**QUICLORIS: A Slow Denial-of-Service Attack on the QUIC Protocol**

**(Anirudh Srinivas Balaji, V. Anil Kumar, P. P. Amritha, and M. Sethumadhavan)**

**QUIC: Key Features:**

Handshake Process:

* + QUIC combines cryptographic (TLS 1.3) and transport parameters into a single handshake.
  + The client sends a ClientHello message, and the server responds with a ServerHello, enabling immediate data exchange.

0-RTT (Zero Round-Trip Time):

* + QUIC supports 0-RTT, allowing the client to send application data (e.g., HTTP/3 requests) during the handshake.
  + This reduces latency significantly but requires prior configurations between the client and server.

Reliability and ACK Frames:

* + Unlike UDP, QUIC adds reliability by including ACK frames to acknowledge received data.
  + These acknowledgments enhance loss detection and congestion control.

HTTP/3 Support:

* + Once a QUIC connection is established, application-layer data is transmitted using HTTP/3, leveraging QUIC's multiplexing capabilities for efficient and fast communication.

**Streams in QUIC**

In QUIC, communication between endpoints (client and server) happens through streams, which are of two types:

* Unidirectional streams:
  + Allow only one QUIC endpoint to send data.
* Bidirectional streams:
  + Enable both endpoints to exchange data in both directions.

Connection Identifiers

QUIC uses connection identifiers to ensure seamless communication in case of network path changes. This mechanism:

* Maintains the connection even after changes in network topology or address mappings, such as during Network Address Translation (NAT) rebinding.

Flow Control to Limit Memory Usage

To prevent excessive memory consumption by malicious users, QUIC employs a limited flow control mechanism:

* The receiver announces the total amount of data it can handle for the entire connection or for a specific stream.
* The frames MAX\_DATA and MAX\_STREAM\_DATA are sent by the receiver to inform the sender about these limits.

Stream Identifiers

* Each stream has a unique identifier (stream ID), which is a 62-bit integer.
* Streams initiated by the client have even-numbered IDs.
* Streams initiated by the server have odd-numbered IDs.
* The two least significant bits of the stream ID are used to identify the stream type.

Data Encapsulation with STREAM Frames

Application data is encapsulated in STREAM frames, which include:

* Stream ID: Identifies the stream to which the data belongs.
* Offset: Specifies the position of the data within the stream, enabling correct reassembly.

If a QUIC endpoint receives duplicate data for the same offset in a stream:

* The duplicate data is ignored.
* The offset value for a specific piece of data must remain unchanged if the data is retransmitted.

Security Considerations

QUIC incorporates several mechanisms to ensure secure communication:

1. Handling idle timeouts:
   * Applications using QUIC must provide appropriate guidance for suspending or managing timeouts when the connection becomes idle.
2. PING frames:
   * Excessive transmission of PING frames can have catastrophic performance impacts by unnecessarily overloading the network.

QUIC aims to reduce latencies in client-server communications by minimizing RTTs through simplified handshakes. Distributed Denial-of-Service (DDoS) attacks target QUIC servers by overwhelming them with massive requests, rendering services unavailable for legitimate users. These DoS attacks are categorized as buffer overflow or flood-based attacks.

* Buffer overflow attacks consume server resources like disk space, CPU, and memory, potentially crashing the system.
* Flood-based attacks inundate servers with excessive packets, saturating capacity and disrupting services.

While some mitigation techniques against QUIC floods are in use, deeper studies are still needed. Mechanisms to reduce latency in QUIC introduce potential security vulnerabilities, such as a lack of forward secrecy and susceptibility to replay attacks.

Smurf DoS Attack

* Description:  
  The Smurf DoS attack exploits the broadcast address of a vulnerable network to send a large number of spoofed packets, flooding the target’s IP address with traffic.
* Mechanism:
  + The attacker sends ICMP packets (used for Ping requests) to the broadcast address of a network.
  + The source address of these packets is spoofed to match the target’s IP address.
  + All devices in the network respond to the broadcast by sending their replies to the target’s IP, overwhelming it with traffic.
* Impact:  
  The target is flooded with a volume of traffic that exceeds its capacity, causing performance degradation or complete service unavailability.

