Network Security Using Job Oriented Architecture (SUJOA)

Tariq Ahamad[1], Abdullah Aljumah[2]

College Of Computer Engineering & Sciences
Salman Bin Abdulaziz University, KSA

ABSTRACT

In the modern world operating system, various security systems (Collection of various security components) are already installed or can be directly installed in it. They are responsible for checking the complete node for suspicious behaviour. There are some intrusions having the ability to hide themselves from being checked called armouring intrusions. In this research article we present alternative organisation of security systems. To distinguish operating system applications and security systems, the node is completely virtualized with current virtualized systems. The node is then checked by security systems from outside and the right security components are provided through job oriented architecture. Since they run on a virtual machine the infected nodes can be halted, duplicated and moved to other nodes for further analysis and legal aspects. The coordinated architecture analysed in this research article and the results of a preliminary implementation with positive results are discussed.

Key Words: ***Network Security, Network Security Architecture.***

INTRODUCTION

A lot of network security systems protect a computer network against various types of network or electronical attacks. The attacks are e.g. worms and viruses but also hacker or internal attacks performed by the normal users of the network. The security systems are a collection of various security components as antivirus software, firewalls, and intrusion detection systems [5] and in the modern world operating systems these security components are directly installed in and monitor the nodes for suspicious behaviour. The components lack from cooperative workflows in order to correlates events for abnormal behaviour detection [2, even redundant checks lead to an increase in the required resources. This organisation of security components is a major weakness especially in coping with upcoming intrusions [6]. E.g. the Bradley virus is computational hard to detect when such an organisation is used [1]. We concern with the organisation of security components and introduce a more sophisticated way. The whole node is virtualized using an virtualization system as VMware. The operating and security systems run in a node in different virtual machines so that the security components check the node from outside. The features of current virtualization systems are used with the ability to halt, duplicate, and move virtual machines. This leads to a different handling of infections because the infected virtual machine is duplicated and saved for further analysis and legal aspects. Job oriented architecture (JOA) provides on demand the right security components, which is implemented through the exchange or adding of virtual machines. We concern the advantages and disadvantages of such an environment and conclude with the current project status.

ARCHITECTURE

Architecture for each node is organised in four layers: the first layer is the hardware and the second layer is a core operating system providing the kernel to access the hardware and the virtualization system to run several virtual machines simultaneously[3][5]. The third layer contains the virtual machines with different operating systems and one virtual machine for security. This implements the security environment, which ensures the access to all other virtual machines for scanning purposes. The fourth layer is the application layer with user's applications and installed security components, which are connected to the security environment[4]. In the network, one or more security servers exist, which provide the right virtual machine with installed security environment and security components. This server provides the virtual machine to new nodes and ensures that each node is properly secured. Furthermore, it contains more analysis systems to scan virtual machines deeply. With this implementation, the security components are seen as different services provided by the security server and distributed to the node where they are required. This is expanded to a service oriented architecture where the right security components are provided on demand according to the current situation in the network [6]. Especially with the novel more and more different node types as mobile handhelds connected to the network and thin clients, the required security components in a node has to be adapted due to different available resources and security issues. With the introduced architecture, the security components can be easily adapted to the security requirements of the node. The maintenance of security systems changes according to the architecture. If a new node connects [4][6] to the network, the security server checks the node if the right architecture is installed and provides an virtual machine with the security environment and installed security components. Through integrating this workflow in the DHCP, a network is properly secured because only when the node has a running security system it receives access to the network. New security components or changes in the required components in a node are quickly resolved: the security server provides a new virtual machine that exchanges the current machines on the node [14].

IMPLEMENTATION ISSUES

The implementation of the architecture is feasible. Current virtualization systems as VMware or KVM provide most of the required features. VMware also provides main boards where the layer one and two is directly installed. The only missing feature is the ability that the security components of the security environment are able to access the other virtual machines for scanning purposes. However, this can be implemented through an extension of the virtualization system [7][10]. With the right implementation, well known security components - e.g. antivirus software, firewall, and intrusion detection system - are facilitated. These are installed in the security environment and a guard measures the required data from the operating system and presents it to the security component. This analyses the data and defines the response, which is executed by the guard accordingly. Consequently, all existing security components are reusable [17].This architecture provides an improvement in the implementation of security components. These are platform independent running in the security environment and must consequently not adapted when the used operating system or used hardware platform changes. In addition, the security environment gathers the data to analyse and perform the response accordingly, which must not be implemented in the security component. This leads to a faster deployment of novel approaches.

The project status is that the architecture of the nodes and of the network is designed and theoretically analysed. Different proof-of-concept implementations of the virtualization system have been realised to analyse the features of these. Various parts of the implementation are still missing due to the early stage of the project. The preliminary results are discussed in the next section [8].

PRELIMINARY RESULTS

The first results are promising. The implementation of the node is feasible where the only hard task is to ensure that the security components are able to access the operating system. The installation of the security components in the security environment installed in a virtual machine is also challenging because this influences the performance and the security of the security system[9]. Security issues of this organisation are analysed and they are solvable when up-to-date approaches from cryptography are used. Especially the more and more emerging integrity checks and the complete installation of a distributed public/private key infrastructure are important. With this, an adversarial is not able to use the distributed security system for attacking the network [11].

