Lessons Learned from WebSockets (RFC6455)

Or, "The World is a Cruel, Broken Place"

Ian Fette, Google

Overview

- Assumed Security Boundaries
- Cross-Protocol Attacks
- Deployed Infrastructure with bugs

Assumed Security Boundaries

- Many security boundaries are undocumented and "assumed"
- Internal traffic often assumed to be authenticated or otherwise wholesome
- WebSocket traffic originates from the browser, which is on the internal network
- Solution: Origin header (RFC 6454) to indicate the true origin of the request

Cross-Protocol Attacks

- WebSockets provided a way to send semi-arbitrary data
- Many interesting targets - SMTP, DNS (on tcp)
- Solution: Make sure you're talking to WebSockets server: Sec-WebSocket-Key + Computation -> Sec-WebSocket-Accept

Deployed Infrastructure with Bugs

- WebSockets uses Upgrade from HTTP
- After "Upgrade" the connection is no longer HTTP
- ... yet some proxies will try to continue to interpret the stream
- malicious.js: "Open WebSocket to evil.com. Send "GET http://www.google-analytics.com/urchin.js", receive "DoEvilStuff()"
- Solution: Masking

Conclusions

- Consider the environment into which you're deploying
- Existing attacks may be "close enough" to something possible with your API
- Lowest common demoninators are sadly important for security