;login:
DECEMBER 2013 VOL. 38, NO. 6
ELECTRONIC SUPPLEMENT

Conference Reports

HotSec '13: 2013 USENIX Summit on Hot Topics in Security

Washington, D.C.
August 13, 2013

HotSec Reboot

Summarized by Michael Rushanan (micharu1@cs.jhu.edu)

Michael Bailey, Deputy Program Chair, and Casey Henderson, Co-Executive Director of USENIX, introduced the reboot of HotSec this year. What started as the brainchild of Matt Blaze has evolved from a workshop that solicited paper submissions to a summit that fosters discussion and participation. There was a total of seven discussions, each covering a hot topic in security that is relevant to both academia and industry.

The Death of Passwords

Joseph Bonneau, Google

Joseph started with a few startling statements with respect to passwords, though one in particular captivated me: "Passwords aren't going to die soon. At least, not completely." This statement felt counterintuitive; had not recent work in this domain converged on the same ominous conclusion—the complete elimination of passwords? Joseph told us that while the aforementioned conclusion is true, the consensual approach is to shift toward the reduction of the number of passwords and the number of Web sites collecting passwords.

Joseph continued by enumerating the reduction of bad ideas. Specifically, academic literature has rejected expiration policies, leveraging personal knowledge, and implementing complex password measurements (e.g., computing Shannon's entropy on input password). As it is, there is no silver bullet to fix one of the irrevocable pieces of the password model, the human memory. Joseph does not claim that helping people choose passwords is ineffective, but rather that password strengthening processes, such as two-factor authentication, cause the dilution of the password-based factor. People tend to scale down the verbosity of the password if they believe the second factor will protect them. There are still proposals in the space, such as cognitive one-way schemes where a functional challenge/response is triggered by cognitive recognition of outputs, though the adoption of such protocols has been limited.

Economics are tricky. Joseph implicated the effect of Web sites that do a bad job on Web security. There is a negative externality caused by Web sites that do not implement SSL or allow infinite password guesses without rate limiting—Web sites that do have good password policies are subject to any and all negative exposure. Exacerbating the global problem is market competition. If a Web site doesn't have any direct competition, it is found to have a fairly loose password policy. Because of this, Joseph likened passwords to the Pareto equilibrium; if we fix one thing we make something else worst.

All is not lost, Joseph consoled us. There are non-trivial deployments in the password space that have seen success. For instance, there is two-factor authentication on mobile phones provided by industry leaders such as Google and Facebook. There is OAuth, an open authorization protocol that combines simplicity and standardization for applications. And there are more complex implementations, such as Facebook federated authentication, a decentralized approach to provide user authentication across distinct systems.

Open research questions. Joseph enumerated the following research questions that excite him: What is the best password policy? Does password strength matter? Which passwords are least convenient? What are the results of password policy? Following on the question of best password policy, Joseph mentioned that NIST ranking/policy efforts have been shown to not correlate very well to the strength of the password being cracked (Weir et al., "Testing Metrics for Password Creation Policies"). His other thoughts streamed into a collective blacklist of bad passwords shared by all services, an increased minimum length policy that doesn't constrain anything else, and special cases where users can provide alternative vectors of input (e.g., touchscreen). Of course, there are varying attack surfaces with these approaches, such as a smudge attack, where an attacker might use finger residue to limit the possible combinations of a password.

Joseph also probed the possibility of random passwords by depicting an xkcd comic, "Correct Horse Battery Staple," that illuminated random-word passwords as difficult to guess and easy to remember. This lead to a short discussion about implicit memory and how a study by Bojinov et al. attempted to train people to memorize passwords without explicitly telling them the password. The result, though long training time was required, was a correct replication of the password under the correct stimulus. This approach would be useful when the "rubber hose" technique is employed to gain information.

Before moving on to the contentious areas of research in this area, Joseph introduced backup authentication. Most people synonymously agree that an email account is used to provide backup authentication. However, if you're running the Web mail provider, this becomes useless. Useful schemes in social networking attempt to relax the email backup by employing trustees (e.g., friends) and tokens to provide an out-of-band authentication. Joseph admits, however, that there has been prior work

that provides context as to why this scheme is weak against an attacker who knows you.

