to day online services and facilities. Hence all of these mission critical computing machines are very critical and organization doesn't expect downtime of those systems due to virus attacks and hacking of those systems. High assurance systems are now in demand and everybody wants extra security on top of general Antivirus systems available in the market. These days' hackers and viruses coming on internet are too smart hence mission critical systems only having an antivirus are not sufficient. They want extra security at execution level to avoid any virus attack and system downtime. Now day's typical operating system kernels enforce minimal restrictions on the applications permitted to execute, resulting in the ability of malicious programs to abuse system resources. Malware running as stand-alone processes, once installed, may freely execute privileges provided to the user account running the process. Hence Operating System level secure computing is now playing critical role for high assurance systems. On top of antivirus solutions to give more security, A well-known approach to protecting systems from malicious activities is through the deployment of mandatory access control (MAC). Such systems provide the kernel with access monitoring mechanisms as well as policy specification platforms. The user decides on the policies and the various access rights on system resources. Existing MAC systems such as SELinux, grsecurity and AppArmor.Enable the user (or the system administrator) to express detailed and powerful policies. They can be implemented using the Linux Security Modules to monitor access to selected system resources, and apply the specified policies to the corresponding processes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

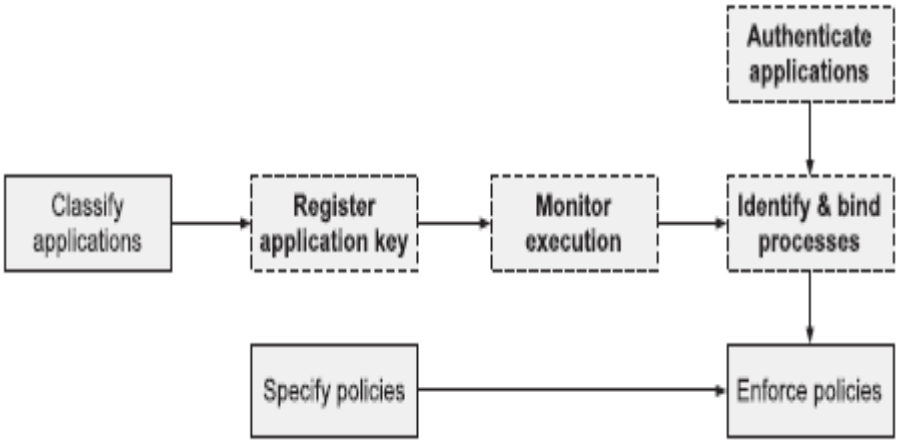

Figure1.Secure computing policies.et.al [2]

II. LITERATURE SURVEY

High system assurance with process level authentication and secure computing has received considerable attention in current years as promising approaches for securing mission critical high computing machines.On top of antivirus solutions to give more security, a well-known approach to protecting systems from malicious activities is through the deployment of mandatory access control (MAC). Such systems provide the kernel with access monitoring mechanisms as well as policy specification platforms.

H.M.J. Almohri et al. [1] proposed the concept of Identifying Native Applications with High Assurance. Proposed concept provides we address the identification problem by proposing a novel secure application identification model in which user-level applications are required to present identification proofs at run time to be authenticated to the kernel. The secret key of an application is registered with a trusted kernel at the installation time and is used to uniquely authenticate the application.

P. Loscocco et al. [2], proposed the Integrating Flexible Support for Security Policies into the Linux Operating System. Proposed concept gives approach to analyze and compare quality of protection offered by different MAC systems .Approach introduced rge nation of vulnerability surface under attack as measurement of protection quality and implement a tool called VulSAN for computing such vulnerability surfaces.

C. Wright et al. [5] have proposed Linux Security Modules (LSM). Computer security is a chronic and growing problem, even for Linux, as evidenced by the seemingly endless stream of software security vulnerabilities. Security research has produced numerous access control mechanisms that help improve system security; however, there is little consensus on the best solution. Many powerful security systems have been implemented as research prototypes or highly specialized products, leaving systems operators with a difficult challenge: how to utilize these advanced features, without having to throw away their existing systems? The Linux Security Modules (LSM) project addresses this problem by providing the Linux kernel with a general purpose framework for access control's enables loading enhanced security policies as kernel modules. By providing Linux with a standard API for policy enforcement modules, the LSM project hopes to enable widespread deployment of security hardened systems.

