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. 3, Issue 6, June 2015

Design and Implementation of Kernel Based Process Validation for High Level System Assurance

Pradnya Patil

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

ABSTRACT: Now in today's modern operating system kernels level security is not present and a well-known approach to secure systems from malicious activity is through the deployment of Firewall or Anti-Virus .Existing solutions belongs to authorization mechanism however authorization mechanism along is not sufficient for achieving high level system assurance. Today's modern computing era operating system Kernel should have process level validation mechanism, where process of user level application proves its identity to kernel. Current process validation 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 in malware may impersonate to other processes thus violating of system assurance can occur if we don't prevent untrusted process execution. In our proposed an Enhanced Process Level Validation system where user-level applications are required to present issued secrete credentials as proofs at runtime to be validated to kernel. The enhanced process validation system will not allow untrusted processes or code to execute and kernel level validation is provided. We believe that this will be second level measure for secure computing.

KEYWORDS: Operating System Security, Process Validation, Secure Computing, Secrete Credentials.

I. INTRODUCTION

In Today's changing computing requirements in every field we are heavily relying on mission critical high computing machine to get most of our day to day online services and facilities. As a result all of these mission critical computing machines are very critical and organization or companies doesn't expect downtime of those systems due to virus attacks and hacking of those systems. The modern operating system major problem is kernel do not enforce more restriction on application before execution or making request for making system call to access resources.High level assurance systems are now in demand and everybody wants extra security on top of general solutions available in the market. Coming days' hackers and viruses those are coming on internet and on the systems are too smart and that's reason mission critical systems only having an antivirus or firewall are not sufficient. In most of the cases it's been observed that malware names are the same as the name of valid process and they impersonate to other processes. In this use case present antivirus solutions and firewall are unable to catch these threats and they abuse the system resources that's cause violating the system resources. So now day's users want extra process level security at execution level to avoid any virus attack at execution level so that maximum system assurance can be expected.In many cases Malware running as stand-alone processes, once installed, they may freely execute privileges provided to the user account running the process and may damage system level assurance. Hence Operating System process level secure computing is now playing critical role for high assurance systems.

DOI: 10.15680/ijircce.2015.0306172

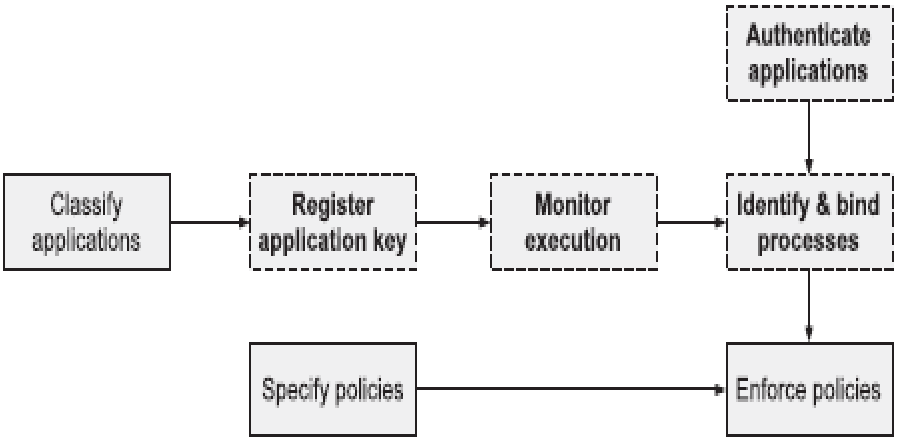


Figure1.Secure computing policies.et.al [2]

II. RESEARCH PERSPECTIVE

In research area Dependable and Secure Computing main focus is on Dependability aspect and Dependability ability to provide service that can justifiably trusted. This means that to do a secure computing need a trusted factor where we can do computing. In this dissertation we are proposing a system that will do process level validation before execution. In the proposed research trusted component is considered as operating system Kernel and validation is based on the secrete credentials issued by kernel to every process. For every process once considered as trusted then it will generate trusted credentials for the same and one copy of credentials is stored with kernel. When process execution gets triggered then both credentials i.e. with process and with kernel gets compared. If both credentials matched then conclude as its validated process and allowed for execution.

III. ACTUAL IMPLEMENTATION

In the implanted system provides a component approach to achieve secure computing and high system assurance through kernel level process validation. This system has Credential Registar component's to generate secrete key for each process and Process Authenticatorvalidate secrete credential issues for very process a key before process execution. All the generated secrete credentials are stored into Credential List for the reference of kernel. In every process execution triggers system gets credentials from the respective process and also gets credentials from kernel and do compare the both. If both of secrete credentials gets matched then process validation system conclude as validated or trusted process and allowed for the execution. If credentials does not match then considered as untrusted process and execution is blocked by the system. This is how process validation system ensures system assurance of the system through process level validation. Below are main operation does by the Enhanced Process Validation system:

- **Application Classification**
  When user installs a new application on Windows operating system then Application Classification done based on initial assessment of executable. This operating conclude that whether user trying application is malicious or not. This module reports that application is trusted based on initial assessment then it will be allowed for the installation otherwise it will not be allowed for installation. This first level security provided by the system.