Known points of contention. The first contention is with the user studies—are they ecologically valid? Joseph argued that Mechanical Turk studies are gaining popularity, though there may be a sampling bias due to the compensation of participants. Second, does password strength matter at all? If I were to select a strong password but have no real risk, is it necessary to have a strong password? Third, should we hash passwords? Traditionally, the answer would be yes, however the benefit is increasingly lowered as password cracking libraries become more efficient. This stirred up some commotion in the audience as someone in the audience asked, "Do you think we should get rid of hashing then?" Gene Rackow responded, "No, use scrypt. If you could guarantee that no one could get a cleartext password file, then no problem. No one can give this guarantee."

Final points of contention included: writing passwords down, shoulder surfing, and graphical passwords. Gene Rackow told the audience, "Writing down passwords is good . . . but write it as a puzzle—obfuscation helps." Eric Wustrow and Steven Bellovin agreed that shoulder surfing is still problematic and perhaps more so with the introduction of Google Glasses.

Contentious points aside, Joseph continued on with topics that are seemingly more targeted by industry and less by academia. For example, can account types be evaluated in a way that weakens/strengthens password policy? Can we take this example a step forward and consider the quantifiable risk introduced between user and site? Additionally, Joseph reported that password cracking/hashing has drifted away from academics; for example, a room poll of who has heard of hashcat resulted in a mere five to six people. It is John the Ripper's (password cracker) competition.

Joseph concluded his presentation by covering user strategies. He asked the audience what user strategies actually work (e.g., password manager). Steven Bellovin responded, "I would not put any of my passwords in a browser-based password manager as that increases the accessibility for an attack." Someone else concluded that he "would exclude password managers from browsers." The discussion then shifted toward the threat model of large authentication providers. Joseph believed that there is no good sense of how malware and malicious attacks affect LAPs (Large Authentication Providers), and he reported that he believes the lack of collaboration between industry and academia has caused this knowledge gap to occur.

Joseph publicly posted his slides to Google Drive. The link to those slides can be found here: http://goo.gl/c6An04.

Eroding Trust and the CA Debacle

Jeremy Clark, Concordia University

Jeremy reported that there is eroding trust in the current certificate authority (CA) model due to increased breaches, decreased baseline validation, and revocation and TLS protocol issues. To elaborate on this, he first provided due diligence to cryptographic attacks. Aging primitives such as MD2, MD5, and RC4 coupled with weak keys and bad randomness have left protocol-level implementations sloppy and broken. Lucky Thirteen, a cryptographic timing attack against TLS, and BEAST, CBC mode exploitation in TLS, play on these weaknesses. Additionally, version downgrade attacks have exacerbated the problem by allowing the utilization of aging primitives for backwards compatibility that greatly diminish security. So, Jeremy asked the audience, why haven't servers upgraded to TLS 1.2?

The general consensus was a combination of legacy support and laziness. Jeremy wondered whether patch-working TLS 1.0 was an acceptable approach then. Or, perhaps they should push in new and fresh specifications? These thoughts didn't generate much debate, and Jeremy went on to discuss the CA model.

The CA model prerequisite is simple: provide a valid public key to set up the secure client. This is, however, not as simple as it would seem; for instance, how do you figure out whether you are talking to a real domain? As for certificate distribution, Jeremy provided the usual daunting response: there are 150 root certificates from approximately 50 trusted organizations; each root certificate can then authorize thousands of intermediate CAs, all of which can issue a certificate for any site. Jeremy followed this fact with a question: is it reasonable to trust one million sites "automagically"? Nothing in the real world is analogous to this.

This last comment stirred up some debate as someone from the audience stated that we are not trusting people but rather CAs. Someone else retorted that CAs were invented when people first started doing transactions online and thus there should be explicit trust. This same person also asked what the real problem was? Specifically, he wanted to know whether users or businesses should be responsible for managing trust—with the assumption that users do not have the resources to know better. Finally, Gene Rackow said something that resonated well: "[this discussion] highlights that [eroding trust with respect to the CA model] is a problem because it is not considered a risk."

