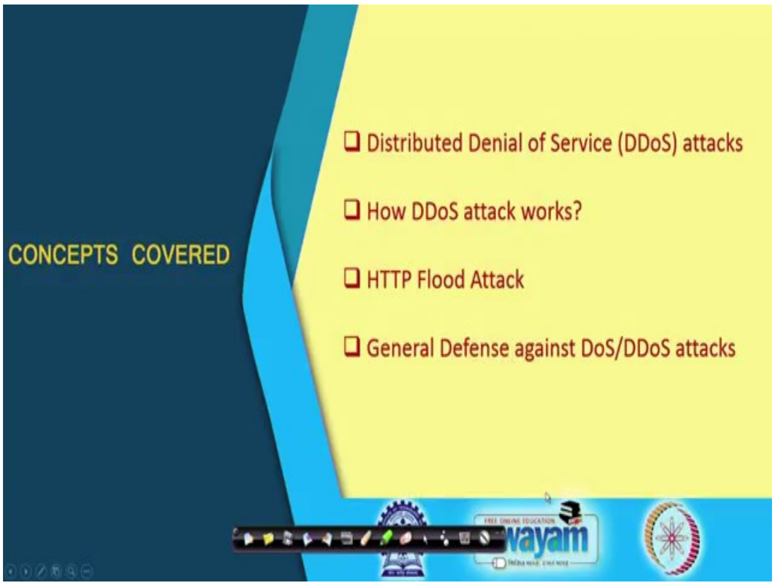
Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 39
Network Based Attacks (Part – II)

In this lecture, we continue with our discussion on Network Based Attacks. If you recall in our previous lecture, we had talked about some kinds of network based attacks, in particular the denial of service attack and some of the mechanisms using which such an attack can be mounted. So, we continue with our discussion.

(Refer Slide Time: 00:39)



CONCEPTS COVERED

❑ Distributed Denial of Service (DDoS) attacks

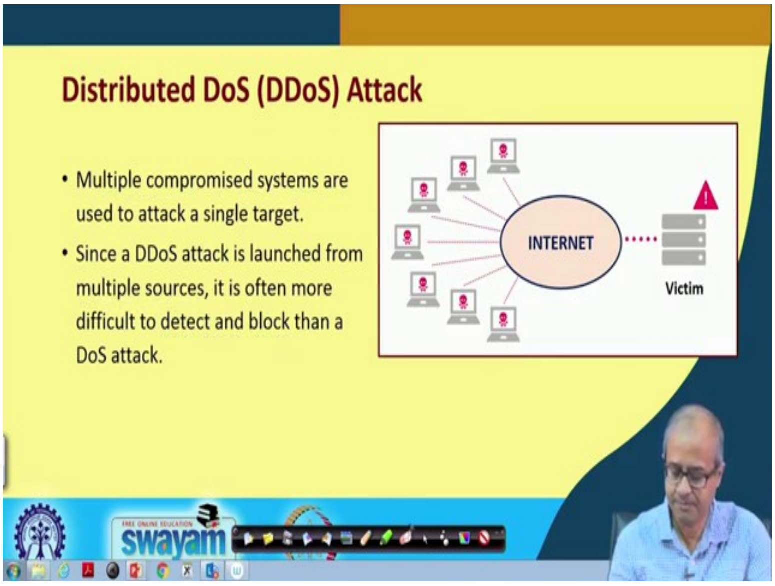
❑ How DDoS attack works?

❑ HTTP Flood Attack

❑ General Defense against DoS/DDoS attacks

So, in this lecture, we shall mainly be covering and enhanced version of denial of service called distributed denial of service attack in short DDoS. We shall see how such attacks work specifically, some HTTP based flooding attack, called HTTP flood attack, and finally, we shall be talking about some of the general safeguards on or defence against such attacks, fine.

(Refer Slide Time: 01:10)



Distributed DoS (DDoS) Attack

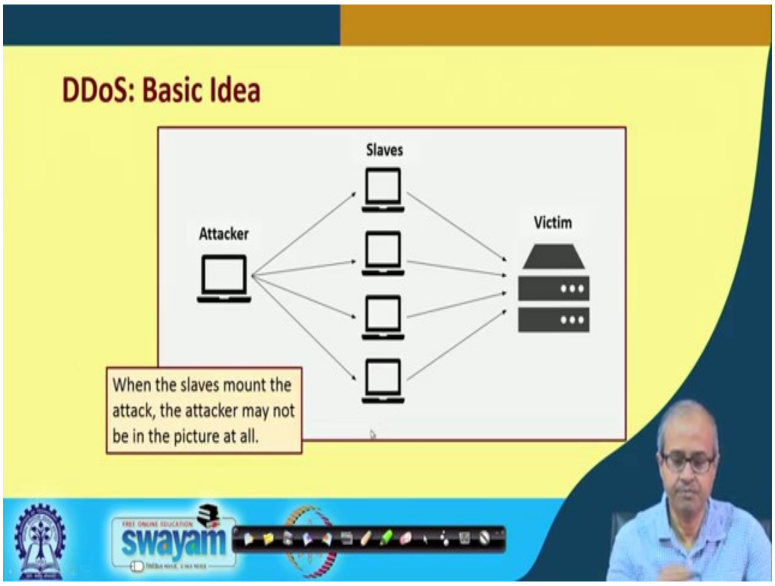
- Multiple compromised systems are used to attack a single target.
- Since a DDoS attack is launched from multiple sources, it is often more difficult to detect and block than a DoS attack.

