An Attack Surface Metric

Pratyusa K. Manadhata and Jeannette M. Wing
Carnegie Mellon University
{pratyus, wing}@cs.cmu.edu

1 Motivation and Goals

Measurement of security, both qualitatively and quantitatively, has been a long standing challenge to the research community, and is of practical import to software industry today. Software industry has responded to demands for improvement in software security by increasing effort into creating "more secure" products and services. How can industry determine if this effort is paying off and how can consumers determine if industry's effort has made a difference? Our work is motivated by the question faced by both industry and consumers today: How can we quantify a software system's security? We propose to use the measure of a system's *attack surface* as an indication of the system's security. While it is very difficult to devise metrics that reliably measure the security of software, prior work has shown that a system's *attack surface measurement* serves as a reliable proxy for security. Howard et al. have measured the attack surfaces of seven different versions of Windows [1], and we have measured the attack surfaces of four different versions of Linux [3]. The results of both the Linux and Windows measurements confirm perceived beliefs about the relative security of the different versions. The measurement methods, however, are based on intuition and are hard to replicate. Our current work is focused on defining a metric to *systematically* measure a system's attack surface. We envision our *attack surface metric* to be useful to both industry and consumers. Software designers and developers can use our *attack surface metric* as a tool in the software development process; they can measure their system's attack surface periodically during the software development phase, and compare the results with previous measurements. They should strive towards reducing their system's attack surface from one version to another to mitigate the security risk of their system. Software consumers can also use our metric to compare and differentiate between alternative and competing software systems. For example, system administrators can compare the attack surface measurements of different available web servers in choosing one for their organization.

2 Attack Surface Metric

Intuitively, a system's attack surface is the set of ways in which an adversary can attack the system. We know from the past that many attacks, e.g., exploiting a buffer overflow, on a system take place by sending data from the system's *operating environment* into the system. Similarly, many other attacks, e.g., symlink attacks, on a system take place because the system sends data into its environment. In both these types of attacks, an attacker connects to a system using the system's *channels* (e.g., open port), invokes the system's *methods* (e.g., API), and sends *data items* into the system or receives data items from the system. We collectively refer to a system's methods, channels, and data items as the system's *resources*. Given our above observation, we define a system's attack surface in terms of the system's resources. A system's attack surface is the subset of resources that an attacker can use to attack the system. Intuitively, the more resources available to an attacker, the larger the attack surface, hence the more insecure it is.

Not all resources, however, contribute equally to the measure of a system's attack surface. Informally, a system's attack surface measurement indicates the level of damage an attacker can

potentially cause to the system, and the effort required for the attacker to cause such damage. In order to measure a system's attack surface, we need to identify the resources that contribute to a system's attack surface, and determine the contribution of each such resource to the system's attack surface. We have introduced a formal *entry point and exit point framework* to identify these relevant resources. We have introduced the informal notions of *damage potential* and *effort* to estimate a resource's contribution [4]. A resource's contribution to a system's attack surface depends on the resource's damage potential, i.e., the level of damage the attacker can cause to the system in using the resource in an attack, and the effort the attacker spends to acquire the necessary access rights in order to be able to use the resource in an attack. The higher the damage potential, the higher the contribution; the higher the effort, the lower the contribution. We estimate a resource's contribution to the attack surface in terms of the resource's *attributes*. We estimate a method's contribution in terms of the method's *privilege* and *access rights*. Similarly, we estimate a channel's contribution in terms of the channel's *protocol* and *access rights*, and a data item's contribution in terms of the data item's *type* and *access rights*. We measure a system's attack surface along three dimensions: method, channel, and data. We measure the total contribution of the methods, the total contribution of the channels, and the total contribution of the data items to a system's attack surface. Given two systems, we measure their attack surfaces, and compare their attack surface measurements to indicate, along one dimension of many, whether one is more secure than another. We have demonstrated the use of our attack surface metric by measuring the attack surfaces of two open source IMAP servers: Cyrus 2.2.10 and Courier-IMAP 4.0.1.

3 Current Status and Future Work

A key challenge in security metric research lies in devising appropriate techniques for validating a security metric. Our current work is focused on developing both formal and empirical validation techniques for our attack surface metric. Using I/O automata [2], we have formally established a relation between a system's attack surface and the number of *executions* allowed by the system that an adversary can use to attack the system; we have shown that if a system, A, has a larger attack surface compared to a system, B, then A allows a larger number of such executions compared to B. We are also exploring three different empirical validation techniques. First, we plan to establish a correlation between a system's attack surface and the number of vulnerability bulletins released for the system; if a system, A, has a larger attack surface compared to a system, B, then we expect to see a larger number of bulletins for A compared to B. In collaboration with the Idaho National Laboratory (INL), we have measured the attack surfaces of two open source FTP servers: ProFTP 1.2.10 and Wu-FTP 2.6.2. We have also counted the number of times these two FTP servers are mentioned in CERT bulletins, MITRE CVEs, and the Bugtraq vulnerabilities database. The vulnerability bulletin counts are as expected; Wu-FTP has a larger attack surface compared to ProFTP and the number of bulletins for Wu-FTP is more than the number of bulletins for ProFTP. Second, we are using machine learning techniques to show that the six attributes (method privilege and access rights, channel protocol and access rights, and data item type and access rights) used in our measurement method are good indicators of a resource's damage potential and effort. Third, we are analyzing the data collected from honeypots to establish a correlation between a system's attack surface and the number of observed attacks on the system.

In the future, we plan to extend our work in three directions. First, we plan to measure the attack surfaces of two popular open source database servers in collaboration with INL. Based on our experience, we plan to implement an attack surface measurement toolkit that will reduce the number of manual steps and manual inputs required in our current method. Second, our current

attack surface measurement method requires the source code of a system; but it may not be always feasible to obtain the source code of a system (e.g., commercial software). Hence we plan to extend our method so that we can approximate a system's attack surface measurement in the absence of source code. Third, our current notions of damage potential and effort are based on intuition. We plan to characterize the notions of damage potential and effort more formally.

References

[1] M. Howard, J. Pincus, and J. M. Wing, "Measuring Relative Attack Surfaces," *Proceedings of Workshop on Advanced Developments in Software and Systems Security*, Taipei, December 2003.

[2] N. Lynch and M. Tuttle, "An introduction to Input/Output automata," *CWI-Quarterly*, 2(3):219–246, September 1989.

[3] P. K. Manadhata and J. M. Wing, "Measuring a system's attack surface," *CMU School of Computer Science Technical Report CMU-CS-04-102*, January 2004.

[4] P. K. Manadhata and J. M. Wing, "An Attack Surface Metric," *CMU School of Computer Science Technical Report CMU-CS-05-155*, July 2005.