Jeremy moved on to discuss CA validation. The term was used generally to cover a few topics, the first of which is the authentication process of a certificate recipient. The process of issuing a certificate was envisioned as a human transaction but quickly moved toward automation. Therefore, a CA needs to validate that the potential recipient of a certificate is who she purports to be. A common technique for this is to use domain validation

(i.e., email confirmation from the registered address listed on the domain WHOIS record). Jeremy noted that this validation is susceptible to invalid certificate issuance, detracting from our trust assumptions.

The next segment of the discussion was described as "problems beyond validation." Jeremy pointed out that if the CA is subverted, then all certificates issued by that CA are at risk. Therefore, trust in the CA itself is indeed difficult to quantify. Even if human verification is provided, Verisign accidentally issued two certificates to an ex-Microsoft employee. Also, oppressive state actors could compel CAs to issue fraudulent certificates. Steven Bellovin extended this last statement to include software giant Microsoft, which is included in many government CAs, thus having the ability to issue certificates for any domain around the world. Additional problems mentioned were CA revocation inadequacies (due diligence with publishing CRLs), and SSL stripping (detecting HTTPS links and redirects from HTTP traffic such that a MITM can sniff all traffic).

Lastly, Jeremy opened up the discussion as to what we think should be done to address eroding trust. Someone suggested altering the certificate model to be more strict. For example, if a CA should become subverted, the browser should stop functioning until the end user appropriately disables the affected CA (and subordinates down the chain). Obviously, this approach is not tractable because it completely discards any notion of usability. Joseph Bonneau described his work on S-links, a method to express security policies in links which provide a trust model that is close to what an end user actually understands (hyperlinks). Brian Warner commented on YURLs, describing it as a method to specify keys in the URL.

Privacy Considerations of Genome Sequencing

Jean-Pierre Hubaux, École Polytechnique Fédérale de Lausanne

Jean-Pierre told us that genetic sequencing is following a curve faster than that of Moore's Law. The emerging use of genomics is being driven by services such as patientslikeme.com and 23andme.com, which aim to provide early diagnosis of health problems and the fine tuning of treatment. Prior to examining the privacy concerns of genomic data, Jean-Pierre informed us that we should first cover some background.

A single nucleotide polymorphism (SNP, or "snip") is a sequence variation in a long DNA molecule, where most of the encoded values are exactly the same. The human genome is approximately 3 billion nucleotides long and packed into 23 pairs of chromosomes. Genetic variations are simply varied positions in the genome. This is useful for understanding why humans are different from one another, similar to their ancestors, and the basis for how it is possible for correlating genomes to genetic predispositions.

The process of collecting a full genome involves a sample (e.g., blood or tissue) that is placed in a sequence machine, outputted as raw bits, and stored as a file. The threat of this digital acquisition of the human genome is the revelation that genomic data is privacy-sensitive because it uniquely identifies the person whom it was sampled from. Jean-Pierre recounted the misconceptions about genome privacy that attempt to derail the privacy-sensitivity observation as follows: 1) genome privacy is hopeless because we all leave behind biological cells—this is incorrect because the collection and sequencing of samples is expensive, prone to mistake, and generally not scalable; 2) genome privacy is irrelevant because genetics are nondeterminstic—this is incorrect because, for example, genetic paternity is highly related to BRCA1 and BRCA2 genes; 3) genome privacy should be considered only by bioinformaticians—this is incorrect because the bioinformaticians have yet to consider security and privacy; 4) genome privacy will be guaranteed by legislation—this is incorrect because state actors may use it themselves; 5) privacy enhancing technologies are a nuisance with respect to genetics—this is incorrect because simple anonymization and deidentification techniques may lose valuable information; and 6) encrypting genomic data is unnecessary due to the complicated structure of genomic data—this is incorrect because it doesn't assume a determined active attacker.

During Jean-Pierre's recounting of misconceptions, someone asked whether legislation for genomic privacy was actually bad. From this person's experience in the medical context, legislation was the only method for driving adoption and change. Jean-Pierre then addressed the issue of genomic data anonymization as mentioned above. Recent research by M. Gymrek has proven it possible to identify personal genomes by surname inference. This result would implicate the simplicity of the deanonymization of genetic data over time.

