An IoT security solution based on 10 years of experience in VoIP technology

Tim Panton - CTO
|pipe|
@steely_glint
My background

- 2FA app for Y2k rollouts
- Web security startup
- Sidetracked into VoIP
- Web telephony - before webRTC
- WebRTC ‘done’ - what next?
IoT security is a Joke in Germany
Seemed familiar...

- Connected device
- Minimal GUI
- No keyboard
- Couple of sensors
- ‘improved on’ existing thing
Same problems
Same solutions?
10 years of knowledge going to waste?
Pretty much.

@steely_glint
Don’t believe me?

VoIP

IoT

- Hideous config means users just keep defaults

@steely_glint
Don’t leave ports open

VoIP

IoT

Remotely hacking ships shouldn't be this easy, and yet ...

• Open ports mean you’ll get hacked

@steely_glint
Think it won’t be found?

VoIP

IoT

• The hacking tools are even better

@steely_glint
Round-trip via the cloud?

<table>
<thead>
<tr>
<th>Perceived Quality</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>0 to 150 ms</td>
</tr>
<tr>
<td>Good</td>
<td>150 ms to 300 ms</td>
</tr>
<tr>
<td>Poor</td>
<td>300 ms to 450 ms</td>
</tr>
<tr>
<td>Unacceptable</td>
<td>Above 450 ms</td>
</tr>
</tbody>
</table>

- Try and keep traffic local (SIP Reinvite).

@steely_glint
VoIP lessons learned by VoIP.

- Be at least as secure as best-of-breed web security
- Low latency
- Transport security
- Minimize network attack surface (no open ports)
- Use standard protocols and known crypto
WebRTC design

- Strong decentralised crypto
- P2P
- web-based
- realtime
- voice/video/data
- in 2bn+ browser endpoints
- open source impl available
- signalling + identity unspecified
- W3C +IETF standard
  backed by Google, Mozilla, AT+T, Cisco many more…
WebRTC Protocol soup

- STUN - to find your public IP
- ICE - to find a direct path between peers
- TURN - if there isn’t a direct path use a reflector in cloud
- DTLS - encryption + identity (self signed x509s)
- SCTP - multiplexed data streams over DTLS ((un)ordered,(un)reliable)
- SRTP - for low latency media streaming
- continuing consent - for detecting browsers going away
- well documented, implementable, IETF standards
IoT Goals

- Be at least as secure as best-of-breed IoT security
- Network independence (assume IP - it always wins)
- Be easy to use (so users won't dodge around it)
- No username/passwords
- Provide identity management
- Provide transport security
- Minimize network attack surface (no open ports)
- Use standard protocols and known crypto
What many webcams do

- Local on device service
- Cheap for vendor
- Local Auth
- Works on LAN
- Low(ish) latency
- Ports avail to hackers
- Passwords weak or default
Current IoT best practice (Cloud Client)

- Route via a cloud service (or VPN)
- Costly for vendor
- Vendor holds Auth
- Useless without cloud
- High latency
- Breach 1, breach all GDPR risk.
Want a combination (Peer - like VoIP)

- Local on device service
- Cheap for vendor
- Local Auth
- Works on LAN
- Low(ish) latency
- Ports and passwords closed to hackers

@steely_glint
|pipe|

- WebRTC data channel protocol - standards based cryptography
- Zero install client in 4 main browsers (maintained by Google, Apple, Moz + MS)
- Identity managed by self-signed (local) x509s
- Identity verified by proximity with QR or BTLE
Duckling protocol

* Described by Ross Anderson in 1990s
* Device trusts first thing it sees
* We flip this and the device shows QRcode which includes a nonce
* Smartphone then ‘calls’ this address and verifies nonce
* First to connect claims ownership (TOFU)
https://youtu.be/d8Xxiar4_ig
Paired.

- The two ends have now done a full x509 exchange
- Over an e2e encrypted link
- Verified by an out-of-band token
- Takes ~3sec in real life
- Once only
- Extendable to lend/borrow devices
|pipe| fingerprint as address

- Each |pipe| endpoint creates a Public/Private key pair
- It self-signs an X509 cert containing the public key
- It calculates the SHA256 hash of the cert
- It uses this hash as an address/id in a p2p mesh virtual network
- All communication is e2e over DTLS using this key pair
pipe resulting properties

- Mathematical near certainty of uniqueness in namespace -> 2 unrelated endpoints can federate later if introduced

- Endpoints can instantly know provenance of incoming requests and drop traffic from untrusted sources (DTLS hello contains sender’s cert)

- Endpoints are their own auth servers - no central authority needed

- Using Hash as an ID means just 32bytes received over a trusted channel is enough to extend web-of-trust (fits in QRcode, BTLE beacon etc)
|pipe| Cypher suites

- RSA 2048 bit Keys
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

This means that |pipe| supports Perfect Forward Secrecy
Data Channel

- E2E encrypted
- multiple named datagram channels
- Single port pair
- bidirectional
- selectable reliability
- selectable ordering
- zero round-trip creation
- simple web-socket like API
Code for a basic webcam

```javascript
var img; // an `<img>` tag on page
var pipe; // a pipe session

function makeDc(deviceId) {
    pipe.setTo(deviceId);
    camChannel = pipe.createDataChannel("camera", {});
    camChannel.onopen = function() {
        camChannel.send(JSON.stringify({command: "read"})));
    }
    camChannel.onmessage = function(evt){
        var jMess = JSON.parse(evt.data);
        img.src='data:image/jpeg;base64,' +jMess.frame;
    }
}
```
What’s in the user’s hand?
What’s at the other end
Small ARM Linux - running our stack and logic
Some things we built and can talk about

- WebCam
- Doorbell
- Meeting room Presenter
- Interactive sculpture
- Secure proxy for home control
- E2E channel over 2g or IoT NB
- Single purpose podcast box for elderly

@steely_glint
Webcam

- Basic test of pairing etc
- Supports Lend/Borrow
- No passwords, no open ports
- 85% bandwidth saving over cloud
- Pi 3
Doorbell

- Used BLTE to pair and set wifi credentials
- In effect a range-extender for BTLE :-)
- 1 button IVR for set up
- WebRTC audio
- Sends still image
- Pi Zero W
IVR for doorbell

Thanks to
allison@theivrvoice.com
Meeting room presenter

- Displays QR on large HDMI screen
- Short-term pairing
- Screencast from phone
- Phone on user’s 4g device on wifi
- No state left on device
- Pi 3 or Beaglebone black
Interactive sculpture

- Public ‘control’
- Short-term pairing
- Remote access for maintenance
- Random network connections at galleries
- Realtime
- Beaglebone black

@steely_glint
Secure proxy for home control

- Remote access
- Non-cloud
- In progress
- Minimal changes to device code
- Device web server securely proxied via |pipe|
- Beaglebone pocket
E2E channel over 2g or IoT NB

with Lora and SigFox by providing an E2E encrypted mode without the need for a spec APN/VPN setup (in progress)
Podcast player

- 1 button box
- Plays from pre-selected podcasts
- Press button to skip to next
- `|pipe|` for remote podcast list update
- Pi 3 with Google AIY kit
Summary.

- Lots of lessons from VoIP are applicable to IoT
- In particular WebRTC data channel
- Let’s not standardize exclusively around cloud
- Come and see some of our PoC on the booth
- Looking for Investors and paid PoCs
Questions?

- Tim Panton
- tim@pi.pe
- https://pi.pe/
- https://GitHub.com/pipe