TURN Server Exploitation

Tristan Wiesepape

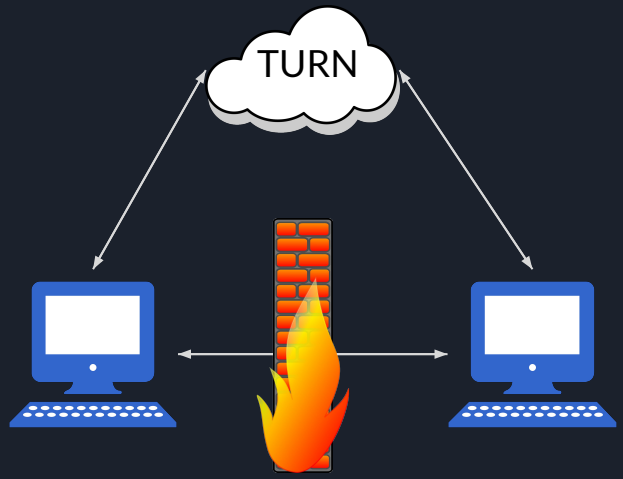
What is TURN

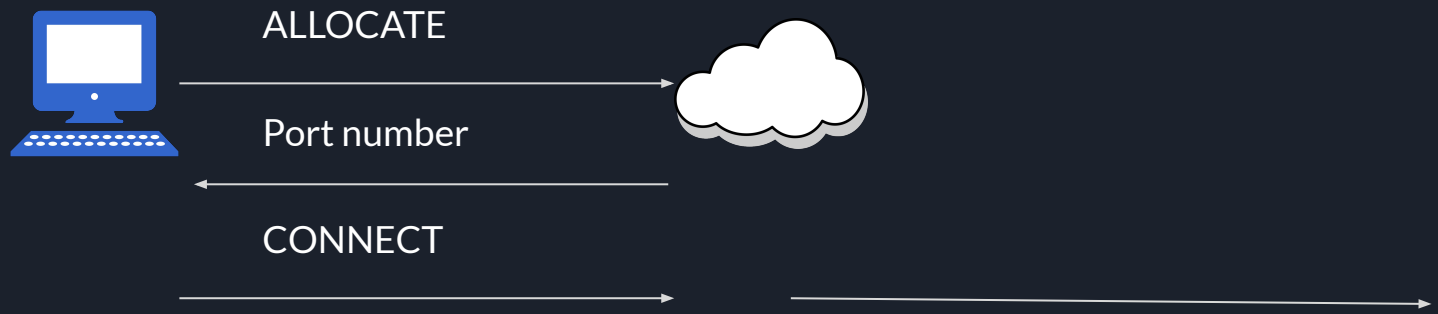
- Proxying service used for VOIP and WebRTC
- Deployed as part of a STUN server
- coturn is the main implementation of TURN
- coturn is vulnerable to several vulnerabilities
- Slack was vulnerable to TURN exploits

What does TURN do?

Normal Peer to peer connection

TURN proxied connection




ALLOCATE
Port number
CONNECT

More about TURN

- TURN servers must have a public IP
- Client devices must have credentials for the TURN server
  - Hardcoded
  - API based time sensitive credentials
- TURN servers can proxy arbitrary traffic
  - TCP, TLS, UDP
- Default coturn configuration is often insecure

Exploit 1: Tunneling

- TURN servers can tunnel arbitrary traffic into the corporate LAN
- Targets can include
  - Cloud provider metadata endpoints (AWS, GCP, Azure)
  - localhost
    - Coturn local management interface
    - Other services running on a shared computer
  - Other vulnerable services running on the LAN
- Often endpoints that aren't supposed to be publicly accessible often have far more lax security practices
- Can be fixed by changing coturn configuration to disallow certain IPs

Exploit 1.5: Tunneling

- TURN servers can tunnel arbitrary traffic to a public endpoint
- This cannot be disallowed
  - It's the reason TURN exists
- This can be used like a public, free VPN
- Illegal traffic could be tunneled through your server to hide its source
- Illegal traffic could seem to have originated from your IP
- Solution is to keep logs of connections

Exploit 2: Port Scanning

- Timing attacks in the behavior of TURN servers
- If you try to connect to an IP and port, there are 3 different possibilities
  - Connection -> Port is open
  - Instant Failure -> Port is closed
  - 30 second timeout -> IP doesn't exist
- This can be used to enumerate internal services

Exploit 3: DOS

- Each connection on a TURN server uses a port
- Default ephemeral port range:
  - 32768-60999
  - About 28,000 ports
- If all ports are consumed the TURN server is no longer functional
- This can be fairly easily done
- The easiest mitigation is to enforce per user limits on allocations

Practical use

- Look for TURN servers on any audio based app (especially WebRTC)
- Check for connections to localhost and management endpoints
- Default coturn configuration is insecure