HITPC-HITSC Joint API Task Force
FINAL
Report of the January 26, 2016, Virtual Hearing

Names of ONC Staff Liaisons Present: Michelle Consolazio and Rose-Marie Nsahlai

Purpose of Hearing: Not stated

Review of Agenda and Opening Remarks

Task Force Co-chairpersons Josh Mandel and Meg Marshall thanked the invited panelists.

Panel 1: Consumer Technologies

Questions

1. Are there any well-known threats or vulnerabilities associated with APIs themselves that should be addressed (e.g., security engineering considerations/best practices)?
2. As APIs are gaining adoption, are there steps organizations need to take to mitigate any additional threat vectors to data?
3. Are these just specific to APIs in general? What might be unique or specific to health care?
4. 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?
5. How are APIs distributed in a way that the recipient/end user of the API can trust the API is authentic?
6. 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?
7. Is there a catalogue or store of tools that are built for the APIs for third parties to access?
8. Are there known compliance implications with the use of APIs?
9. What are the perceived and actual security concerns or barriers to the adoption of APIs?
10. How can these risks be mitigated/how are you addressing this?

David Wollman and Marty Burns, NIST, showed presentation slides, including depictions of the ecosystem, and described the Green Button Initiative, which enables electronic consumer access to energy data and supports development of an ecosystem. The purpose is to enable consumers' access to their energy use data. Green Button is available to customers in the United States and Canada as a result of collaboration among the White House, NIST, the U.S. Department of Energy, utilities, vendors, state regulators, UCA International Users Group, the Smart Grid Interoperability Panel, and the North American Energy Standards Board. Certification is not fully in place. Energy data are very complex. Exchange occurs between the utility company, the third party, and the retail customer. Authentication identifies the client to the server and allows communications over a secure channel. Authorization identifies access rights to an authenticated party. OAuth allows management of the conveyance of rights to data for a specific individual or account to a third party that is already authenticated to a data custodian. No personal information is included. OAuth 2.0 uses bearer tokens to indicate authorization, uses scope negotiation to allow tailoring of relationship to a subset of data that may be available for a customer, and extends outside itself to enable long-lived authorizations with short-lived tokens. It also adds bulk access via client access tokens of collections of data from authorized individual customers. The

utility company bears primary responsibility for privacy and security. Many resources, such as an API sandbox for developers, are available at http://www.greenbuttondata.org.

Stephan Somogyi, Google Inc., who had no slides or written testimony, talked about the protection of data and API engineering. There is protection of the data themselves and of access to them. Protection of the data is most efficiently done by encryption, both direct and in transit. Google has best practices for implementing encryption, conducting key management for the encryption, and making sure that decisions are pragmatic for the entire ecosystem. Best practices require allocation of considerable resources to maintain systems and to perform continuing engineering to ensure that best practices are current. Data in transition must be protected by best-practice encryption standards. They should be protected by modern browsers and authorize best practices for certificates. Technical control is necessary but not sufficient to build a secure system. The data-hosting party must have sufficient internal processes, policy control, and organizational security programs to ensure a common, high level of training and knowledge about the issues that create risk. Security must be considered holistically and systematically. APIs are not uniquely insecure or vulnerable. Any system that is open to the Internet has vulnerabilities. Devices should check encrypted signatures or software. Data should be accessible only to those with a need. A healthy engineering culture provides the preconditions for the design patents that make secure implementation of API a matter of course rather than an anomaly. Data from the outside should be considered untested.

David Ting, Imprivata, submitted written responses to the questions. He said that APIs and browser-based applications have different but overlapping sets of threat vectors. In general, APIs are easier to secure, because they have less dependency on third-party components such as the browser itself and Web page development tools and frameworks. Cybersecurity standards and best practices are well-documented, such as OWASP and NIST and NSA standards documents. Ting said that security for all computer interactions must cover the following:

- Confidentiality: When the data are 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 the subject of the data. For an API, especially in health care, the identity and permissions of users are critical for privacy.
- Authentication and Authorization: Ensuring that the requesting party of the API has been authenticated by an identity provider service that is trusted (typically cryptographically) by the API issuer, and that the requesting party has been granted the right to use the API. Plus the reverse; that the requesting party can verify that the API service it is calling is authentic (not being impersonated).