A. Infection Handling

One main advantage of the proposed architecture is the infection handling. If some security component identifies a virtual machine of a node as infected, the following workflow is processed: the security environment of the node halts the infected virtual machine to prevent propagation [12]. It duplicates the virtual machine and sends this to the security server to analyse it more deeply and to save the evidences for legal aspects - this is a weakness in current systems: either the node is disinfected and the evidences are destroyed or the evidences are saved but the node is still infected. Afterwards, the security server provides a clean virtual machine with the desired applications installed in order to limit the downtime of the node [17].

B. Security Issues

The proposed architecture is used to implement a distributed security system with integrated components. The security components are able to roam through the network and cooperative workflows enable the detection of unforeseen intrusions. This has a major drawback that adversaries may use the system to attack a network, i.e. to propagate intrusions through it. This must be prevented through the design of the architecture and is discussed now [13]. The layer one containing the hardware is not an aim of adversaries. Layer two with the core operating and virtualization system is furthermore highly dependent on the facilitated hardware and changes therefore only when the hardware changes. This is ensured through public/private key signatures used in cryptography (this is also integrity checking called). When an adversarial installs an intrusion in this layer, it is immediately recognised and prevented [14]. Layer three and four contains the operating system with application of the user, which are protected as the operating system in nowadays implementations. The virtual machine containing the security environment and the security components is additionally secured: the security environment does not change and is therefore protected using a cryptographic signature with integrity checking [15].The security components access resources of the node. These are secured using public/private keys organised in a distributed public key infrastructure. Only when the security components have the right keys, they receive access to the resources where security components initiated by the adversarial does not have these. To summarise the security issues, the adversarial is still able to install intrusions in the operating system and in the security system when the implementation provides bugs or when the adversarial knows internal knowledge.

CONCLUSION

In this research article we discussed how to increase the performance of a network security system by features of virtualized system with job oriented architecture. The advantages especially help to identify novel more and more intelligent intrusions and provides a more sophisticated infection handling. Furthermore, the article faces several unsolved problems, which are of interested for novel network security systems. The next step in the project is to implement a prototype of the architecture and to build up a testbed with some nodes to simulate the whole workflow. For this, a VMware implementation is first desired due to the reduced time for setting up all of these nodes. The architecture is also usable in current approaches of network security systems facilitating artificial immune systems , multi-agent systems, and distributed systems to distinguish the normal operating system and the security system on each node .

References

[1] A. Aho, R. Sethi, J. Ullman, “Compilers, Principles, Techniques, and Tools”. Addison-wesley Publishing Company.

[2] G. Ammons, J. Larus. “Improving Data-flow Analysis with Path Profiles”. In the Proceeding of the 1998 ACM SIGPLAN Conference on Programming Language Design and Implementation, Montreal Canada, June 17-19, 1998.

[3] T. Ball, J. Larus. “Efficient Path Profiling”, In the proceeding of MICRO-29, December 2-4, 1996, Paris, France.

[4] T. Ball, J. R. Larus. “Optimally Profiling and Tracing Programs”. ACM Transactions on Programming Languages and Systems, Vol 16, No. 4, July 1994, pp1319-1360.

[5] S. Berkovits, J. Guttman, V. Swarup. “Authentication for Mobile Agents”, in: Giovanni Vigna (Ed.): Mobile Agents and Security. pp 114-136. Springer-Verlag, 1998.

[6] C. Cifuentes. “Structuring Decompiled Graphs. Personal Communication. Proceedings of the

International Conference on Compiler Construction (CC'96), Lecture Notes in Computer Science 1060. Linkoping, Sweden. 22-26 April 1996, pp 91-105.

[7] C Cifuentes, M Van Emmerik, and N. Ramsey, *The Design of a Resourceable and Retargetable Binary Translator*. Proceedings of the Sixth Working Conference on Reverse Engineering, Atlanta, USA, October 1999, IEEE-CS Press, pp 280-291.

[8] Cloakware Systems. “Building Cloakware”. Five minute presentation at the 2000 IEEE Symposium of Security and Privacy. May, 2000. Berkeley, California.

[9] D. Chess. “Security issues in mobile code systems”. in: Giovanni Vigna (Ed.): Mobile Agents and Security. pp 1-14. Springer-Verlag, 1998.

[10] S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, S. Staniford-Chen, R. Yip, D. Zerkle. GrIDS: “*A Graph-Based Intrusion Detection System”*. National Information System and Security Conference, Baltimore, 1997.

[11] T. Cormen, C. E. Leiserson, R. Rivest, “Introduction to Algorithms”. The MIT Press, 1993. Tenth edition.

[12] M. Elder, J. Knight, “Security Attacks on Critical Infrastructure Systems”. Computer Science Technical Report. CS-98-23.

[13] M. Hennessey, J. Riely, “Type Safe Execution of Mobile Agents in Anonymous Networks”, in: Jan Vitek; Christian Jensen (Eds.): Secure Internet Programming, LNCS 1603, Springer-Verlag, pp. 95-116, 1999.

[14] F. Hohl. “Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts”. In Lecture Notes in Computer Science, vol. 1419, Mobile Agents and Security. Edited by G. Vigna. Springer-Verlag, 1998.

[15] J. Knight, K. Sullivan, M. Elder, C. Wang. “Survivability Architectures: Issues and Approaches” In Proceedings: DARPA Information Survivability Conference and Exposition. IEEE Computer Society Press. Los Alamitos, CA, January 2000, pp. 157-171.

[16] B. Yee. “A Sanctuary for Mobile Agents”. Technical Report CS97-537. Computer Science Department, University of California in San Diego, USA.

[17] http://gita.state.az.us/enterprise_architecture/NEW/Security_Arch/