As for research, Jean-Pierre provided us with a list of recent work in this area. This list includes work on an efficient and secure framework for testing genomes without revealing sensitive pieces of the data; an Android application, GenoDroid, that provides genome sequencing on a smartphone device; and, a complete model for privacy-preserving computation of disease risk that performs preventative health tests for numerous system actors without directly accessing the genome sequence (this work was accepted by HealthTech this year and was authored by Jean-Pierre and colleagues).

Jean-Pierre concluded with the assertion that there is much work to be done in this area. Someone from NIST asked about the difference between genomic data and typical data (e.g., banking data). Jean-Pierre answered that genomic data is statically defined once; even after 10 decades the genome sequence is still the same. Also, the genome is more sensitive because it can be correlated across family members. As for traditional approaches

to ensure confidentiality of sensitive data (encryption), genomic data is tricky as it is approximately 300 GB per patient and thus gets really big. Someone else asked about the genomic data going to the cloud. Jean-Pierre agreed that the cloud would be a logical place to store genomic data given the scalability of cloud storage.

Mark Nunnikhoven asked about non-human genomic data and whether or not any work has been done for ensuring the privacy of this data? Jean-Pierre joked that no one concerns themselves with the privacy of mice. Mark then clarified that he is more worried about privacy controls bolting onto existing infrastructure that extends to human. Matthew Green commented that current protocols are not efficient enough to keep up with the genomic data. Jean-Pierre agreed, saying that we have to work closely with geneticists to best understand how to efficiently protect this type of data and to minimize the inefficiencies. Matt then wondered whether geneticists cared to work with computer security researchers. Jean-Pierre said yes. Joseph Bonneau asked about the barrier to collecting and sequencing DNA samples. Jean-Pierre replied that there are enormous costs associated with collecting samples. For example, is a hair from someone's house the target person's hair or his wife's?

You can check out genomic privacy work here: http://lca.epfl.ch/projects/genomic-privacy/. And here: http://sprout.ics.uci.edu/projects/privacy-dna/.

Crypto APIs

Matthew Green, Johns Hopkins University

Matt introduced crypto APIs as a broad topic that is often debated to decide whether it merits academic research or is an exercise in engineering. For this discussion, Matt wanted to answer the following questions: Is there a problem? What can we do about it? How can we address this problem?

To kick off the discussion, Matt provided a little historical context regarding cryptographic software. Crypto software is not a new thing; it's been around since the late '70s and, at the time, was only used by people who actually intended to encrypt things. When software stacks such as PGP and OpenSSL came about, the original APIs continued to target expert users who wanted to encrypt. This resulted in interfaces that were poor for general use.

Matt rhetorically asked, what's changing in software today? The answer—cryptography is scaling to applications that are not traditional security applications (e.g., a note-taking application). This would be fantastic . . . if developers were not employing outdated primitives and protocols: 64-bit RC2 and unauthenticated encryption is laughable to a cryptographer, but not so much to that note-taking application developer. Couple this with developers who learn about password protection (i.e., salt) via StackOverflow and there are more problems than just crypto primitives.

Matt believes this is our fault. Security experts have decided that this is not our problem, that we should not care about tools, and we should only be interested in building new and interesting crypto systems. He also does not believe education to be the leading cause of poor implementations but rather the lack of good tools. Jean-Pierre asked why Matt didn't think education was a useful solution. Matt responded that due to the frequency of bad implementations, he can see the use-case for an advanced API that just doesn't fail silently on bad inputs.

Strengthening his affirmation, Matt reported that the primary place you interact with crypto, as a developer, is with the API (e.g., OpenSSL, NaCl, MS Crypto). This interaction is frequent and it is never short on complexity; there are a lot of ways to get things wrong and the documentation is just short of being useless (i.e., short of examples of how to encrypt correctly and occasionally incorrect short examples). The biggest problem, however, is that most people are unaware of this problem.