According to W. Dai [7], Digital signatures are an important mechanism for ensuring data trustworthiness via source authenticity, integrity, and source nonrepudiation. However, their trustworthiness guarantee can be subverted in the real

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

world by sophisticated attacks, which can obtain cryptographically legitimate digital signatures without actually compromising the private signing key. This problem cannot be adequately addressed by a purely cryptographic approach, by the revocation mechanism of Public Key Infrastructure (PKI) because it may take a long time to detect the compromise, or by using tamper-resistant hardware because the attacker does not need to compromise the hardware.
The concept of 'Runtime Execution Monitoring (REM)' is given by A.M. Fiskiran and R.B. Lee et al. [9] and according to him many computer security threats involve execution of unauthorized foreign code on the victim computer. Viruses, network and email worms, Trojan horses, backdoor programs used in denial of service attacks are a few examples. Proposed architectural technique, which we call runtime execution monitoring (REM), to detect program flow anomalies associated with such malicious code.

III. RELATED WORK

Major problem is typical Operating systems kernel doesn't enforce more restriction on the applications before execution and resulting in the ability of the malicious program to abuse system resources. Malware running as standalone processes, once installed, may freely execute and damage mission critical systems. Process authentication is different from process identification. However, the information such as process names or executable paths that is conventionally used by OS to identify a process is not reliable. Existing solutions are not up to the mark to give secure computing and high assurance.

Main goal is to publish solution works closely with Operating system kernel to assure that any unauthenticated process will not work. In the proposed solution different modules takes care of any process from its installed in the system to its execution and continue watch the behavior of process. It also prevent unwanted system call generation to access any hardware resources.

IV. RESEARCH PERSPECTIVE

In secure computing main focus on the process authentication before execution. This problem can be solved by using authenticate legacy applications using a helper program to the Verifier. To authenticate a newly started process the Authenticator checks if the process has already been verified by looking its process id up in the status list. If process id belongs to status list, the Authenticator sends to the Verifier. We describe the general operations needed for process authentication solutions, including credential generation, process authentication, and runtime monitoring. Proposed solution works closely with Operating system kernel to assure that any unauthenticated process will not work.

V. PROPOSED WORK

The proposed model provides a component approach to achieve secure computing and high system assurance through kernel level processauthentication. The model has different component's to generate secrete key for each process and modules that can validate a key before process execution. In proposed approach has component that issues a secrete key for every created process and authenticator component is authenticate on first time creation of the process. Two different key list are maintained in the system for credentials and status lists.Proposed solution will create secret credentials for newly created and already created processes. Expected user that will manage proposed solution is expected to be root user only.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

