Network Security

Professor Gaurav S. Kasbekar
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Week - 02

Lecture - 09

Different Types of Attacks on Networks

Hello, in this lecture we discuss different types of attacks on networks. This slide summarizes the different types of attacks on networks. One type of attack is, intruders can infect hosts with malware such as viruses, worms, and so on. Then another class of attacks is Denial-of-Service, (DoS) attacks. There are different types of DoS attacks as we'll see.

One example of a DoS attack is as follows. The attacker floods the access link of a host. That is the link that connects a host to the rest of the internet with a large number of bogus packets. So, this prevents legitimate packets from reaching the host. So, hence, network service is denied to the host.

Another example of an attack is, obtaining secret information such as, passwords and credit card information from legitimate users. Then another attack is, impersonating some other user. For example, Trudy might pretend to be Alice. Some malicious user might, for example, try to access the email account of a legitimate user. Then, another attack is Illegal access to resources.

For example, using subscriptions to some general articles to which a user does not have legitimate access but illegally accessing those resources. And another attack is modification and replay of messages. An intruder can intercept messages being sent from Alice to Bob and modify them and/or replay them before forwarding them to the recipient Bob. We now discuss the above attacks in detail. So, we start with infection of host with malware.

In this attack, the attacker puts malware, such as viruses and worms into a host via different means, such as over the network or through a USB flash drive, and so on. Once this malware is introduced into a host, it can perform different kinds of harmful activities. For example, it may delete files. So, useful files are not available to a legitimate user. The

malware may acquire some hard disc space or CPU time. So, this hard disc space is occupied, so, it cannot be used for storing legitimate files.

Or CPU time is blocked, so, the processes of a legitimate user run slowly. Then, the malware might send some spam email to contacts. The attacker might also install some spyware that collects secret information, and sends it over the network to the attacker. This secret information might include, for example, passwords, credit card information, keystrokes, and so on. So, this spyware is installed on a user's computer, and it collects all this kind of secret information, and sends it over the network to the attacker.

So, this secret information is leaked out. Another kind of harmful activity that can be performed by malware is, because of the installation of this malware on a host, it can get enrolled into a network of thousands of similarly compromised hosts, and this network is called a botnet. And that is used by the attacker for various malicious activities, such as spam email distribution, distributed denial of service attacks, and so on. We'll discuss this in more detail later. So, in this attack, the intruder first installs malware into a network of thousands of hosts and then, sends commands to those hosts to perform different malicious activities.

Malware can be self-replicating, that is, it produces a copy of itself. In particular, once the malware infects one host, it does not stop there, but from that host, it seeks entry to other hosts. And from those other hosts, it seeks entry to yet other hosts and so on and so forth. So, it keeps on spreading. The spread of malware is often exponentially fast.

That's because, as a simple example, suppose the malware initially infects one computer, then from that computer it infects three other computers, and from each of these three other computers it infects three other computers and so on. So, the rate of the spread is exponential in the number of rounds. So, the number of infections is of the order of 3 to the power n, where n is the number of rounds that have taken place. So, malware can spread exponentially fast. Some examples of malware are as follows.

Virus, worm, and Trojan horse or simply Trojan. These are some popular kinds of malware. So, there are some differences between these three types. A virus is a set of malicious instructions that are inserted into a program when the program is infected. So, notice that a virus is a part of a program.

A virus is a set of instructions that are inserted into another program. So, it's not a standalone program. A virus requires some human action to spread. For example, if a

human runs an infected executable file, then the virus runs, and then it spreads. When the infected program is executed by a user, the virus also executes.

Since the virus instructions are part of the program, and then the virus performs its malicious activities as well as infects other programs. In contrast to a virus, a worm is a standalone program. It can spread across a network without any human action unlike a virus. So, in particular, once a worm is running in a machine on a network, then it can explore the network by scanning the network and then send itself to other machines. And a Trojan horse is a set of instructions that are hidden inside an otherwise useful program that performs some malicious activities.

