Chapter 2
Attack Graph Techniques

2.1 An example scenario

Modern attack-graph techniques can automatically discover *all* possible ways an attacker can compromise an enterprise network by analyzing configuration information of the hosts and network [7, 12, 13, 19, 20, 24, 26, 27, 37, 38, 39, 41, 44, 46, 47, 50, 52]. We will use the MulVAL logical attack graph [38, 39] as the foundation to build the metric models. A logical attack graph directly encodes the logical causality relationship among configuration settings and potential attacker privileges. It shows "why an attack can happen", instead of "how an attack happens" as in some earlier attack-graph works [41, 46, 47, 50]. Its semantics is similar to the "exploit dependency attack graph" in the Cauldron project [7, 20, 35], and to a lesser degree also similar to the "multiple-prerequisite attack graph" [19] in the NetSPA project [27]. Thus our methodology could also be applied in combination with the other attack graph models. The advantage of MulVAL is that the logical relationship is clear and explicit in the attack graph representation and the graph generation is scalable. The asymptotic complexity of MulVAL attack graph generation is $O(n^2)$ where n is the number of machines in the network, and it can generate attack graphs with a thousand machines in minutes [38]. Recently Saha shows that by generating MulVAL logical attack graphs directly in the XSB [42] logic engine the graph generation time can be further reduced [44].

The semantics of MulVAL attack graphs is best explained with an example. In the small enterprise network shown in Figure 2.1, there are three subnets mediated by an external and an internal firewall. The web server is in the DMZ subnet and is directly accessible from the Internet through the external firewall. The database server is located in the Internal subnet and holds sensitive information. It is only accessible from the web server and the User subnet. The User subnet contains the user workstations used by the company's employees. The firewalls allow all out-bound traffic from the User subnet. The web server contains the vulnerability CVE-2006-

X. Ou and A. Singhal, *Quantitative Security Risk Assessment of Enterprise Networks*,
SpringerBriefs in Computer Science, DOI 10.1007/978-1-4614-1860-3_2, © The Author(s) 2012

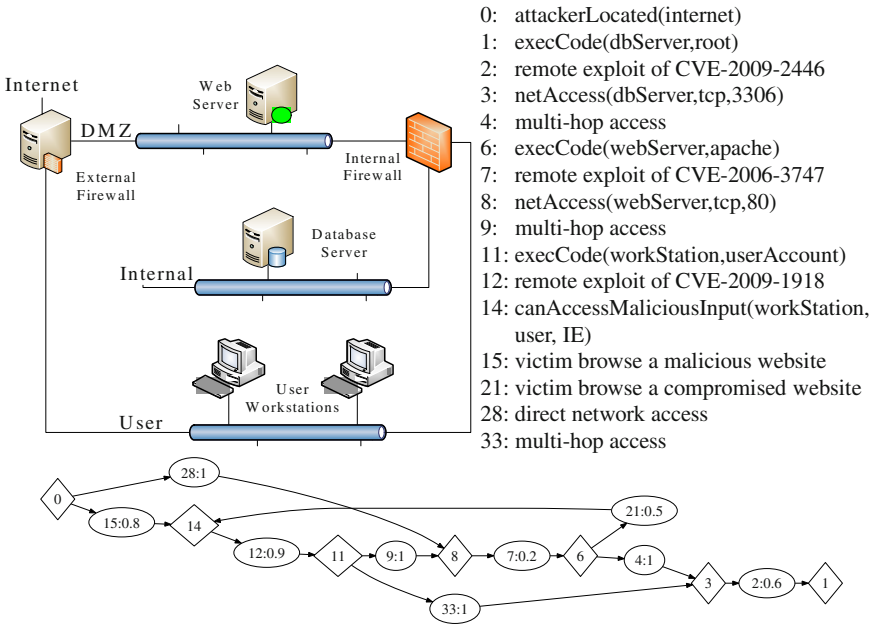

Fig. 2.1 Example scenario and attack graph

3747[1] in the Apache HTTP service which can result in a remote attacker possibly executing arbitrary code on the machine. The database server contains the vulnerability CVE-2009-2446 in the MySQL database service which could allow administrator access. The user workstations contain the vulnerability CVE-2009-1918 in the Internet Explorer. If a user accesses malicious content using the vulnerable IE browser the machine may be compromised. After analyzing the configuration of this network, MulVAL outputs an attack graph shown below the network diagram. The labels of the graph nodes are displayed at the righthand side of the network diagram.

There are two types of vertices in the attack graph[2]. The diamond vertices represent privileges an attacker could obtain through exploiting the vulnerabilities in the system. An elliptic vertex represents an attack step that can lead to a privilege. Node 0 is the attacker's initial privilege which in this case is a vantage point at the Internet. An attack can only be accomplished when all its pre-conditions are met; thus the incoming arcs to an attack-step vertex form a logical AND relation. For example, node 7 (shown as "7:0.2" in the graph) represents the exploit of the web server vulnerability and the exploit can only happen when the attacker can access tcp port 80 on the web server (node 8). Multiple incoming arcs to a privilege vertex indicates

[1] Common Vulnerabilities and Exposures (CVE) is a dictionary of common names (i.e., CVE Identifiers) for publicly known information security vulnerabilities http://cve.mitre.org/

[2] MulVAL attack graph also has a third type of vertices which are facts about system configuration. They are omitted for presentation clarity.

more than one way to obtain the privilege and thus form a logical OR relation. For example, privilege 3 (network access to the MySQL service on the database server) can be obtained either through compromising the web server (6) or the workstation (11).