So, let us start by talking about distributed denial of service attack. Well, in a traditional denial of service attack, you can recall, there is an attacker who is trying to flood a victim machine by usually a large number of junk packets. Distributed means something similar happens, but the attack is mounted not from a single machine, but from many machines.

So, you will see here, the idea is, there are multiple compromised systems that are used to mount an attack. If you look at the diagram here, you see the victim is out here. This is the victim computer or computers which are attacked, obviously over the internet, and there are multiple computers which are all compromised and are under control of the so called attacker.

The attacker will be controlling these machines and all these machines, they can be thousands in number. They will be mounting a distributed or parallel attack on the victim; they will all be sending large number of packets so that normal service gets affected. Now, the issue here is that this kind of a distributed denial of service attack as I said, is launched from multiple sources, not one, but several computers can mount this kind of a attack in parallel. And because it is mounted from multiple sources, it becomes that much more difficult for the system, system administrator to detect that this kind of an attack is happening, ok. And generally detection and preventing this kind of an attack becomes more difficult, fine.

(Refer Slide Time: 03:21)



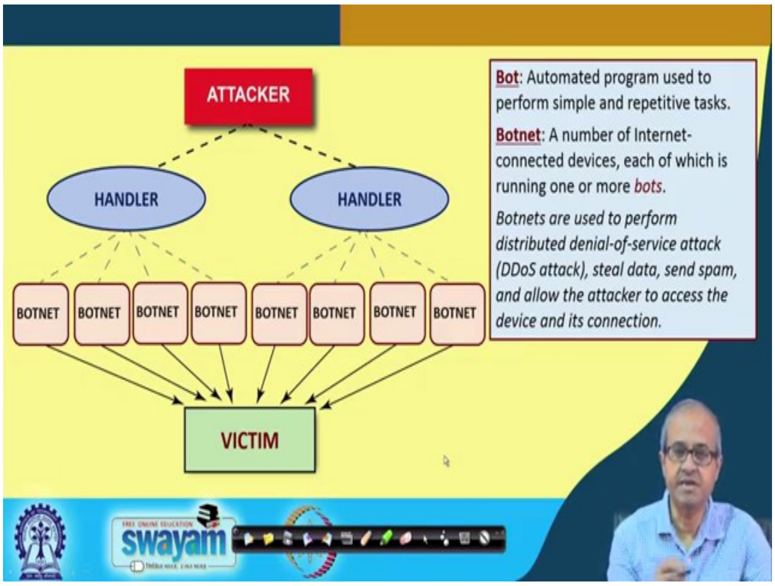
DDoS: Basic Idea

When the slaves mount the attack, the attacker may not be in the picture at all.

So, let us look at the basic idea using another diagram. Here let us assume that the attacker is here, this is the attacker machine. There is the hacker. We sitting on some machine and somehow the attacker has compromised a number of computer systems on the internet. These are the so called slave machines. And attacker is directing the slave machines to mount some kind of a distributed attack on the victim so that each of them will be sending a large number of some kind of packets to the victim. And the victim will be flooded with all those request and finally, a denial of service scenario will take place.

Now, the issue here is that the reason that this kind of attack is much more difficult to detect and pinpoint, you see the attacker might have compromise the slaves, and the slaves are made to mount the attack. But at the point at the time, when the slaves are mounting the attack maybe the attacker is no longer in the network, the attacker may be detaching or removing itself from the network. So, it becomes almost impossible for anyone to look at the attacker when an attack is being taking place, because the attacker is not actually mounting the attack, there are number of slave machines which are mounting the attack on behalf of the attacker, right.

(Refer Slide Time: 05:08)



Bot: Automated program used to perform simple and repetitive tasks.

Botnet: A number of Internet-connected devices, each of which is running one or more *bots*.

Botnets are used to perform distributed denial-of-service attack (DDoS attack), steal data, send spam, and allow the attacker to access the device and its connection.

