Conference Reports

In this issue:

MULTIMEDIA at USENIX

Did you know that all USENIX conference videos and MP3s are now free and open to the public? Check out the videos and MP3s of these events: http://www.usenix.org/publications/multimedia/

Plus, don't forget to subscribe to the USENIX YouTube channel for the latest conference highlights and greatest hits: http://www.youtube.com/usenixassociation

20th USENIX Security Symposium (USENIX Security '11)

August 8–12, 2011
San Francisco, CA

Opening Remarks, Awards, and Keynote Address

Summarized by Rik Farrow (rik@usenix.org)

Tadoyoshi Kohno (University of Washington), the chair of Security '12, stood in for David Wagner, who was sick, and announced two Outstanding Paper awards: to Clark et al. for "Why (Special Agent) Johnny (Still) Can't Encrypt," and to Caballero et al. for "Measuring *Pay-per-Install*." He also announced that Security '12 would be held in Bellevue, Washington, an edge city of Seattle.

Network Security in the Medium Term: 2061–2561 AD

Charles Stross, Author of award-winning science fiction

Charlie Stross pointed out that, by 2061, networking will have been around about as long as steam engines have been today, but that we ourselves might not be around, having been wiped out by some global kernel panic or a nearby cosmic ray burst. And if we don't solve the energy crisis, we won't have a network to secure—there will be no power.

Stross covered many possible future scenarios. He decried the notion of the AI Singularity, the point of human-equivalent artificial intelligences, saying this was a fantasy akin to a steam-powered tin man. Reading his speech from his iPad (no graphics), Stross spoke eloquently, sometimes dancing closer to his supposed target, network security. I highly suggest listening to the MP3 of his speech on the USENIX Web site.

Stross posited that advances in both computing and bandwidth will allow complete lifelogging. Not only will cameras and microphones record everything we see, background processing will convert printed text and spoken voice into searchable text, and face recognition will identify anyone we come into contact with. Lifelogs would be an incredibly pre-

cious resource, one that would require protection, both while being transmitted and then later, when stored.

Stross made another point that I considered very significant in the near term. Currently, service providers cap our data transfers instead of supplying the networking infrastructure that would support practically unlimited access. He said that this expense of data transfer had pushed him into turning his iPhone into a dumb phone. This bandwidth-limiting by today's providers suggests that we need to keep computation local instead of moving masses of data into the cloud for computation (the network infrastructure model). With bandwidth caps, the cloud may remain just as distant as real clouds are to earthbound humans.

Web Security

Summarized by Tamara Denning (tdenning@cs.washington.edu)

Fast and Precise Sanitizer Analysis with Bek

Pieter Hooimeijer, University of Virginia; Benjamin Livshits and David Molnar, Microsoft Research; Prateek Saxena, University of California, Berkeley; Margus Veanes, Microsoft Research

Pieter Hooimeijer presented BEK, a formal language for defining browser input sanitizers and a back-end system for supporting that language. The work is prompted by the inability to make formal determinations about the behaviors of current Web input sanitizers; for example, it is not trivial to determine whether applying a sanitizer twice—or applying two different sanitizers—may result in unsafe output. Specifying a sanitizer via BEK allows one to check whether specific strings (e.g., XSS attack) are potential outputs of the sanitizer. In addition, BEK allows one to check for properties such as commutativity, idempotence, equivalence, and reversibility. The back-end of BEK is supported by a symbolic state transducer model of the sanitizer that can be used to run analysis or generate sanitizer code.

The authors evaluated BEK using 35 currently deployed sanitizers: 76% of tested sanitizers could be ported to the BEK language without modifying the language (90% with multi-character lookahead). The authors found that BEK could check for equivalence between sanitizers in under one minute. Lastly, the authors used BEK to determine whether or not the sanitizers were capable of allowing any XSS attack strings as sanitized output.

During the questions, Hooimeijer clarified that sanitizers were manually ported to BEK, but that the BEK language was designed to resemble the way that current sanitizers are written so that coders will be able to write sanitizers in BEK.

Toward Secure Embedded Web Interfaces

Baptiste Gourdin, LSV ENS-Cachan; Chinmay Soman, Hristo Bojinov, and Elie Bursztein, Stanford University

Elie Bursztein presented work analyzing the security of Web interfaces on embedded devices and subsequently developing a framework for secure Web interfaces. Bursztein discussed the prevalence of Web interfaces for customization of consumer electronics such as routers, printers, VoIP phones, and digital photo frames: there are at least twice as many Web interfaces on embedded devices as there are traditional servers hosting Web sites. Additionally, these Web interfaces tend to be custom-developed on a tight deadline and feature-driven, leading to many vulnerabilities. The authors audited the security of over 30 devices from a variety of brands and categories, and found vulnerabilities in all devices tested.

In an effort to improve the bottom line of security for embedded Web interfaces, the authors developed WebDroid, a framework built on Android for providing secure embedded Web interfaces. More specifically, WebDroid protects against the vulnerabilities revealed in the motivating security audits. The authors performed benchmark testing to evaluate the performance of WebDroid's security features, and found that WebDroid has a 10–15% loss in performance (requests per second and process time of requests) when using security features.

During questions, Bursztein clarified that most of these embedded devices could use WebDroid after installing Android, since they mostly have ARM processors. Bursztein was also asked whether the team looked at open source router firmware such as DD-WRT and OpenWrt; they did not.

Zozzle: Fast and Precise In-Browser JavaScript Malware Detection

Charlie Curtsinger, University of Massachusetts Amherst; Benjamin Livshits and Benjamin Zorn, Microsoft Research; Christian Seifert, Microsoft

Benjamin Livshits presented Zozzle, a low-overhead in-browser method for detecting malware. Zozzle, which does static, online analysis can be contrasted with Nozzle, which performs offline runtime analysis to detect heap sprays. Zozzle detects JavaScript malware via machine learning;it was trained using one thousand malicious samples and seven thousand benign samples. Zozzle uses hierarchical features and Naive Bayes classification; it deals with obfuscated code by unfolding the code using the JavaScript runtime in the browser, then reclassifying it.

When using 300 features, Zozzle has a throughput of 1 MB of code per second. When run over 1.2 million samples of JavaScript code, this resulted in four false positives and a

false negative rate of 9%, both of which are comparable to the results given by antivirus engines.

One person asked if it is possible for an attacker to overwrite Zozzle's weight table in order to avoid detection; Livshits answered that this is a possibility but that Zozzle provides the benefit of online analysis for sites behind paywalls and other similar situations, due to its in-browser nature. In answer to another question, Livshits said that the team compared the ongoing results of Nozzle against Zozzle, and found that Zozzle identifies new malware threats before they are encountered by Zozzle. In another answer, Livshits clarified that Zozzle identifies threats beyond heap sprays.

Invited Talk

The Three Cyber-War Fallacies

Dave Aitel, CEO of Immunity, Inc.

Summarized by Nathaniel Husted (nhusted@indiana.edu)

Dave Aitel defined the three fallacies in the current understanding of cyberwarfare as being that cyberwar is asymmetric, non-kinetic, and not attributable. He gave examples of these fallacies from sites such as *The Economist* and CNAS. However, the Pentagon has defined "cyber" as a new warfare domain, thus making it a fact that can't be ignored, and modern hackers are now part of this domain.

Dave Aitel first attacked the fallacy that cyberwar is non-kinetic. The term kinetic, in this sense, is used to refer to bombs, ammunition, and other objects causing physical damage. For example, disabling a smart grid or the water pumps of New Orleans would have dire physical consequences. Another example of the kinetic nature of cyberwar is that it can change nation-state behavior. For example, sites like WikiLeaks can affect the policy and actions of a country as large as the United States. Also, considering the number of Fortune 500 companies that are most likely compromised in some way, shape, or form (Dave suggested many might not even know), it would be possible to affect their supply chain.

As for the second fallacy, attribution happens all the time in cyberwar. Dave mentions articles from McAfee, *The Economist*, and a number of other news sources. That organizations from China commenced "Operation ShadyRat" has been published in a large number of publications after McAfee's original statements. Such declarations lead to attribution.

The final fallacy is that cyberwar is asymmetric. In this case, Dave discussed the cost of both attacking and defending. The popular view is that attacking is cheap while defending is very expensive. The phrase, "An attacker only needs to find one hole while a defender has to defend many," is a prime example of this. But creating a worldwide strike capability not only requires a large network of computers but also the ability to maintain a large amount of up-to-date information regarding targets and exploits. Such a large network incurs large costs, and only an enormous organizations, like Google, Amazon, or the US government, has this sort of computing infrastructure. Moreover, creating a 0-day exploit for a piece of software takes roughly 450 hours, according to metrics obtained by Dave's company, Immunity. It takes 18 hours to run a modern exploit against a machine. If an exploited machine is discovered by the defender, i.e., cleaned up, the attacker must also assume that all their information on this machine is compromised and they must start over.