Another problem, Matt described, is that developers have too many options to choose from. We often see examples of unauthenticated CBC/CTR mode block ciphers and wonder, "Why are people doing this?" Well, API developers argue, it's because we need to support legacy applications. But are we willing to allow a developer to use PCKS#1 v1.5 and be subjected to a padding oracle attack just to support legacy applications? Not to mention, standard APIs such as javax.crypto allow cipher objects to be instantiated without passing algorithm/mode/padding to the constructor, thus leaving the host system to decide the default values.

Matt informed us of additional problems with interfaces and documentation that are not useful and even misleading. For example, non-intuitive interfaces for curl, OpenSSL random functions, and bcrypt randomness can cause confusion to what the output type is and what are acceptable inputs. Additionally, Matt provided the audience with a meaningless comment found in the Microsoft Crypto API regarding certificate chain validation: "The return value indicates whether the function was able to check for the policy, it does not indicate whether the policy check failed or passed." Other problems enumerated by Matt include: few tools for key management and poor quality for the highly anticipated W3C cryptography specification.

Matt went on to discuss what we might do to make these APIs more secure with respect to misuse by referencing Dan Bernstein's crypto library, NaCl. All types are strings, and this is useful as it removes the freedom to do things incorrectly with types. However, the downside to this approach is that we have to do whatever Dan tells us (i.e., crypto blackbox API). Matt would like to see a slightly different approach where API layering is applied such that there are "crypto boxes" that allow more expressiveness than the aforementioned.

Realizing this approach is a challenge, however, because the community is not entirely sure how to treat this as a research problem. One way that researchers are trying to address this problem is to utilize a domain-specific language approach to prevent a developer from doing bad things. This is when Matt opened up the floor for discussion—how would someone from the audience address this problem in a research context?

Thomas Ristenpart (University of Wisconsin-Madison) commented that there is not a large pool of applied cryptographers, so it'd be more beneficial to get developers and cryptographers to interact and facilitate the discussion of how to move forward with crypto APIs in a meaningful way. Gary Belvin asked whether new crypto systems would be resilient to developer mistakes. Matt replied that such crypto systems haven't been well received as most academics believe that this is a simple engineering problem and not a real world problem. Someone asked Matt's thoughts on an ideal outcome of the W3C efforts. Matt responded that initially the W3C should only release the high-level API and wait on the low-level API as he is concerned that too many developers will pick the low-level over the high-level.

David Wagner asked whether we should start teaching cryptography at the lowest level: for example, moving away from block ciphers and hash functions, and starting first with the notion of a secure channel and storage. Matt liked the idea of starting the educational process in the abstract, working up to more granular implementations later on in the curriculum. He drew on the experience of first learning about the file system—having no concept of an inode but in the abstract understanding what a file was. Steven Bellovin (Columbia University) also remarked about cryptographic education. He believes that big bright warnings of "Do not roll your own . . . you will get it wrong" is useful until you've reached the point where the students start to explore crypto APIs for themselves. Of course, as a developer, you are expected to know more than one small API and thus it's not necessarily as simple as providing one good crypto API.

A crypto API developer mentioned that his team initially approached the problem of building a crypto API that was lightweight and efficient. However, as he went through the design process he kept running into contradictions and misleading assumptions. For example: How do you encrypt something securely? What database are you using? What are the key length constraints for this particular system? It ultimately ended up being a huge API.

Balancing Academic Freedom and Responsibility in Security Research

Discussion Leaders: Dan Wallach, Rice University; Kurt Opsahl, Senior Staff Attorney, Electronic Frontier Foundation

Dan informed the audience that he was looking for lessons learned. Specifically, he was interested in thinking about how it is that we do what we do, weakening known security assumptions, and deal with that in the broader context of security. To kickstart the discussion, Dan gave an intimate introduction to some research that found itself under the careful watch of the Recording Industry Association of America (RIAA).

In 2001 the Secure Digital Music Initiative (SDMI) opened a public challenge to help name a preferred digital watermarking technology that would thwart users from copying music. All four proposed technologies were broken, the RIAA was informed of the results and the intent to publish, and the work was accepted in the Proceedings of the International Information Hiding Workshop. That is precisely when the RIAA decided that it would threaten to sue the researchers for publicly disclosing their findings. The Princeton and Rice researchers' first reaction was to concede to the threat and contact Electronic Frontier Foundation (EFF). This reaction resulted in EFF legal representation and the acceptance of the paper later at USENIX 2001.