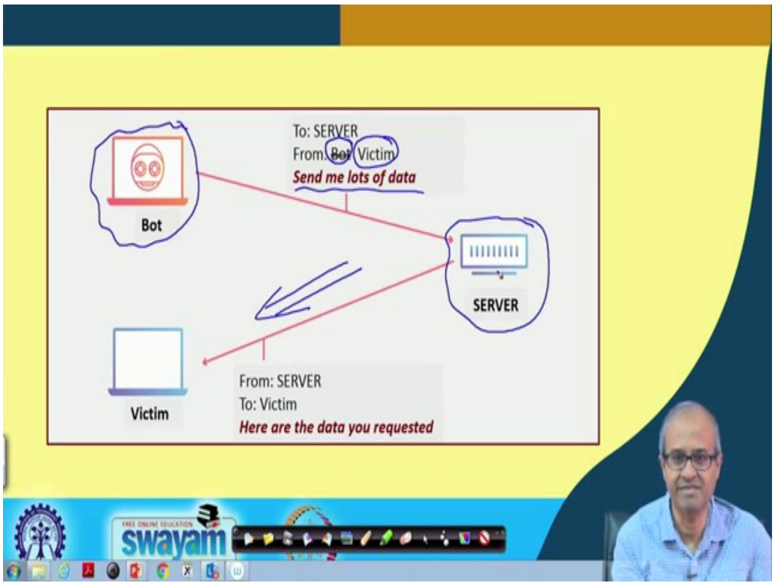
So, this is a slightly more detailed picture. And here we talk about some of the terminology or terms that are frequently used in this context. Well, here on top you see the attacker that is a machine where the hacker is sitting and actually is coordinating the attack. There are some computer systems which are called handlers. Well, handlers again are machines which are under control of the attacker and these handlers they have so called botnets, a large number of botnets under each of them. It is, this handler will be having several botnets, this handler will be having several botnets and so on. And all these botnets together will be mounting the attack on the victim machine, ok.

Now, let us understand these terminologies. Well, attacker is a machine as I told you. Handlers are also computer systems which the attacker will first compromise, all maybe this handlers are machines of some of the attacker’s friends. They have, they have allowed the attacker to use or access those machines. Now, there are some terminology bot. What is meant by bot? Bot is nothing but a small segment of code, a program code which typically runs in a repetitive fashion that is the dentition of a bot. Usually, a bot carries out some repetitive tasks like sending a packet to a victim host, a large number of such packets. In an alternative loop, some repetitive task is carried out. Typically the operation of a bot is relatively simple in nature.

And this botnets that I have shown in the diagram, what is a botnet? Botnet is a number of system, computer systems or any device. It can be your mobile device also, any internet or network connected system you can say and each of these machines are running one or more bots. Somehow this handler has inserted or installed some bots on those machines. Depending on some weak points, the machines are hacked into and bots are installed and this bots are running on those machines. So, the machines on which the bots are running, these are called botnets.

Now, this large number of botnets, botnets can be thousands and even more in number. These botnets are actually used to perform this distributed denial of service attack, the DDoS attack. Large number of botnets are firing packets to the victim, ok. And on behalf of the attacker, all these botnets are working. They are usually mounting an denial of service attack or sometimes can also gather some information from the victim machine depending on the scenario, ok. This is how roughly things work.

(Refer Slide Time: 08:40)



To: SERVER
From: ~~Bot~~ Victim
Send me lots of data

Bot

SERVER

Victim

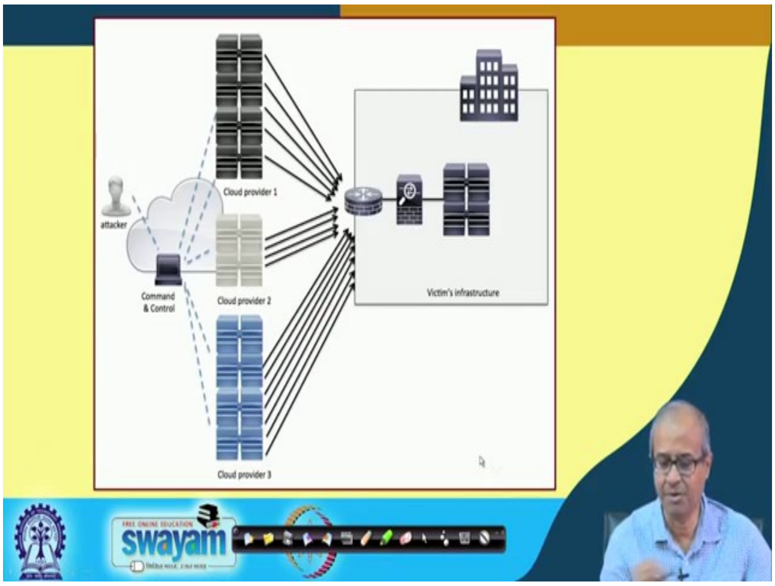
From: SERVER
To: Victim
Here are the data you requested

Now, let us look at a single bot or a botnet. How the bot typically attacks a victims machine out here. You see what a bot typically does is something like this. Let us say bot will not directly send a packet, because it can, but if a bot directly sends a packet then it will be easy to detect. The victim machine, system administrator can detect the packets are coming from that particular IP address and so that particular IP address must be a malicious entity.

