Android in the Healthcare Workplace: A Case Study

Thomas Richards

g13net@gmail.com

OWASP

04/05/12

About me

- My name is Tom
- Twitter: @g13net
- Website: www.g13net.com
- Independent Vulnerability Researcher(at night!)
  - 13 Published vulnerabilities with 5 CVE-IDs assigned
- IT Support Analyst (by day)

Why this talk?

- With the growth of mobile devices, companies are looking to capitalize on this for business purposes
- Very rapidly writing applications for these mobile platforms
- Security is a concern

What this talk is

- This talk is about a security assessment performed on a product my company uses
- This is not about the Android platform itself, or security issues with it

Background

- Home Health Company (visiting nurses)
- Transitioned from Laptop with thick client software to Mobile platform
  - Flexibility and Mobility are huge for a 75% mobile workforce
- New product, rewritten for Android from Windows Mobile

HIPAA Concerns

- HIPAA is the word in Healthcare
- Need to Protect PHI
- Encryption!
  - At Rest
  - In Transit

Deploying Devices

- Deployed 250 Android Tablets
  - Running Froyo (2.2)
- MDM Solutions
- No Imaging

About the software

- Runs on Android
  - Not available in the market
- Clinicians sync to get data
  - Patient data(records) are kept on the device
- Vendor stated data on the device was encrypted as well as data in transit

How did I perform this assessment?

- Android Emulator!
  - Able to observe traffic in real time
- Used OWASP Mobile Top 10 and Web Top 10 as guidelines

Authentication and Authorization

- Only two pieces of information were needed to configure a device: Server name and Agent ID
- Agent IDs are sequential
- No way to validate an approved device is being configured
- Finding Server name and Agent ID would lead to complete compromise

Password

- User’s password was configured and stored locally
- No complexity requirements

Data at rest

- I was able to determine that the data in the local database was encrypted (yay!)
- It was protected by the user's password and using built in SQLite APIs for encrypting a database

Data in Transit

- No SSL!
- Using HTTP, they used POST methods to retrieve data from the server
- Now treat this as a web app also

Interesting Side Note

- Going to sync1.vendor.com/falcon showed form based login prompt.
- Also not in HTTPs, tried to connect to it via
- Going to: sync1.vend.com/falcon/mobiledevicehandler.fal
  - Displayed custom encoding

Session Handling

- No Cookies present.
- The server would not know if the request was proper which could lead to Replay attacks.

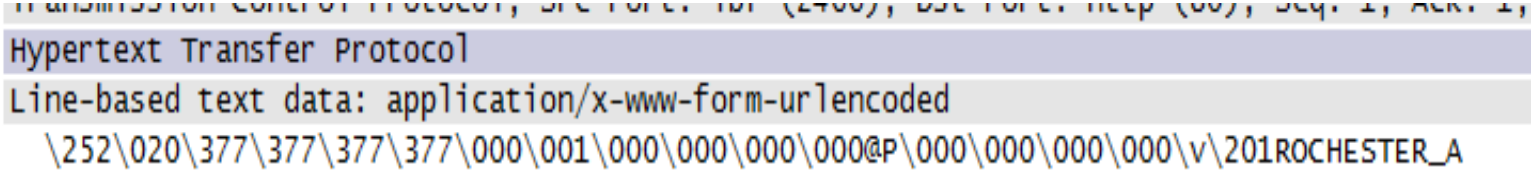
Insufficient Transport Layer Protection

- Obvious Issues

- Custom “encoding”
  - After some RE, not encryption! (no key present)
  - Some Plaintext available
  - I was able to analyze their protocol.

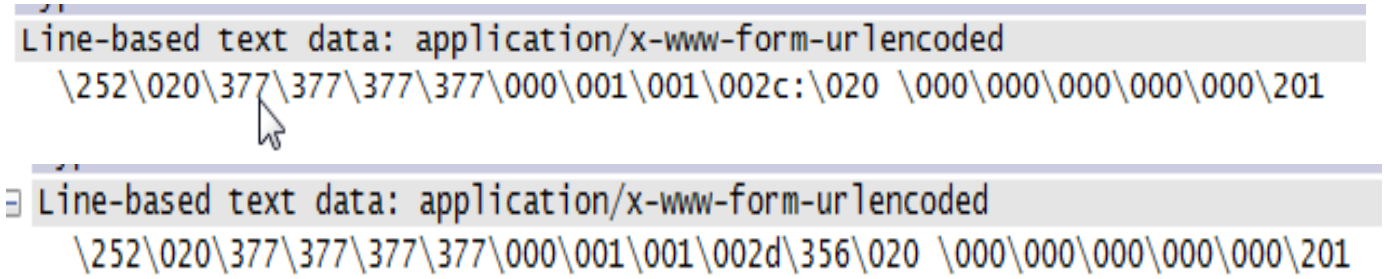
Server name Identification

- Plaintext!



Agent ID Identification

- After observing traffic with different Agent IDs, I was able to determine where in the string it lived

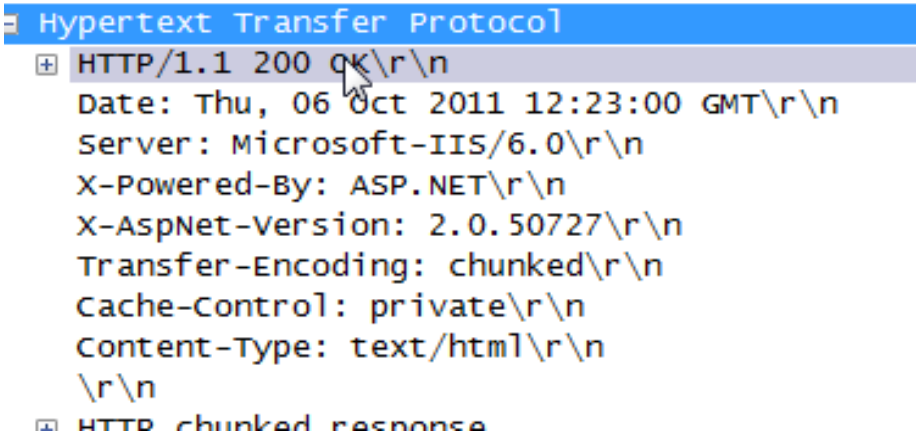


Agent ID Identification Cont.

- Raw Hex:
  - aa10ffffffff00010102633a1020000000000081
  - aa10ffffffff0001010264ee1020000000000081
- Converting the hex “633a” and “64ee” to decimal revealed the Agent IDs.
- This coupled with Server Name in plaintext could lead to complete compromise of data

Server Identification

- No attempts were made to disguise the identity of web server and technology used



Notifying the Vendor

- Brought this to the attention of my boss who asked me to write it up
- Submitted write-up to the vendor
- CTO Came and stated they were aware of these issues (they lied to us in the beginning)

Vendor's Plan

- Setup Codes
  - Unique 8 character string generated on server end before setting up a device
- SSL (eventually)
  - As of current version, still no SSL present. They stated it would have been in Dec 2011 release

Protecting Ourselves

- Ask vendor if the app has been independently assessed for security issues (companies specialize in this!)

- Assess the software yourself.

Thank you!