h2c Smuggling: Request Smuggling Via HTTP/2 Cleartext (h2c)

How Upgrade is done: For Long lived WebSocket Connection, Upgrade header is often used. Proxy supports this by keeping the original client connection alive and simply proxying TCP traffic to BackEnd Server. Now Proxy is not content aware and can not apply any access control rules

Steps:
1. Client Sends header -> Upgrade:h2c
2. Proxy/Nginx Server forward it to Backend Server
3. Backend Server responds with 101 Switching protocol to Proxy and Proxy forward it to client
4. Proxy keeps the original TCP Connection alive and then refuses it to communicate with the server over the newly build session
5. Now a Persistent is maintained and Proxy no longer monitors the content.

Along with Upgrade header client also needs to send HTTP2-Settings and Connection header
HTTP2-Settings -> Base64 encoded string which contains the value of HTTP2 Connection parameter

Why Upgrade was required :
HTTP/1.1 is plain text, short lived protocol while HTTP/2 is long lived Binary Protocol which allows the client and server to communicate for longer duration over a single connection.

Problem :
Once the connection is established the proxy no longer monitors the content and can not apply any access control rules . Here attackers can take advantage of this and access high privilege content

Example:
Let’s say an application has restricted certain end points for a particular user group. Now if an attacker is able to establish an H2c connection, he can easily access those endpoints.

Prevention : The first involves mandating WebSocket support for HTTP/1.1 upgrade headers. The second is to disable WebSocket support altogether and disable forwarding Upgrade headers.