Ting said that APIs are distributed by using public key cryptography. The solution is the same one used to allow us to trust banking and e-commerce sites. There are hundreds of tools, frameworks, and forums available for building secure APIs. 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. Security risks can be mitigated through compliance with best practices, such as code reviews, security reviews, automated code analysis, and extensive testing.

Greg Brail, Apigee, submitted written testimony. API use has grown rapidly. According to Brail, at the simplest level, an API is a contract. The contract specifies how a software developer accesses an API and

tells the developer what to expect. A well-designed API makes this contract clear through documentation and specifications that describe not only what kinds of requests the API expects but what kinds of security controls have been put in place and what set of security credentials a developer must acquire before she or he builds an application that uses the API. Since an API is a contract, it is possible for the organization that offers the API to completely document and understand the interaction between the API and the application that uses it. Tools and techniques are available from commercial software vendors and the open-source community. The tools can be used to ensure that API access is not allowed unless the client follows the contract. These tools may also be used to monitor API usage and gather data to understand exactly who is using the API and how. This contract-driven interaction model makes it possible for the organization that provides an API to add policies and security controls on every interaction. An API team can regulate which applications and end users are authorized to use an API and which parts of the API they are allowed to use. The team can also control what an authorized user can do, including limits on the number of API calls or when they can be made. Finally, the team can follow the trail of API calls to understand exactly what authorized API users did and what unauthorized attempts may have been made. As a result, APIs, rather than being a new security risk, provide a well-documented, popular way for organizations to share access to data and services with third parties while maintaining strict security controls. Especially compared to other ways of sharing data, such as via website, file transfer, email, or printing, a well-implemented API offers a stronger set of security controls. There are a variety of security best practices that API providers should follow, and a great deal of information on these topics is available from various books and blogs.

Eve Maler, ForgeRock, submitted written testimony, saying that APIs present some unique circumstances regarding data tagging to track provenance. When creating fairly static, nonvolatile data, such as asking an individual to fill out a form or recording details of a visit to a health care provider, tagging the data creates no problem. But what if an API endpoint can report out a live feed of data coming from a device that has a sensor for blood oxygen levels? The most upstream point of provenance is the API or the device. Maler said that a solution would be to identify the points where the API or device is on-boarded to its service ecosystem, formalize that on-boarding ceremony, and apply security tags to elements of the metadata used in that ceremony. Standardizing an API within an industry is valuable when interoperability—removal of business and technical friction—is needed for some large subset of interactions among players. FHIR is one example of an industry movement kick-started through a standard API. The Open Bank API effort in the United Kingdom is another example. Maler noted the following as select reasons for using standardized mechanisms for security, identity, and consent to the extent possible:

- Complexity and variation are enemies of security. Standardization simplifies.
- It is hard to separate data from different parts of a person's life. A standard way of identifying the person across those worlds could help bring the data together for their benefit.
- A standard mechanism has likely been well-vetted by others.
- Standards can generally be implemented by multiple parties, and those implementations usually strive for interoperability with each other, so it may be more possible to buy rather than build at a favorable price.

According to Maler, APIs are a good idea for designing standard mechanisms for security, identity, and privacy as well. This is where the innovative emerging technologies OAuth 2.0, OpenID Connect, and User-Managed Access (UMA) come into play; in part, their specifications include definitions of APIs, and they are extremely well-suited for use with APIs. OAuth is an API enabling a client app to call an API on behalf of a resource owner (typically an individual) and with consent, without ever having seen the credentials (such as a username and password). An access token stands for the consent and the list of actions (scopes) that the client app can perform, which may not be the entire possible list. The resource

owner can always go back to the API publisher and withdraw the consent, revoking the token. OpenID Connect is effectively a simple OAuth-protected API that does single sign-on and identity data retrieval jobs. Its main innovation is to use lightweight technology to remove friction from tasks that the older SAML standard proved too heavy to tackle in practice. UMA is innovative because it puts the individual resource owner and the authorization service that executes the owner's policies for access at the center of the equation. It enables use cases from proactive delegation (share by user choice) to reactive consent (access approval when asked) to any time monitoring and adjustment of access (denial and withdrawal), all with a choice to adjust scopes of access at any time. Its architecture enables the resource owner to manage these choices in a central location (where central is relative to some identity ecosystem that the services used by the individual are willing and able to participate in). Maler concluded by saying that the Health Relationship Trust (HEART) standards effort is key, because it specifically focuses on patient-centric, privacy-sensitive health data-sharing use cases and seeks to tighten both the security of the above three standards and their interoperability when applied to the FHIR API.