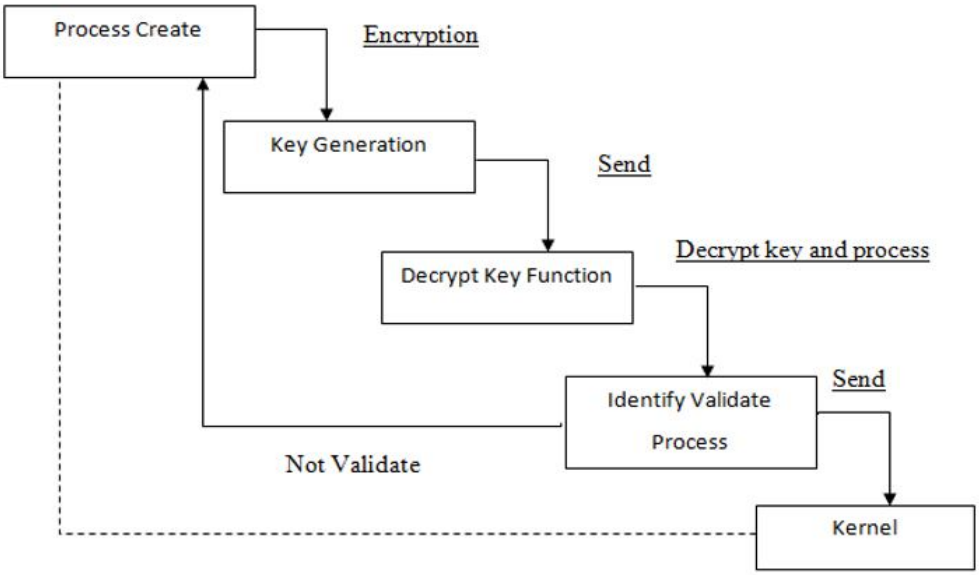

Figure2. System Flow Diagram

VI. SCOPE OF WORK

Proposed concept can be used in Linux kernel to do a process level authentication for every process before executing it. In the proposed concept also proposed authentication protocol also helps us to securely authenticate with process and record or maintain authentication status of the process. Secure computing can be achieved with proposed approach for Linux as well on Windows OS. Through adoption of this approach helps to keep mission critical systems with zero downtime.

VII. DICUSSION AND FUTURE WORK

Above discussed approaches helps to provide high security with secure computing to achieve high system assurance. In future, proposed model can be used with SE Linux models to leverage more security to the mission critical systems. Also proposed approach can be used in Android operating system for mobile devices to support the process level authentication for mobile applications. The consumption of this model and protocols can make more secure computing OS.

REFERENCES

1. H.M.J. Almohri, D. Yao, and D. Kafura, "Identifying Native Applications with High Assurance," Proc. ACM Conf. Data and Application Security and Privacy (CODASPY '12), Feb. 2012.
2. Hussain M.J. Almohri,Danfeng (Daphne) Yao, and Dennis Kafura "Process Authentication for High System Assurance" IEEE Trans on. Dependable and Secure Computing, vol. 11, no. 2, MARCH/APRIL 2014
3. P. Loscocco and S. Smalley, "Integrating Flexible Support for Security Policies into the Linux Operating System," Proc. USENIX Ann. Technical Conf., 2001.
4. "grsecurity," http://www.grsecurity.net/, 2013.
5. Z.M.H. Chen and N. Li, "Analyzing and Comparing the Protection Quality of Security Enhanced Operating Systems,"Proc. 16th Ann. Network and Distributed System Security Symp. 2009.
6. C. Wright, C. Cowan, S. Smalley, J. Morris, and G. Kroah-Hartman, "Linux Security Module Framework," Proc. 11th Ottawa Linux Symp., 2002.
7. K. Xu, H. Xiong, D. Stefan, C. Wu, and D. Yao, "Data-Provenance Verification for Secure Hosts," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 173-183, Mar./Apr. 2012.
8. W. Dai, T.P. Parker, H. Jin, and S. Xu, "Enhancing Data Trustworthiness via Assured Digital Signing," IEEE Trans.Dependable and Secure Computing, vol. 9, no. 6, pp. 838-851, Nov./Dec. 2012.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

9. G. Xu, C. Borcea, and L. Iftode, "Satem: Trusted Service Code Execution across Transactions," Proc. IEEE 25th Symp. Reliable Distributed Systems (SRDS '06), pp. 321-336, 2006.
10. A.M. Fiskiran and R.B. Lee, "Runtime Execution Monitoring (REM) to Detect and Prevent Malicious Code Execution," Proc. IEEE Int'l Conf. Computer Design: VLSI in Computers and Processors (ICCD '04), pp. 452-457, 2004.
11. T. Jaeger and R. Sandhu, Operating System Security. Morgan & Claypool, 2008.
12. K. Xu, P. Butler, S. Saha, and D. Yao, "DNS for Massive-Scale Command and Control," IEEE Trans. Dependable and Secure Computing, vol. 10, no. 3, pp. 143-153, May/June 2013.
13. X. Shu and D. Yao, "Data-Leak Detection as a Service," Proc. Eighth Int'l Conf. Security and Privacy in Communication Networks (SECURECOMM '12), Sept. 2012.
14. K. Xu, D. Yao, Q. Ma, and A. Crowell, "Detecting Infection Onset with Behavior-Based Policies," Proc. Fifth

BIOGRAPHY

Pradnya Kerba Patil has done B.E in Information Technology in year May-2011 from T.K.I.E.T Warnanager, Kolhapur. She is perusing M.E Computer Engineering from Savitribai Phule Pune University, Pune India. Her area of research is High Level System Assurance, etc.