- **Secrete Credential Generation**
  Once Application Classifier module conclude as this is trusted application then while installation Credential Registar module generates a secrete credentials. The secrete credentials issued with help of operating system kernel module and its gives a pseudo random number to the process.

- **Formations of Capsule of Secrete Credentials**
  Code capsule is created for each trusted process with combination of process name and generated secrete credentials Code Capsule is encrypted

- **Maintain List of all process credentials**
  For all the trusted processes code capsules are maintained into Credentials List and which is stored at secure place as kernel data structure. This will be system file which is will be placed at system files location i.e. C:\Windows\System32 folder.

- **Process Validation**
  Before every process execution Process Authenticator (PA) validate process secrete credentials. PA will first demand secrete credentials to process and takes that credentials. In parallel PA gets credentials from Credentials list which is maintained by the system and compare two credentials.

- **Runtime Monitoring**
  Once process gets validated as trusted process then process name is maintained in Status List where list of all validated and trusted process names are listed. When already validated and trusted process validation will not do again and system checks only process name is available in the Status List.

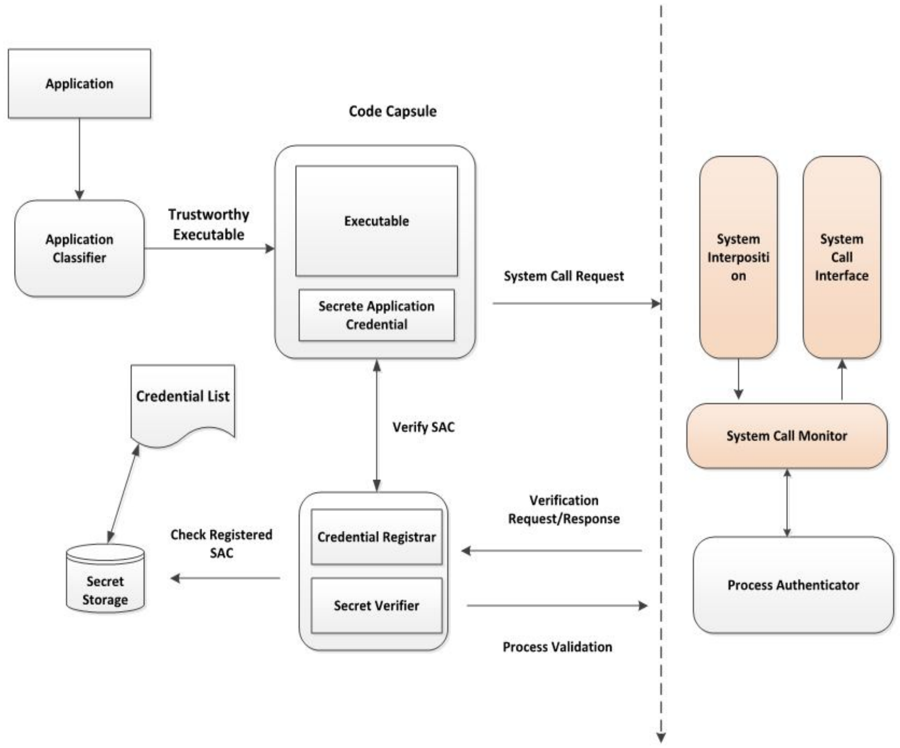
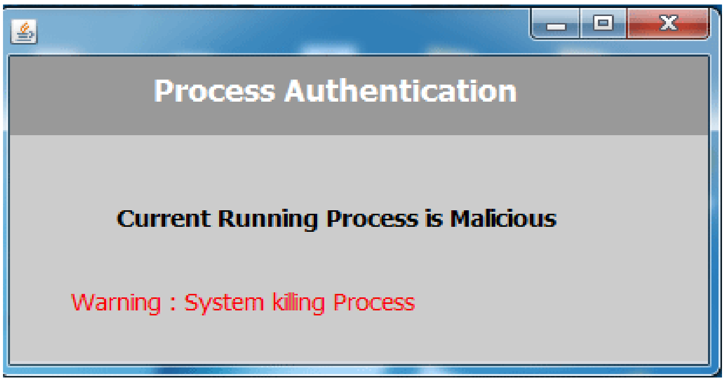


Figure2. Proposed System Architecture

DOI: 10.15680/ijircce.2015.0306172

IV. EXPERIMENAL RESULTS

Below are results of actual implementation system. The application which concludes as malicious or untrusted for those applications will be not allowed for the installation. At initial phase application classification is based on assessment of installation executable of the application. Module checks that whether application has digital signature and as of trusted authority, does application has a proper version and publisher information.