Defending against attackers is also cheaper than we have come to believe. Aitel says that the attackers are winning because they have a much better strategy. Defenders are hampered by the culture. For example, law enforcement is very successful against hackers with economic motives, but very bad about deterring anyone without a financial motive. The academic community is not a serious player in this area: many of their discoveries do not keep pace with reality. Defenders also continue consistently to underestimate their attackers' abilities. Finally, defenders more often than not continue to use software with serious vulnerabilities. Dave asked, "How many issues do you have to come up with before your company will stop using a product?"

Rik Farrow wondered about the ability of organizations to avoid using insecure software, as all software has some insecurities. Dave answered that there are relatively secure options, such as Google Chrome in the browsing market, for example, but companies don't choose them. How can organizations that completely misunderstand cyberwar use this new information to change their strategies? One of the biggest things they can do is run new purchases and products through a security team. If the security team says it isn't secure, don't use it or release it. Carson Gaspar (Goldman Sachs) said that businesses think that being secure is more expensive than being insecure and asked how this relates to Dave's talk. Dave replied that, viewed on a quarterly basis, they may be right. However, in the long run not being secure costs far more. Adam Drew (Qualcomm) asked Dave what advice he'd give to help students in academic research become more effective in this area. Dave replied that they need to be taught to think like attackers, but it's complicated. Many attackers are "crazy people" who have ingrained characteristics that make them very skilled at what they do but are not easily taught (or managed). However, he also said there are good people in academia doing good work.

The slides for Dave Aitel's talk are available at http://prezi.com/vunircise2q8/three-cyber-war-fallacies/.

Analysis of Deployed Systems

Summarized by Shane Clark (ssclark@cs.umass.edu)

Why (Special Agent) Johnny (Still) Can't Encrypt: A Security Analysis of the APCO Project 25 Two-Way Radio System

Sandy Clark, Travis Goodspeed, Perry Metzger, Zachary Wasserman, Kevin Xu, and Matt Blaze, University of Pennsylvania

Awarded Outstanding Paper!

Matt Blaze presented this security analysis of the APCO Project 25 (P25) radio system. P25 is a digital radio standard used by law enforcement groups and the US Secret Service. It provides a radio system that is backwards compatible with existing analog solutions while also supporting encryption for sensitive communications. Blaze and his colleagues identified vulnerabilities in P25 radios, including susceptibility to tracking attacks and efficient denial of service. They also found that legitimate P25 users often unknowingly transmit in the clear because of usability issues that make it difficult to verify when encryption is in use.

Both active and passive tracking attacks are possible. Active attackers can "ping" a radio with a malformed frame to which it responds whenever in range, without the victim's knowledge. Radios can be passively tracked while in use, because they transmit a unique ID in the clear with each message, though the protocol specifies an option to encrypt the ID. Denial of service attacks can be launched with a 14 dB energy advantage given to the attacker. By jamming only a 64-bit subfield, an attacker can render an entire 1728-bit voice frame unreadable. The researchers prototyped a jamming device using a $15 child's toy. While a realistic attack would require an amplifier, the prototype highlights the simplicity of the attack.

Finally, Blaze addressed usability issues and mitigation techniques. The radios tested give users little feedback about whether outgoing traffic is being encrypted, and also demodulate and play any incoming traffic without giving the user an indication of whether the traffic is encrypted. The over-the-air rekeying protocol that the radios use also fails regularly, forcing users to communicate in the clear until their radios can be rekeyed successfully. The researchers purchased hardware to measure the sensitive voice traffic transmitted in the clear in several metropolitan areas. They observed an average of 20 minutes of sensitive cleartext per city per day. This sensitive cleartext included information such as confidential informant names. Eavesdropping attacks could be mitigated by using the over-the-air rekeying protocol less frequently and by preventing unencrypted voice traffic from mixing transparently with encrypted traffic.

Questions from the audience addressed government reaction to the research and further details about mitigation. Blaze responded that the researchers approached the government "very politely" and that the government employees they interacted with all understood that identifying an attack was not equivalent to launching one. He also pointed out that all of the passive attacks they identified could be effectively stopped by improving user awareness of radio state. According to Blaze, however, the active attacks that the researchers identified are fundamental to the protocol and require a redesign to mitigate effectively.

Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space

Martin Mulazzani, Sebastian Schrittwieser, Manuel Leithner, Markus Huber, and Edgar Weippl, SBA Research

Martin Mulazzani presented this work on cloud storage system vulnerabilities. Mulazzani and his collaborators identified three attacks, all of which they launched successfully against the popular Dropbox cloud storage system.

The first vulnerability takes advantage of Dropbox's use of SHA-1 hashes for data deduplication. If a file hash already exists in the system, then the file is linked to the account rather than uploaded. An attacker can thus check for file existence or get a copy of a file, assuming knowledge of its SHA-1 hash. This attack is applicable to any cloud storage system that implements client-side data deduplication without requiring a client-side data possession proof. Attackers can also steal entire Dropbox folders if they steal a user's "Host ID," which is a unique credential used for authentication at setup time. It is stored in cleartext on the user's computer. Finally, an attacker can upload unlimited data not linked to an account by taking advantage of a vulnerability in the upload/download system. The data will eventually be reclaimed as garbage, but were reliably available for at least four weeks, according to Mulazzani's experiments. The researchers suggested several techniques to prevent the data deduplication attack by requiring client-side proofs. Since the researchers notified Dropbox of these attacks, the company has removed data deduplication, fixed the upload/download system vulnerability, and encrypted the Host ID. Mulazzani noted that the plaintext Host ID is still resident in RAM, so the folder stealing attack is more difficult, but not impossible.

During the Q&A, Ian Goldberg suggested that one approach to data deduplication is to challenge the client to compute a MAC of the file. Mulazzani agreed, but speculated that it might be slower in the case that the file already exists. Mark Seiden (Yahoo!) asked if the researchers had verified that

Dropbox computes the SHA-1 hash at the server for each upload. Mulazzani confirmed that the system does so.

Comprehensive Experimental Analyses of Automotive Attack Surfaces

Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, University of California, San Diego; Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno, University of Washington

Cars are increasingly complex systems that typically contain 10+ electronic control units (ECUs), embedded systems. In this work, the researchers extracted firmware from several ECUs, reverse-engineered it to identify vulnerabilities, and finally created a series of attacks, all of which give the attacker complete control of the vehicle (brakes, engine, locks, etc.) without requiring physical access.

Stephen Checkoway described attacks using the car's media player, Bluetooth interface, and telematics unit (used for systems such as OnStar). The media player attack used a specially crafted WMA file, the Bluetooth attack used a trojaned Android phone, and the telematics attack could be triggered via the audio in a phone call. Checkoway played a video demonstrating a compromise via the telematics unit in which a remote researcher was able to unlock the car, disable the anti-theft system, and start the engine, allowing an onsite researcher to simply drive the car away. A second video demonstrated surreptitious tracking and audio recording via the onboard GPS and telematics systems. Checkoway attributed the proliferation of vulnerabilities mainly to a lack of past adversarial pressure and the heterogeneous multi-vendor development of modern car systems. Almost all the bugs they found appeared at component boundaries, often through incorrect assumptions made by suppliers. Finally, Checkoway said that relevant stakeholders such as SAE, USCAR, and the US Department of Transportation have been notified of the vulnerabilities and are taking action in response.

Bill Cheswick asked if modern military gear used the same equipment. Checkoway said that he did not know. J. Alex Halderman asked if a monoculture might be worse for security than the heterogeneous status quo. Checkoway responded that he is not sure and clarified that the researchers did not mean to suggest that one vendor should make all systems, but that fewer vendors should be used for each car. Mike Ryan (ISI) asked if the researchers could steer the car remotely. Checkoway said that they could not steer the car and did not test acceleration because of the risk involved. Rik Farrow asked how widespread the compromised telematic unit is. Checkoway answered that they only tested one make and model so he is unsure, but his understanding is that each manufacturer uses at least one unique telematics unit.

Forensic Analysis

Summarized by Lakshmanan Nataraj (lakshmanan_nataraj@umail.ucsb.edu)

Forensic Triage for Mobile Phones with DEC0DE

Robert J. Walls, Erik Learned-Miller, and Brian Neil Levine, University of Massachusetts Amherst

A typical crime scene investigation includes lots of digital evidence such as computers, mobile phones, etc., and it takes law enforcement agencies quite a while to extract data from these devices. In order to acquire evidence quickly and on-scene, Robert Walls proposed a system called DEC0DE for forensic triage of mobile phones. The authors chose mobile phones instead of computers since phones are not only ubiquitous but also contain key information (address books, images, etc.). For this work, the authors specifically dealt only with feature phones, which account for 60% of the phones used in the US. This system extracts digital information directly from the phone storage in, at maximum, around 20 minutes. The interesting point to be noted is that this system is agnostic to the file system and operating system of the phone. This is important, since it allows the possibility of handling phone models that have not been previously seen without any extra work (critical for triage).