So, the difference between a Trojan and a virus is that, we use the term Trojan, if the malicious instructions are installed at the time the program is written. That is, the intention of the program is to add these malicious instructions to a victim. So, the program is written in order to infect computers. And the term virus is used if the instructions are added to the program later on. So, in the case of a virus, the program is initially a legitimate program, but then some intruder infects that program later on.

Trojan horses typically do not infect other files or copy themselves to other machines, unlike viruses and worms. There are other types of malware as well. Examples are trapdoor, logic bomb, spyware, and rootkit. From the point of view of this course, we are not so much interested in exactly what kind of malware it is, but our objective is to defend against all kinds of malware. Now, to gain insight into malware, let's see how a virus looks like.

A virus can be installed in most programs by doing the following. The attacker replaces any instruction, say, the instruction at memory location x by a jump to some free space in memory, say, location y. So, suppose, this is the memory of a victim computer. So, there is some instruction at location x. The attacker replaces this instruction by a jump to some free space in memory, say, location y. So, from this point the program jumps to some other location, say, location y. And then, the virus instructions are added starting at memory location y. So, these are the virus instructions. And then, the instruction that was originally at this location x, that is placed at the end of the virus instructions, followed by a jump to x + 1.

This is x + 1 and from here the program jumps to x + 1. So, what essentially happened is that, the program was supposed to go from x to x + 1 but instead, from x it jumped to y and at y the malicious instructions that are part of the virus, they are written. The program then performs these malicious instructions and then, jumps back to point x + 1. That is the next

memory location from where it started. So, apart from doing whatever damage the virus does, it may replicate itself by looking for other executable files in any directory and infecting them.

So, the virus performs some malicious activities, such as deleting files, blocking CPU time, and blocking hard disc space, and so on. Besides, it also infects other files. Once an infected program is run, the virus is executed again to do more damage, and to replicate itself to more programs. So, how may a virus appear on a computer? There are different ways it can appear on a computer.

One way is, a user might run an infected program received as an email attachment or copied from a USB flash drive. So, once this infected program is run, then the malware gets installed into the computer. Or a user might just purchase software from some company and that software might turn out to be infected. So, how did malware get into that software? The malware may have been planted into the software by some malicious employee of the company or by some malicious external user, who broke into the computers of the company and installed the malware.

Or it may even be planted by a non-malicious employee whose office computer was infected. So, the office computer of the non-malicious employee was infected and when the employee used the computer, the malware gets added to the software, which is then later on purchased by a legitimate user. So, this concludes our discussion of malware. Next, we discuss Denial-of-Service (DoS) attacks. A DoS attack causes some network infrastructures to be unusable by legitimate users.

Examples of network infrastructure are web or email server, link, host, corporate or university network. So, such network infrastructure might become unusable by legitimate users because of a DoS attack. The different types of DoS attacks are listed here. One is a vulnerability attack, then bandwidth flooding, and connection flooding. So, in a vulnerability attack, the attacker sends some messages, which are well designed to a vulnerable application or operating system on a host.

These messages cause a service to stop. For example, some web or email server running on the host might stop running, or the host may crash. So, because of this, service cannot be provided to legitimate users. Then, in the bandwidth flooding attack, consider a host connected to the rest of the internet via some access link. So, this is a host, that is connected via an access link to the rest of the internet.

In the bandwidth threading attack, an attacker sends a large number of bogus packets to the access link of the host, which might be a server. This causes the link to clog, and that prevents legitimate packets from reaching the host. So, this is the access link. And the attacker sends a large number of bogus packets to this link, so this link clogs, and legitimate packets are not able to reach the host. And because of that, service cannot be provided to legitimate users.

In the connection threading attack, the attacker establishes a large number of bogus TCP connections with the target host. So, because of this, the target host's resources are consumed and it stops accepting legitimate connections. So, these are different types of denial of service attacks. Let's discuss these in more detail. Here is an example of a bandwidth flooding attack.

