Testimony of David Ting, co-founder and Chief Technology Officer at Imprivata
API Task Force Virtual Hearing 1/26/16

1. Are there any well-known threats or vulnerabilities associated with APIs themselves that should be addressed (e.g. security engineering considerations/best practices)?

   a) As APIs are gaining adoption, are there steps organizations need to take to mitigate any additional threat vectors to data?
   APIs and Browser-based applications have different (but overlapping) sets of threat vectors. API's are easier to secure, in general, because they have less dependency on 3rd party components (such as the browser itself and web page development tools and frameworks).
   Cybersecurity standards and best practices are well-documented. Here are a sampling of both public sector (OWASP) and governmental (NIST and NSA) standards documents.
   https://www.owasp.org/index.php/REST_Security_Cheat_Sheet
   https://www.owasp.org/index.php/REST_Assessment_Cheat_Sheet
   https://www.nsa.gov/ia/_files/support/guidelines_implementation_rest.pdf
   http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf

   b) Are these just specific to APIs in general? What might be unique/specific to healthcare?
   Security for all computer interactions must cover:
   Confidentiality: When the data is exchanged it must be done confidentiality, so it cannot be read while in transit between the sender and the receiver.

   Integrity: Integrity of the data being exchanged must remain. There should be assurances that the received data has not been altered.

   Availability: Security must cover prevention against attackers, rendering the API inaccessible by authorized users. This is often called "denial of service".

   Privacy: Ensuring that the requesting party does not receive personal information beyond that which has been authorized by subject of the data. For an API, especially in healthcare, the identity and permissions of users are critical for privacy.

c. How does the issuer of the API ensure that the API won't become a tool used for malicious activity which could compromise the data source?
   An API can ensure that it won't be a tool used for malicious activity the same way as a web application issuer does. Compliance with best practices, code reviews, security review, automated code analysis, and extensive testing are all commonly used and effective methods.

2. How are APIs distributed in a way that the recipient/end-user of the API can trust the API is authentic?
   APIs are distributed using public key cryptography. The solution is the same one as used to allow us to trust our banking and e-commerce sites today.

3. Are there existing metrics or is there a need to develop metrics to measure the maturity of security and privacy controls in the use of APIs?
   I am not familiar with any existing, broadly adopted, metrics for measuring maturity of an API.

4. Is there a catalogue or store of tools that are built for the APIs for third parties to access?
   There are hundreds of tools and frameworks and forums available regarding building secure API's. The public sector security and development community is continually vetting these offerings. The fact that most of them provide source code means that it is feasible to automatically analyze the security of the code.

5. Are there known compliance implications with the use of APIs?
   Not in general. If an API operates within a regulated environment (e.g. healthcare, banking, etc), then its design and testing must assure that it complies with relevant requirements.

6. What are the perceived and actual security concerns or barriers to the adoption of APIs? How can these risks be mitigated/how are you addressing this?
   Risks can be mitigated through compliance with best practices such as code reviews, security reviews, automated code analysis, and extensive testing.