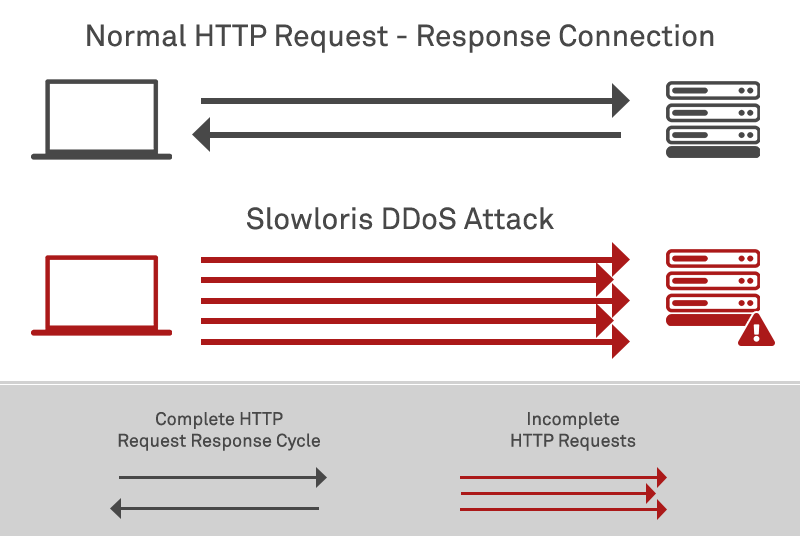
Ping Flood DoS Attack

* Description:  
  The Ping Flood DoS attack aims to overwhelm a target by flooding it with ICMP (Ping) packets. The target becomes unable to process requests faster than its capacity, leading to service disruption.
* Mechanism:
  + The attacker continuously sends Ping requests (ICMP packets) to the target.
  + The target is forced to respond to each request, consuming its network and processing resources.
* Escalation to DDoS:
  + As illustrated in Figure 3, the Ping Flood DoS attack can be amplified into a DDoS (Distributed Denial of Service) attack.
  + The attacker infects other devices (zombies) within the network, which also send Ping requests to the target.
  + This creates massive traffic from multiple sources, making the attack harder to mitigate.
* Impact:  
  The target is overwhelmed by the Ping requests, potentially leading to complete service interruption or system failure.

In addition to overwhelming a target with a large number of requests, another method for executing DoS or DDoS attacks involves keeping the target busy by sending prolonged, incomplete requests. The R.U.D.Y (R U Dead Yet) attack works by sending form data at an extremely slow pace, keeping the target's web server alive and eventually making it unavailable to legitimate traffic. This attack exploits form submission processes, making any web service accepting form inputs vulnerable.

A variant of R.U.D.Y is the Slowloris attack. This involves opening and maintaining multiple simultaneous HTTP connections with partial requests. Using minimal bandwidth, the attacker consumes server resources by sending slow, incomplete HTTP requests. Since servers have a limited number of threads to handle concurrent connections, these threads remain occupied indefinitely, blocking new connections.

1. By sending multiple partial HTTP request headers and HTTP Keep-Alive headers, the attacker first establishes multiple connections to the targeted server.
2. The target server creates a thread for every request sent by the attacker and closes it once the connection is finished. To optimize resources, if a connection takes too long, the server applies a timeout to free up the thread for other incoming requests.
3. By sending Keep-Alive headers, the attacker communicates with the server to prevent it from timing out. If the server does time out, the attacker creates a new connection thread and continues the same process.
4. While the server waits for the requests to complete, it cannot release any of the open partial connections. When all threads are occupied, the server becomes unable to respond to additional requests, resulting in a slow denial-of-service attack (Slowloris).



**The system of the article:**

Locally Configured QUIC Client-Server Architecture:

* + Two clients (Client A and Client B) establish communication through a multi-connection server.
  + Each client sends a CA certificate to initiate a secure connection with the server.

Connection Process:

* + Clients send connection requests to the server.
  + The server responds with an ACK frame, confirming that the communication is successfully established.

Exploitation of PING Frames:

* + PING frames in QUIC are used to check the status of active connections. However, excessive or unnecessary usage of these frames can create a vulnerability.
  + An attacker can exploit PING frames to launch a Slowloris attack, maintaining numerous open connections with slow and incomplete requests.
  + This tactic blocks the server's resources, leading to degraded performance or denial of service for legitimate users.

QUIC is designed for faster communication, particularly with HTTP/3, leveraging its modern architecture based on UDP. However, the Slowloris attack, which keeps multiple connections open with slow requests, can exploit QUIC’s resources and disrupt service availability. Applications using QUIC must address potential security vulnerabilities that can degrade performance. Implementing countermeasures, such as early detection of suspicious activity, limiting suspicious connections, and proactive defense mechanisms, is crucial to prevent denial-of-service (DoS) attacks and maintain reliable service performance.