CS525N: Network Security

Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks

Argyraki† & Cheriton, Stanford U.
(† now at EPFL)

Presented by Can Tatar

February 13, 2012

The problems that render DDoS traffic hard to filter

- Source address spoofing: An attack source often uses multiple fake source IP addresses to send its traffic.
- Large number of attack sources: Each hardware router has only a limited number of filters that can block traffic without degrading the router's performance (i.e., filters operating at wire speed).
- Pushing filtering into the Internet core does not scale: It introduces end-to-end filtering state into core routers.

Motivation

- There are enough filtering resources in the Internet to block such large-scale attacks.
  - An attack coming from thousands of different networks involves thousands of routers; assuming each router contributes a few thousand filters, there are millions of filters available to block attack traffic.
  - The closer we get to the attack sources, the larger the amount of filtering resources available per attack source.

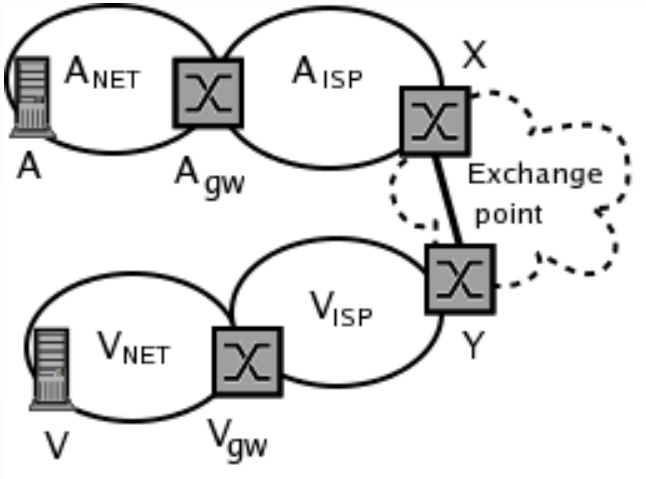
Solution: AITF

- A DDoS filtering mechanism that overcomes these problems.
- An AITF-enabled receiver uses the routes recorded on incoming packets to identify the last point of trust on each attack path and causes attack traffic to be blocked at that point—i.e., as close as possible to its sources.

Route record

- A router that participates in a route record (RR) scheme writes its IP address on each packet it forwards.
  - Only border routers participate in RR.
- As a result, each packet carries the identities of a sub-list of the border routers that forwarded it.

Route record



Packets sent by host A to host V carry recorded route $\{A_{gw}\ X\ Y\ V_{gw}\}$.

Identifying distinct flows

- A DDoS victim feeds recorded paths into a local policy module, which classifies incoming traffic in distinct flows, decides which ones are undesired and forms filtering requests against them.
- It is up to the policy module to classify incoming traffic in multiple "flow levels," in order to identify undesired flows in the face of source address and path spoofing.

Identifying distinct flows

- Consider that attack source A is sending high-rate traffic to victim V.
  - If network A_{NET} prevents source address spoofing, V can easily identify F1{A A_{gw} X Y V_{gw} V} as a high-rate flow and, thus, undesired.
  - If A is able to spoof multiple source IP addresses, V can only identify F2{* A_{gw} X Y V_{gw} V} as the undesired flow.

Blocking close to the attack source

- AITF involves 4 entities:
  1. The victim V sends a filtering request to V_{gw}, specifying an undesired flow F.
  2. The victim's gateway V_{gw}:
     1. Installs a temporary filter to block F for T_{tmp} seconds.
     2. Initiates a 3-way handshake with A_{gw}.
     3. Removes its temporary filter, upon completion of the handshake.

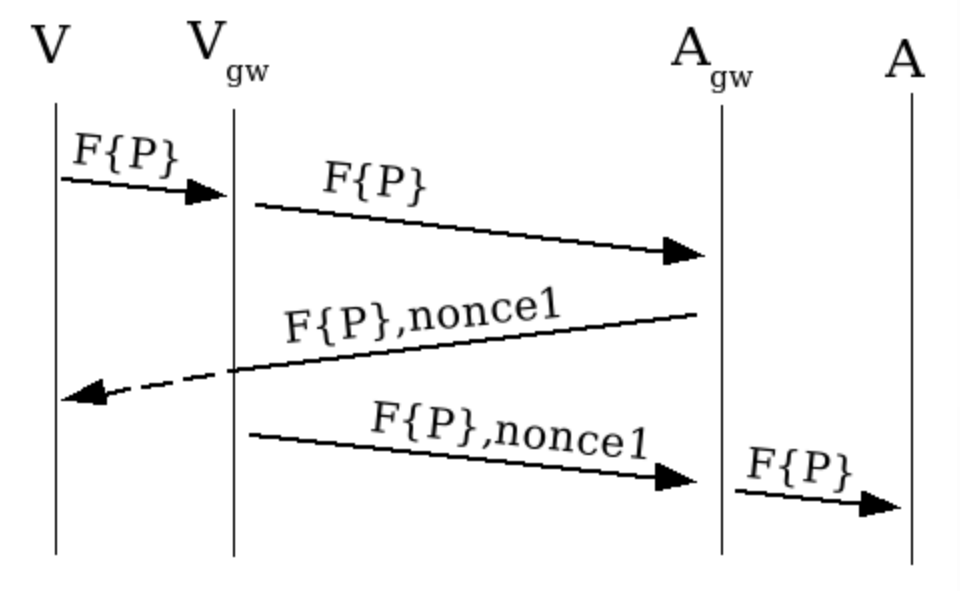
Blocking close to the attack source

3. The attack gateway A_{gw}:
    1. Responds to the 3-way handshake.
    2. Installs a temporary filter to block F for T_{tmp} seconds, upon completion of the handshake.
    3. Sends a filtering request to the attack source A, to stop F for $T_{long} >> T_{tmp}$ minutes.
    4. Removes its temporary filter, if A complies within Ttmp seconds; otherwise, it disconnects A.
4. The attack source A stops F for T_{long} minutes or risks disconnection.

