Module 5: Risk Management

CSC3207 Computer Security

Outline

- Risk and threat modeling with Attack Trees
- Vulnerability analysis
- Compliance: Audits reviews and inspection, Vulnerability scanners, Penetration testing

References

- Security in Computing; 5th Edition; Charles P. Pfleeger and Shari Lawrence Pfleeger

Risk Management

- **Risk management**: the process of identifying risk, as represented by vulnerabilities, to an organization's information assets and infrastructure, and taking steps to reduce this risk to an acceptable level
- a management activity at the heart of security planning.
- Risk analysis is an organized process for identifying the most significant risks in a computing environment, determining the impact of those risks, and weighing the desirability of applying various controls against those risks
- Good, effective security planning includes a careful risk analysis.
- A risk is a potential problem that the system or its users may experience

Risk

- The risk impact: a loss associated with an event
  - The event must generate a negative effect: compromised security, lost time, diminished quality, lost money, lost control, lost understanding, and so on
- The likelihood that the event will occur
  - The probability of occurrence associated with each risk is measured from 0 (impossible) to 1 (certain)
- When the risk probability is 1, we say we have a problem
- The degree to which we can change the outcome
  - We must determine what, if anything, we can do to avoid the impact or at least reduce its effects
- Risk control involves a set of actions to reduce or eliminate the risk

Components of Risk Management

- Risk management involves three major undertakings: risk identification, risk assessment, and risk control
- Risk identification is the examination and documentation of the security posture of an organization's information technology and the risks it faces
- Risk assessment is the determination of the extent to which the organization's information assets are exposed or at risk
- Risk control is the application of controls to reduce the risks to an organization's data and information systems