Q&A

Mandel asked about experience with APIs to allow customers to bring their tools to health care providers: Are there any special considerations for security? Somogyi responded that, as with a Web browser, what matters is protection by technology and policy. Technically, there is nothing special with API protection. What is different is the greater sensitivity of the data handled by the API. Maler referred to business, legal, and technical boundaries and opportunities. Technical security issues focus on the nature of API access and the business risks of those transactions. According to Ting, health care organizations' concerns center on the validity of data and the fact that the patient data are those submitted by the actual patient. Integrity may need to be verified or validated via third party. Ownership and the right to distribution are other issues. Brail responded that APIs provide an opportunity to verify the identities of the organization and the end user. Organizations can set policies regarding which developers can access the API according to terms of service and use an authentic application. Maler added that OAuth log-in offers protection with a powerful set of technologies. Both the client application and the user have identities to which security protections can be applied. Wollman responded to a question about Green Button by saying that Green Button allows the separation of personal information from the energy use data streams.

Leslie Kelly Hall noted the great opportunities for consumers and wondered about security and privacy modules to defend points of entry. Ting's colleague responded that API layers can be put in front of existing EMRs. There is a need to find ways to make information available to apps. UMA and other consent systems are critically important. FHIR can be put to creative use. Ting talked about prescribing controlled substances via an API. Authentication is done through the API, which leaves an audit trail. Maler interjected that FHIR enables hackathons. Restful APIs allow user-managed access. Web style programming can be applied to devices. Data sources are increasing. OAuth enables sharing with another party with whom one does not share credentials. Kelly Hall concluded that APIs give opportunity for more security and privacy. Maler agreed that API platforms provide robust solutions. Lessons from Google can be applied. Wollman referred to granular access permissions and controlling privacy. Maler talked about policy on access permissions.

Mandel referred to the engineering culture, saying that it requires buy-in by organizations. However, not all health care organizations buy in; they are not always motivated to expose data and host APIs. Maler said that it is a business model challenge. App developers are motivated by what their customers want to do with their data. Health care organizations are subject to considerable regulation. Customers can pressure health care providers to be more responsive. Someone said that one issue for clarification is whether the provider is the owner or the custodian of the patient's data. Wollman said that, rather than

data ownership, access can be based on the need for access to data for operations. Regarding culture, efficiencies of access have moved the culture. Maler wanted to talk to Wollman offline about solutions in development. Alisoun Moore talked about HIPAA and HITECH; the latter opened access. Providers are concerned about consumers' APIs being a possible violation of regulations. According to Maler, ownership means control of access.

From his perspective as a hospital executive, Aaron Miri inquired about data provenance and difficulty in maintaining the quality of data. Somogyi responded that data quality and APIs are not related; there is nothing inherent in APIs that affects quality. Ting replied to another question about standards development on data citation unique to health care by saying that EMR transactions are well-documented according to metadata policy regarding source, subject, and permission. The metadata allow for audit information. Miri summarized that an API is a tunnel and a trusted construct through which data travel. The hospital then has the information to provide to an auditor.

Marshall wondered about Google's best practices and support of advancements. Noting that a good answer would require days, Somogyi said that, regarding security and encryption, Google makes a great effort to access risk, stay abreast of new technology, and be robust in deprecating old standards. Best practices do not remain best forever. This can be scary for many organizations because of their legacy systems and resource limitations. The protection of users and data is paramount. Google aggressively disallows outdated practices as a cost of doing business. Culture is more than buy-in by organizations; one must do right by users. With regard to customers' understanding, Somogyi said that when old practices are disallowed correctly, the user never notices. Browser developers agree among themselves when a new practice will happen.

