Getting Into Mobile

Without Getting Into Trouble

Greg Kliewer
Senior Solutions Strategist

October, 2014
The good old days

- Network separation
- No programmatic access from the Public Internet
- Safety through total isolation and control
Canadian smartphone ownership in 2013. Up from 33% in 2012.

- 2012 Mobile devices outnumber workstations by 2:1.
- 2013 Time spent on non-voice mobile surpasses time spent on workstations.
- 2014 Smartphone and tablet sales surpass workstations.

Today
Threats

- Inauthentic user
- Careless user
- Poorly coded apps
- Spoofed apps
- Lost or stolen devices
- Hijacked or tampered with devices
What are you going to do about it?

building blocks for secure mobile access
Reverse proxy in the DMZ using a hardened platform

- Terminate inbound connection
- Establish a new connection to fulfill the request

- The only open port to the public Internet should be 443
- All extraneous ports should be closed
- All services not strictly necessary should be stopped or removed, preferably
- Management ports are on separate network not accessible from the public Internet
- Be ready to patch at a moment’s notice (e.g. heartbleed, bash shellshock)
Validate user credentials before making the backend connection

- Extract credentials
- Validate them against the existing user directory
- THEN make the backend connection
PROBLEM

It’s the app that calls the API.

Do you want it holding on to the user’s username and password?
Introduce an OAuth 2.0 Authorization Server ...

- Authenticate User
- Authenticate App (request user to grant access to the App)
- Issue an Access Token that represents an ephemeral session with the client

• NOTE the requirement for a new set of credentials, identifying the app
...and an OAuth 2.0 Resource Server

- Validate Access Token
- Retrieve permissions
- Authorize access to requested resources
- Rinse and repeat

- The session eliminates the need for an app to cache or store the user credentials
- Instead, you keep a temporary-use token for a short period of time
PROBLEM

Sessions can be hijacked.
This one, at least, has a fairly simple solution.
Keep the Access Token private

- The Access Token is a secret and must be treated as such.
- Use HTTP keep alives to limit resource utilization impact.
- You may want SSL acceleration for this.
- Did we mention? Be ready to patch / step up key strength / cipher suite requirements at any time

• 443 ONLY
• Strong crypto only

SSL / TLS

- App Creds
- Token Sessions with permissions

© 2014 CA. ALL RIGHTS RESERVED.
PROBLEM

Aren’t mobile devices and apps just inherently unsecure?
• Relaxes device security
  (no screen lock)
  (no PIN / passphrase)
• Leaves device behind at
  the gym or on the bus

• Caches or persists user
  ID credentials
• Stores sensitive data
• Stores sensitive data
  without safeguards

• Tricks user into
  downloading fake app
• Steals phone, bypasses
  protections, unlocks it
• Relaxes device security (no screen lock) (no PIN / passphrase)
• Leaves device behind at the gym or on the bus

• Caches or persists user ID credentials
• Stores sensitive data
• Stores sensitive data without safeguards

• Tricks user into downloading fake app
• Steals phone, bypasses protections, unlocks it
SDKs can provide a lighter-weight solution

- Takes security out of the hands of app devs (they will thank you!)
- Tie in to device safeguards
  - Key stores (hdw backed)
  - Native Containers
  - Biometrics
  - Geolocation
  - Remote control (for wipe)
  - Manufacturer Attestation
- Tie enhanced crypto to user authentication and consent
  - Dynamic app secret provisioning
  - Dynamic certificate provisioning
Solutions for secure enterprise mobility
1. Securely gate all access from mobile platforms

**MOBILE API GATEWAY**

- Appliance
- Pre-hardened for exposure to public Internet
- Onboard OAuth 2.0 Authorization and Resource Servers
- Control sign on and sign off across apps, devices, and web properties
- Enforce multi-factor authentication (e.g. OTP, Biometrics, Attestation)

**Secure Platform**

**Comprehensive Access Control**
2. Extend your security reach onto the devices themselves

**PLATFORM SDKs**

- Take security out of the hands of application developers
- Bind to device security primitives to leverage native capabilities (e.g. keystores, containers, hardware-backed crypto, biometrics, etc.)
- Block access from inauthentic applications
- Detect device and OS tampering
Come see us at booth 603

“The most radical and transformative of inventions are often those that empower others to unleash their creativity – to pursue their dreams.” - Jeff Bezos