But in the contrast, what the bot might do in an intelligent way, a bot is sending some packet to some server, server on the network, but before sending what it does, you know that IP spoofing can be done, it is not a very difficult thing. That in the source address instead of specifying the address of itself, it will specify the address of the victim machine which it is trying to attack. And bot is telling or asking the server to send a lot of data, let us say it is asking it to download a very huge file, let us say a file of 1 gigabyte size. Server who received this request, and it will try to fulfil it, but the destination address is not the address of the bot, but the address of the victim, so that data it will be sending directed to the victim machine.

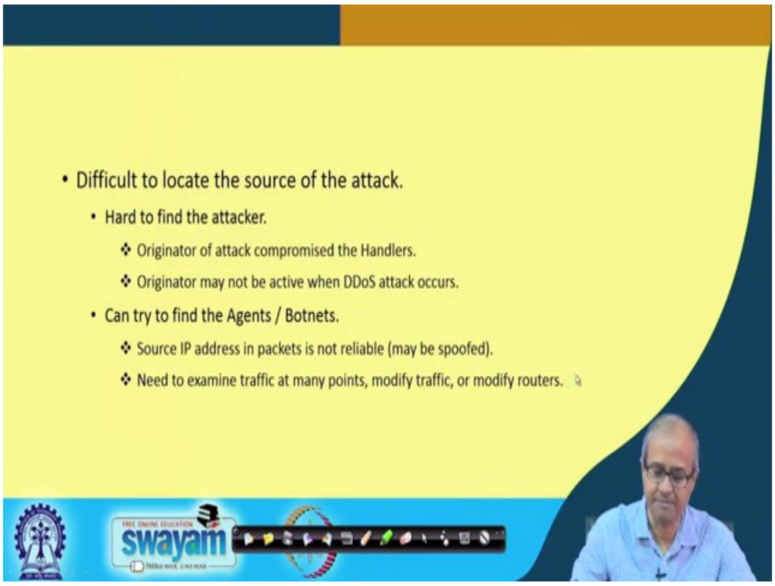
So, there will be many such bot doing something similar and if you try to tress back the attack you will land up into the server, you will not know exactly who was the origin or who was the originator of the attack. This is how in a distributed fashion this kind of attacks can be mounted.

(Refer Slide Time: 10:54)



And the situation can become worse if you have a cloud kind of a scenario; you know in a cloud there are very large number of servers available. So, instead of one server, if the attacker using some kind of command and control mechanism can send that kind of a request to the cloud providers, then all these servers will start sending huge volumes of data to the victim network or the victim machine. So, this will become a much serious kind of attack, because this cloud servers are much more powerful; they can send packets much faster and the victim machine will be overwhelmed with packets or requests in a much more effective and faster way, fine.

(Refer Slide Time: 11:45)



- Difficult to locate the source of the attack.
  - Hard to find the attacker.
    - ❖ Originator of attack compromised the Handlers.
    - ❖ Originator may not be active when DDoS attack occurs.
  - Can try to find the Agents / Botnets.
    - ❖ Source IP address in packets is not reliable (may be spoofed).
    - ❖ Need to examine traffic at many points, modify traffic, or modify routers.

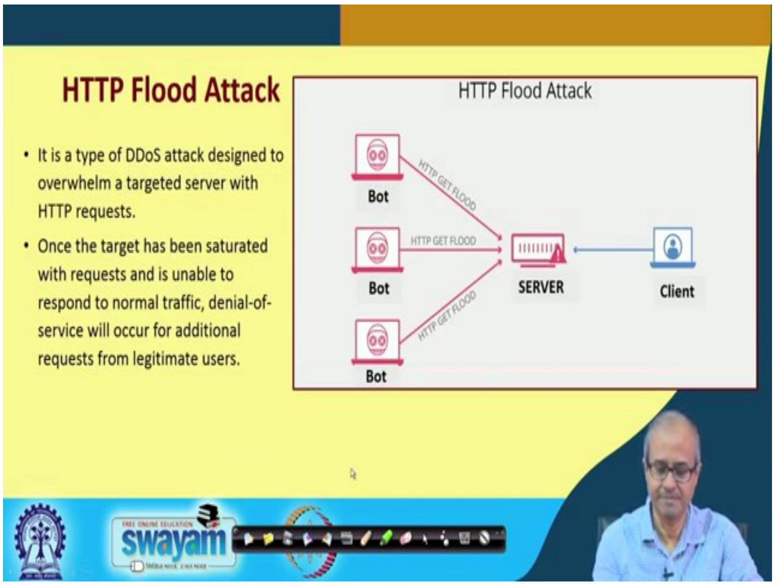
Now, as I said this kind of distributed denial of service attack is difficult in terms of locating the source who was the originator of the attack, where the attacker is located. Because the attacker is mounted by other machines, the attacker is mounted by other machines, the original attacker may not be in the picture at all when the attack is being mounted, ok. So, there are some approaches which may be followed to try and locate the attacker.