Recall that in this attack, the attacker sends a large number of bogus packets to the access link of a host and that causes the link to clog. Suppose the access link has rate R bits per second. So, to block an appreciable fraction of the link, the attacker must send packets to the host at rate approximately R. But if R is large, for example, 10 Gbps, then only one host controlled by the attacker may not be able to send packets at this large rate of R or close to R. Also, if the attack packets are sent from only one host, then it would be easy to detect this attack and block it by a firewall.

So, if all the attack packets are being sent from a particular IP address, say I, then a firewall can detect that there are a large number of packets being sent from a particular IP address, say I, and it can block those packets. So, such an attack would be easy to detect. To counter this, an attacker can launch a more sophisticated attack, that is, a Distributed Denial-of-Service, (DDoS) attack. In this attack, there are two steps. First, the attacker infects a large number of machines with some malware.

Then, because of the installation of this malware on these machines, which are shown here, these machines start acting under the commands that are sent by the attacker. So, whatever commands are sent by the attacker are followed by these computers because of the malware installed on them. So, this set of computers, slave computers, they are known as a botnet. It's a network of thousands of compromised hosts on which malware has been installed. And once the attacker then sends a command to all the infected hosts that are part of the botnet and asks them to send traffic to a target host.

So, at the same time, all these slaves, they start sending bogus packets to the access link of this victim computer. So, the access link gets clogged and useful packets do not reach the

victim computer. So, this is a distributed denial of service attack. Since bogus packets are sent from a large number of computers, the aggregate rate at which packets reach this victim computer are large. And also, this attack is more difficult to detect than the case where only one attacker computer sends bogus packets.

Because packets are coming in from a large number of IP addresses, so it's difficult to detect by a firewall. Now, let's discuss the vulnerability attack in some more detail. So, it is typically caused by a poorly written application or system software. In particular, the problem is caused by code that is very trusting of user input. Examples are buffer overflow attack and SQL injection attack.

Let's discuss these attacks in some more detail. These are examples of vulnerability attack. In the buffer overflow attack, as an example, suppose a program declares an array of 100 elements with 100 bytes reserved for storing each element. And the program populates the array with input from the user. But the program does not check the length of the user input string.

If the user types an input which exceeds 100 bytes, then the buffer overflows because the buffer is only 100 bytes long. So, if the input is more than 100 bytes, then the buffer overflows. Since the buffer overflows, the program writes into memory past the end of the buffer. And this memory at the end of the buffer may hold data or executable code. So, using this buffer overflow attack, the attacker may write into areas known to hold executable code and replace it with malicious code.

So, the attacker writes into a part of memory in which executable code is placed, and the next time that executable code runs, since it has been replaced with the malicious code, some malicious activities are performed by the code. So, this is the buffer overflow attack. Now, we discuss SQL injection attack. SQL is Structured Query Language. It is popularly used to perform operations on a database, such as querying a database, modifying, inserting information into a database, or deleting information from a database.

Now, consider a webpage that has two fields to allow users to enter a username and a password. When the user enters a username and a password, the code behind the page generates an SQL query to check the password against a list of usernames, to check whether the username and password are legitimate or not. This code is shown here. SELECT UserList.Username, FROM UserList, WHERE UserList.Username = 'Username', where this is the username provided by the user, AND UserList.Password = 'Password'. This is the password that is typed in by the user.

If this query returns a row, that means there is a line in the database, in which, the username is the username typed by the user and the password is the password typed by the user. So, then access is granted. So, this is how the code is supposed to run. But a malicious user can attack this code in the following way. Suppose the malicious user enters a valid username; it is not difficult to find a valid username.

Often it is the same as the email address of a legitimate user, which is well known. So, a malicious user enters a valid username, and in place of the password field, which the attacker does not know, the malicious user injects some valid code, specifically password' or '1 = 1'. Then in the place of this password, this text typed by the malicious user gets substituted, and the resulting query becomes the following. SELECT UserList.Username, FROM UserList, WHERE UserList.Username is Username, AND UserList.Password = Password, or '1 = 1'. Now, this first comparison is not true.