The input to the system is raw storage (stream of bytes) from a phone. In order to remove unwanted bytes that need not be parsed, the raw storage is first filtered using a technique called block hash filtering (BHF), which preserves important fields such as timestamps and phone numbers. The system later locates these fields and interprets a combination of fields as a record. As the name suggests, BHF is carried out by dividing the storage into blocks and computing a hash on every block. Duplicate blocks are filtered out by comparing the hashes against a library of known hashes so that only important data is retained. Experimental evaluations on different phone models show that this filtering helps in removing lots of extraneous data (69% on average). There was also a lot of overlap between phones of the same model.

Once the filtering step is completed, the next step is inference. The system uses machine learning algorithms for this step, with the assumption that similar phone models have similar call logs. The formats are encoded using probabilistic finite state machines and then parsed using a dynamic programming algorithm (Viterbi). The state machines differ depending on the record of consideration. In the end, a decision-tree-based classifier helps to remove false positives. The whole system was evaluated by manually selecting known models from different manufacturers and known records and verified on models that closely match the former. Around 93% of the records were recovered. The main limitation of

this system is that the authors assume raw storage can be acquired, which itself is a great challenge. Another limitation is that success depends on the quality of the state machine.

In Q&A, Rik Farrow mentioned that often service providers can transfer contacts from an old phone to a new phone with ease, so is it not a better idea to use those same tools? Robert replied that even their own tools don't do a good job. He illustrated this by describing how his lab mate had the service provider transfer contacts for him, but the process only transferred a small fraction of the contacts; his system, however, was able to extract all the contacts without any changes to his system. Ben Fuller asked about the poor performance working with two LG phones. Robert answered that the system could get better as more phone models become available.

mCarve: Carving Attributed Dump Sets

Ton van Deursen, Sjouke Mauw, and Saša Radomirović, University of Luxembourg

Sjouke Mauw started off by saying that there is a general feeling that MIFARE transportation cards can be easily hacked. In this work, the authors verified this by finding vulnerabilities on a Luxembourg-based transportation card called the e-go card. To kick-start the project, they first used standard data carving tools from digital forensics, which gave them a lot of dumps along with attributes such as identity of the card, purchase date, and number of rides left. They then posed a research question: given a series of dumps with many attributes, is it possible to map a dump attribute to the set of dumps? They made some strong assumptions, though, such as equal length dumps, same location of attributes in every dump, and that the encoding of an attribute is the same or deterministic. Although these assumptions seem very strong, they are acceptable given that they are dealing with dumps of transportation cards.

Sjouke discussed strategies for finding all possible mappings of an attribute to a dump set. The first strategy is based on commonalities. It is done by XORing the set of dump bits and finding the common indices. The second strategy is a little more complex and is based on dissimilarities. The whole methodology was validated by making several trips on a bus and manually noting known attributes such as date, rides left, etc. On applying the above algorithms, several attributes were found and matched with the manually noted set. And all this took only a few seconds per card. Sjouke concluded with future work such as automatic encoding and better algorithms to improve robustness. The current version of the tool is open source and available for download from http://satoss.uni.lu/mcarve.

Do the bits need to be consecutive for the dissimilarity algorithm to work? The consecutive bits were only shown for illustration and are not required by the algorithm. Someone asked about applications besides transportation cards and what knowledge was needed by the system about a card. Sjouke said they need contextual information about what is represented in the bit stream. Another application could be protocol reverse engineering, for which memory dumps are not needed but the communication can be inspected.

ShellOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks

Kevin Z. Snow, Srinivas Krishnan, and Fabian Monrose, University of North Carolina at Chapel Hill; Niels Provos, Google

Code injection attacks are one of the most common methods of gaining control over a computer. Readily available exploit kits are making it very easy to deploy exploits. At a higher level, a code injection attack transfers the application control flow to a code supplied by the attacker. In most cases, though, it transfers the control to a shell code regardless of the method of exploitation.

Detecting the shell code itself aids in subverting a code injection attack. Lightweight emulation based on dynamic analysis is the state of the art in detecting shell code, but it is very slow. Emulators have also proven to be detectable, which makes lightweight emulation-based analysis weak. Kevin Snow described how the authors designed a new, faster, and more efficient dynamic analysis technique which is not based on emulation. For this sole purpose, they built an operating system, called ShellOS, that executes code streams. The entire OS consists of approximately 2500 lines of code written in C and assembly code.

The most interesting and useful part of this system is that it can run on a standard OS as a guest OS using hardware virtualization. When ShellOS first boots, it creates a suitable environment to execute shell code by allocating memory given by some user-supplied process memory snapshot. The host OS then supplies the ShellOS with a code stream to analyze through a shared memory region, which triggers ShellOS to execute the code stream from every triggered offset. ShellOS observes faults and timeouts in the code. In order to trace these to memory, ShellOS catches page faults at memory addresses that are defined by the above heuristics. To determine the effectiveness of their system, they conducted some experiments on throughput and detectability and compared them against the state-of-the-art shell code detection system called Nemu. The experiments showed ShellOS to be faster and more efficient than Nemu.

They did a case study on a real world scenario where shell codes are injected in PDF files. They used 374 suspicious PDF documents provided by Google that were collected between 2007 and 2010. They also had a benign set to test

false positives, which consisted of 179 PDFs from previous USENIX conferences. The system was initially able to detect 325 PDFs from the malicious set and the remaining were detected after unpacking. None of the PDFs from the benign set was flagged as malicious. Although ShellOS is fast and detects shell code effectively, it does have some limitations. First, it is not easy to extract shell code. Second, hardware virtualization may also be detectable. However, future versions of ShellOS may not need hardware virtualization. The authors plan to release the source code of ShellOS soon.

What would be the effect if the shell code invokes an API call that affects the external environment, such as file manipulations or process creation? Is another process created in the virtual machine? The results of the API calls are simulated within ShellOS, and process call creation is not currently supported. The shell code will still be detected but their system will not be able to follow it in the diagnostics. What if an attacker has access to their system and keeps tweaking his code till the system does not detect it? That would certainly be a problem, as with all other approaches.

Invited Talk

Crossing the Chasm: Pitching Security Research to Mainstream Browser Vendors

Collin Jackson, Assistant Research Professor at Carnegie Mellon University

Summarized by Mihir Nanavati (mihirn@cs.ubc.ca)

Collin Jackson addressed some of the reasons why very few ideas proposed in academia for increasing browser security ever get adopted by the browser community. He discussed some of the fundamental differences in the goals of academia and publishing papers, and those in building browsers for mass market adoption. Using real-world illustrations, he also gave some "rules" to keep in mind to try to increase the possibility of getting a feature added to a browser.

Jackson started by quantifying what crossing the chasm to mass market adoption really means. Jackson argued that even very popular add-ons such as NoScript are only used by a small fraction of the total user base. To really make an impact, a feature needs to make it into the browser's main code. The easiest way to do this is to be picked up by at least one of the major browsers, which is usually followed by adoption by the other browser vendors.

Getting a few people interested in an idea is easy, and even getting several technically oriented early adopters on board is not too hard a feat. It is at the next stage, that of making it acceptable to ordinary users, that most flounder. This is a fundamental difference between browser add-ons and the core engine—add-ons are aimed at power users. Features that require extensive user interaction or break a large proportion of the Web are perfectly acceptable for add-ons but are impractical for core adoption.

While it is tempting to attribute lack of adoption to the inertia or general laziness of browser vendors, this is unfair. Browser vendors can move very quickly if the feature meets certain criteria and is deemed necessary to the community—clickjacking defenses like x-frame-options were implemented in every major browser in under two years, while history privacy has largely been adopted in under a year.

Jackson then considered the differences between ideas that had successfully been adopted and those that had failed to make the cut. Generally, browser vendors tend to favor small, simple features that fix something that is badly broken. Ones that can be implemented in a verifiable way, preferably across multiple browsers, and don't break existing Web pages are preferable. This is in contrast to academia, which tends to reward novel ideas that open new avenues for research and often involve complex and significant implementations.

He outlined a set of general guidelines on how best to select ideas for browser adoption. Ideally, features should attempt to make themselves indispensable by solving a real world problem, especially those that are receiving a fair degree of media coverage. Getting adopted by a single Web browser and championed by large Web sites are good ways of getting noticed. Same-origin policy is an example of such an indispensable feature, as is PostMessage, which was originally introduced by Opera to allow different browser windows to communicate without making a round trip to the server.

Second, sometimes even imperfect solutions may be preferred over more complex solutions if they are easily implementable in a standard way and can be deployed unilaterally. Features that require cooperation from Web sites often break a lot of functionality, since Web sites are very slow to react to changes in browsers.

