**Details of attacks from the article: "A hands-on gaze on HTTP/3 security through the lens of HTTP/2"**

**1. HTTP3-flood**

* Description: This attack overwhelms an HTTP/3 server with a massive number of requests or connections, aiming to exhaust its resources (CPU, memory, or bandwidth).
* How it works: Attackers exploit HTTP/3's fast and multiplexed connections to generate excessive traffic that overwhelms the server.
* Global Impact: If successful, legitimate users experience service unavailability or significant delays, potentially leading to a complete denial of service (DoS).

**2. Fuzzing**

* Description: This technique sends malformed or random data to the server to test its robustness and discover vulnerabilities.
* How it works: Malformed requests are processed by the server, potentially causing crashes or exploitable behaviors.
* Global Impact: Successful fuzzing attacks can lead to server crashes, data corruption, or the discovery of security flaws that can be further exploited.

**3. HTTP3-loris**

* Description: A slow-rate attack where attackers open multiple connections with incomplete requests to deplete server resources.
* How it works: The server allocates resources for these connections, keeping them open without finalizing, which prevents handling legitimate requests.
* Global Impact: The server may run out of memory or processing power, leading to a denial of service (DoS) for legitimate users.

**4. HTTP3-stream**

* Description: This attack targets HTTP/3’s stream multiplexing, sending numerous simultaneous streams to disrupt legitimate ones.
* How it works: Attackers overload the multiplexing capabilities by generating many streams, causing contention and delays for legitimate traffic.
* Global Impact: Legitimate users may experience degraded performance or partial service disruption.

**5. QUIC-flood**

* Description: Similar to HTTP3-flood, but this targets the underlying QUIC protocol by sending a high volume of QUIC packets.
* How it works: Attackers exploit QUIC's rapid connection setup (based on UDP) to overwhelm servers with connection or data requests.
* Global Impact: Server resources are overwhelmed, leading to degraded service or a complete denial of service (DoS).

**6. QUIC-loris**

* Description: A slow-rate attack on QUIC, where attackers keep many QUIC connections open with incomplete exchanges.
* How it works: The server allocates resources (e.g., memory, cryptographic states) for these incomplete connections, depleting its capacity.
* Global Impact: Legitimate connections may fail to establish, leading to service unavailability.

**7. QUIC-enc**

* Description: This attack targets the encryption mechanisms of QUIC, attempting to overload the server’s computational resources needed for encryption/decryption.
* How it works: Attackers send numerous connection requests, exploiting the CPU-intensive nature of encryption operations in QUIC (based on TLS 1.3).
* Global Impact: Servers may experience slowed performance or resource exhaustion, potentially leading to a denial of service (DoS).

**8. HTTP-smuggle**

* Description: This attack exploits desynchronization between client-server communication to inject malicious requests.
* How it works: Attackers craft requests that are parsed differently by the server or intermediaries, allowing them to bypass security controls or execute unauthorized actions.
* Global Impact: Sensitive data can be leaked, unauthorized operations performed, or security measures bypassed.

**9. HTTP2-concurrent**

* Description: This attack targets HTTP/2’s concurrent connections, overwhelming the server by opening and maintaining a high number of simultaneous connections.
* How it works: Attackers exploit the server’s ability to handle multiple requests to create contention for resources.
* Global Impact: Legitimate users experience degraded performance or denial of service (DoS).

**10. HTTP2-pause**

* Description: This attack involves sending partial requests and deliberately pausing their completion to hold server resources hostage.
* How it works: Attackers create connections that remain active but incomplete, preventing the server from freeing resources for other users.
* Global Impact: Legitimate requests are delayed or dropped, leading to a significant degradation of service.