Network Security

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

Week - 10

Lecture - 57

Firewalls and Intrusion Detection Systems: Part 7

Hello, recall that in the previous few lectures, we discussed the concept of an intrusion detection system, and we discussed the different types of IDSs. We will now discuss distributed denial-of-service attack detection and prevention. So, we can either take preventive measures at the host, which is the victim of the distributed denial-of-service attack, or we can take measures in the network. So, first, let us discuss preventive measures at the host. Later, we'll discuss measures that can be taken in the network.

So, we will focus on the SYN flood attack. Recall that in the SYN flood attack, the attackers send a large number of TCP SYN packets, which is the first packet in the TCP connection setup. So, recall that to set up a TCP connection, three packets are sent: a TCP SYN packet, then a SYN-ACK packet, and an ACK packet. So, in the SYN flood attack, the attackers send a large number of TCP SYN packets without completing the third step of the TCP three-way handshake. Because of this flood of SYN packets, the server's connection resources become exhausted since they are allocated but never used for half-open connections.

This causes legitimate clients to be denied service. So, this is a SYN flood attack, and we now want to discuss measures to defend against and prevent this attack. So, one way the host can defend against them is the host just drops incoming requests for TCP connections if it suspects that it is being sent by an attacker. But the key question is, how can a host distinguish between legitimate TCP connection requests and those from attackers? So, the host classifies source IP addresses into different categories.

For example, almost certainly genuine IP addresses, probably spoofed IP addresses, and so on and so forth. So, the host classifies source IP addresses into these categories. So, what are almost certainly genuine IP addresses? These are those addresses with which normal connections were established and terminated in the past. So, if a particular host had received earlier a TCP connection request from a particular IP address and then there was

a normal connection established and terminated in the past, then if another connection request is received from the same IP address, it is likely that it's a legitimate request for a TCP connection.

- One way host can defend against them:
  - ❑ host drops incoming request for TCP connection if it suspects that it is being sent by an attacker
- Host classifies source IP addresses into:
  - ❑ "almost certainly genuine", "probably spoofed", etc.
- "Almost certainly genuine" addresses are those with whom normal connections were established and terminated in the past
- Under moderate load conditions, all incoming SYN requests are served

So, such IP addresses are in the category "almost certainly genuine." So, under moderate load conditions, all incoming SYN requests are served. That is, SYN requests from source IP addresses of all categories. They are served under moderate load conditions. That's because plenty of bandwidth is available to serve all the SYN requests.

But when the load rapidly increases, then SYN requests with unfamiliar source addresses are discarded with high probability. So, if there is a SYN request from an IP address with which no normal connection was established in the past, then it is a suspicious IP address. It may not be legitimate. And since the load is anyway high, so such SYN requests are discarded with high priority. But there is a shortcoming of this strategy that SYN requests from some legitimate clients who are connecting for the first time may be dropped.

So, there may be some legitimate clients who have not interacted with this host in the past, but they are legitimate clients. So, SYN requests from such legitimate clients who are connecting for the first time may be dropped. So, that's the shortcoming of this strategy. So, the core principle is that it's difficult in general to distinguish between legitimate source IP addresses and malicious source IP addresses. So, hence, this strategy suffers from this shortcoming.

Another different strategy is the following. The receiver of a SYN packet allocates buffers for the TCP connection request only upon completion of the three-way handshake. That is, once the SYN packet, SYNACK packet, and ACK packet have been exchanged, only after that point does the receiver of the SYN packet allocate buffers for the TCP connection request. While the connection is still half open, that is when the SYN packet has been sent, but the SYNACK and ACK packets have not been sent. So, while the connection is still half open, minimal information about it is stored in a data structure called the SYN cache.

So, just some limited amount of information is stored, which includes the TCP sequence numbers and source/destination IP addresses and port numbers. So, the complete information about this connection is not stored to save space. So, this strategy reduces the amount of storage required for each half-open connection. So, a shortcoming is that a small amount of information still needs to be stored for each half-open connection, which occupies space at the server. So, if attackers open a large number of connections with the victim, in that case, since a small amount of information is stored for each half-open connection, the half-open connections will collectively exhaust the buffer space of the victim.

So, that's the shortcoming of this strategy. A better solution is to use SYN cookies, which we discussed earlier, under which no information whatsoever needs to be stored for a half open connection at the server. So, recall that in that solution the recipient of a SYN request calculates a cookie and then the initial sequence number of the response of the host is the cookie. So, we discussed that no information whatsoever needs to be stored for a half-open connection when SYN cookies are used. So, that is a better solution to defend against SYN flood attacks.

We have discussed preventive measures at the host, that is, at the host to which the SYN requests are sent. Now let us discuss preventive measures in the network. So, the schemes which we discussed on the previous slides, they help prevent memory exhaustion at the victim's machine. But they do not help reduce incoming attack traffic, which may cause the victim's network link to saturate. So, only after the SYN requests have been received by the host, some measures are taken to defend against SYN flood attacks.