Mandel had another question about user permission. Wollman and Burns explained scope negotiation in Green Button. When the customer logs in, an interview regarding purpose of use commences. Options include agreement to no personal information being released. The scope must be acceptable to all three parties. If accepted, the authorization sequence starts or the consumer is redirected back to the data custodian to approve what the third party needs. Authorization and access tokens are used for granular details. Mandel wondered whether customers have used this to express permissions in the real world. Wollman indicated that it is too soon to tell. He repeated that customers are not presented with choices; they respond to interview questions. Marshall referred to several use cases, including the HITECH right to restrict data to payers if the patient self-pays and 42 CFR Part 2. Maler talked about Google Docs, Google Apps, and the UMA share button. "Sharing" can mean viewing, editing, or both. If the user wishes to scope down or correct something that went wrong, UMA allows scope design. Maler observed that it is not unusual to want to withdraw permissions.

Aaron Seib wondered whether, as scope and other methods improve, technology will eliminate business and legal components. Maler said that that is not likely. Building relationships with business partners is important, and businesses prefer static partnerships. Business trust is more difficult to establish than technical trust. The Common Accord is a way to make legal agreements. Miri agreed regarding risk management in the hospital, where everything in privacy and security is highly documented and technology will never be enough.

Kelly Hall commented on the importance of educating patients about the risks and rights to use data as they wish.

Panel 2: Consumer Technologies

Questions for Panel 2: Consumer Technologies

1. Does your organization use APIs for apps which are available internally or to third parties? If so…
2. Do you publish your documentation online or make it available to third-party developers?
3. How do you determine who can get access to your API?

4. Do they need to be "certified" for privacy or security standards by your organization to use?
5. Are there terms of use that include specific language for privacy and security?
6. Are there production deployments of these APIs/third-party applications using APIs?
7. What are the perceived and actual privacy and security concerns or barriers to the adoption of APIs?
8. How can these risks be mitigated/how are you addressing this?
9. How to improve consumer experience with the third-party apps using the APIs?
10. Are there third-party certifying authorities in non-health care industry that we can leverage?

Alisoun Moore, LexisNexis, submitted written responses. She said that LexisNexis provides risk mitigation services and data to many industries, including health care. Her company assimilates information from more than 10,000 public record sources to determine correct identities for individuals, businesses, and health care providers. This information is used by thousands of businesses to ensure that transactions can occur securely and to protect consumers. XML and secure batch processing of customer data are routinely offered against data to verify information. APIs are allowed with strict controls and licensing for clients who need to upload or download the company's data. Some documentation is published online. All clients that wish to procure access to LexisNexis data must follow a process for technical integration and conform to data usage agreements that protect their data and guide their use of LexisNexis data. All clients must abide by pertinent federal and state laws. Access to the API is based on the clients' specific needs and what they want access to. LexisNexis works with clients to ensure that they understand permissible use. When agreements and licenses are signed, secure access via XML is set up. There are specific requirements for privacy and security. The data centers are FISMA High compliant. A proprietary technology called LEXID, an internally created unique identifier for each correctly resolved identity, masks the real identity of people and is often used by clients who do not wish to use a Social Security number or another unique identifier.

Moore said that, as with any technology, if the APIs are not developed or governed with strict security controls and data usage policies, then security and privacy will be compromised. This can lead to data breaches, and developers, companies, the government, and consumers should all be wary of poorly designed APIs and a lack of governing documentation. LexisNexis has a strict process on the development of its internal APIs, data usage, and licensing agreements for access to its products and services. If clients want specific integration of the system to theirs, agreements are documented. Applications must also be easy to use, secure, seamless, and useful. Use of focus groups to test the apps and development of intuitive user interfaces are key to ensure these characteristics.

