Detecting AAA Vulnerabilities by Mining Execution Profiles *

Zhan Xu David Leon Andy Podgurski Vincenzo Liberatore

Security vulnerabilities are often due to latent software defects, which can be exploited by a malicious attacker. For example, a buffer overflow vulnerability can derive from a program's failure to check the size of its input. Security vulnerabilities are often present in complex software after deployment, and these defects must be identified early so as to minimize the damages that attackers can inflict. Meanwhile, Authentication, Authorization, and Accounting (*AAA*) services, such as Kerberos, Pubcookie, and Shibboleth, address the growing security demands for network access control. Distributed applications rely on AAA platforms to protect users and other stakeholders. However, AAA middleware potentially contains latent defects, which in turn can cause security holes. If such a vulnerability is exploited, the attacker can compromise any application that relies on the AAA services. Therefore, it is crucial to identify and correct latent AAA defects.

We are pursuing an integrated methodology for the identification of defects in AAA platforms. Our approach is based on execution capture, replay, profiling, and on mining the resulting execution profiles. We have used (or plan to use) a suite of complementary execution profiling techniques to reveal program or user behavior associated with exploits. Such techniques include code coverage profiling, object state profiling, information flow analysis, event sequence profiling, and temporal profiling. Our array of profiling techniques is more comprehensive and has a finer granularity than those used in other security platforms, such as Intrusion Detection Systems (IDS) and firewalls. In particular, we monitor the source code, the back-end database, and various event sequences. We have used statistical methods borrowed from multivariate visualization, such as correspondence analysis and multidimensional scaling.

*Division of Computer Science, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106. E-mail: {zxx10, dzl, hap, vxl11}@cwru.edu. This work has been supported in part under NSF grant ANI-0123929.

Our preliminary work includes code coverage profiling of *OpenLDAP*, a platform that is often employed to authenticate users through the Cyrus's *SAML* (Simple Authentication and Security Layer) libraries. We use modified versions of GNU `gcc` and GNU `gcov` to compile OpenLDAP and obtain profiles containing counts of run-time function calls. We replay against this instrumented LDAP server a set of synthetically generated requests, including one that causes a known security failure. The profiles can be visualized by multidimensional scaling, which produces a graphical representation of the pattern of proximities among a set of executions and can reveal their relationships, as shown in Figure 1. The troublesome request is isolated as an outlier in the bottom left corner of the display. In general, outliers denote unusual executions, which are more likely to reveal security vulnerabilities.

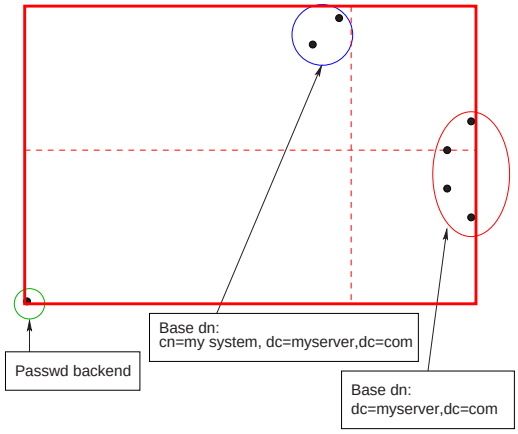

Figure 1: Sample profile of ldapsearch requests (red: search root, blue: search subdirectory, green: search password backend)

Our future work includes: (1) obtain more realistic data, e.g., massive logs of real LDAP requests, (2) combine multiple profiling techniques, (3) test other AAA platforms, and (4) study specific profiling technique against AAA software to identify cooperating distributed attacks.