Components of Risk Management

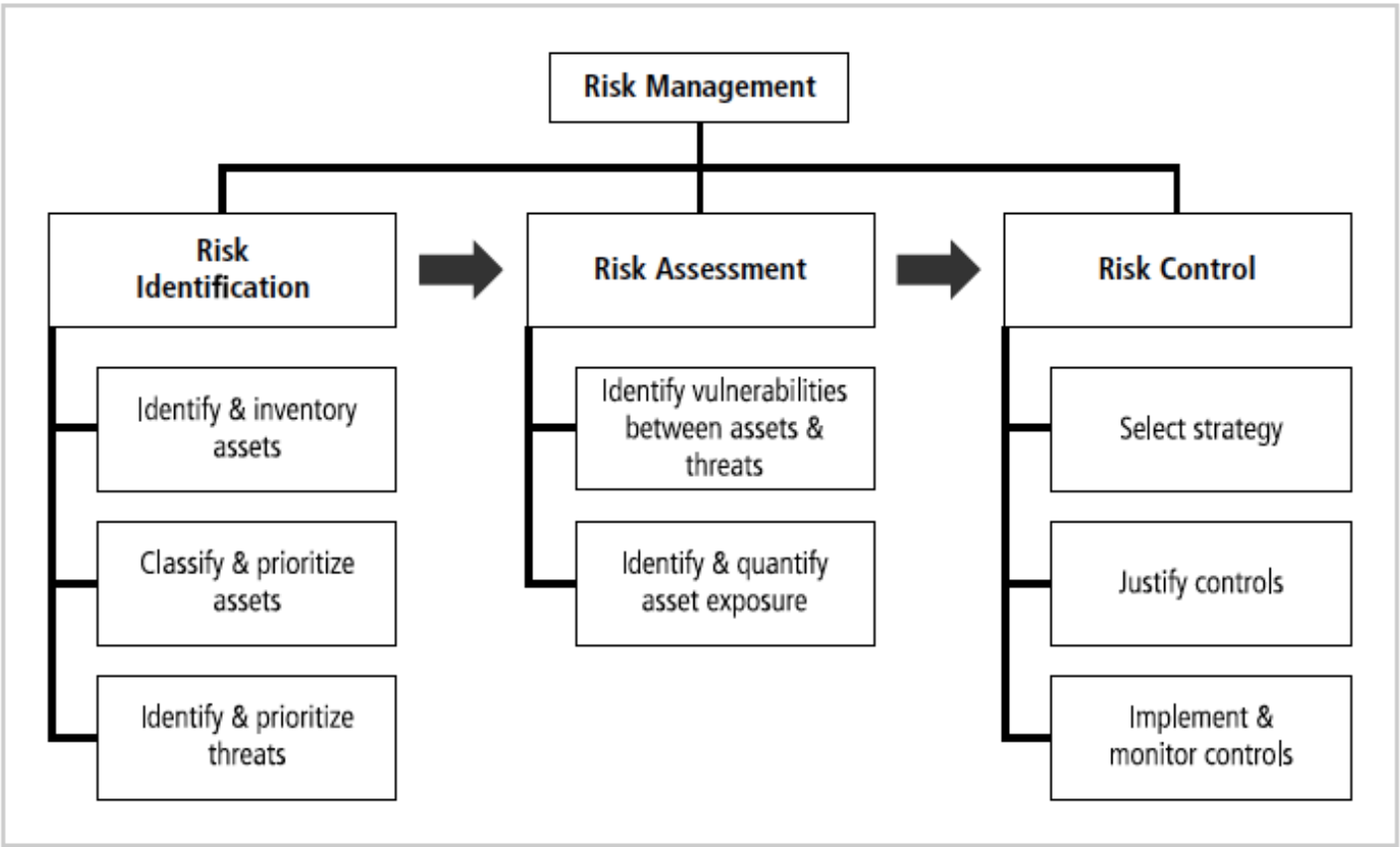


Figure 4-1 Components of Risk Management

Risk Exposure

- We usually want to weigh the pros and cons of different actions we can take to address each risk
- We can quantify the effects of a risk by multiplying the risk impact by the risk probability, yielding the **risk exposure**.
- For example, if the likelihood of virus attack is 0.3 and the cost to clean up the affected files is $10,000, then the risk exposure is $3,000.
- So we can use such a calculation to decide that a virus checker is worth an investment of $100, since it will prevent a much larger expected potential loss
- Risk probabilities can change over time, so a risk analysis activity should track them and plan for events accordingly

Risk Exposure

- We can identify, limit, avoid, or transfer risk but we can seldom eliminate it
- 3 strategies for dealing with risk:
  - avoid the risk by changing requirements for security or other system characteristics
  - transfer the risk by allocating the risk to other systems, people, organizations, or assets by buying insurance to cover any financial loss should the risk become a reality
  - assume the risk by accepting it, controlling it with available resources and preparing to deal with the loss if it occurs

Risk Leverage

- costs are associated not only with the risk's potential impact but also with reducing it
- Risk leverage is the difference in risk exposure divided by the cost of reducing the risk

$$\frac{\text{(risk exposure before reduction)} - \text{(risk exposure after reduction)}}{\text{(cost of risk reduction)}}$$

- The leverage measures value for money spent: A risk reduction of \$100 for a cost of \$10, a 10:1 reduction, is quite a favorable result
- If the leverage value of a proposed action is not high enough, then we look for alternative but less costly actions or more effective reduction techniques
- Risk leverage is the amount of benefit per unit spent

Risk Leverage

$$\frac{\text{(risk exposure before reduction)} - \text{(risk exposure after reduction)}}{\text{(cost of risk reduction)}}$$

Attack surfaces and Attack trees

Attack Surfaces

An attack surface consists of the reachable and exploitable vulnerabilities in a system. Examples of attack surfaces:

- Open ports on outward facing Web and other servers, and code listening on those ports
- Services available on the inside of a firewall
- Code that processes incoming data, email, XML, office documents, and industry specific custom data exchange formats
- Interfaces, SQL, and Web forms
- An employee with access to sensitive information vulnerable to a social engineering attack

Categories of Attack surfaces

- **Network attack surface**: refers to vulnerabilities over an enterprise network, wide-area network, or the Internet. Includes network protocol vulnerabilities, such as those used for a denial-of-service attack, disruption of communications links, and various forms of intruder attacks.
- **Software attack surface**: This refers to vulnerabilities in application, utility, or operating system code. A particular focus in this category is Web server software.
- **Human attack surface**: This category refers to vulnerabilities created by personnel or outsiders, such as social engineering, human error, and trusted insiders.

Attack surface analysis

- An attack surface analysis is a useful technique for assessing the scale and severity of threats to a system.
- A systematic analysis of points of vulnerability makes developers and security analysts aware of where security mechanisms are required.
- Once an attack surface is defined, designers may be able to find ways to make the surface smaller, thus making the task of the adversary more difficult.
- The attack surface also provides guidance on setting priorities for testing, strengthening security measures, or modifying the service or application.

