ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 1, January 2013

Attack Surfaces as a Metric for Version Change

Sukanth Sistla
Department of CSE, JNTUA College of Engineering, Anantapur, India

Abstract - In software metrics, developers increasingly focus on counting bugs at code level are no of vulnerabilities encountered at system level generally but we focus on surface attack as a measure to differentiate a version of two compare- able systems to justify which is better than other. We measure a system's attack surface in terms of three kinds of resources used in attacks on the system. The attack surface metric is validated by conducting a user survey. Our measure can be used as a parameter by software developers in the development process and consumers in their decision making process.

***Keywords*- Attack Surface, Attack Surface Metric, Parameter Sensitivity, User Survey.**

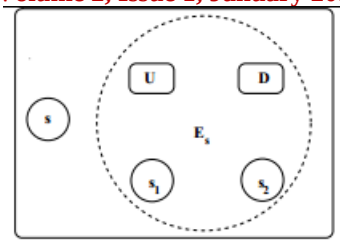
I. INTRODUCTION

Software industry has responded to demands for improvement in software security by increasing effort for creating "more secure" products and services. A system's attack surface [1] is the subset of the system's resources that an attacker can use to cause damage to the system. We introduce an entry point and exit point framework to identify these relevant resources. Moreover, not all resources contribute equally to the measure of a system's attack surface. A resource's contribution to the attack surface reflects the likelihood of the resource being used in attacks in this paper; we propose a metric to determine whether one version of a system is more secure than another. Rather than measure the absolute security of a system, we measure its relative security: Given two versions, A and B, of a system, we measure whether version A is more secure than version B with respect to their attack surface. We do not use the attack surface metric to determine whether a version of a system is absolutely good or bad, rather to determine whether one version of a system is relatively better or worse than another. Intuitively, a system's attack surface is the ways in which the system will be successfully attacked. We define the attack surface of a system in terms of the system's resources. An attacker uses a system's resources to attack a system; hence a system's resources contribute to the system's attack surface. Intuitively, the more resources available to the attacker, the more exposed the attack surface. The more exposed the attack surface [2], the more ways the system can be attacked and hence the more insecure it is. Given two versions, A and B, of a system, we compare their attack surface to determine whether one is more secure than another. The attack surface measurements might be incomparable because of the way we define attack surface along multiple dimensions. We can, however, use the attack surface measurements along with the knowledge of the usage scenario of the system to determine whether version A is more secure than version B.

II. ATTACK SURFACE DEFINITION

We use the entry point [3] and exit point framework to identify the resources that are part of a system's attack surface. Informally, entry points of a system are the ways through which data "enters" into the system from its environment, and exit points are the ways through which data "exits" from the system to its environment. The entry points and exit points of a system act as the basis for attacks on the system. Our paper contains a formal description of the entry point and exit point framework [21].Consider a set, S, of systems, a user, U, and a data store, D. For a given system, s 2 S, we define its environment, Es = h U, D, Ti, to be a three- tuple where U is the user, D is the data store, and T = S \ {s}, is the set of systems excluding s. Every system in S has a set of methods. A method of a system receives arguments as input and returns results as output. Examples of methods are the API of a system. Every system also has a set of communication channels. The channels of a system s are the means by which the user U or any other system in the environment communicates with s. Examples of channels are TCP/UDP sockets, RPC end points, and named pipes. The user U and the data store D are global with respect to the systems in S. The data store is a collection of data items. Examples of data items are strings, URLs, files, and cookies. We model the data store D as a separate entity to allow sharing of data among all the systems in S. For simplicity, we assume only one user U is present in the environment. U represents the Adversary who attacks the systems in S.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 1, January 2013



A. Entry Points

The methods of a system that receive data items from the system's environment are the system' sentry points. A method of a system can receive data directly or indirectly from the environment [3].A method, m, of a system, s, receives a data items directly if either (a) the user U or a system, s1, in the environment invoke m and passes data items as input to m, or (b) m reads data items from the data store, or (c) m invokes the API of a system s1 in the environment and receives data items as results returned. A method is a direct entry point if it receives data items directly from the environment. Few examples of the direct entry points of a web server are the methods in the API of the web server, the methods of the web server that read configuration files from the file system, and the methods of the web server that invoke the API of an application server. A method, m, of s receives data items indirectly if either (a) a method, m1, of s receives a data item, d, directly, and either m1 passes d as input to m or m receives d as result returned from m1, or (b) a method, m2, of s receives a data item, d, indirectly, and either m2 passes d as input to m or m receives d as result returned from m2. A method is a indirect entry point if it receives data items indirectly from the environment. For example, a method in the API of the web server that receives login information [5] from a user might pass the information to another method in the authentication module; the method in the authentication module is an indirect entry point. The set of entry points of a system is the union of the set of direct entry points and the set of indirect entry points.

B. Exit Points

The methods of a system that send data items to the system's environment are the system's exit points. For example, a method [4] that writes into a log file is an exit point. A method of a system can send data directly or indirectly into the environment. A method, m, of a system, s, sends a data items directly if either (a) the user U or a system, s1, in the environment invoke m and receive data items as results returned from m, or (b) m writes data items to the data store, or (c) m invokes the API of a system s1 in the environment and passes data items as input to s1's API. A method m of s is a direct exit point if m sends data items directly to the environment. A method, m, of s sends data items indirectly if either (a) m passes a data item, d, as input to a method, m1, of s or m1 receives d as result returned from m , and m1 passes d directly to s's environment, or (b) m passes a data item, d, as input to a method, m2, of s or m2 receives d as result returned from m,and m2 passes d indirectly to s's environment. A method m of s is a indirect exit point if m sends data items indirectly to the environment. The set of exit points of a system is the union of the set of direct exit points and the set of indirect exit points invoking s's direct entry points (direct exit points). Since the direct entry points (direct exit points) of s act as a basis for attacks on s, we do not consider the transient data items as a basis for attacks on s.