Finally, low-risk proposals have the best chance of adoption. Generally, people are annoyed when Web sites do not load, and rather than understanding the reasoning behind it or filing a bug report, they tend to just switch to another browser. If it is imperative to break functionality, Jackson strongly suggested minimizing the impact by analyzing how necessary the break was, and whether the feature could be made opt-in and Web sites gradually be persuaded to use it.

Switching back to the theme of academia vs. industry, Jackson noted that feature evaluations in research tended to be far short of what is expected in industry. Simply verifying the front page of sites on the Alexa Top 100 or that the browser

can still play YouTube videos is completely insufficient. Evaluations involving a deep crawl of the Web site and using client-side measurements are far more likely to reveal compatibility problems due to the addition of a feature.

Jackson then asked whether there was any point in pursuing bold and complex solutions, since the probability of them ever getting mainstreamed is so low. He concluded that even if such ideas are never mainstreamed, they push the boundaries of research and could be successful even if only a very small subset of the entire system they proposed gets adopted.

The questions revolved around whether browser vendors were being too conservative and trying to protect users too much. Couldn't Web browsers just leave the decision of trusting a Web site or not to the user? Jackson explained how Web developers would like to provide users with rich functionality, often using JavaScript, regardless of the user's trust perception of the Web site, making sandboxing and protection in Web browsers necessary. Jackson was then asked if a way to improve research evaluations would be to release better benchmarks to the community. While wholeheartedly approving the idea, he had doubts about whether this was possible, due to copyright issues. The session concluded with the observation that there was a disconnect between developers and users, and that a feature that required any amount of effort from a user would be unpopular and likely to be turned off. For this reason, the importance of keeping features simple cannot be overstated.

Static and Dynamic Analysis

Summarized by Samee Zahur (sza4uq@virginia.edu)

MACE: Model-inference-Assisted Concolic Exploration for Protocol and Vulnerability Discovery

Chia Yuan Cho, University of California, Berkeley, and DSO National Labs; Domagoj Babić, University of California, Berkeley; Pongsin Poosankam, University of California, Berkeley, and Carnegie Mellon University; Kevin Zhijie Chen, Edward XueJun Wu, and Dawn Song, University of California, Berkeley

Domagoj Babić introduced their new tool for dynamic symbolic analysis, MACE. Although many companies already use such tools for some of their software development projects, he noted that testing remains the most widely used means of weeding out software bugs and vulnerabilities. While most existing automated tools do not remember anything from one iteration to the next, MACE improves on this by learning an approximation of the application's state space and then using that approximation to guide further search. Its effectiveness was demonstrated particularly well for programs implementing various network protocols, such as Vino and Samba, where their abstract execution pattern naturally fits into the finite automaton structure used by their models here.

In the rest of the presentation, Domagoj outlined how dynamic symbolic analysis normally works, and how their approach differs. Normally, the process starts by running through the program with some concrete input sequence and collecting a trace of every single branch condition. This produces a set of constraints that the input sequence must satisfy in order to follow through the same path in the program. Other paths are then explored by negating the last constraint of each prefix of each constraint sequence. Their procedure, however, takes an additional input from the user: an output abstraction function. This groups all program outputs into coarse-grained categories or abstractions, and this is what determines the quality of the learnt program model. Using this, and a variation of the L* learning algorithm, they were able to deduce the sequences of inputs that cause significant program state changes. Thus MACE would iteratively build up a deterministic finite automaton (DFA) modeling internal state changes of the program, and also produce exact sequences of inputs that will cause transitions between the states. As new states and transitions of this DFA are discovered, they are fed back into the learning algorithm. Further exploration is then guided by this model, as MACE can now use the known input sequences for transitioning between states to start further exploration at any given state. It can also filter out redundant input sequences that cause the same transitions, making the analysis more tractable.

This enabled them to find a number of vulnerabilities in network programs, ones Domagoj called "deep vulnerabilities," that is, those hard for an unguided analyzer to find. He explained this by showing a graph that demonstrated how an unguided analyzer quickly loses its ability to explore deeper states. At the end, however, one of the audience members noted that the improvement in code coverage over an unguided search seemed to vary—6.5% for Vino versus 59% for Samba—and asked why it was so. Domagoj conjectured that since Vino implemented a simpler protocol, it is likely that the baseline was already able to explore a large part it. Session chair Sam King asked what was the most surprising discovery they made during development. Domagoj answered, "It was surprising to see that it works!".

Static Detection of Access Control Vulnerabilities in Web Applications

Fangqi Sun, Liang Xu, and Zhendong Su, University of California, Davis

Fangqi Sun presented their work on static analysis of Web sites to detect access control violations. Even large companies like Bloomberg and Disney often have such vul-

nerabilities on their Web sites, where they make implicit assumptions about access control policies and simply forget to place guards on sensitive Web pages. This often allows an attacker to gain access to restricted parts of a Web site by just typing a URL in a browser. She pointed out that frequently used methods of code review are neither comprehensive nor efficient, and automated detection of access control vulnerabilities is often hard in the absence of formal specifications. Their approach, instead, was to automatically explore hyperlinks produced by PHP scripts to produce role-specific sitemaps. Once they obtain such sitemaps for normal users, sysadmins, etc., their tool can automatically attempt to explore pages that should be accessible by one role and not another (e.g., by sysadmins and not normal users). If it succeeds, the tool flags a vulnerability.

Rationales behind various design choices were given. For example, Fangqi described how static analysis provided better code coverage compared to dynamic analysis techniques, but also required the use of context-free grammars to approximate various links produced by PHP scripts. The presentation ended with evaluations and some limitations of their tool. It was able to find vulnerabilities in both traditional and Web 2.0 applications. The evaluation also demonstrated the usefulness of a specialized tool: it can scan through 12,000 lines of code in two minutes.

Someone asked how their tool explores privileged pages not found in the code. Fangqi said developers can manually specify additional Web pages to explore. Session chair Sam King asked how dynamically generated links were being handled. Fangqi said that JavaScript-generated links are indeed a limitation, as she had already pointed out, and they intend to address it in the future.

ADsafety: Type-Based Verification of JavaScript Sandboxing

Joe Gibbs Politz, Spiridon Aristides Eliopoulos, Arjun Guha, and Shriram Krishnamurthi, Brown University

As advertisements and other mashup components in today's Web applications often require the use of third-party JavaScripts, they also require sandboxing to provide safety guarantees. Arjun Guha said that the authors' work tries to verify safety properties of various trusted sandboxing libraries often employed, for which they specifically focus on Yahoo!'s ADsafe library as a typical example. Arjun showed how common such third-party scripts are, how hard it is to verify sandboxing libraries that are trusted to provide safety, and how even a single mistake can provide attackers with the upper hand. The rest of the presentation consisted of the details of how a type-based static checker can provably guarantee the safety of a sandboxing library, and how they used it to verify the ADsafe library.

Arjun described, one by one, how their type system guarantees each of the claimed safety properties, e.g., not being able to load arbitrary code at runtime, not being able to affect DOM outside a designated part, etc. One of their key observations was that ADsafe already requires JavaScript codes to pass an existing static checker, JSLint. They could therefore design their type system to be a superset of everything JSLint accepts, allowing them to significantly simplify their design of the type system.

Arjun ended with several bugs they found in ADsafe with their automated system, as well as a bug in JSLint. Sam King asked whether we should move to a better language that is easier to check, or to a subset of JavaScript. Arjun answered that JSLint and other static checkers already do require code to be rewritten in a restricted subset of JavaScript, to the point where it is almost a new language. David Evans (University of Virginia) noted that the use of a keyword whitelist (as opposed to the keyword blacklist of identifiers, as used here) may be more effective in filtering unsafe attributes. Since browsers are adding new features and keywords all the time, some of them may be unsafe. Arjun replied that if the external environment cannot be relied on, even a whitelist-based filter can be defeated, pointing out that redefinition of built-in keywords by the hosting page is always a problem. A questioner sought clarification on which bugs were empirically found and which proven by their type system. Only the JSLint bug was empirically found.

Invited Talk

I'm from the Government and I'm Here to Help: Perspectives from a Privacy Tech Wonk

Tara Whalen, Office of the Privacy Commissioner of Canada

Summarized by Julie Ard (julieard@gmail.com)

The Canadian Office of the Privacy Commissioner (OPC), established in 1983, has a mandate to oversee compliance with the Canadian Privacy Act, covering governmental privacy, and its expansion in 2000 to protect and promote the privacy rights of individuals. The OPC acts as an ombudsman. Its powers include investigation, audit, and the ability to pursue court actions, publicly report on information handling practices, promote public awareness, and to support research on privacy issues (they awarded $350,000 in grants last year for privacy research and public education projects). The Technology Analysis Branch in the OPC supports investigations, among other duties. Technologists undertook two major technical investigations which Tara discussed in detail, emphasizing the importance of government employees

and investigators having technical expertise. One investigation concerned Facebook's privacy policies and possible conflicts with Canadian privacy law. The second investigation was into Google's inadvertent collection of data from WiFi networks while taking pictures for their Street View service.