Attack Trees

- **An attack tree** is a branching, hierarchical data structure that represents a set of potential techniques for exploiting security vulnerabilities.
- The security incident ie. the goal of the attack is represented as the root node of the tree
- The ways that an attacker could reach that goal are iteratively and incrementally represented as branches and subnodes of the tree.
- Each subnode defines a subgoal, and each subgoal may have its own set of further subgoals, etc.
- The final nodes on the paths outward from the root, i.e., the leaf nodes, represent different ways to initiate an attack. Each node other than a leaf is either an AND-node or an OR-node.

Attack Trees

- To achieve the goal represented by an AND-node, the subgoals represented by all of that node's subnodes must be achieved; and for an OR-node, at least one of the subgoals must be achieved.
- Branches can be labeled with values representing difficulty, cost, or other attack attributes, so that alternative attacks can be compared.
- The motivation for the use of attack trees is to effectively exploit the information available on attack patterns.
- Security analysts can use the attack tree to document security attacks in a structured form that reveals key vulnerabilities.
- The attack tree can guide both the design of systems and applications, and the choice and strength of countermeasures.

Attack Tree Example

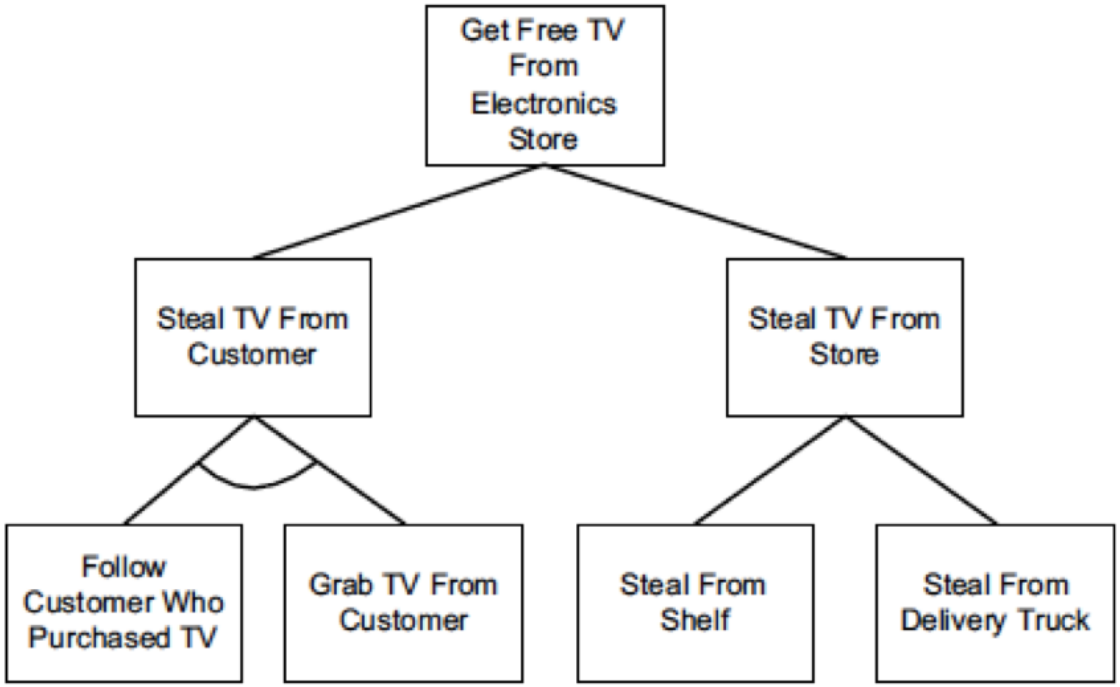


Figure 1. Example attack tree

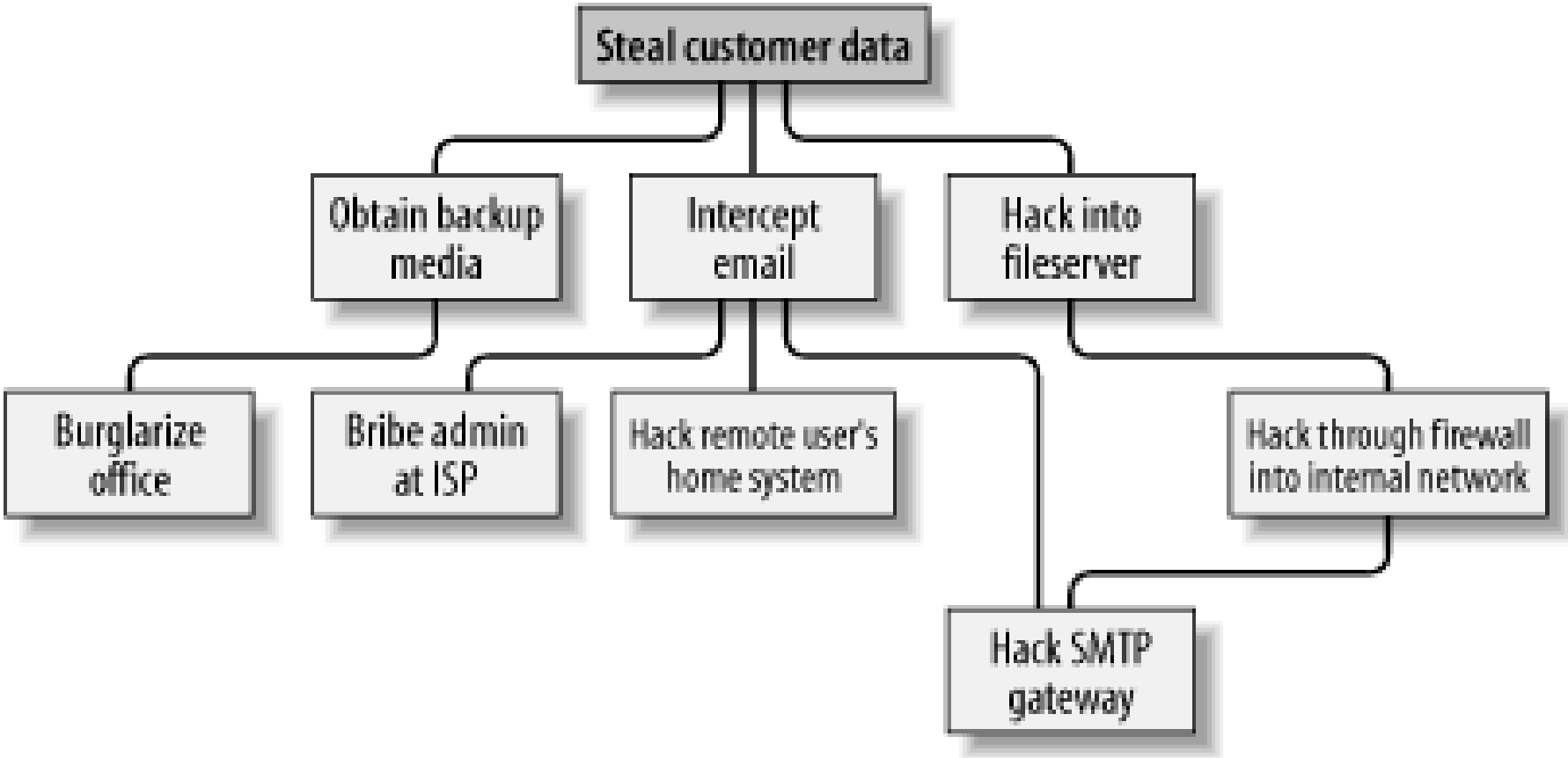
Attack Tree Example

- The attacker's goal is to get a free television from a retail electronics store.
- The attacker can accomplish the goal by either stealing the television from a customer or by stealing it directly from the store.
- To steal the television from a customer, the thief must identify and follow a customer who has purchased a television AND actually take the television from the person. This branch is an example of an AND relationship.
- To demonstrate the OR relationship, the thief has two ways to steal the television from the store. He can either take it from the shelf OR from the delivery truck.