C.Attack Surface

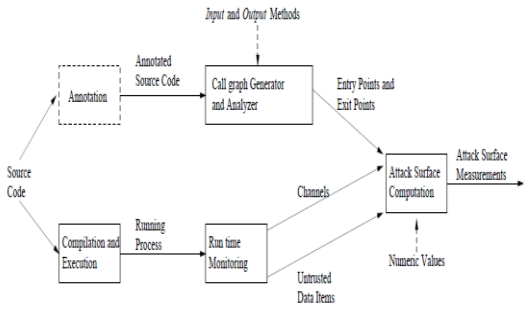


Figure 1: Attack surface measurement steps.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 1, January 2013

The set of entry points[4] and exit points, the set of channels, and the set of un trusted data items are the resources that the attacker can use to either send data into the system or receive data from the system and hence attack the system. Hence given a system, s, and its environment, Es, s' s attack surface[1] is the triple, h M,C, Ii, where M is the set of entry points and exit points, C is the set of channels, and I is the set of un trusted data items of s. Attack Surface measurement. A naive way of measuring a system's attack surface is to count the number of resources that contribute to the attack surface. This naive method is misleading as it assumes that all resources contribute equally to the attack surface. In real systems, not all resources contribute equally to the attack surface. For example, a method, m1, running as root is more likely to be used in an attack than a method, m2, and running as non-root; hence m1 contributes higher to the attack surface than m2. We estimate a resource's contribution to a system's attack surface as a damage potential-effort ratio [5] where damage potential is the level of harm the attacker can cause to the system in using the resource in an attack and effort is the amount of work done by the attacker to acquire the necessary access rights in order to be able to use the resource in an attack.

III. RESULTS AND CALCULATIONS

The surface metric calculation is as follows

- Each surface is sum of weights of each type of vector
- Total surface is sum of all the parameters used.
- I assume this is the RASQ (they don't make an explicit connection)
- Values of weights are not explained

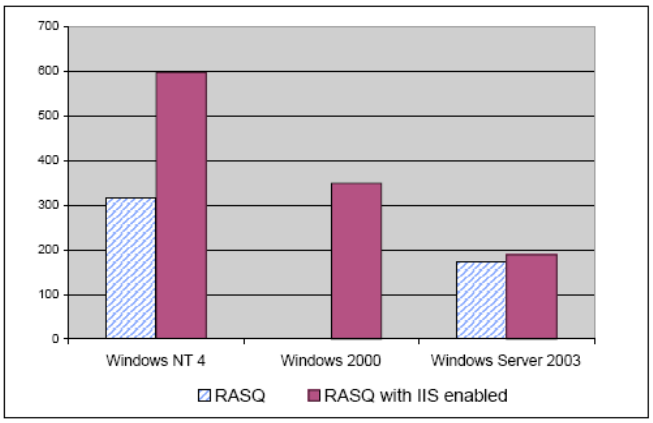


Fig 1 Attack Surface Measurement in Different Operating Systems

We analyzed the Likert scale responses using descriptive techniques [30]. We combined the strongly agree and the agree responses and the strongly disagree and the disagree responses, and computed the proportion of the subjects who agree (strongly or otherwise), disagree (strongly or otherwise), and are neutral with the steps in our measurement method. We performed one sample t-tests to determine the statistical significance of our findings. We chose a p-value of 0.05 as the threshold; findings with p-values less than 0.05 are statistically significant. We below summarize the findings of the survey.

1. Methods, channels, and data are the right dimensions of the attack surface.
2. A resource's damage potential-effort ratio is an indicator of the likelihood of the resource Being used in attacks.
3. A method's privilege is an indicator of damage potential and a method's access rights, a channel's access rights, and a data item's access rights are indicators of attacker effort.
4. The findings are not conclusive with respect to channel protocol and data item type.

IV. CONCLUSION

Software industry has responded to demands for improvement in software security by increasing effort for creating "more secure" products and services. In this paper, we propose to use the measure of a software system's attack surface as an indicator of the system's security. Finally, a system's attack surface is creating

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 1, January 2013

ways in which an adversary can enter the system and potentially cause damage. Hence the larger the attack surface, the more insecure the system.

REFERENCES

[1] J. Alves-Foss and S. Barbosa. Assessing computer security vulnerability. ACM SIGOPS Operating Systems Review, 29(3):3–13, 1995.

[2] S. M. Bellovin. On the brittleness of software and the infeasibility of security metrics. IEEE Security and Privacy, 04(4):96, 2006.

[3] P. K. Manadhata, J. M. Wing, M. A. Flynn, and M. A. McQueen. Measuring the attack surfaces of two FTP daemons. In ACM CCS Workshop on Quality of Protection, October 2006.

[4] P. H. Rossi, J. D. Wright, and A. B. Anderson, editors. Handbook of Survey Research. The Academic Press, New York, NY, USA, 1983.

[5] B. Schneier. Attack trees: Modeling security threats. Dr. Dobb's Journal, 1999.

AUTHOR BIOGRAPHY

Sukanth Sistla, sukanth.886@gmail.com, Department of CSE , JNTUA College of Engineering Anantapur, India has three publications

1. A Structural and Behavioral Analysis Approach for Process Model Evaluation, International Journal on Computer and communication Technology Vol.3 pp 81-87.
2. Consistency Evaluation Based On Consistency Evaluation and Structural Analysis. Special issue national Conference on Information and Computation Technology pp.177-179.JNU-New Delhi.
3. Process Model Consistency Measurement International Organization of Scientific Research-Journal on Computer science and Engineering Vol.7 pp.40-44