Let me see what actually happens. It is really hard to find the attacker is the reason, is the originator of the attack. Firstly, they have used some of the handlers. Those are intermediate machines, and handlers in turn they are using so called botnets, ok. So, this is some kind of a distributed kind of a scenario coming in where the attacker is not in the picture. So, the, and I said the originator may not be active at all when the attack is actually mounted, ok. So, these are some reasons why this attack is typically very difficult to locate and prevent.

Now, what we can do, we can try to find that which are the agents or the handlers which are mounting the attacks and which are the botnets. But again, this is not easy because as the example I showed just a couple of slides back that the source IP address may not be the original IP address of the botnet. It may be spoofed. The IP address may be changed to the IP address of the victim. So, this server, if you look at the server log what are the packets that came to the server, you cannot locate the botnet machines because the source address was modified, ok.

So, in order to find out you may need to examine traffic at many points in the routers, different ports of the routers, on which port more number of packets are coming. So, you will be using that link as a suspect, go to the next level router, again find out in that router from which point more number of packets are coming, in that way, you may try to systematically try and locate where the botnets are, ok, but it is not easy, it is rather difficult. And it requires some kind of collaboration between a number of organization, because all these routers may not be belonging to you, right.

(Refer Slide Time: 14:48)



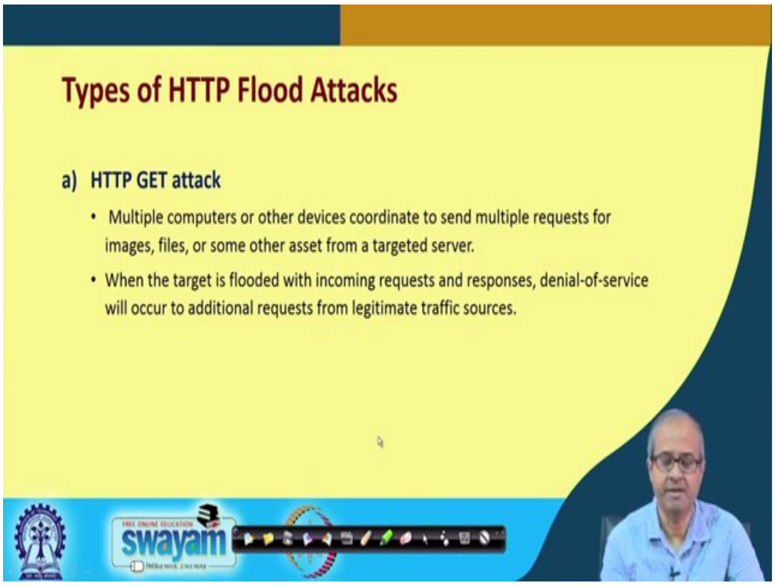
HTTP Flood Attack

- It is a type of DDoS attack designed to overwhelm a targeted server with HTTP requests.
- Once the target has been saturated with requests and is unable to respond to normal traffic, denial-of-service will occur for additional requests from legitimate users.

So, there is another kind of an attack which is based on HTTP request and response. This is called HTTP flood attack. This is also some kind of a denial of service attack, distributed denial of service. So, as it said, this is also a type of DDoS attack, but here the idea is we are flooding the target server with HTTP requests. We are asking for some websites or we are submitting some forms to a web server something like that. Typical HTTP requests that are formed; we are using something like this.

Now, the concept is similar by sending a large number of HTTP requests like in this picture you see, these bots are all sending HTTP requests to a server. And the server is becoming flooded with these requests. And if there is a legitimate client on the other side who is actually trying to use the server, the client will find that the server is not accessible. It is become very slow, because of this large number of HTTP requests that these bots are sending to the server, ok. So, it is mentioned here once the target has been saturated with this kind of HTTP requests, denial of service will occur; legitimate requests cannot be serviced, fine.

(Refer Slide Time: 16:26)