Data protection authorities exist in over 40 countries (predominantly in Europe). The most similar governmental organization in the US is the Federal Trade Commission. A very active discussion included questions regarding how both Facebook and Google responded to the OPC's complaints. Over 30 countries were involved in the Google complaint. Many simply requested that data associated with their citizens be deleted. Some (including Canada) requested access to the data so that they could perform their own investigation. Canada's investigation was performed on-site; copies of the data in question were not made. American lawsuits are ongoing. A question from the audience initiated a discussion about secure deletion and technical assurance.

Another member of the audience observed that companies tend to push the line on privacy, characterizing Facebook's privacy policies as a moving target and suggesting that they tend to beg for forgiveness rather than ask permission. For example, the Facebook CEO stated publicly that our notions of privacy are obsolete. The audience recognized that the choices these companies make affect society and privacy standards. Governmental data protection authorities expend vast resources investigating, and making complaints and recommendations. This process may take several months to a year, depending on complexity. The OPC does not think that it is a losing battle or a "done deal" that privacy is obsolete.

Someone asked whether it's possible to use location services for oneself but not share that information with Big Brother."One can disable location services altogether, but it would be more desirable for the individual to choose which applications can use location data. This preference can vary based on the application. For example, in the US government employers do want information from BlackBerry to track their government employees. A member of the audience asked whether that happens in Canada; Tara said that such actions are covered by established legislation.

Another topic presented was that of lawful intercept based on a case study covered in the presentation of a German who published his own mobile data in order to see what could be determined from that data. A discussion ensued on the topic of lawful intercept for law enforcement and national security. In Canada, numerous bills have been proposed but none have been passed. Someone asked about data crossing borders into the US, for example, where organizations are required to retain Patriot Act data. Tara said that to established legislation covers those situations.

Tara encouraged researchers to make their work presentable, visualizable, and accessible so that it can influence the world of politics. Evidence is vital for informing the policy debate. She applauded the creation of tools like Tor to empower citizens. Canada's role in establishing the facts of the Google and Facebook cases heavily influenced their outcomes: Facebook's initial reaction was to implement Canada's requests, and Google's press release essentially mirrored elements of Canada's complaint by promising to implement changes that the OPC suggested. Tara reiterated that these debates have the power to shape company policies, as evidenced by Google and Facebook's reactions to the complaints, and that Canada values its role in fostering global cooperation.

Understanding the Underground Economy

Summarized by Robert Walls (rjwalls@cs.umass.edu)

Measuring Pay-per-Install*: The Commoditization of Malware Distribution*

Juan Caballero, IMDEA Software Institute; Chris Grier, Christian Kreibich, and Vern Paxson, University of California, Berkeley, and ICSI

Chris Grier began the session with an in-depth look at the ecosystem that has built up around pay-per-install (PPI) services. PPI services provide a way for clients to quickly install their malware on a large number of pre-compromised hosts by simply purchasing installs from these services. To measure the PPI ecosystem, they infiltrated four programs and set up a number of hosts, in geographically diverse locations, to automatically download the malware provided by the PPI service. This allowed the team to perform real-time monitoring, infer the types of clients using PPI services, and estimate the financial impact of a botnet takedown.

They drew three major conclusions from their study. First, they found that PPI services are popular: 12 of the 20 most common malware families are at least partly distributed by PPI services. Second, they found that malware regularly performs repacking to avoid detection, every 11 days on average. Third, clients target specific geographic locations for their malware, resulting in differing demand and therefore different install rates for each country. Chris attributed this demand to the client's ability to monetize their malware in each particular country. For example, their measurements indicate that spambots tend to be installed uniformly across different countries, while click-fraud binaries largely focus on Western countries. Finally, Chris mentioned that they observed instances of PPI arbitrage, where individuals would exploit price differences between PPI providers by buying installs from one provider and selling them to another.

Dan Farmer asked if someone could exploit the PPI system by cheaply acquiring virtual hosts through cloud services,

selling them to the PPI providers, and then turning off the hosts. Chris replied that this is probably possible, but the PPI services will eventually detect the abuse and withhold payment. One audience member asked how the volume of PPI installation of malware compares to other distribution mechanisms. Chris said they are working on expanding their study to include this data, but they do know that most of the popular malware uses multiple installation vectors. Another attendee inquired about the specific payment mechanisms. Chris explained that most programs advertised payouts using WebMoney and some mentioned PayPal. Finally, Jelena Mirkovic (ISI) wanted to know about the implications for researchers. Chris suggested that researchers should be aware of how malware fits in the PPI ecosystem. He reiterated that malware using PPI services may not include any infection mechanisms. Anecdotally, Chris said they found PPI loaders that were misclassified as another family of malware. Such loaders might exhibit different behavior each time they are run. Co-author Vern Paxson helpfully added to Chris's comments by pointing out that the PPI loaders can be used as a source to acquire new samples of malware.

Dirty Jobs: The Role of Freelance Labor in Web Service Abuse

Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, University of California, San Diego

Marti Motoyama continued the session with his work on the role of freelance labor in abusing Web services. He argued that scammers, spammers, and other Internet denizens can leverage the large labor pool provided by sites such as freelancer.com and Amazon's Mechanical Turk to cheaply and effectively abuse free Web services. Marti used Gmail and spamming as an example, pointing out how one can use outsourced human labor to circumvent many of Gmail's technological protections against creating bulk accounts. They estimate that about 30% of the jobs on freelancer.com are abusive. Marti primarily covered three different job types commonly submitted to freelancer.com: account registration, online social network linking, and search engine abuse. For each of these types, he commissioned his own job on freelancer.com to measure the quality of the workers' responses.

The goal of the first job type, account registration, is to obtain access to a large number of accounts on a target Web service. After looking at seven years' worth of job data on freelancer.com, Marti and his colleagues found that Gmail and Craigslist were the most targeted Web services for this job type. Marti commissioned the task of creating Web-based email accounts to 10 workers, the majority of whom delivered valid accounts. He observed that the accounts in many of the delivered sets were fairly old, indicating that they were stockpiled and not created on demand. The second job type he analyzed was online social network (OSN) linking. He defined OSN linking as buying friends, followers, or subscribers on sites such as Facebook, Twitter, and YouTube. They found that the commissioned workers were generally unable to deliver high-quality social links, since many of the delivered links came from fake OSN accounts. Finally, Marti covered search engine abuse, specifically jobs for creating written content that contains certain links or keywords. He observed that 10% of the jobs seen on freelancer.com fall into this category. The freelancer.com workers delivered mixed results for this task, with some doing very well and others ignoring job requirements.

Marti concluded that the large, cheap labor pool available to abusers changes the threat model to Web services and that traditional security mechanisms are not sufficient to stop abuse. However, he claimed it is possible for outsourcing sites to detect and remove abusive jobs. During the Q&A, Tyler Moore asked whether sites like freelancer.com are actually interested in filtering abusive jobs, given that they earn revenue from these jobs. Marti responded that they do very little enforcement, especially when compared to similar sites such as Amazon Mechanical Turk. He then added that freelancer.com is a legitimate business, so they might be willing to address the issue if the scope of the problem is brought to their attention. Finally, another attendee commented that even if freelancer.com is taken down, it is likely that another site will be created to provide the same abuse service. Marti agreed that this is a possibility.

Show Me the Money: Characterizing Spam-advertised Revenue

Chris Kanich, University of California, San Diego; Nicholas Weaver, International Computer Science Institute; Damon McCoy and Tristan Halvorson, University of California, San Diego; Christian Kreibich, International Computer Science Institute; Kirill Levchenko, University of California, San Diego; Vern Paxson, International Computer Science Institute and University of California, Berkeley; Geoffrey M. Voelker and Stefan Savage, University of California, San Diego

Chris Kanich started his presentation by pointing out two questions he is commonly asked about spam: who buys this stuff and how much money do the spammers make? In beginning to address these questions, Chris said that, at its core, spam is about advertising goods. The spammer—"affiliate marketer," in spam parlance—earns a commission on every sale they can provide to their affiliate program. Chris found that the order IDs for many affiliate programs appeared to be sequential and thus he was able to measure the IDs over time and estimate the sales throughput for those programs. Overall, the throughput varied from as low as 49 to nearly 900 orders per day. By combining the throughput with an estimated cost per order, Chris was able to calculate the

average revenue per month for each program. This revenue varied from $200,000 per month to as high as $2,400,000 per month for the larger spam pharmacies.

Chris explained that they inferred information about the purchasers using the Web logs of a compromised image hosting server. The spam sites received views from all across the world, but sales were concentrated in the United States and Western Europe. Chris estimates 91% of all customers to be located in Western countries. While the vast majority of purchases are for recreational drugs such as Viagra, 29% are for non-recreational pharmaceuticals. US-based customers are four times more likely to buy non-recreational drugs than other Western customers.