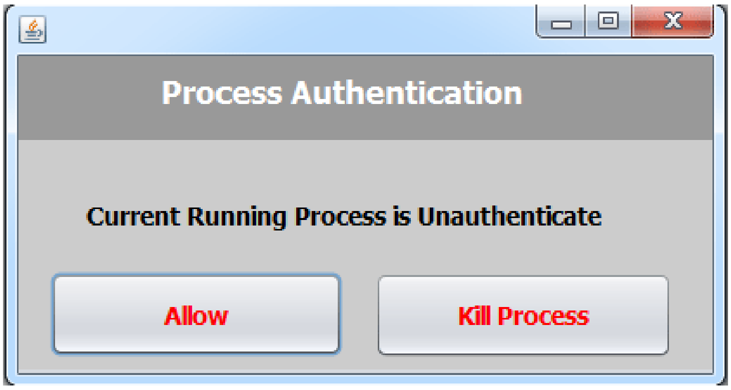
Process Authentication

Current Running Process is Malicious

Warning : System killing Process

Figure 3: Process Validation System observes Application is malicious

In other case if already installation application process validation fails in case where secrete credentials are not matching in that case system reports to the user as this process is not trusted and its malicious. Takes preferences from the user regarding the user still want to execute or kill that process.



Process Authentication

Current Running Process is Unauthenticate

Figure 4: Process Validation System observes running process is malicious

Below graph shows after deployment of system on Windows machine the system performance is not degraded as shown in below graph. Some decrease in performance is observed but it is minimal as compared with other antivirus solution.

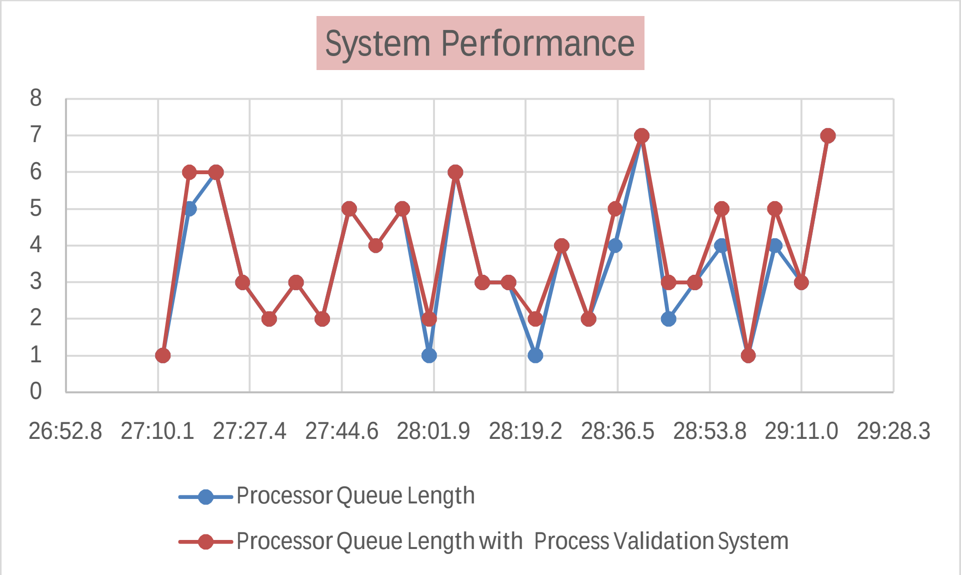


Figure4: System performance results with Process Validation system

V. CONCLUSION

In this paper the system implemented work done as much as possible using self-designed process validation techniques; in this system we have shown totally different work than existing research of providing high level system assurance through secure computing. A lot of research is possible in this area where a secure computing using dependability aspect. Anyone can do number of variation in this system in future. In this paper we worked only on kernel based process validation as dependability aspect. With the implemented system we can assure that addition level of secure computing will provide user a high level of system assurance.

VI. DICUSSION AND FUTURE WORK

In above implemented system approaches and helps to provide high security with secure computing to achieve high level system assurance. In future, proposed system can be implemented and extend to the mobile operating systems like Android since its most widely used smart phone operating system. As growing use of Android the threats also increased. This system also extended with using Firewall that are based on some policies.

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.

DOI: 10.15680/ijircce.2015.0306172

[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.
[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

DOI: 10.15680/ijircce.2015.0306172