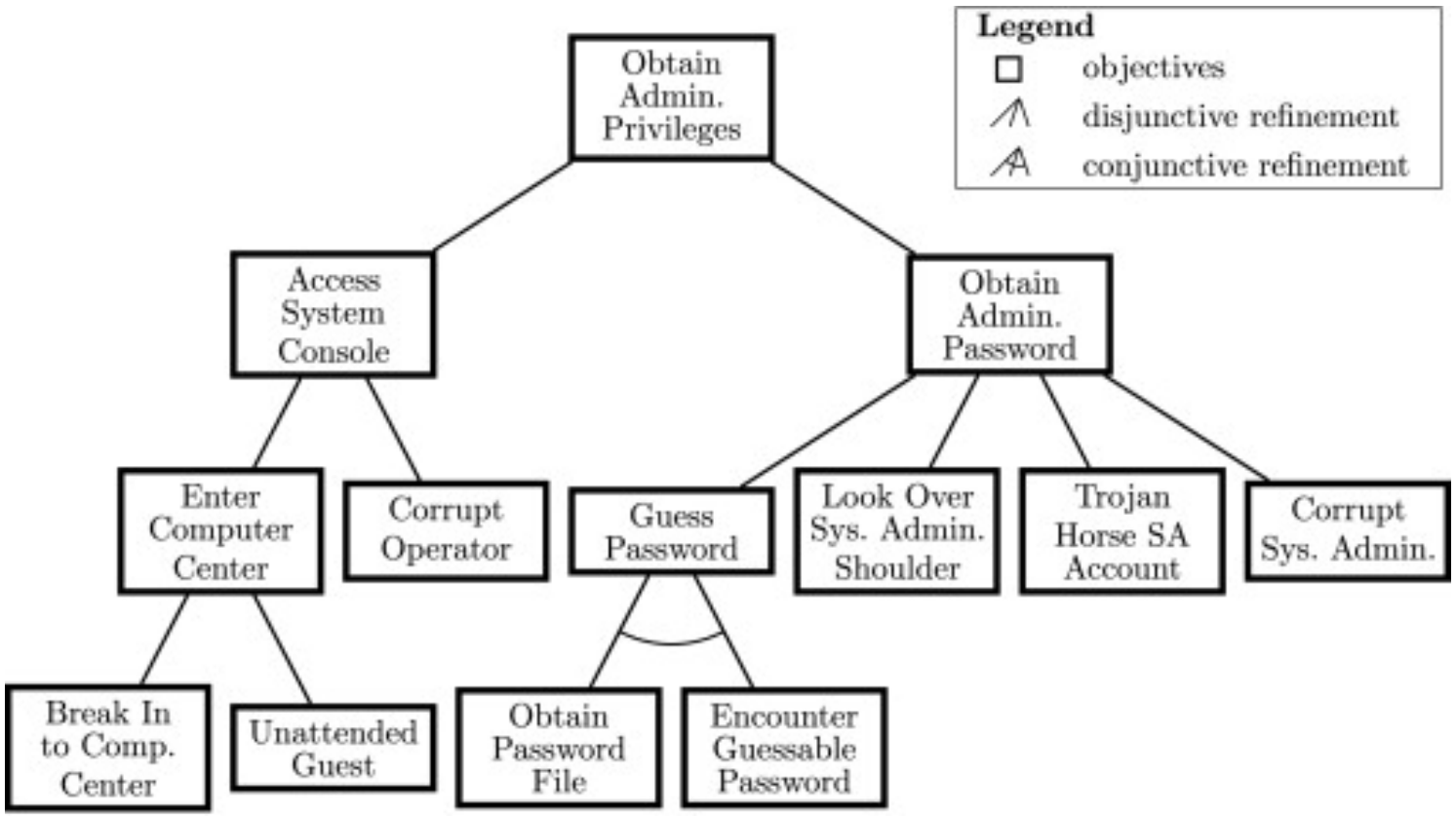
Attack Tree Example 2

- To create an attack tree, you must first define the root node.
- For example, one attack objective might be "Steal ABC Corp.'s Customers' Account Data."
- Direct means of achieving this could be as follows:
  - Obtain backup tapes from ABC's file server.
  - Intercept email between ABC Corp. and their customers.
  - Compromise ABC Corp.'s file server from over the Internet.
- These three subgoals are the leaf nodes immediately below our root node
- Next, for each leaf node, you determine subgoals that achieve that leaf node's goal. These become the next "layer" of leaf nodes. This step is repeated as necessary to achieve the level of detail and complexity with which you wish to examine the attack.

Attack Tree Example 2



Attack Tree Example 3



Exercise

Develop an attack tree for the following scenarios

- gaining access to customer account details from the database of a bank.
- reading someone's e-mail.

Vulnerability Analysis

Vulnerability Analysis

- **Vulnerability**: specific failures of security controls (procedures, technology or management)
  - Errors in code
  - Human violators
  - Mismatch between assumptions
- Vulnerabilities arise from computer system design, implementation, maintenance, and operation
- **Exploit**: Use of vulnerability to violate policy
- **Attacker**: Attempts to exploit the vulnerability

