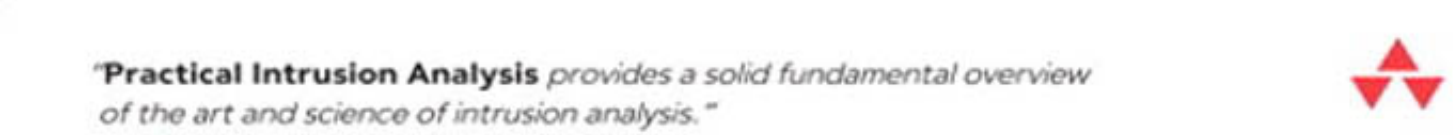
"**Practical Intrusion Analysis** *provides a solid fundamental overview of the art and science of intrusion analysis.*"

—**Nate Miller,** Cofounder, Stratum Security



PRACTICAL INTRUSION ANALYSIS

Prevention and Detection for the Twenty-First Century

RYAN TROST

Practical Intrusion Analysis

Security is not a one-time single-point fix; it's a continuous process, as exemplified in the *protect-detect-react* lifecycle. To *protect* from attacks, you take steps to prevent them from succeeding. Still, you must understand that not all attacks can be averted in advance, and there must usually remain some residual vulnerability even after reasonable protective measures are applied.

Indeed, the more important question is not the vulnerability itself, but the magnitude of damage in case of an incident. You rely on the *detect* phase to identify actual attack instances. But, the detection process must be tied to residual vulnerabilities, especially ones that lie on paths to critical network resources. After attacks are detected, comprehensive capabilities are needed to *react* to them based on vulnerability paths. You can thus reduce the impact of attacks through advance planning and by knowing the paths of vulnerability through your networks, based on preemptive analysis of network vulnerability scan results. To create such a proactive stance, you must transform raw data about network vulnerabilities into attack roadmaps that help you prioritize and manage risks, maintain situational awareness, and plan for optimal countermeasures.

This chapter describes the latest advances in an innovative proactive approach to network security called *Topological Vulnerability Analysis (TVA)*.[1,2] By analyzing vulnerability interdependencies, TVA builds a complete map that shows all possible paths of multistep penetration into a network, organized as a concise attack graph. The TVA attack graph then supports proactive network defenses across the entire protect-detect-react lifecycle. This includes identifying critical vulnerabilities, computing key security metrics, guiding the configuration of IDSs, correlating and prioritizing intrusion alarms, reducing false alarms, and planning optimal attack responses. You can also implement the TVA approach as a working tool, available commercially through limited distribution.

The remainder of this chapter is organized as follows:

- **Topological Vulnerability Analysis (TVA)**. Reviews the TVA approach and provides a visual example.
- **Attack modeling and simulation**. Describes the process of capturing network attack models in TVA to simulate multistep penetrating attacks.
- **Optimal network protection**. Discusses how to apply attack graphs for optimal network protection.
- **Intrusion detection and response**. Covers the application of attack graphs to intrusion detection and response.
- **Summary**. Summarizes our approach and suggests possible future advances.

TOPOLOGICAL VULNERABILITY ANALYSIS (TVA)

Because of vulnerability interdependencies across networks, a topological attack graph approach is needed, especially for proactive defense against insidious multistep attacks. The traditional approach that treats network data and events in isolation, without the context provided by attack graphs, is clearly insufficient. TVA combines vulnerabilities in ways that real attackers might, discovering all attack paths through a network, given the completeness of scan data used for your analysis. Mapping all paths through the network provides defense-in-depth, with multiple options for mitigating potential attacks, rather than relying on mere perimeter defenses.

This section overviews the TVA attack graph analysis and gives an example attack graph as an illustration. It then discusses the limitations of this modeling/simulation approach to attack graphs analysis.

OVERVIEW OF APPROACH

Figure 5-1 shows the overall flow of TVA. It begins by building an input attack model, based on the network configuration and potential attacker exploits. Network configuration data might include vulnerability scan reports, hosts inventory results, and firewall rules. Because you *model* network penetration versus actually exploiting vulnerabilities, you need to represent the fact that a given vulnerability can potentially be exploited. In fact, assume the worst case and model exploitation cause/effect, even if working exploit code is yet unreported for a given vulnerability. This model is explained in the section, "Attack Modeling and Simulation."

From this input attack model, TVA matches modeled exploits against vulnerabilities to predict multistep attacks through the network. From the resulting attack graph, it generates recommendations for optimal priority of hardening vulnerabilities, as described in the section, "Vulnerability Mitigation." The attack graph can also be explored through interactive visualization. (For more in-depth risk analysis, including what-if scenarios, see the section, "Attack Graph Visualization.") The TVA attack graph also supports computation of various metrics for measuring overall network security (see the section, "Security Metrics").