Userlist.Password = Password, this is false because password is just some random string typed by the attacker. But this part, or '1 = 1', this is always true, and hence a row is written, thereby allowing access. So, instead of typing the password, the malicious user types some code, which is this password, or '1 = 1'. Since this comparison is always true, 1 is always equal to 1, so hence the result of this is a row, is returned as a result of this query, and hence, access is granted to the malicious user. So, this is one example where the vulnerability attack is launched.

And this is possible because the code does not check the password that is typed by the malicious user. So, if the code had checked the password, then it could have found that it is not a legitimate password, but it is some bytes in a program. Then another kind of attack is obtaining secret information, such as passwords and credit card information. To illustrate this attack, consider a host, such as a laptop or smartphone communicating over a wireless link. The wireless link might be Wi-Fi or cellular.

Now, when a wireless transceiver transmits some information, that information propagates and reaches all the other hosts that are in the vicinity of the transmitter. So, an attacker can place a passive receiver near the host that records a copy of every packet that is sent on the wireless medium. Such a receiver is called a packet sniffer. It collects all the packets that are sent out on the wireless medium in a particular vicinity in which it is placed. Packet sniffers can be used to collect information from not only wireless media but also, they can be connected to wired broadcast media, such as Ethernet or cable internet.

In this case, recall that there is a cable which is shared and a transmission on the cable reaches the entire cable. So, if a packet sniffer is attached to a wired broadcast medium, then it collects all the packets that flow on the cable. These sniffed packets are then analyzed to extract sensitive information from the sniffed packets. Now, how do we defend against such packet sniffing attacks? So, a solution is to encrypt messages before sending them.

And then only the legitimate receiver is able to decrypt the messages and recover the transferred information. We'll study cryptography later and we'll discuss different schemes for cryptography. Now, we just remarked that a packet sniffer is not only used for malicious activities, but it is also a useful tool for several non-malicious activities as well, such as collecting statistics for network monitoring, and management and debugging network protocols. But malicious users can use a packet sniffer to gather all the information that is sent on a broadcast medium, like a wireless medium or shared cable, and extract confidential information from the transmitted packets. Then, another kind of attack is traffic analysis.

This attack can be performed even if the information is encrypted before transmitting it. Suppose again, that the attacker uses a packet sniffer and records all the communication taking place between legitimate users Alice and Bob. Also suppose that all information is encrypted before transmission by Alice and Bob. But the attacker may still be able to gather some useful information. For example, the attacker may be able to determine the location and identity of the communicating host.

The IP addresses of the transmitter and receiver are sent without encryption. So, using these, the identities of the communicating hosts can be inferred by the attacker, who gathers all the packets flowing between Alice and Bob. The location can be inferred from the IP address. The attacker may be able to observe the frequency lengths and timings of the messages being exchanged, and this may result in the leak of useful information. So, this information, such as frequency of messages, lengths of messages, and timing of messages, may be useful in guessing the nature of the communication that is taking place.

Some examples will illustrate this. Suppose, each time Alice sends a long message to Bob, Bob sends a short reply back to Alice and a long message to five other people. The contents of these messages cannot be read by the attacker, but from this pattern the attacker can infer that Alice is probably Bob's manager, and each time Alice sends some instructions to Bob, Bob sends a short reply, which may be an acknowledgement, and then Bob sends a long

message to five other people, and delegates tasks to these five other people who are Bob's subordinates. So, this may indicate that Alice is sending orders to Bob and Bob is relaying them to his subordinates. So, this, some information is leaked out because of this pattern.

Now, if Alice sends regular short messages to Bob and then suddenly sends a series of long ones, then it can indicate to the attacker that something has changed. Then, the attacker can further investigate and try to find out what has changed. So, again, some information is leaked out. Another example is, if Alice telephones a known terrorist every week, then this fact may be enough to charge her without knowing the details of the conversation. So. here, just the identities of the users who are communicating, that is enough to leak out useful information.

