BOSTON UNIVERSITY

Exploiting WebRTC: An Analysis of Videoconferencing Vulnerabilities

Apollo Lo, Wiley Hunt

Background

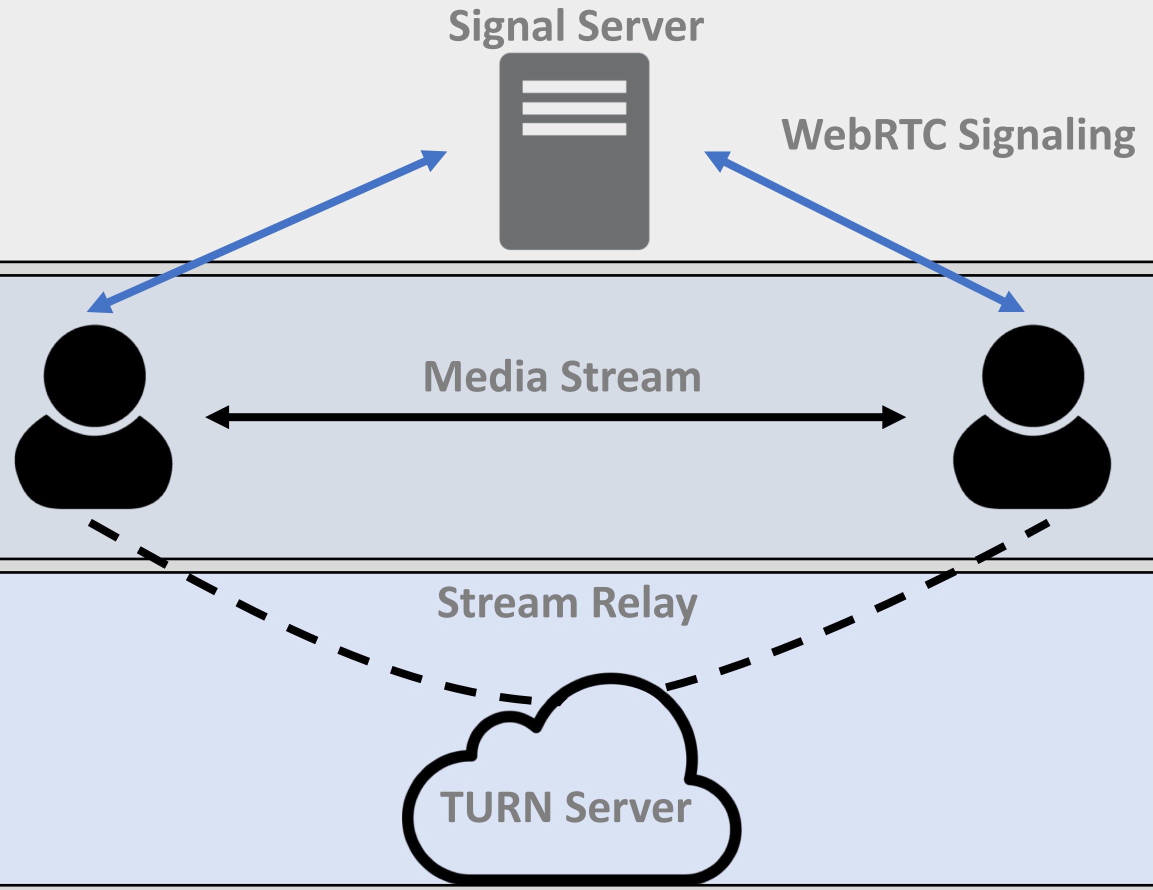
- WebRTC is used by many prominent video-conferencing platforms (Google Hangouts, JitsiMeet, etc.)
- Built-in to most popular browsers (Chrome, Firefox, Safari, etc.)

Focus

- Where is WebRTC vulnerable?
- What sort of attacks could be performed?
- What information could these attacks leak?

Conclusions

- **Vulnerabilities** across signaling, media, and application layers
- Users provide WebRTC services with potentially sensitive information – **trust and system integrity** is key!
- Installing/using open-source packages/tools can lead to **privacy compromise** (monitoring/recording, IP leaks, continued access to camera/mic)
- Future exploration needed of exploits associated with **screen sharing or chat features**



Signaling Attack

- Created compromised signaling server
- Instead of directly connecting peers, connected them to attacker

Results

- Attacker views users' media without them knowing
- Attacker can implement audio/video recording

Session Termination Exploit

- Initiate calls in pop-up window
- On hang-up, minimize and hide window without terminating media session

Results

- Session remains open (though appears closed)
- Other user retains full-access to camera and microphone

TURN Server Compromise

- Noted tendency of "leaked" private TURN servers to appear in open-source code
- Studied information that could be captured by a rogue TURN server

Results

- Traffic passing through server leaks users' IPs and open ports
- Studying packet size indicates number of users and type of media shared