The attack graph guides optimal strategies for preventing attacks, such as patching critical vulnerabilities and hardening systems and services. However, because of realistic operational constraints, such as availability of patches or the need to offer mission-critical services, there usually remain some residual attack paths through a network. At this point, the residual attack graph provides the necessary context for dealing with

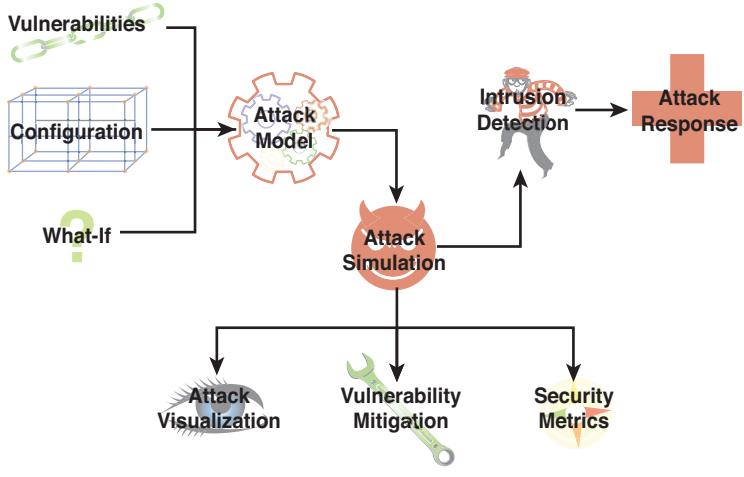


Figure 5-1 Visual representation of the Topological Vulnerability Analysis (TVA) overview

intrusion attempts. This includes guidance for the deployment and configuration of IDSs, correlation of intrusion alarms, and the prediction of next possible attack steps for an appropriate attack response.

For example, the attack graph can guide the placement of intrusion detection sensors to cover all attack paths, while minimizing sensors redundancy. As in all cases for TVA analysis, the attack graph must be kept current with respect to changes in network vulnerabilities. The attack graph then can filter false intrusion alarms, based on known paths of residual vulnerability. The graph also provides the context for correlating isolated alarms as part of a larger multistep attack penetration. It also shows the next possible vulnerabilities that an attacker might exploit, and whether they lie on attack paths to critical network resources. This in turn supports optimal planning and response against attacks, while minimizing the effects of false alarms and purposeful misdirection by an attacker.

ILLUSTRATIVE EXAMPLE

As a simple illustration of the attack graph approach, consider the small network in Figure 5-2. In this network, assume that the mail server and file server are only for internal use. However, outside access to the Web server is needed. Thus the firewall allows incoming Web connections to the Web server and blocks all other traffic from the outside. In this attack scenario, you want to know if an attacker on the outside can compromise the mail server through one or more attack steps.

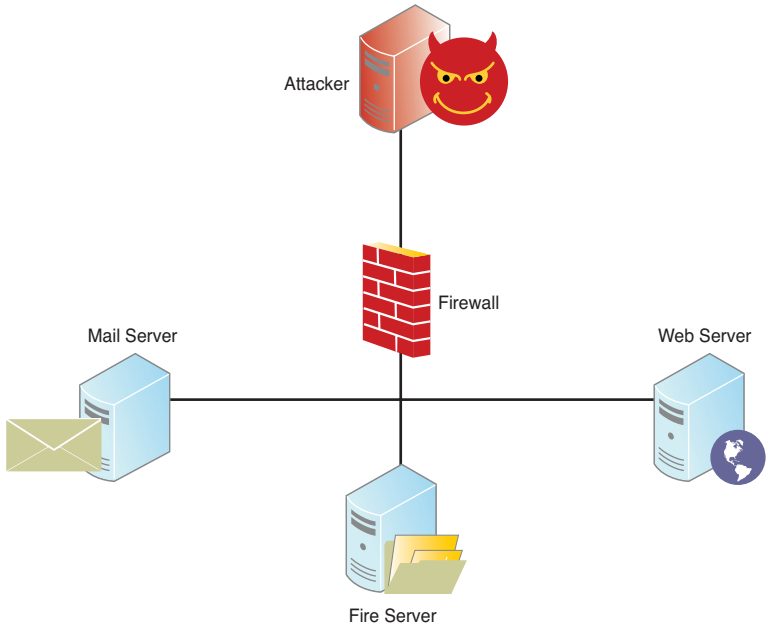


Figure 5-2 Small example network. The firewall allows Web traffic to the Web server, and blocks all other incoming traffic.

To model this scenario, you need to capture elements of the network configuration relevant to attack penetration. This includes the existence of vulnerable software (services) on hosts and the connectivity allowed to vulnerable services. You also need a set of potential attacker exploits that might work against the vulnerable services. In general, you rely on existing security tools to scan the network and build the input model.

For example, you can run a vulnerability scanning tool, such as Nessus,[3] against the hosts in the internal network to map their vulnerabilities and feed this into the TVA model. You then rely on your database of modeled exploits, which is prebuilt to cover exploitable vulnerabilities detected by Nessus. Assume the worst case, such as a vulnerability is exploitable (leads to an exploit) as long as it is reported as giving sufficient control over the victim machine. This is independent of any particular code or procedure that might actually carry out such exploitation.