Mark Seiden (Yahoo!) questioned the legitimacy of the drugs. Chris said that the drugs they tested contained the active ingredient in the correct amount, but they couldd not make any claims about other aspects of the drug. Another attendee asked about the percentage of purchases that arrived. Chris said that most arrived; the ones that did not were likely due to errors on his part. John Spring wanted to know to what extent this becomes a public health problem. Chris commented that their goal is to bring this issue to light and they are currently in contact with the FDA. Finally, an attendee brought up the issue of credit card fraud, pointing out that selling these drugs is already illegal, so why don't the programs go that extra step? Chris replied that these are businesses, and it is trivial for customers to contact the credit card company to cancel orders. In fact, the customer service for these programs tends to be very good.

Invited Talk

Privacy in the Age of Augmented Reality

Alessandro Acquisti, Associate Professor of Information Technology and Public Policy at Heinz College, Carnegie Mellon University

Summarized by Nathaniel Husted (nhusted@indiana.edu)

Alessandro Acquisti started his talk by discussing Cincinnatus, a Roman consul who, after being called back to service, defended Rome from northern invaders. A photograph of his statue was shown, in which the general is returning a symbol of military power with one hand and retrieving his agriculture tools with the other, dramatizing his choice to return to private life after his military victory. This story was used to indicate the importance of private life in ancient Rome. Alessandro then retold a story regarding a man who destroyed the legendary Temple of Artemis in Ephesus so that his name would be recorded for all history. The individual was captured and killed by the citizens of Ephesus. His captors also attempted to purge his name from history, but failed. We know the individual as Herostratus. Together, the two stories illustrate that throughout the course of human history we have been concerned with both our public and our private lives, as well as with controlling the information about us in the public sphere.

Alessandro's talk revolved around four major research experiments performed by Alessandro and his co-authors: the inconsistency of privacy evaluation, the paradoxical nature of privacy control, humans' ability to discount past information, and the use of social networks and face recognition for individual re-identification. These experiments try to look at how technology affects our privacy decisions and how privacy decisions affect our technology; how we make trade-offs and decisions regarding what information we want to keep private and make public; and what are the cost-benefit trade-offs in revealing private information.

The first experiment focused on whether people's evaluations of privacy can be manipulated. The experiment contrasted the willingness to accept cash to reveal personal data versus the willingness to pay cash to protect personal data. The results showed that participants' valuations of privacy changed significantly based on the priming and framing of the offer. If they started with less privacy, they valued privacy less; if they started with more privacy, they valued it more.

The second experiment focused on the paradoxical nature of control and its relation to privacy. Traditionally, control over personal information is believed to be a means of protecting privacy. Their experiment investigated whether more control can lead to less privacy. For this experiment, more than 100 students were asked to perform an online survey where a portion of the questions asked were sensitive in nature. One version of the survey stated that the answers to the survey, if provided by the subjects, would be published by researchers; the second version of the survey allowed individuals to choose what answers would be published. When allowed explicit control via the added box, individuals not only answered more questions but also allowed more answers to be made public. The results of the experiment show that making people feel more in control over their privacy can lead to more public disclosures of sensitive information.

The third experiment focused on how we judge individuals for past and present behavior, both good and bad. The experiment consisted of a survey in which individuals were asked to read a story about Mr. A, who either found a purse and kept a large sum of money or returned the purse with the money. This event either occurred five years ago or 12 months ago. Individuals formed very negative impressions of Mr. A when he was presented as having kept the money, no matter how long ago that event happened. However, when Mr. A was presented as having returned the money, individuals thought positively of him—but only if his good deed happened

recently. If the good deed happened five years ago, there was no positive impact.

The fourth, and most recent, experiment concerned Alessandro's work on combining Facebook, personal information facial recognition software, and data mining. It is this portion of the talk where Alessandro's group is bridging the gap between science fiction and modern society. In this project they were able to compare images from Facebook's searchable profiles with head shots taken manually on the CMU campus or with images coming from dating Web sites, with the goal of identifying individuals online and offline. The culmination of this project was a sample augmented reality application that can perform the transfer from face to personal information on a mobile smartphone.

Alessandro discussed how these new technologies will affect our views on privacy. We can view our social network profiles as real IDs. In fact, social networks have, in some ways, turned into an inadvertent national ID. The convergence of various technologies also creates a "democratization of surveillance," because in a world where all personal information can be gathered from a face, we all become each other's big brothers with the aided use of a mere smartphone.

Some questions focused on whether younger people value privacy less; on whether the amount of value a person places on privacy changes if the loss is more concrete; on whether gender affects our decisions regarding discounting individuals' behavior; and on what the take-home message from the talk was. Alessandro tackled the first question by mentioning that what is most likely to happen is that views on what should be private and what public will change with time. Alessandro found that the concreteness or abstractness of the reward did not affect a person's behavior in the first experiment. He also found that, in the current studies, gender did not have a significant effect. Finally, the positive message from this talk was the hope provided by research advances in privacy enhancing technologies (PETS).

Defenses and New Directions

Summarized by Ben Ransford (ransford@cs.umass.edu)

Secure In-Band Wireless Pairing

Shyamnath Gollakota, Nabeel Ahmed, Nickolai Zeldovich, and Dina Katabi, Massachusetts Institute of Technology

Shyamnath Gollakota presented a protocol that allows a user to pair two wireless devices. When two devices are paired, they share a secret and can authenticate each other's transmissions. Some consumer-grade wireless devices, such as WiFi routers and Bluetooth audio equipment, establish a shared key via Diffie-Hellman (DH) key exchange when the user presses a button on both devices within a certain time window. However, on a wireless medium, simple DH is vulnerable to man-in-the-middle (MITM) attacks. Past academic work has proposed pairing protocols that require trustworthy out-of-band channels to bootstrap mutual trust, but Gollakota argued that out-of-band channels are difficult to incorporate in devices such as medical and home sensors, for reasons of both cost and size.

Gollakota described a new pairing protocol, Tamper-Evident Pairing (TEP), that is secure against MITM attacks and uses only in-band communication. The key idea is that, because an adversary cannot create radio silence by transmitting, pairing devices can reliably detect MITM tampering. TEP surrounds DH packets with a long leading synchronization packet and a specially constructed trailing hash. These modifications make TEP secure against adversarial message alteration, message hiding, and channel hogging. Because TEP messages are tamper-evident, if each pairing device receives exactly one untampered-with pairing request during the designated time window, they have successfully authenticated each other and can pair. The authors implemented TEP in the driver of a mainstream 802.11 card and evaluated it on a network test bed at MIT.

An audience member suggested that an attacker could selectively overpower the silent hash bits. TEP uses a balanced hash with an equal number of ones and zeroes to ensure that every one corresponds to a radio-silence zero. Zack Weinberg asked whether users would be willing to wait for the timeout period. A user with physical access can press a hardware button to preempt the timeout. Carson Gaspar (Goldman-Sachs) asked whether an attacker could overpower both the synchronization packet and the balanced hash. Such behavior would be detectable. Someone pointed out that TEP devices might interpret cross-technology interference from other products (e.g., microwave ovens) as TEP synchronization packets; Gollakota responded with some empirical data to demonstrate that only certain devices would interfere, and that those interfering devices would delay the pairing by only a few minutes. If a device persistently interferes, it breaks the paired devices' ability to communicate at all.

TRESOR Runs Encryption Securely Outside RAM

Tilo Müller and Felix C. Freiling, University of Erlangen; Andreas Dewald, University of Mannheim

Tilo Müller described a Linux kernel patch that allows AES operations to occur entirely outside of RAM. The patch addresses a well-known shortcoming of most cryptography implementations targeting microprocessors: secret keys are stored in RAM, where they are vulnerable to attacks that allow a miscreant to dump system memory—for example,

the "cold boot" attacks presented at USENIX Security in 2009. Even popular full-disk encryption (FDE) implementations store secret keys in vulnerable memory. TRESOR implements AES in such a way that key material is stored in processor registers rather than RAM. It uses the large debug registers that are available only to processes running at the highest privilege level. Under normal operation, these registers are unused; they are available for breakpoints and rarely accessed configuration information. To circumvent RAM storage of keys, the authors implemented AES in x86 assembly, avoiding putting runtime variables in data segments and using 2-kilobit x86 SSE registers for intermediate states.

Müller compared TRESOR to an AES implementation using only generic x86 instructions, which was too slow, and to an implementation using Intel's new AES-NI instruction set, which provides fast AES instructions but stores round keys in insecure RAM. TRESOR, in comparison, generates round keys on the fly and does not store them in RAM. Müller noted that the kernel's normal context switching stomps on the debug registers, inspiring the authors to make TRESOR run in an atomic section. In the authors' evaluation under QEMU, their search of emulated RAM with the open-source aeskeyfind tool failed to find the key under TRESOR but succeeded under the alternative schemes. In future work the authors plan to store keys in the Trusted Platform Module (TPM) or x86 machine-specific registers (MSRs). TRESOR is open source software available at http://www1.cs.fau.de/tresor.