A careful examination of the attack graph reveals a number of intrusion paths leading to the compromise of the various hosts. An attacker could first compromise the web server and use it as a stepping stone to further attack the database server (0, 28, 8, 7, 6, 4, 3, 2, 1). Or he could first gain control on a user workstation by tricking a user to click a malicious link, and launch further attacks from the workstation (0,15,14, 12, 11, ...). There are many other attack paths. In general if we enumerate all possible attack paths in a system the number will be exponential. However, the privileges and attacks on all these paths are inter-dependent on each other and the number of pair-wise inter-dependencies is quadratic to the size of the network. Instead of enumerating all attack paths, a logical attack graph like MulVAL enumerates the inter-dependencies among the attacks and privileges. This provides an efficient polynomial-time algorithm for computing a compact representation of *all* attack paths in a system.

Although the example attack graph is computed from known vulnerabilities, attack graphs are equally powerful in reasoning about unknown (zero-day) vulnerabilities [18, 39], by introducing hypothetical vulnerabilities in the input. Such hypothetical vulnerabilities can be marked in the produced attack graphs and handled accordingly in the subsequent analysis.

Attack graphs are often perceived to offer a deterministic view of enterprise network security: an attack can succeed as long as all its preconditions are met, and a privilege can be obtained as long as the graph shows a path leading to it from the attacker's initial privilege. This type of deterministic semantics is certainly valuable and one can use it to conduct various types of useful analysis [13, 16, 22, 25, 27, 36, 44, 55]. However, the reality of practical enterprise security management is far from a clear-cut zero/one view. Take the vulnerability CVE-2006-3747 on the web server as an example. The official description found on the CVE website (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3747) says:

> Off-by-one error in the ldap scheme handling in the Rewrite module (mod_rewrite) in Apache 1.3 from 1.3.28, 2.0.46 and other versions before 2.0.59, and 2.2, when RewriteEngine is enabled, allows remote attackers to cause a denial of service (application crash) and possibly execute arbitrary code via crafted URLs that are not properly handled using certain rewrite rules.

The word "possibly" highlights that the true consequence of exploiting the vulnerability is far from certain. Since this vulnerability is one of the first stepping stones for the subsequent attacks, the likelihood for an attacker to obtain the other privileges are also affected by the likelihood he can succeed at this first stage. A system administrator would typically conduct some research on the web to "get a sense" on how likely an attacker, given access to the vulnerability, would be able to successfully exploit it. He then combines this with the specific situation in his own network to gauge the risk. This is an important process since most organizations operate under limited resources and cannot afford to fix all potential security

problems. Without an understanding of the likelihood a vulnerability can lead to real damage, it will be hard to see how the potential damage compares to the costs incurred by the various countermeasures (*e.g.* down time due to patching) and make sensible decisions. Unfortunately there is currently no quantitative models that can help administrators make such decisions, and as a result security management of enterprise networks is still a "black art". Our proposed research attempts to transform this field into a science by designing objective quantitative security metrics built upon attack-graph techniques. Logical relations encoded in attack graphs are highly important in gauging security risks, but one must go beyond the deterministic view and admit the inherent uncertainty in risk assessment.

2.2 Tools for Generating Attack Graphs

- TVA (Topological Analysis of Network Attack Vulnerability) In [33, 35, 20] the authors describe a tool for generation of attack graphs. This approach assumes the monotonicity property of attacks and it has polynomial time complexity. The central idea is to use an exploit dependency graph to represent the pre and post conditions for an exploit. Then a graph search algorithm is used to chain the individual vulnerabilities and find attack paths that involve multiple vulnerabilities.
- NETSPA (A Network Security Planning Architecture) In [19, 18] the authors use attack graphs to model adversaries and the effect of simple counter measures. It creates a network model using firewall rules and network vulnerability scans. It then uses the model to compute network reachability and attack graphs representing potential attack paths for adversaries exploiting known vulnerabilities. This discovers all hosts that can be compromised by an attacker starting from one or more locations. NETSPA typically scales as $O(nlogn)$ as the number of hosts in a typical network increases. Risk is assessed for different adversaries by measuring the total assets that can be captured by an attacker.
- MULVAL (Multihost, multistage, Vulnerability Analysis) In [38, 39] a network security analyzer based on Datalog is described. The information in vulnerability databases, the configuration information for each machine and other relevant information are all encoded as Datalog facts. The reasoning engine captures the interaction among various components in the network. The reasoning engine in MULVAL scales well ($O(n^2)$) with the size of the network.

Skybox security [4] and Red Seal Systems [2] have developed a tool that can generate attack graphs. Risk is calculated using the probability of success of an attack path multiplied by the loss associated with the compromised target. Nessus [1] and Retina [3] are vulnerability management systems that can help organizations with vulnerability assessment, mitigation and protection.

http://www.springer.com/978-1-4614-1859-7

Quantitative Security Risk Assessment of Enterprise Networks

Ou, X.; Singhal, A.

2011, XIII, 28 p., Softcover
ISBN: 978-1-4614-1859-7