But SYN requests are transmitted on the victim's network link, that is the link connecting the victim's computer to the rest of the internet. So, the victim's network link may become saturated. So, that's a limitation of preventive measures inside the host. So, we discussed preventive measures inside the network. We investigate approaches that seek to throttle attack traffic near the source or in the core of the network well before it enters the victim's network.

So, this way we can prevent the victim's network link from saturating. So, one technique to perform filtering inside the network is egress filtering. Recall that most distributed denial of service attack packets use spoofed source IP addresses. We discussed that initially the attacker compromises a large number of machines and enrolls them in a botnet, and then simultaneously attack packets are sent from all the machines in the botnet. This is used to

make it difficult for intrusion detection systems to pinpoint the true source of the attack and block it.

The egress router is the last router encountered by any packet generated inside the network before it exits that network and enters the internet. So, to illustrate this, consider a network and this is the gateway router of the network. This is connected to the rest of the internet. So, the internet is here. So, this is the internet and this is the organization's network and this is the last router that is encountered by any packet generated inside the network before it exits that network.

For example, consider some hosts over here. They send packets via this path, and then this router is the last router encountered on the path towards the destination. So, egress router is this last router, that is this one in this picture, which is encountered by any packet generated inside the network before it exits that network and enters the internet. Let $\mathcal{A}$ be the set of all externally visible IP addresses within the network that is behind the egress router. So, for example, this host might have an externally visible IP address.

So, the IP address of this host is included in the set $\mathcal{A}$. The egress router examines the source IP address of each packet leaving the network. That is, it examines the source address of each packet that is going to be transmitted on this link. And if the address does not match any address in the set $\mathcal{A}$, then it drops the packet. So, it's a very simple scheme. $\mathcal{A}$ is the set of all externally visible IP addresses within the network.

So, the source IP address of every packet leaving the network should belong to $\mathcal{A}$. But if the source IP address does not match any address in $\mathcal{A}$, then the egress router drops the packet. So, if any spoofed packets are sent, then they are dropped by this egress router. They do not travel into the internet and cause harm to victim computers. So, by thus detecting and filtering spoofed packets, it helps prevent distributed denial of service attacks. Notice that not all spoofed packets will be detected.

For example, if a host in the set $\mathcal{A}$ sends a packet with a spoofed IP address and the spoofed IP address is another address that belongs to the set $\mathcal{A}$, then that packet won't be detected because the egress router will examine the source IP address and find that it belongs to the set $\mathcal{A}$, so it will not drop the packet. So, even though the packet is spoofed, it will not be detected in this case; so hence, not all spoofed packets are detected. So, what is the shortcoming of egress filtering? The shortcoming is that it is unlikely to be universally deployed. Not all internet service providers will introduce egress filtering in their gateway routers because there is not always sufficient motivation for an ISP to implement egress

filtering because the ISP sees no incentive in forestalling a distributed denial of service attack on someone else's server.

So, if egress filtering is implemented at this router, then that will prevent packets from causing harm to victims on the internet. But there is no direct benefit to the ISP itself. So, for this reason, there is little incentive for the ISP to implement egress routing in this gateway router. So, there is this shortcoming of egress routing. There isn't enough motivation for ISPs to implement it.

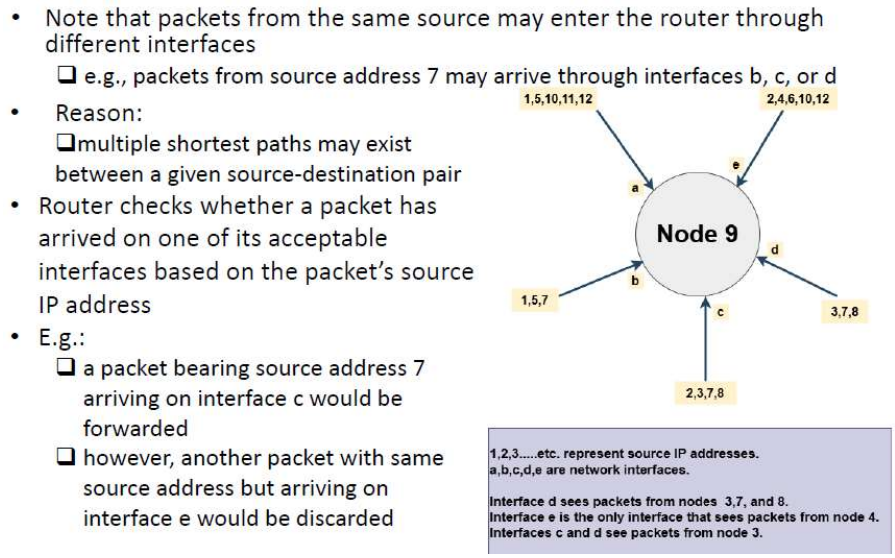
The idea of egress filtering has also been extended to routers in the core of the internet. That is, routers through which packets travel when they are sent into the internet. So, this idea is called distributed route filtering (DRF). We'll discuss DRF next. We call a core router that performs filtering of spoofed packets a filter.