An audience member noted that TRESOR keeps secrets in RAM briefly before they are moved to the debug registers, and asked whether the secrets had to be in RAM at all. Müller replied that passwords on Linux can be much larger than any available register. Frank Stajano asked about the performance impact of TRESOR and whether off-the-shelf FDE is usable under TRESOR; Müller said that TRESOR's overhead compared to AES-NI was not huge. John Criswell noted that an attacker could change the kernel in memory or on disk to attack TRESOR, which Müller acknowledged. Another audience member asked whether the authors had studied other side channels; Müller reported that TRESOR should be resistant to timing attacks because the code does not use input-dependent branches; he pointed out that the authors had not yet considered power side channels.

Bubble Trouble: Off-Line De-Anonymization of Bubble Forms

Joseph A. Calandrino, William Clarkson, and Edward W. Felten, Princeton University

Bubble forms are machine-readable pieces of paper on which people place marks to indicate their preferences or opinions. Will Clarkson pointed out that bubble forms are used for voting in some precincts and then posted online, making them an attractive target for malefactors who wish to know how certain people voted. Clarkson's group used a set of 92 anonymized surveys to train a classifier on several features of the markings such as shape, radius, center, and color distribution. They trained the classifier on 12 (out of 20) filled bubbles per person, then tested the classifier's performance on the remaining eight. The results invalidated the common assumption that people cannot be identified by their markings on bubble forms: their classifier ranked the true respondent over all others more than half the time on 1200 dpi scans. Clarkson reported that the scans were robust against downsampling, with 45% of respondents correctly identified at only 150 dpi. Clarkson suggested several applications of bubble-form de-anonymization, such as the detection of cheaters on standardized tests. He concluded by suggesting several ways of making bubble forms more robust against their attacks; for example, making bubbles' borders thicker decreased the classifier's accuracy.

An audience member asked whether the authors' chosen features were robust against changing the environment in which a person filled in the form; Clarkson explained their attempts to avoid overfitting by their classifier and remarked that people seem to be consistent as conditions vary. Another person asked about varying the scanner; Clarkson said that their blurring step normalized for scanner variations. Adrian Mettler suggested that users could use felt-tip pens to confound de-anonymization. An audience member asked whether stress affected bubble-form filling, and Clarkson acknowledged that it might. Another audience member asked whether the authors' techniques could de-anonymize users from a much larger set; Clarkson agreed that further testing was necessary but said that voting, for example, often occurs in smaller precincts in which their classifier could work. Peter Neumann (SRI) suggested that a malicious party could de-anonymize voters using other techniques, such as marking ballots with invisible ink. Clarkson clarified the authors' assumption that the parties that receive the bubble forms are not tampering with them.

Securing Search

Summarized by Ed Gould (summary@left.wing.org)

Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade

Nektarios Leontiadis, Carnegie Mellon University; Tyler Moore, Harvard University; Nicolas Christin, Carnegie Mellon University

Nektarios Leontiadis described their work measuring and analyzing specific attacks against search results, namely those used by illicit online pharmacies. This work seeks to determine the size, effectiveness, and weak points of the attacks. The specific choice of drug sales was motivated

by the potential dangers of improper use of the drugs. As a form of illicit advertising, email spam is inefficient. Social network and blog spam are better, but Search Engine Optimization targets users more directly. A Google search for "no prescription cialis" will produce some odd results; some are legitimate, some are malicious.

To collect data, the team issued more than 200 queries daily during the experimental period. They note that SEO attacks are growing, while blog and email spam are declining. The number of pharmacies (both legitimate and illicit) is constant. Infections on the attacked systems tend to be long-lived, and seem to persist the longest on .edu sites. The researchers identified 34 connected components of the attack. They noted seven organized groups, loosely connected, that represent 50% of the infected nodes. Eleven ASes hosted most of the redirect servers.

They conclude that there is one major group of affiliates perpetrating these attacks, and that .edu sites are popular to attack.

Neil Schwarz asked why .edu domains take longer to disinfect. It is difficult to notice the infections. Lucas Ballard asked about query selection: when looking for good sites, how often do bad sites outperform good ones? They only have aggregated results, not differentiated by type of query. Lucas followed up by asking why the domain count is rising. Is the count limited to domains (domain rotation) or does it also include IPs? They just counted domains, not IPs. Stefan Savage pointed out a source of possible bias by using page access as an estimator. Some sites use on-site billing, others off-site. Off-site billing involves an extra redirect, and their data do not include payment sites.

deSEO: Combating Search-Result Poisoning

John P. John, University of Washington; Fang Yu and Yinglian Xie, MSR Silicon Valley; Arvind Krishnamurthy, University of Washington; Martín Abadi, MSR Silicon Valley

John described a tool, called deSEO, to combat Search Engine Optimization (SEO) attacks. The "malware pipeline" is, roughly, find vulnerable servers, compromise them to host malware, and spread links via search results that point to the malware. A Google search for "flintstones pictures myspace" yields a "scareware" link as the first result, claiming that the user's computer is infected by one or more viruses. About 40% of popular search terms are infected (changing, as what's popular changes). There is an estimated $150M profit in the scareware market.

John described how the mechanisms work, what the research can show, and the development of deSEO. Their analysis of attacks was carried out from August to October 2010. E-commerce sites often contain credit card information, which offers an additional incentive to compromise them. Files are uploaded to the compromised servers, and use PHP to generate malware pages. They were able to download a PHP script from a buggy server, even though this is not usually possible, and were thus able to analyze the script.

The general pattern is that there is a dense link structure on the pages generated, linking to more than 20 other sites with some 40 million pages generated, poisoning 20,000 popular search terms. Ninety-five percent of Google Trends terms are poisoned, targeting 100,000 victims over 10 weeks. It is difficult to detect and blacklist these pages, because they are typically cloaked to crawlers and search analysis, as the PHP scripts detect crawlers and produce benign results. Often it takes user interaction (e.g., mouse movement or clicks) to produce the malware. Features that can cause a site to get noticed are diverse behavior before compromise and similar behavior among pages after compromise.

The deSEO tool uses history-based filtering, cluster analysis of suspicious domains, and similarity analysis. They found about 1000 domains, with 15,000 URLs infected.

Alva Couch pointed out that John had just told us how to defeat his technique: use sparsity. Do they have any fallback mechanism? John replied that relevant keywords are still necessary, and clustering is needed to get the rankings. It is possible that the bad guys could use this information, but it seems unlikely.

Someone asked how the malware keeps lists of pages to link with up-to-date, and John answered that the malware server includes a list of sites to link to, and the PHP script selects from this list.

Securing Smart Phones

Summarized by Italo Dacosta (idacosta@gatech.edu)

A Study of Android Application Security

William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri, The Pennsylvania State University

One of the reasons for the increasing popularity of smartphones is the great number of mobile applications available. For example, Google Android, one of the most popular mobile OSes, has hundreds of thousands of mobile applications available in the Android market. However, these applications are not security certified, due to their large numbers and the lack of a common definition for security. As a result, malicious applications can be found in the Android market. This paper describes a breadth of security properties in a large set of popular Android applications to characterize their security and provides a better understanding of mobile applications' and developers' behaviors.

The sample set used in this study consisted of the top 10 most popular applications in each of the Android market's categories—a total of 1,100 applications. To analyze the security and behavior of the applications, access to their source code was required. Android applications are written in Java but use a different bytecode (.dex files) and runtime (Dalvik virtual machine); therefore, existing Java decompiler tools cannot be used directly. Hence, the authors built a Dalvik decompiler, ded, which takes the application's .dex files as input and returns the corresponding Java source code. The ded decompiler works as a multistage process (retargeting, optimization, and decompilation) and it is available on the project's Web site. Using ded to decompile the selected applications produced a total of 21 millions lines of code. The authors performed static and manual analysis of this code to look for dangerous behavior and vulnerabilities and to understand how applications handle sensitive information. The static analysis used Fortify SCA, a commercial tool for Java vulnerability analysis. The authors created custom rules to analyze the applications' source code, using different techniques such as control flow, data flow, structural analysis, and semantic analysis.

This study reports 27 findings which provide insight into the applications' and developers' behavior. In the area of phone identifiers, the authors found that 33 applications leak phone IDs. Regarding location data, 13 applications were found with location data flows to the network. The authors also found that 51% of the applications include an ad or analytics library. In many cases, applications have more than one third-party library. In addition, this study shows evidence of the use of developer kits, which hide the identity of the original developers and may include some dangerous functionalities. Also, several Android-specific vulnerabilities were detected. William Enck described some of the limitations of this study, such as the focus on popular applications, code recovery failures, limitations of the static analysis tool, and obfuscated code in some applications. Finally, Enck noted that this study offers the opportunity for a more automated security certification process for mobile applications.