Types of HTTP Flood Attacks

a) HTTP GET attack

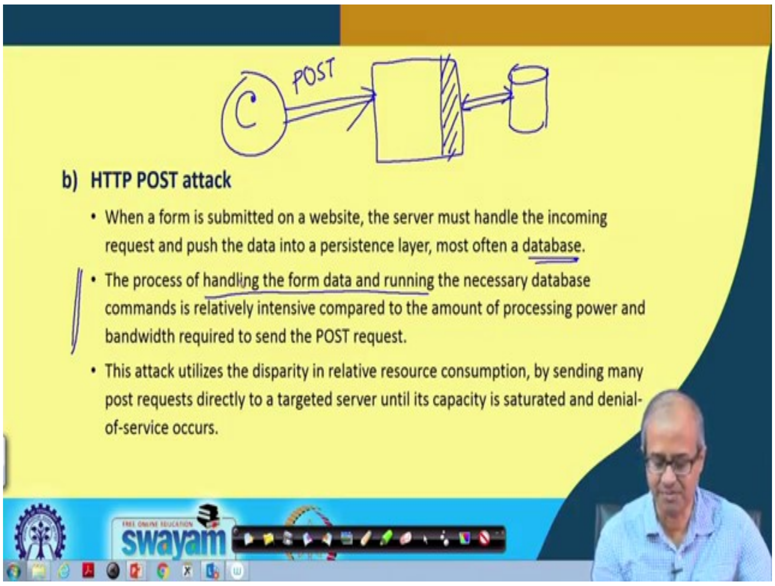
- Multiple computers or other devices coordinate to send multiple requests for images, files, or some other asset from a targeted server.
- When the target is flooded with incoming requests and responses, denial-of-service will occur to additional requests from legitimate traffic sources.

Now, there can be two types of HTTP flooding that you can think of. You know in HTTP if you know about the HTTP request mechanism, there are two kind of HTTP requests; HTTP GET, HTTP POST. In GET you are requesting for a web page from a server; in POST you are submitting a form. Like when you see a webpage, there is some form some time you are asked to type username, password, you type and click on enter or login. So, these data gets submitted on the web server and there is a back end program or database there, which handles these formed requests, that is called POST, ok.

Now, this GET attack is simpler; multiple computers or other devices just as it happens in distributed denial of service, they coordinate among each other, ok. And they sent multiple GET request to a web server; it is a, thousands of computers are sending requests to download some web pages. So, the web server will become very busy to just return those web pages to all those requests, all those clients who are sending the requests, ok.

Now, in this way, the target web server will become flooded with incoming requests, and in this way denial of service can happening. Now, here also this bots are, botnets can be used to mount this kind of attack, because in a loop it will be just generating a large number of HTTP GET request, thousands and millions of requests, ok. So, the server will get flooded.

(Refer Slide Time: 18:15)



b) HTTP POST attack

- When a form is submitted on a website, the server must handle the incoming request and push the data into a persistence layer, most often a database.
- The process of handling the form data and running the necessary database commands is relatively intensive compared to the amount of processing power and bandwidth required to send the POST request.
- This attack utilizes the disparity in relative resource consumption, by sending many post requests directly to a targeted server until its capacity is saturated and denial-of-service occurs.

Now, the POST attack is sometimes a more effective kind of an attack. Why, because you see in a GET attack, you are just asking the web server for a web page, the web server will locate that requested entity from the file system and it will return it. But in PSOT as I said, what happens? Suppose this is your web server; this is your web server, and in the backend there is a database; there can be a database. And there is a client machine out here which is sending a request or submitting a form.

So, when the client submits a form to the web server, the web server retrieves the data that was submitted in the form, and there is a backend program or code which runs. It can be written in Java, JavaScript, many ways are there, PHP. This code is used to access the backend database, possibly the database is updated or some information is retrieved from the database.

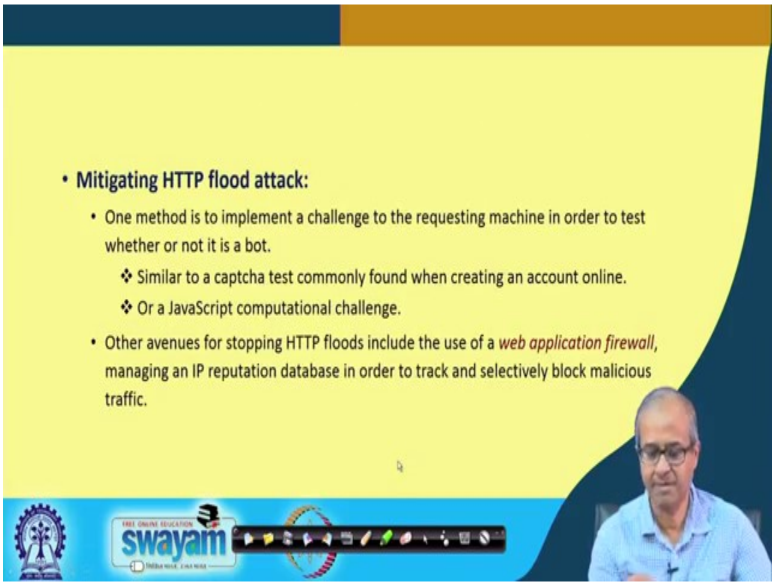
So, you see there are lot more number of operations involved in a POST request as compared to a GET request. GET means just simply load a file and return it, but here you have to run a program, access a backend database and do the needful. It takes much longer time.

So, this is what is mentioned here. When a form is submitted on a website, the server must handle the incoming requests as I said and push the data temporarily in a persistence layer which is typically a database. Data are normally stored in a database when this kind of form submission is implemented, ok.