Blocking close to the attack source

- The reason for the *temporary filter* on the victim's gateway is to immediately protect the victim until the attack gateway takes responsibility.
- The reason for *directly contacting the attack gateway* is to avoid creating a filtering bottleneck in the Internet core.
- The reason for the *3-way handshake* is to enable the attack gateway to verify that the requester of the filter is indeed on the path to the alleged victim.

Securing edge-to-edge communication



V_{gw} sends to A_{gw} a request to block F; A_{gw} sends to V a message that includes F and a nonce; V_{gw} intercepts the message and sends it back to A_{gw}.

Identifying liars

- There are two entities that can lie:
  1. An attack source can pause an undesired flow (to avoid disconnection) and resume as soon as the attack gateway has removed its temporary filter.
  2. An attack gateway can pause an undesired flow and resume as soon as the victim's gateway has removed its temporary filter.

Shadow filtering table

- Every time a gateway removes a temporary filter from its TCAM, it creates a copy in DRAM that expires after T_{long}.
- This "shadow filter" helps check whether the corresponding undesired flow is released prematurely (before T_{long}) by its source.

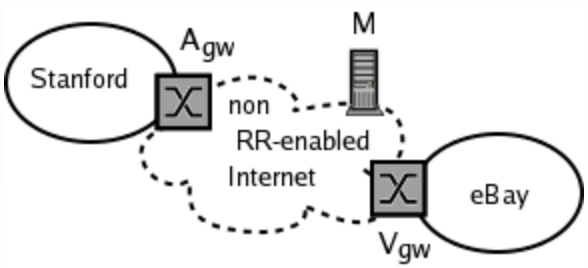
Shadow filtering table

- The victim's gateway uses the same technique to check whether the attack gateway keeps the undesired flow blocked for T_{long} minutes.
- The only difference is that the attack gateway has to be caught violating the filtering agreement twice to be classified as "lying"—the first time could be due to a lying attack source, so the attack gateway is given the benefit of the doubt once.

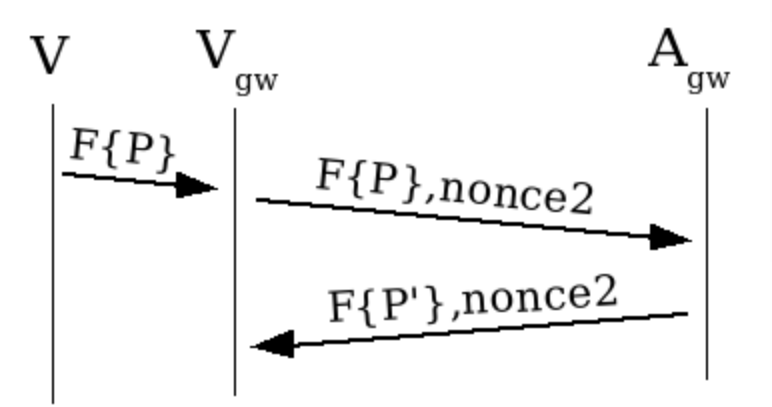
Dealing with non-cooperative gateways

- An attack gateway is classified as “non-cooperative,” if it does not respond to the handshake or responds, but is caught violating the filtering agreement twice.
  - In that case, the victim's gateway can “escalate” the filtering request to the border router that follows the non-cooperative attack gateway on the flow's path.
- The new attack gateway is requested to block all traffic from the last non-cooperative attack gateway to the victim.

Adding resistance to spoofing



Malicious node M pretends to be at Stanford and sends undesired traffic to eBay; its packets carry (spoofed) recorded path {* A_{gw} V_{gw} eBay}.



V_{gw} sends a filtering request against spoofed flow F that appears to be coming from A_{gw}; A_{gw} responds with the authentic path P', which includes the correct random value inserted by A_{gw} in all packets addressed to V. One more nonce is added, to enable V_{gw} to verify that the response with the authentic path is indeed coming from A_{gw}.

Results

- AITF offers filtering response time equal to the one-way delay from the victim to the victim's gateway—i.e., a victim can have an undesired flow blocked within milliseconds.
- It also offers filtering gain on the order of hundreds of blocked flows per used filter—i.e., a router can block two orders of magnitude more flows than it has wire-speed filters.
  - For example, suppose eBay is receiving a million undesired flows; with 10,000 filters, eBay's gateway can have all flows blocked within 100 seconds. In the worst-case scenario, eBay's gateway blocks all traffic from each domain that hosts attack sources and refuses to filter their traffic, which requires a few tens of thousands of filters.
- A set of malicious nodes can practically not abuse AITF to disrupt communication from node A to node B, as long as they are not located on the path from A to B.
  - This holds even during initial deployment, where most Internet domains are AITF-unaware.