Evan Cooke, US Digital Service, submitted written testimony. He emphasized the incredible power of APIs. The Department of Education recently launched an updated college scorecard tool built on top of an open API with data from 7,000 colleges and universities going back 18 years. This API makes it easier for software developers and researchers to extract, customize and build upon the data to support students and families to make better college choices. The result has been a diverse ecosystem of partners that supports better college search and choice tools, better advising and support for students, and more comprehensive rankings with new outcomes data. APIs are collections of technologies and standards rather than monoliths. A common way to describe APIs is as software contracts between parties. Those parties could be private companies, individuals, or government entities. APIs can capture almost any form of business process or exchange of information if the data can be represented in digital form that can be exchanged over a network.

Cooke said that, rather than a single entity, APIs are composed of many parts, such as network protocols, security mechanisms, authentication and authorizations means, request and response methods, and serialization formats. Those parts may change at different rates based on their maturity

and broader changes in the products that the APIs support. Since the requirements for each part of any API can be different, the specificity of guidance may also need to be adjusted depending on what component of the API is referenced. For example, staff might decide to dictate a specific technical format for a mature serialization format but provide higher-level guiding principles for the request and response approach. As an illustration of possible levels of abstraction, Cooke said to consider the NIST Cybersecurity Framework, which describes four different levels of specificity, including function (identity, protect, detect, respond, and recover), categories (e.g., governance, data security), subcategories (e.g., "Organizational information security policy is established"), and informative references (e.g., NIST Special Publication 800-53).

David Berlind, ProgrammableWeb, provided written testimony. He stated that his responses were from the point of view of an independent observer of the API industry and focused on real-world API security exploits. ProgrammableWeb does not currently offer an API; it offers a directory of APIs. It also publishes articles for API technicians. Many API providers offer programmable documentation. When an API provider is looking to attract as many providers as possible, it usually does not concern itself with who can or cannot access the APIs. Developers are sometimes required to have certain certifications to use an API. For example, PayPal says that API users must comply with the payment card industry data security standards, and payment application data security standards and other documentation evident in its compliance must be provided upon request. While thousands of organizations are rushing to join the API gold rush, very few are interested in securing them. Since 2014 many of the biggest Internet companies have either fallen prey to or discovered a major API vulnerability. This includes Google, Apple, Facebook, and Pinterest, all well-resourced companies. Mobile applications involve a great many API cases. The majority are shared between a mobile application, and they are easily discoverable, even with data security technologies. The most advanced solutions for running APIs are sometimes out of step with the latest standards. Berlind suggested the use of a distributed, constantly evolving checklist to inform key stakeholders how to maintain the best possible security, taking into account the latest exploits.

Marc Chanliau, Oracle, submitted written testimony. He referred to two types of APIs: internal and external. Internal APIs are exposed by product vendors to allow customers to customize or extend the vendor's product and integrate the product with third-party applications. This type of APIs is used by the vendor's customers and by third-party vendors wanting to integrate their products with the API provider (e.g., Oracle). There are two subcategories of external APIs: APIs exposed by companies to allow other parties to leverage their services (e.g., FedEx, Walgreen) and APIs exposed by companies to allow other companies to integrate functionality without having to develop it themselves (e.g., Twilio, SendGrid). Typically, APIs are made public in open source or vendor documentation. For example, Twilio exposes its API publicly (https://www.twilio.com/docs/api/rest/making-calls). By making APIs publicly available, enterprises can improve partner connectivity (mash-ups) and cloud integration. Oracle provides only what are referred to as internal APIs. Its products expose APIs that allow customers to integrate, customize, and extend its products. Oracle API documentation is publicly accessible. In addition, Oracle offers products designed to manage and secure external APIs. These products are sold to customers seeking to improve API security and management in their companies. Oracle APIs are available to third-party developers wishing to customize, extend, or integrate Oracle products. These APIs are subjected to the same intellectual property laws as Oracle products, since they are designed and owned by Oracle.