Now, this process as I said, this takes more time, ok. So, the computation and complexity of handling the form data is much more, because you have to run a program, you have to access a database. But in the contrast, the client sending a post request; this is not difficult at all. Just a single command, it is sending.

So, the complexity of the command the client is sending is very simple, but the complexity of the task the server has to do is more complex. So, there is some kind of an asymmetry. And this asymmetry is taken to advantage, to mount this kind of an attack. Large number of clients can send a large number of such post requests and the database and this backend engine becomes extremely busy and the web server becomes almost unusable, ok.

(Refer Slide Time: 21:18)



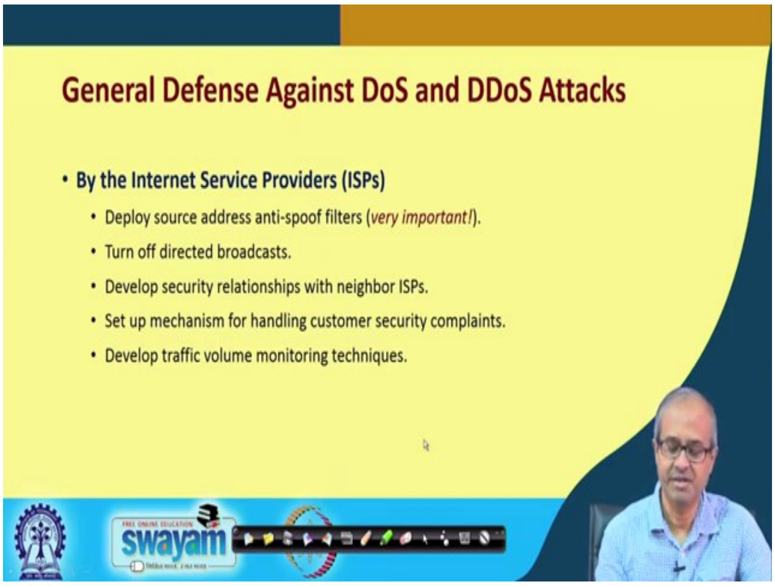
- **Mitigating HTTP flood attack:**
  - One method is to implement a challenge to the requesting machine in order to test whether or not it is a bot.
    - ❖ Similar to a captcha test commonly found when creating an account online.
    - ❖ Or a JavaScript computational challenge.
  - Other avenues for stopping HTTP floods include the use of a *web application firewall*, managing an IP reputation database in order to track and selectively block malicious traffic.

So, in order to mitigate this kind of HTTP flooding attack, well one method maybe to implement some kind of a challenge to the machine who is requesting for service, because you have to detect, try and detect whether it is a legitimate client or it is some kind of an automated system like, a botnet. So, to identify whether it is a bot, there are multiple ways you already know or you are familiar with like, you must be familiar with the captcha.

So, in many case, in many websites whenever you want to do something, submit a form, the system displays an image and ask you to type what you see. That is called an captcha. Or in some other machine like in online reservation system, you have seen some time that there is something similar to a captcha, but it is trying to ask you a simple mathematical puzzle. 2 plus 4 is how much? You have to type 6, so usually that is implemented using JavaScript, ok. This kind of a thing if you do, then automated bots may not be able to I mean, answer to that captcha questions on or this kind of a mathematical puzzles, ok. So, they might get mitigated that way.

And the other thing is that you can use some kind of a web application firewall, ok. And this firewall can maintain a reputation database that, which IP addresses are trustworthy, which IP addresses are suspect depending on past history, and reputation this database is maintained. So, whenever such a request comes, you can check against the database whether it is coming from a reliable source or a source which you do not trust that much, ok. These are some ways in which these kind of distributed attacks can be at least reduced, not eliminated, reduced.

(Refer Slide Time: 23:29)



General Defense Against DoS and DDoS Attacks

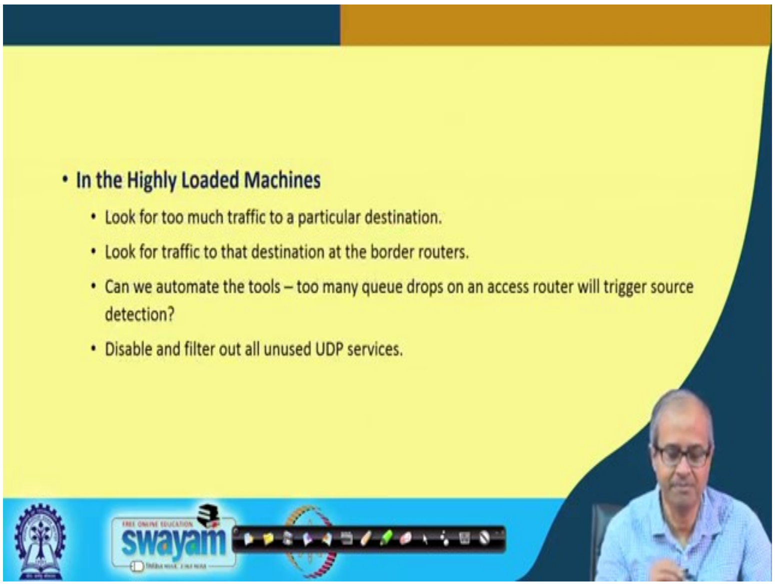
- By the Internet Service Providers (ISPs)
  - Deploy source address anti-spoof filters (*very important!*).
  - Turn off directed broadcasts.
  - Develop security relationships with neighbor ISPs.
  - Set up mechanism for handling customer security complaints.
  - Develop traffic volume monitoring techniques.

