- Entities that are declared as *final* must not be changed.
- Variables may not be used before they are initialized.
- Array bounds must be checked during all array accesses.
- Objects cannot arbitrarily be casted into other object types.

The program simply declares a character pointer, and without allocating any memory, accepts user input in that pointer. This can cause havoc if an attacker finds intelligent ways to exploit such code. This is not possible in Java.

1.5.4 Specific Attacks

On the Internet, computers exchange messages with each other in the form of small groups of data, called *packets*. A packet is like an envelope that contains the actual data to be sent and the address information. Attackers target these packets as they travel from the source computer to the destination computer over the Internet. These attacks take two main forms: (*a*) *packet sniffing* (also called *snooping*) and (*b*) *packet spoofing*. The protocol used in this communication is called the *Internet Protocol (IP)*. Other names for these two attacks are (*a*) *IP sniffing* and (*b*) *IP spoofing*.

Understanding the Two Attacks

a. *Packet sniffing:* Packet sniffing is a passive attack on a conversation. An attacker need not hijack a conversation, but instead, can simply observe (*i.e.*, sniff) the packets as they pass by. To prevent an attacker from sniffing packets, the information that is passed needs to be protected in some ways. This can be done at two levels: (*i*) The data that is traveling can be encoded in some way or (*ii*) the transmission link itself can be encoded. To read a packet, the attacker needs to access it in the first place. The simplest way to do this is to control a computer that the traffic goes through. Usually, this is a router. However, routers are highly protected resources. Therefore, an attacker might not be able to attack it and instead attack a less-protected computer on the same path.

b. *Packet spooling:* In this technique, an attacker sends packets with an incorrect source address. When this happens, the receiver (*i.e.*, the party who receives these packets containing a false source address) would inadvertently send replies back to this forged address (called the *spoofed address*), and not to the attacker. This can lead to three possible scenarios:

(i.) The attacker can intercept the reply: If the attacker is between the destination and the forged source, the attacker can see the reply and use that information for the hijacking.
(ii.) The attacker need not see the reply: If the attacker's intention was a Denial Of Service (DOS) attack, the attacker need not bother about the reply.
(iii.) The attacker does not want the reply: The attacker could simply be angry with the host, so it may put that host's address as the forged source address and send the packet to the destination. The attacker does not want a reply from the destination, as it wants the host with the forged address to receive it and get confused.

Another attack, which is similar to these attacks, is the *DNS spoofing* attack. People usually can't identify Websites using the *Domain Name System (DNS)* because they are not really memorable (for example, 120.10.1.67). For this, a special server computer called as a DNS server maintains the mappings between domain names and the corresponding IP address. The DNS server could be located anywhere. Usually, it is with the *Internet Service Provider (ISP)* of the users. With this background, the DNS spoofing attack works as follows.

1. Suppose that there is a merchant (Bob), whose site's domain name is www.bob.com, and the IP address is 100.10.20. Therefore, the DNS entry for Bob in all the DNS is www.bob.com.
2. The attacker (Trudy) manages to hack and replace the IP address of Bob with her own (say 100.20.20.20) in the DNS server maintained by the ISP of another user, Alice. Therefore, the DNS server maintained by the ISP of Alice now has the following entry: www.bob.com, 100.20.20.20.
3. When Alice wants to communicate with Bob's site, her web browser queries the DNS server maintained by her ISP for Bob's IP address, providing it with the domain name (*i.e.*, www.bob.com). Alice gets the replaced (*i.e.*, Trudy's) IP address, which is 100.20.20.20.
4. Alice then starts communicating with Trudy, believing that she is communicating with Bob.

Such attacks of DNS spoofing are quite common and cause havoc. Even worse, the attacker (Trudy) does not have to listen to the conversation on the wire. She has to simply be able to hack the DNS server of the ISP and replace a single IP address with her own.

A protocol called the *DNSSec (secure DNS)* is being used to thwart such attacks. Unfortunately, it is not widely used.

EXERCISES

1. Find more examples of security attacks reported in the last few years.
2. What is the key principle of security?
3. Why is confidentially an important principle of security? Think about ways of providing security. (*Hint:* Think about the ways in which children use a secret language.)
4. Discuss the reasons behind the significance of authentication. Find out the simple mechanism of authentication. (*Hint:* What information do you provide when you use a free e-mail service such as Yahoo or Hotmail?)
5. In real life, how is the message integrity ensured? (*Hint:* On what basis is a check honored?)
6. What is repudiation? How can it be prevented in real life? (*Hint:* Think what happens if you issue a cheque, and after that, tell the bank that you never issued that cheque).
7. What is access control? How different is it from availability?
8. Why are some attacks called passive? Why are others called active?
9. Discuss a passive attack.
10. What is a masquerade? Which principle of security is breached because of that?
11. What are replay attacks? Give an example of replay attacks.
12. What is a denial of service attack?
13. What is a worm? What is the significant difference between a worm and a virus?
14. Find out more about some recent worms.
15. Write a small virus-like program in plain English that accepts a file name and changes every character in the file to an asterisk.