Shue-Jane Thompson, IBM, did not submit written testimony. She referred to general or customized APIs. Identity and trust are primary considerations. APIs are commonly used for internal applications. Data at rest and data in the air must be secure. Although people are afraid of exposing their data, the technology increasingly enables ease of sharing. Thompson referred to common principles to secure APIs: generalization or customization, partnerships, leveraging of social data, IT environment, and big

data analytics. IBM has products that are designed specifically to tackle API requirements and challenges. In addition to the endpoint, IBM built an enterprise management solution for the creation of APIs by leveraging existing API blocks. Individual companies can quickly create API-based applications. The API marketplace is a tool that allows developers to integrate APIs into the application for health care privacy. To maintain patient confidentiality, the API can anonymize the data as they move from the patient through the API. The use of session tokens and API keys can prevent unauthorized access to the information. The identity of the user can be stored in the IBM directory. Thompson reminded the members that the educated customer is the best customer. APIs can be dangerous without proper precautions.

Gray Brooks, General Services Administration (GSA), had no written testimony. He explained that every GSA project, internal or external, that involves actual development begins with an API. It is the core premise of the team that the API is built first, made available, and then is built on top of. Documentation is published online for third party developers. Undocumented APIs are of little use to anyone. Documentation should exist as actual Web pages, public by default. In determining who gets access, two layers are considered: access to the documentation and access to the material. A team member can access the documentation without having to make a request. Those who have to write to the data source have access. API.data.gov is available to government agencies. Privacy and security standards are not distinct from the standards that guide development. APIs do not represent a unique or different perspective from existing development. Brooks noted that the government's legal rights are strong. Too much legalese presented to developers is to be avoided. Regarding privacy and security concerns, APIs do not present new ones. API keys are provided for the services. The default API rate limit is 5,000 hits a day and can be adjusted on demand. The limit is not based on the fragility of the system. By enhancing and maximizing the developer experience, GSA hopes to drive as much internally and externally as possible. Staff avoids getting too involved with the third-party developments, assuming that a multitude of mash-ups in third-party applications provides the best user experience.

Q&A

Kelly Hall inquired about ways other than APIs to allow patient control and access. A panelist mentioned static data, bulk export, and portals, all of which are suboptimal. Berlind observed that providers' portals are not interoperable. Not all of the patient information is in one place. Drawing from multiple sources is required. Although no technology is infallible, the benefits of APIs are greater than the risks. Chanliau explained that the back-end application handles the confidentiality via authentication and authorization in two layers. Thompson talked about Care Everywhere, by which Epic Systems established a closed network and ecosystem for data protection. Berlind said that various security mechanisms apply. Most vulnerabilities arise from human error. Miri said that many sources of data are closed-loop systems. According to Moore, the API is the entry point. In addition to access, authentication that the data pertain to the appropriate patient is necessary. Someone said that that use case governs the OAuth system token. He described an attack that gained access to OAuth and to Twitter and Facebook tokens. Thompson said that, more than an ecosystem, a dynamic environment with continuing learning is essential.

Mandel said that stage 3 will require that providers expose data to some APIs, but not in a standardized way. Different parts of the EHR may use different APIs. He asked about experience with standardization of APIs. A panelist said that security is the way to standardize today by using standards to protect access to APIs. Berlind referred to a standard API for data portability used in the United Kingdom: the Open Bank Project. His organization uses a meta-API to give developers access to multiple APIs. Cooke talked about the framework of APIs as software contracts and replacing the technical API with a contract. The contract would define which part is to be static over a defined period and which part is expected to evolve.

Marshall referred to consumer protection, saying that, due to limited technical savvy, a good housekeeping seal of approval might be a good idea. She requested more details. A panelist referred to watching for events, reengineering the exploit, and developing a list. He referred to a major vulnerability revealed this month with Verizon's pump service. The credentials to access the APIs were visible in the source code of the website. Some exploits involve compromise of source code repositories that are deemed private. Credentials for the APIs are stored in plain text in those source code repositories. It is not just an API itself that, in some cases, might be vulnerable; it is the adjacent things around it. Hackers will orchestrate a very sophisticated attack that involves multiple barriers that have to be breached in order to get to the final objective. A checklist would not only cover known things but allow the addition of new items on an ongoing basis. The program would include a clearinghouse of all exports. An app user would want to know about certification of privacy by a third party. Developers, as well as customers, could use a seal of approval for APIs. Somogyi was concerned about a new level of regulation. Third-party guidelines may be a better approach. Moore said that the task force seems to want a standardized API. Providers who own the data would have to be very secure. There is a lot of security infrastructure and responsibility for breaches. A two-tiered system may be necessary. Many vendors may lack resources for security. Marshall indicated that the task force could work with OCR to obtain clarification of requirements. She invited the panelists to listen to the task force meetings when recommendations were being debated. She asked the staff to inform the panelists of upcoming meetings. Thompson said that ecosystem stress points are evolving; machine learning is important.