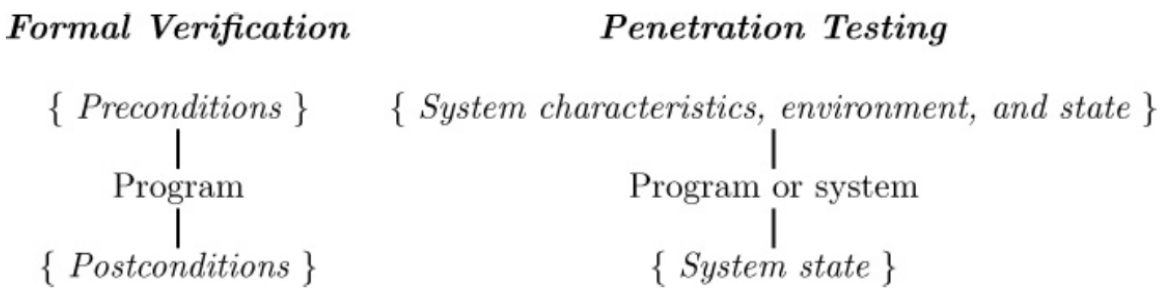
Vulnerability Example

- Many systems have special administrative users who are authorized to create new accounts
- Suppose a user who is not an administrative user can add a new entry to the database of users, thereby creating a new account
- This operation is forbidden to the non-administrative user
- However, such a user has taken advantage of an inconsistency in the way data in the database is accessed
- The inconsistency is the vulnerability
- the sequence of steps that adds the new user is the exploit
- A secure system should have no such problems

Techniques for Detecting Vulnerabilities

- System Verification:
  - Determine preconditions, post-conditions
  - Validate that system ensures post-conditions given preconditions
  - Can prove the absence of vulnerabilities
- Penetration testing:
  - Start with system/environment characteristics
  - Try to find vulnerabilities
  - Cannot prove the absence of vulnerabilities

Comparison of System Verification and Pen Testing



System Verification

- Preconditions place constraints on the state of the system when the program (or system) is run
- The postconditions state the effect of running the program
- What are the problems?
  - Invalid assumptions
  - Limited view of system
  - Still an inexact science
  - External environmental factors
  - Incorrect configuration, maintenance and operation of the program or system

Penetration Testing

- Test strength of security controls of the complete system
  - evaluate procedural and operational controls as well as technological controls
  - Attempt to violate stated policy
  - Works on in-place system
  - Framework for evaluating results
  - Examines procedural, operational and technological controls
- testing security by using various tools and techniques common to attackers
- Typical approach: Red Team, Blue Team, White Team
  - Red team attempts to discover vulnerabilities
  - Blue team simulates normal administration, Detect attack, respond
  - White team injects workload, captures results

Penetration Testing Viewpoints

- External vs. Internal
  - Penetration Testing can be performed from the viewpoint of an external attacker or a malicious employee
- Overt vs. Covert
  - Penetration Testing can be performed with or without the knowledge of the IT department of the company being tested
- The methodology is similar to that of an attacker
  - enumerate/scan the network to discover what machines are attached and operating
  - assess vulnerabilities
  - research vulnerabilities for known exploits, and then
  - use tools available to penetrate the network

Penetration Testing Methodology

- Enumerating a network to discover what machines are attached and operating is a useful task for both an intruder and a system administrator
- The information gained from a network scan assists in the determination of the actual current layout
- Several tools and techniques exist for both Windows and Linux platforms to perform these tests
- Once the devices and their open ports have been identified, a vulnerability scanner can be used
  - The scanner will use its database of vulnerabilities to test if the system has any of them
- These vulnerabilities are further researched online, and then utilities that can be used to penetrate the network are retrieved and executed
- A good penetration test should result in a report that explains the weaknesses found, lists them from most critical to least critical, and provides suggestions for improving the network's security

Types/layers of Penetration Testing

- Black Box (External Attacker)
  - External attacker has no knowledge of target system
  - Attacks often build on human element – Social Engineering
- System access provided (External Attacker)
  - Red team provided with limited access to system, Models external attack
  - Goal is to gain normal or elevated access, Then violate policy
- Internal attacker
  - Red team provided with authorized user access
  - Goal is to elevate privilege / violate policy

Problems with Penetration Testing

- Nonrigorous
  - Dependent on insight (and whim) of testers
  - No good way of evaluating when "complete"
- How do we make it systematic?
  - Try all classes of likely flaws
  - But what are these?
- Vulnerability Classification!

Using Nmap Tool

- Download and install nmap from https://nmap.org/
- Use Nmap to scan a network for hosts that are up
- Use Nmap to enumerate the ports and services available on a host
- Try out activities at https://www.freecodecamp.org/news/what-is-nmap-and-how-to-use-it-a-tutorial-for-the-greatest-scanning-tool-of-all-time/