That is, the fact that Alice is communicating with a terrorist. Another kind of attack is, impersonation of a legitimate user by a malicious user. So, a way this can happen is, it is easy for a malicious user to create a packet with a false source IP address and send it into the internet to a target node. This is called IP spoofing. It is quite easy to do.

IP spoofing can be used to attack a routing algorithm. For example, an attacker can send a packet to a router, claiming that it is another router. The attacker spoofs the IP address of a legitimate user and sends a packet pretending to be a legitimate router. And then, the attacker asks the victim to modify the routing table so as to harm the network or benefit the attacker. For example, the routing table might be modified so that long routes are taken or packets are dropped and so on.

So, the network may be harmed or the attacker may be benefited. Another example is, the routing table might be modified so that all the packets reaching that router are forwarded to the attacker's computer. And then, the attacker is able to read the packets. The malicious user can also use impersonation to access the email account of a legitimate user. So, in these kind of attacks, the attacker pretends to be another user and communicates with a legitimate user.

What are mechanisms to defend against impersonation attacks? One mechanism is end-point authentication. This is a mechanism for a user at one end of a connection to check whether the user at the other end is indeed who he or she claims to be. For example, if Alice and Bob communicate over a network, then Alice can verify that on the other side of the communication link, there is indeed Bob, and Bob can verify that indeed it is Alice who is communicating with him. Then, another mechanism to defend against impersonation attacks is message integrity.

This is a mechanism for a recipient of a message, say Bob, to check whether a message received from a user, say X, was indeed sent by the user X and was not modified during transit. So, if Bob receives a message from Alice, then Bob can check that the message was indeed sent by Alice and not by someone else, and also that the message was not modified during transit by some attacker. So, the difference between end-point authentication and message integrity is that in the case of end-point authentication, there are two users who are live at the two ends of a connection. And in the case of message integrity, a user, say Alice, just sends a message to Bob, and Bob then checks just from the message whether the message was legitimate and whether it was modified during transit. Another kind of attack is illegal access to resources.

Here the attacker obtains free access to paid services. For example, the attacker might obtain free access to paid online products, such as e-books, magazines, journal articles, and so on, to which the attacker does not have legitimate access. The attacker might also acquire some free talk-time on someone else's account or freely use computing power on a supercomputer. So, these are illegal accesses to resources. And these attacks are possible because the attacker is able to circumvent some controls that permit access to only paid subscribers of such services.

For example, a user might have to provide some credentials to access an e-book, but an attacker may be able to either steal those credentials or bypass the procedure to gain authorization and gain access to the e-book. This is the attack where an attacker illegally accesses some resources to which they do not have legitimate access. Another example of an attack is modification and replay of messages. In the case of modification attack, an intruder intercepts one or more messages from a transmitter and modifies the messages, delays them and/or reorders them before sending them to the receiver. So, the information that is sent by the transmitter is changed and modified information reaches the receiver.

An example is, suppose, the original messages allow John Smith to read confidential file x; this might be modified to state allow Fred Brown to read confidential file x. So, because of this attack, Fred Brown gains access to the confidential file x. So, this is an example of a modification attack. Replay attack is where an intruder captures a message traveling from a transmitter to a receiver and then, later, replace the same message to cause unwanted effects. So, one example is, after Bob places an order on a website, the intruder replaced the same order 10 more times, causing a larger quantity of the ordered good to be delivered to Bob. This attack may cause unwanted effects, such as, in this example. The reason this

attack is possible is that, often, there are mechanisms to check whether a message is received from a legitimate user or not and whether it was modified during transit or not.

So, message integrity mechanisms can be used to assure this. But if an attacker captures a legitimate packet and then replays the same packet multiple times, then message integrity checks may not indicate whether the message is legitimate or not. Since they are copies of legitimate packets, so it may be difficult to detect that they are replayed packets. So, using this fact, an attacker can launch a replay attack. This concludes our discussion of attacks on networks.

In the rest of the course, we will discuss different mechanisms for defending against these attacks. Thank you.