What the researchers had experienced firsthand is just how gray the area of responsible disclosure is. However, due to the lack of standardization of responsible disclosure, this doesn't mean that one experience will scale to all situations, no matter how similar. For instance, Dan later worked on hacking Diebold electronic voting machines and knew that he had to be preemptive with discussing his work with university attorneys and the EFF before anything was done. So when Diebold sent a cease-and-desist letter, they were already on top of it.

Kurt provided another example: Massachusetts Bay Transportation Authority v. Anderson, a case where three MIT students found a security vulnerability in the MBTA fare collection system. Days before their presentation at DefCon, the MBTA filed a complaint seeking a restraining order to stop the talk. Unfortunately, the judge handling the case was subject to a misinterpretation of the law believing that the disclosure of this vulnerability would go against the Computer Fraud and Abuse Act. Therefore, the judge denied the availability of the talk. As for the result, the national media coverage caused the public disclosure of the vulnerability anyway.

Edward Felten echoed the examples of Dan and Kurt. Having experienced the RIAA contention firsthand, Ed recommended the following list of critical actions: contact the company about the flaw; talk to their security team; if you cannot reach the company, talk to the people of that company in a non-public but indirect way (e.g., send a copy of the paper); and be sure to tell them who you are and what you're attempting to do. Shane Clark followed up with his experience working on medical device security. He has found that it helps to know the manufacturer culture and anticipate their mindset going in.

Jean Camp of Indiana University asked, "What are the appropriate roles in academia and institutions when students make mistakes?" Kurt answered that he would expect the institutions

to support the students. Another person recommended that students should have contact with the EFF, if only to facilitate the discussion about academic freedom and responsibility. Ed followed up with a recommendation against asking the academic institution for permission. It's ultimately your decision but the institution is more likely to be risk adverse due to liability.

Someone made a valid comment that we all want to be first to do something. Is there any interest in agreeing on academic conduct between research and industry? Dan replied that responsible disclosure seems one-sided in the relationship. Responsible disclosure is executed with the expectation that the company will do due diligence to protect its users and itself. More times than not, however, the company will refuse to believe the problem is there. Nadia Heninger reinforced this idea of a one-sided debate by offering her experiences with responsible disclosure.

Nadia's work with widespread weak keys on network devices lead to the attempted contact of 61 companies. Problem one, who do you disclose to? Thirteen of the companies had a security organization to contact directly. This is when the utilization of the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) comes in handy. The ICS-CERT will handle the contact with companies so that the disclosure can happen. However, a significant set of the contacted companies still has not responded!

Steven Bellovin altered the discussion by asking, when is there too much risk involved to publish? Matthew Green reasoned that devices with no reasonable fix, for example implantable medical devices, might involve too much risk. Denis Foo Kune described his group's approach in the medical device domain, which is to stray from direct public pressure and instead communicate with the manufacturers directly. This involves developing a relationship that will minimize the risk to patients.

Alex Halderman gave an anecdote involving voting. He prefaced his experience with, "There are limits that need creativity." Alex was looking at electronic voting in India. He had found that the hardware could be tampered with and the appropriate facts were divulged to the Indian government and vendor. He later wrote a CCS '11 paper, and a co-author made a video of the exploit on live TV in India. The government immediately denied the exploit and said the device was fake, later adjusting the statement to say the device was real but stolen. The Indian government apprehended the co-author for interrogation without legal representation, and Alex was denied entry to the country. Eventually, everything worked out but it caused Alex to recommend that everyone follow his or her ethical guidance and experienced mentors.

Concluding remarks by Ed involved openly communicating with policy-makers to inform them how laws, such as the DMCA, can hold security researchers back. Jean recommended a service for vulnerability disclosure best practices that is recent and dynamic.

Security, Usability, and Why We Have Neither

Discussion Leader: L. Jean Camp, Indiana University

Jean informed us that people tend to subvert and minimize security efforts. This is precisely why usable security is not usable; we should not be frightened of our machines. We need a usable and transparent design that clearly depicts an action-consequence relationship; if you do x then y will happen, and y is not ideal. To do this, Jean recommended that we define a consequence in our model.