A filter uses a packet's source IP address to make a decision on whether or not to discard the packet. So, to implement DRF, a filter maintains for each of its interfaces the set of all source IP addresses from which packets arrive en route to some destination. So, consider a router; in general, it has many interfaces, which are shown over here. So, to implement DRF, this filter—which is a router—maintains for each of these interfaces (that is, this interface, this interface, this interface, and so on). It maintains for each of these interfaces the set of all source IP addresses from which packets arrive en route to some destination. For example, for this interface, it maintains the list of all source IP addresses from which packets may arrive to this router from this interface.

So, how does the filter get to know which source IP addresses correspond to which interface? So, the filter uses Border Gateway Protocol, which is the routing protocol used on the internet for inter-autonomous system routing. So, the filter uses Border Gateway Protocol (BGP), routing information to obtain the latest mapping between each of its interfaces and the subset of source IP addresses using that interface. So, using BGP routing information, it finds which source IP addresses will send packets that will arrive on this interface. So, now how does the filter perform the filtering decision?

If a packet with source IP address S arrives via an interface it should not have, then the packet is assumed to be spoofed and is discarded. So, suppose a packet with source IP address S arrives via this interface, but this source IP address, S, does not belong to the set of addresses from which packets should arrive on this interface. So, in that case, the packet is assumed to be spoofed and is discarded. So, that's the basis on which filtering is performed in distributed root filtering. So, it's a very simple concept.

So, the router just maintains the set of addresses that is the set of IP addresses which correspond to each interface and if a packet from outside that set is received along that interface then the packet is discarded. So, here's an example to illustrate this concept. This shows an example of a filter implementing DRF. So, this is a filter, and these are its interfaces. So, these numbers—1, 2, 3, and so on—represent source IP addresses.



So, for example, the set of source IP addresses that may be received from this interface are 1, 5, 10, 11, 12. The set of source IP addresses that can be received from this interface are 1, 5, 7, and so on and so forth. A, B, C, D, E - these represent network interfaces. So, in this example, this interface D only sees packets from nodes 3, 7, and 8. So, that is illustrated here, and so on and so forth.

So, each interface marked with the source IP addresses that use that interface en route to some destination. For example, this interface is marked with 1, 5, and 7 as the source IP addresses which use that interface en route to some destination. One observation is that packets from the same source may enter the router through different interfaces. For example, packets from source IP address 7 may arrive through interfaces B, C, or D. So, we can see that 7 is included in the list of interface B; 7 is also included in the list of interface C and interface D. So, packets from the same source IP address may arrive from different interfaces. Why can this happen?

So, the reason is that multiple shortest paths may exist between a given source/destination pair. So, the source IP address is somewhere here. For the source IP address 7 is somewhere

here. There may be multiple shortest paths to a given destination, which is over here. So, one possible route is this one.

Another route is this one, and another route is this one. So, all these routes are the shortest paths to this destination from the source IP address 7. So, there may be multiple shortest paths between a given source/destination pair. So, now the way that DRF operates is as follows. The router checks whether a packet has arrived on one of its acceptable interfaces based on the packet's source IP address.

For example, if a packet with source IP address 7 is received, then it should be received from one of these three interfaces: B, C, or D. If it is received from interface A, then it is a spoofed packet and should be discarded. So, another example is this. A packet with source IP address 7, which arrives on interface C, would be forwarded. So, if a packet with IP address 7 is received from this interface, C, since 7 is included in this list: 2, 3, 7, 8, Hence, the packet would be forwarded.

But another packet with the same source IP address, 7, but arriving on interface E, would be discarded. So, 7 is not included in this list: 2, 4, 6, 10, 12. So, if a packet with source IP address 7 is received from this interface, then it is discarded. But distributed root filtering has some shortcomings. Recent research studies have found that if about 18% of the core routers on the internet implement DRF, then excellent coverage against distributed denial-of-service attacks is obtained.

But since the internet is made up of thousands of core routers, the number of core routers in an absolute sense that need to implement DRF is still a high number. So, it is expensive to implement DRF. So, 18% of all the core routers in the internet need to implement DRF. Since the number of core routers is large, hence, a lot of routers need to implement DRF. So, that increases the cost.

Also, the efficacy of DRF depends on how fast BGP route updates are disseminated. So recall that to use DRF, a router needs to know which source IP addresses correspond to which interface. That is, it needs to know accurate BGP route information. Hence, the efficacy of DRF depends on how fast BGP route updates are disseminated. Routing information changes in response to failed nodes or congested links.

So, if these updates are not distributed fast, then the decision taken on whether to filter a packet or not will be inaccurate. So, this information in turn decides whether an incoming packet at a router should be filtered out or not. Hence, the efficacy of DRF depends on how

fast BGP route updates are disseminated. If a wrong decision is taken, then that could discard legitimate packets in addition to spoof ones. Hence, BGP route updates should be disseminated fast, and if there is a delay in their dissemination, in that case, DRF may take wrong decisions.

So, that's another shortcoming of DRF. So, in summary, we discussed various preventive measures at the host to defend against SYN flood attacks. Then, we discussed preventive measures inside the network. We first discussed address filtering and then distributed route filtering. So, each of these techniques has some shortcomings, which we also discussed.

We'll continue our discussion of distributed denial-of-service prevention and detection in the next lecture. Thank you.