Now talking about general defence; you have, you can take some general measures of course, like the internet service providers from where you take your internet connections.

They can employ some anti spoof filter. IP spoofing must not be allowed. A packet which is coming from a particular IP address must have that particular source address in the source address field of the IP packet. If someone is spoofing, the router can immediately detect if it is configured that way.

Broadcast address is one way in which a large number of packets can be sent to some targeted host. So, the router can be configured to turn off this kind of broadcast things. It will not broadcast a packet to a large number of hosts or something like that, ok. Not only the ISP, the other neighbouring ISPs you have to be in constant touch with them, if a neighbouring ISP finds something suspected they will inform you so that you can also take appropriate measure. And you should continuously monitor traffic volumes; whenever you see the traffic volume on some of the links, are increasing in an abnormal way, you can guess that some kind of attack is going on, is being mounted, ok.

(Refer Slide Time: 24:59)

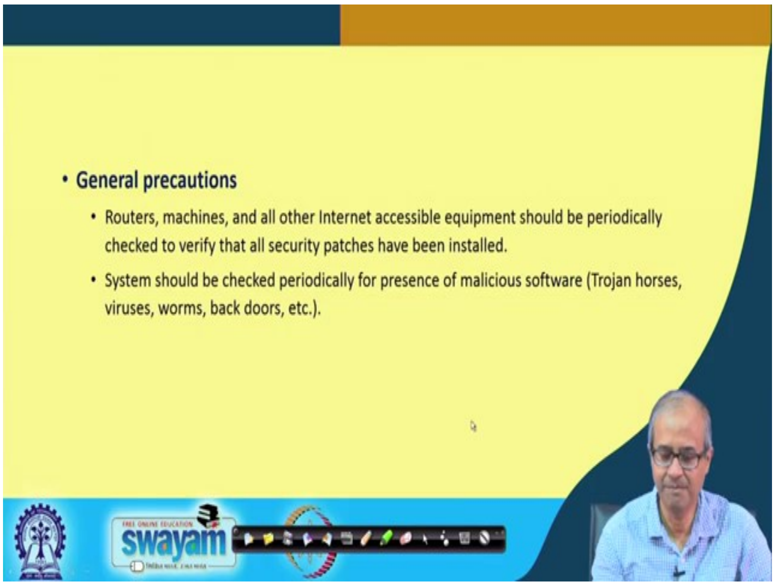


- In the Highly Loaded Machines
  - Look for too much traffic to a particular destination.
  - Look for traffic to that destination at the border routers.
  - Can we automate the tools – too many queue drops on an access router will trigger source detection?
  - Disable and filter out all unused UDP services.

In terms of the machines where the attack has been mounted, the victim machines, well here you can look for a few things like, you can look for too much traffic which is going to a particular, definitely particular destination IP address or some traffic to that destination. You can look at the border routers from the router whether some traffic is coming to that particular target machine. Or you can look at some router queues. You know in the router in all of the links there are some queues. If a large number of packets are coming on one of the, such links of the router, router will not be able to handle all the requests and some of the requests will be discarded, the queue is overflowing.

So, that kind of a thing also you can keep track of suddenly there is lot of packets dropping in some of the queues, must be on that link some attack is being mounted. You can get this kind of a information from this. And usually this UDP services are easier to spoof. So, you can stop all unused UDP services or only the services that are currently being used, leave only those ports open, all other ports you close, these are some of the measures.

(Refer Slide Time: 26:29)



- **General precautions**
  - Routers, machines, and all other Internet accessible equipment should be periodically checked to verify that all security patches have been installed.
  - System should be checked periodically for presence of malicious software (Trojan horses, viruses, worms, back doors, etc.).

General precaution is that all the machines including routers, gateways, everything, you should ensure that proper and latest security patches have been installed on a regular basis. And each individual system on the network, you must periodically check for the presence of malicious software, Trojan viruses, worms. Because those machines may be the platform through which the attacks might get mounted. So, you must ensure that the machines remain clean through regular and periodic checking.

So, with this we come to the end of this lecture, where we discussed some mechanisms for denial of service attack, particularly the distributed version, distributed denial of service attack.

Thank you.