She described a consequence in a risk relationship where trusting someone is a function of consequence. If you don't trust the person, there is no consequence if the person fails. In fact, you will tell more than one person if you're uninterested whether any one of them fails. However, consequences can be catastrophic and Jean recommends evaluating each variable in an action-risk-consequence relationship. To perform this evaluation, though, users need meaningful feedback.

Jean provided numerous examples of meaningless feedback, some of which include: unprotected WiFi connection messages and accessing Web pages with self-signed SSL certificates. Jean compared the certificate warning to smoking a cigarette. The box warns the user about carcinogens rather than simply stating the end result, "This will kill you."

Jean's recommendation for all of this, think translucency. You want to see through enough of the security framework to understand the risks but not enough that it hinders usability. This will provide an understanding of context and thus illuminate the action-risk-consequence relationship, though getting users to calculate online/offline risks may still be a stretch. Jean presents us with nine classic dimensions of risk.

These nine dimensions were described as follows: Voluntary-Uncontrollable (e.g., smoking vs. air pollution); Immediacy (jaywalking vs. global warming); Knowledge (genetically modified crops vs. hot stove); Knowledge of the Risk of Science (pharmaceutical vs. alcohol); Controllability (plane crash vs. automobile crash); Newness (coal-burning facility vs. Catawba nuclear factory); Common-Dream (flu vs. snakebite); Chronic-Catastrophic (two-car wreck or 16-car pile-up); and Severity (sky diving vs. cutting hand). When you provide a summation of any of the above, you instantly see risks and possible mitigations appear.

In conclusion, Jean expressed her personal belief that risk and liability is unfairly being shifted to the user. She then probed the audience on who they felt was doing a good job with usability and security. Someone responded that they thought Silent Circle was doing well as it abstracted the details of its cryptographic background and only presented useful context to the user. Mark Nunnikhoven asked about the individuals who write the security warnings. He reasoned that the atrocious attempts are the

reason why writing for engineers should be encouraged. Denis Foo Kune asked whether embarrassment might be a risk factor. Jean agreed with this suggestion and likened it to how the social implications of drunk driving have completely changed due to the perception of embarrassment.

Security and Privacy for Wearable Computing

David Wagner, University of California, Berkeley

This was quite possibly the most discussion driven presentation of everything we had seen during the HotSec symposium. Which was unfortunate since it came at the end of the day, as there was a bit of a lull in terms of response.

David started by asking the audience, "Who has wearable computing?" Denis Foo Kune raised his hand and informed everyone that he had a FitBit. David went on to enumerate cool new technologies, such as FitBit and Google Glasses, gamification, and devices such as hearing aids. He then shifted to a forward-looking approach and mentioned a belt for epilepsy, shoes that provide health monitoring, and implantable medical devices. The goal of wearable computing was described as providing continuous computer vision, speech recognition, and real-time, unobtrusive interaction with humans.

David moved on to his next query directed toward the audience: what are useful applications of wearable computing? He received a cumulative response from the audience of communications, storage, games, relaxation, authentication, augmented memory and reality, and translation. David followed up with his own: personal assistant, collaborative sensing, health and monitor, sports and exercise, and real-time advice. David then craftily referenced a relevant research pointer, Jana et al.'s "A Scanner Darkly: Protecting User Privacy From Perceptual Applications," to affirm that there are computer security research questions to be answered in the domain of wearable computing.

On that reference, David moved the discussion toward technical problems for wearable computing by asking the audience what their thoughts were on this. José Fernandez commented that we, as engineers, should learn from our past and avoid security-related mistakes by clearly defining the access and controls of a wearable device. José also noted that, unlike current embedded systems, firmware upgrades should be readily available and accessible. David was interested in this remark and thus polled the audience if they, too, believed that wearable devices should be upgradeable. The majority agreed.

The discussion then deviated a bit to alternative revenue streams that take heed of the likes of current mobile applications. The idea, as someone put it, was to have the devices work on our behalf. David and others believed this might create unnecessary confidentiality risk since devices may collect sensitive data about their wearers. As we began to flirt with the idea of security, David directed the talk toward authentication; he asked if we thought wearable computing devices are viable to authentication. Many in the audience agreed but did not provide any further insight.