To incorporate the connectivity-limiting effects of the firewall, scan the firewall. Also, scan behind the firewall to capture vulnerabilities that are available after an attacker reaches the internal network. Alternatively, you can process the firewall rules directly for building the network model.

Figure 5-3 shows the resulting attack graph for this scenario. There is a path from the outside to the inside mail server via a critical vulnerability exposed through the firewall. Figure 5-3(a) is a high-level view of the attack graph. It shows one vulnerability being exploited (implicitly, through the firewall) from the outside to the inside. In other words, the attack graph indicates that one vulnerability is exposed from the outside with the potential to be exploited, which allows the attacker to progress inside. This exploit, along with all others in this model, gives the attacker the ability to execute arbitrary code at an elevated privilege.

Figure 5-3(b) offers a more detailed view. It shows that an attacker can exploit a vulnerability on the Web server from the outside. Then, from the Web server, the attacker can attack the mail server. The box labeled "inside" represents the inside network, and implicitly, all machines on the inside can exploit one another's vulnerabilities. In Figure 5-3, the label 1 in the attack graph edge indicates that there is one exploit (implicitly, one exploitable vulnerability) from the attacker to the Web server. Inside the network, there are three exploits (three exploitable vulnerabilities on the Web server).

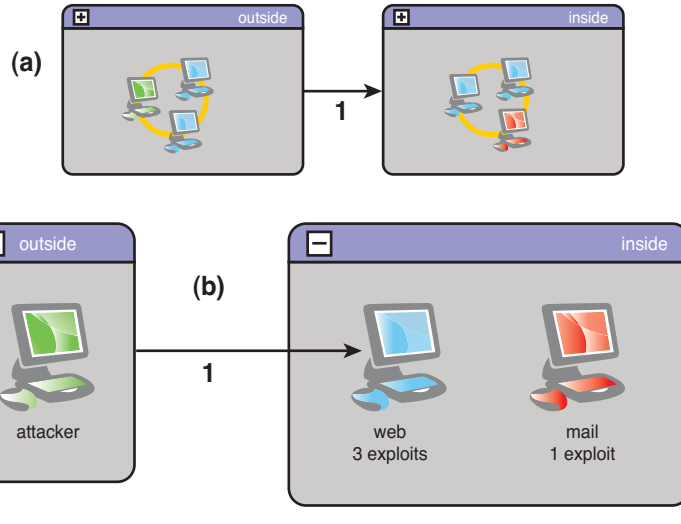


Figure 5-3 The critical vulnerability path from an outside attacker to the inside mail server from Figure 5-2

Of the three exploitable vulnerabilities on the Web server, only one is exploitable from the outside. TVA identifies this critical vulnerability. In other words, if the single vulnerable service from the attacker to the Web server is mitigated, the attacker has no other path

to the mail server. Of course, other vulnerabilities can be mitigated, but the vulnerability from the attacker to the Web server is clearly a high priority.

This simple example shows how hosts on a network can be exploited through multiple steps, even when an attacker cannot directly access them. It is not directly possible to compromise the internal mail server from the outside because of the policy enforced by the firewall. But, TVA shows that the attack goal can be reached indirectly (in this case, through a sequence of two exploits). Furthermore, it shows that addressing a single critical vulnerability from among four within the internal network can prevent this attack scenario.

By constraining the attack graph to particular start and goal points, you focus the analysis on protecting a critical asset against an assumed threat source. For example, the file server does not appear in the attack graph because it does not play a part in this scenario. In other words, there are no attack paths from an attacker to the mail server that involve the file server. Also, Nessus and other vulnerability scanners generate many alerts that are merely informational and not relevant to network penetration. The TVA tool excludes such extraneous alerts from its database of modeled exploits.

In general, many different combinations of critical vulnerabilities might prevent an attack scenario. For enterprise networks, analyzing all attack paths and drawing appropriate conclusions requires extensive analysis.

LIMITATIONS

TVA is fundamentally a modeling/simulation approach. It relies on existing tools to gather network configuration and vulnerability information. It also needs to be prepopulated with a database of modeled exploits that can potentially be applied to a network. So, in this sense, the attack graph results are only as complete as the input model.

The benefits of a modeling/simulation approach include the capability to easily change the model for what-if analysis. But the modeling taxonomy needs to be carefully defined to reflect the realities of the network attack environment, while keeping model complexity manageable. That is, there is a tradeoff between model fidelity and model complexity that you must balance. Also, different analysis tasks might call for variations in model details. For example, the level of detail needed for information-operations support might differ from what is needed for patch management. The TVA tool is written to accept general models, in terms of exploit preconditions/postconditions. The only requirement is to create a database of the modeled exploits needed and to create network models that match exploit conditions.