Mandel related his experience of identifying vulnerabilities in 2014 but not being able to track down the developers to make corrections: What can be done to get vendors to share learnings? Moore described payers' groups formed to share vulnerabilities uncovered by members. Property and casualty markets share data on attacks via special investigative agents. According to Berlind, getting information to the right person as quickly as possible depends on the severity of the problem. He reported that he had talked to several developers who said that they would never conduct research on health care providers' vulnerabilities due to fear of criminal prosecution. He acknowledged that he had no evidence of this happening. Kelly Hall referred him to the new OCR guidance. Someone referred to the NIH Precision Medicine Initiative and its policy on patients' contribution of their data to research.

Kelly Hall expressed approval of eTrust and HITRUST, which are based on a set of minimum standards. Where is the boundary for government regulation when patients are involved? Moore said that there is a role for regulation. However, drawing a line is difficult. Kelly Hall said that the risk of the patient's lack of information is greater than the provider's risk.

Seib asked about consumer safety education materials for sharing sensitive data. Someone talked about email phishing as an area in which education may be needed. Kelly Hall said that her company designs patient materials. Thompson said that continual learning is required. She offered to share materials and lessons learned. All partners must be involved. Moore said that consumer education coupled with a seal of approval would be useful. Another panelist cautioned on giving consumers too much information on APIs. Someone repeated that developers also want a seal of approval. Kelly Hall said that ONC should look at harnessing the work of white hat hackers.

Marshall talked about use cases in which there is authorized access but unauthorized use of the data, such as when an app uses or sells consumers' data in a way of which the patient is not aware or when an API developer uses data for unfair business practices. Are there other use cases of which the task force should be aware? A panelist said that written terms of use can apply to the application itself. Terms of use do not affect hackers. He related an experience with reverse engineering in which an application had no terms of use. Another organization has clear-language terms of use on reverse engineering, circumvention, and other things. Malicious users will not be affected by terms of use, but they may provide legal recourse. He referred to the screen shot in his written testimony for more information.

Moore said that her company has very clear terms of use. Consumers do not necessarily understand how their information may be used. They need to understand the risks when they use APIs.

Closing Remarks

Linda Sanches, OCR, was asked to comment on data ownership. She declined. However, she will work with the task force on the use cases described. OCR recently posted an updated guidance on the HIPAA Privacy Rule and access. More updates are forthcoming. Nsahlai will send the updated access guidance to members. She indicated that ONC staff are also working on the ownership issue.

Mandel summarized points made by the first panel. Green Button is used by utility customers to access information on their energy use. An engineering culture in which security is primary is important. APIs are gateways. Users must be allowed to express and change granular permissions. The Q&A generated descriptions of real-world experiences. Marshall summarized that members of the second panel described apps that use APIs for making data available to third parties. Consumer protections, such as a seal of approval and checklists, may be needed. Standardization and white hat hackers were described as possible prevention efforts.

Next Step: The public hearing will reconvene January 28.

Public Comment: None.

Consolazio invited written comments via email or chat.

Flag to ONC Staff for Coordination: None.

Attendance

Name	01/26/16	01/12/16	12/04/15	11/30/15
Aaron Miri	X	X	X	X
Aaron Seib	X		X	X
David Yakimischak	X	X	X	X
Drew Schiller	X	X	X	X
Ivor Horn	X	X	X	X
Josh C. Mandel	X	X	X	X
Leslie Kelly Hall	X	X	X	X
Linda Sanches	X	X		X
Meg Marshall	X	X	X	X
Rajiv B. Kumar	X	X		
Richard Loomis	X	X	X	X
Robert Jarrin		X	X	X
Rose-Marie Nsahlai	X	X	X	X