David probed further by asking how we might verify authentication intent. No one had a solidified idea for how this may be achieved. Additionally, Steven Bellovin was uneasy about utilizing a device for access control if he could not manually inspect access logs and data transmission. Without a transparent system, data may as well be siphoned off by anyone. Michael Rushanan deviated from the question to ask about coercion. Specifically, he felt that having physical access to a person and a wearable device that authenticates might be easier than forcing someone to relinquish a secret that hasn't been scribed on the front of a computer monitor. David countered with an analogy where a victim has a set of car keys on her physical person; to access the car an attacker need only to coerce the victim.

Moving the flow of the discussion on, David asked the audience about their thoughts on privacy for bystanders. He believes that if we don't solve this problem technologically, it will be solved through regulation. Attendees were torn on this issue as some believed that regulation would be the appropriate mechanism, while others were conflicted on what actually is permissible in the public domain.

David's next question centered on the attacker; what is the attacker's goals and capabilities? Michael Rushanan reasoned that an attacker might have tremendous capabilities, for instance a state actor, and that their goal would be physical falsification. That is, if false data could be injected into the physically sensed world, actuation would also fail. This could potentially be catastrophic for integrated systems that actuate at the city level or even a medical device at the individual level.

The final question posed to the group targeted companies and industry; what did we think industry would be worried about? The general consensus was the inability to protect trade secrets, a lack of privacy when engaging in internal discussion, and the negative externalities associated to any accepted risk or bad practice.

Balancing Academic Freedom and Responsibility in Security Research (CONTINUED)

Discussion Leader: Kurt Opsahl, Senior Staff Attorney, Electronic Frontier Foundation

Kurt returned after the last session for an invited continuation of his and Dan's earlier discussion. Casey Henderson asked what Kurt thought about conference submission and confidentiality. Kurt felt that it was necessary to not only remove biases, but also to provide confidentiality to research that has not yet been publicly disclosed. Casey Henderson then asked about the injunction by the High Court of London on Roel Verdult et al.'s recent USENIX Security submission, "Dismantling Megamos Crypto:

Wirelessly Lockpicking a Vehicle Immobilizer." Kurt agreed that this was a problematic case and that the injunction had come at a time where the work could not be accepted.

Steven Bellovin added a personal experience in which he had been on a program committee where a member contacted a vendor to inform them that some work had been submitted that broke their design. Both Kurt and Steven found this highly unethical and is exactly why certain practices are in place to remove possible conflicts-of-interest. However, Kurt also wondered about the reviewers' role in appropriate disclosure in a similar scenario.

Kurt moved on to discuss what the academic community could do with respect to disclosures. He reported that the researcher is the most familiar expert on what has been circumvented and what the ramifications are. Therefore, it is the researcher who is most qualified to evaluate what needs to be disclosed. However, this decision should be a part of a collaborative effort involving industry engineers, security teams, and academic research groups. Kurt also recommended seeking out program committee members as potential points-of-contact to effectively reach companies.

Someone focused the discussion on disclosure by asking where a researcher draws the line when disclosing information with industry? For example, Nadia Heninger and Alex Halderman went through great lengths to inform manufacturers of vulnerabilities and did not receive a reasonable level of response. Allen Householder of CERT confirmed this result with an anecdotal experience where he posted a SQL Slammer vulnerability notice to 120–130 vendors, and the response was mostly whimsical: "We have MS SQL?" Zakir Durumeric expanded on Nadia's previous comments with disclosure, stating that we didn't know what the expectations were. For instance, should we contact sales to get redirected to the internal engineer staff? Zakir noted that once CERT became involved, the process went more smoothly with respect to contacts.

Householder provided one more comment that Cisco, and a few other industry leaders, provide best practices documentation for disclosures. Unfortunately, this is only a subset of a large set of vendors, and none of this effort is standardized or regulatory. The final question posed to the audience was: what can we do for you? No one responded to the question, but most attendees would agree that the retroactive nature of support is inefficient. Research, as a process, would benefit from interaction immediacy.