Bill Soley (Oracle) asked if there are other implications besides privacy regarding phone identifiers such as IMEI numbers. While other malicious activity is possible, right now the main concern is privacy. David Evans (University of Virginia) said that developers are not being malicious but are just making mistakes and that a possible solution could be to generate unique IDs that do not leak privacy. Enck agreed and pointed out some recent research that follows this approach. Another participant asked if native code was found during the decompilation process. Yes, around 70 or fewer applications had native code; this is an area malware developers are beginning to push. Finally, someone asked how Enck would change the application sample set if he could. Enck would probably obtain a list of all available applications and select applications randomly. Also, the lists of recently added and paid applications can be interesting to study.

Permission Re-Delegation: Attacks and Defenses

Adrienne Porter Felt, University of California, Berkeley; Helen J. Wang and Alexander Moshchuk, Microsoft Research; Steve Hanna and Erika Chin, University of California, Berkeley

In modern client platforms such as browsers and mobile operating systems, applications are untrusted and isolated from each other by using IPC and specific communication mechanisms. They also require explicit permission to access resources such as camera, microphone, and user location data. These permissions are assigned per application to reflect user needs and level of trust in the application. However, a system that uses IPC and per-application permissions can be vulnerable to permission re-delegation attacks, where an application that lacks permissions gains access to additional privileges by communicating with another application (a special case of the confused deputy problem).

Adrienne Porter Felt described how they analyzed the permissions of 872 Android applications to find candidates that could facilitate this type of attacks. They found that 37% of the applications meet the required conditions for a candidate: a dangerous permission and a public interface. To discover the attacks, the authors built an automated tool that uses call graph analysis, and they manually verified the attacks found. The authors found 15 vulnerabilities in 5 system applications; however, other vulnerable applications may not have been detected.

Felt presented IPC Inspection, an OS or browser mechanism to prevent permission re-delegation attacks. When a deputy application (the privileged application) receives a message, the system reduces the deputy's permissions for the length of the session to the intersection of the deputy's previous permissions and the requester permissions. Also, to prevent DoS attacks, the deputy can specify who can and cannot send it messages. IPC inspection was implemented for Android OS and ServiceOS (Microsoft's research browser). The evaluation focused on determining whether IPC Inspection does not break applications and whether it effectively blocks permission re-delegation attacks. The evaluation results showed that 11 out of 20 randomly selected Android applications (from the set of 872) may require minor changes or additional permissions. In addition, the evaluation showed that IPC Inspection prevents all of the permission re-delegation attacks described in this study.

William Enck asked about the case where, in install-time systems, an intentional deputy attenuates authority, which

can lead to permission bloat. You could add a time-of-use check, specifically for this case, that does not need to be necessarily a permission prompt. Felt also noted that people in her research group are working on "user driven access control with access control gadgets" to give the OS a way to know that an action is being approved by the user. Also, someone asked about when a singleton application is used. In this case, the deputy needs to declare itself as a singleton, because otherwise the application could crash. This problem does not happen on the Web, only in the Android OS. Adrian Mettler (UCB) asked about the possibility of escaping from the stack introspection protection option. This is possible, but developers could end up using this for all their messages. However, the option can be added, to make sure it does not break the application and to help application developers.

Quire: Lightweight Provenance for Smart Phone Operating Systems

Michael Dietz, Shashi Shekhar, Yuliy Pisetsky, Anhei Shu, and Dan S. Wallach, Rice University

Android protects applications from each other by using OS security mechanisms. In this model, applications should be slick (i.e., minimal permissions). Instead, however, most applications are complex, due to the use of third-party libraries such as mobile ads and mobile payments. Third-party libraries typically require additional permissions not originally required by the application and that can introduce bugs that affect the application's stability. Also, applications and third-party libraries mutually distrust each other. A simple solution to this problem is to split apart third-party libraries into separate applications. However, this approach introduces a new problem—it increases the risk of confused deputy attacks. In this type of attack, an application lacking a particular permission sends a request through another application that has this permission (confused deputy). The second application then forwards the request to the OS, allowing the first application to evade the permission mechanism.

Dietz described QUIRE, a mechanism that enables the separation of libraries from applications and protects data provenance and integrity, while preventing confused deputy problems. For this purpose, QUIRE introduces the idea of provenance-carrying IPC, where an application can protect itself by quoting the call chain that called it. Quoting only reduces the privileges of the application that chooses to quote the call chain; therefore, confused deputy attacks will not work even if a malicious application lies about the call chain. In addition, QUIRE provides verifiable communication between applications by using simple cryptographic mechanisms to protect data moving over IPC and RPC channels. Moreover, QUIRE does not require changes to the Dalvik virtual machine.

The QUIRE implementation consists of four components: the authority manager OS service, the network service provider OS service, the IPC stub/proxy code generators, and the trusted UI. To evaluate QUIRE, the authors built two demo applications: a secure mobile payment system and a mobile ad service. Through these applications, the authors demonstrated the security benefits and practicality of QUIRE. In addition, the performance evaluation showed that QUIRE overhead is small (80 microseconds per IPC).

Arjun Guha (Brown University) asked what applications' installation looks like under the QUIRE model. Dietz responded that applications will need to use dependencies at installation time to learn what other applications need to be installed. William Enck asked if provenance happens on intents. Dietz explained that QUIRE hooks to the service binding IPC at this point. He has not looked at intents yet but anything using Binder should work well. Paul Pearce asked about ad networks functionality not supported by QUIRE prototype implementation. Dietz responded that he could not think of one at the moment. QUIRE was designed as a system that application developers can use to build a policy on top and, in some cases, it may break functionality. Dave Evans asked what happens when applications use the network instead of IPC for communications. Dietz responded that this will prevent provenance, but it has not been an issue yet. It will be something to consider when the boundary between mobile and Web applications blurs.

Invited Talk

Deport on Arrival: Adventures in Technology, Politics, and Power

J. Alex Halderman, Assistant Professor, Computer Science and Engineering, The University of Michigan

Summarized by Adam Bates (amb@cs.uoregon.edu)

J. Alex Halderman presented stories from three strands of his research—early digital-rights management attempts in audio CDs, security analysis of voting machines in the United States, and security analysis of voting machines in India. Through these stories, Halderman explained the risks for researchers whose work leads them to a stand-off with politically or economically powerful parties. He also demonstrated the importance of being able to explain highly technical security issues in a manner that is palatable to the public.

Halderman's work in digital rights management began at Princeton University as a graduate student in 2003. At the time, companies like Sony were trying to secure their intellectual property that was being distributed in a legacy format, the compact disc. In an early generation of this technology, Halderman discovered that Sony was leverag-

ing the Windows Autorun feature to install software that interfered with the CD Driver. Using the Freedom to Tinker blog as a mouthpiece, he posted that the DRM software could be avoided by holding down the Shift key as the CD was inserted. The "DRM is defeated by the Shift key" story caused the responsible company's stock to drop by 80%. Halderman also spoke out against Sony's infamous DRM-as-rootkit attempts, going back and forth with the company in a "delightfully public" manner. Communicating these issues via a blog helped the Center for Information Technology Policy (CITP) to speak directly to the public. The negative publicity eventually forced Sony to abandon the initiative.

Halderman next related his history with the Diebold voting machines. The move to electronic voting systems was motivated by the voting fraud vulnerabilities of bulky, lever-based machines. Unfortunately, the early generation machines were rushed to market without much regard for computer security. Companies like Diebold had not voluntarily subjected their machines to any kind of independent analysis. In 2006, the CITP lab at Princeton was able to acquire a machine and reverse engineer the hardware for thorough analysis. They discovered and published a number of easily deliverable vulnerabilities, including the ability to infect a machine with malicious software without leaving a trace. It was also possible to create a virus that could spread from machine to machine. This led to another public standoff, where Diebold touted the importance of overlooked security features such as the need for a key in order to gain physical access to machine hardware. However, only one key was used universally and it was easily obtainable commercially.

Halderman went on to work on California's "top-to-bottom" voting machine analysis. California was one of the first states to recognize the threat that insecure electronic voting posed. Under threat of decertification, voting machine manufacturers were required to share their code with the study. However, the fact that the study was being called for by politicians exposed a potential conflict of interest. For this reason, it was important to this research team that they had permission to share their results with the public.

The efforts of Halderman and his students at the University of Michigan also helped to draw attention to very serious issues in Washington DC's prototype Internet voting system. The system was about to go live for an actual election when they were opened for security probing. Halderman's group launched attacks that altered ballots and broke the confidentiality of legitimately cast votes. In spite of the fact that they added the "Hail to the Victors" audio track to the vote confirmation page, their penetration went unnoticed for several days. This work eventually helped to derail Washington DC's use of the online system.

More recently, Halderman became involved in an analysis of India's voting machines. In spite of solid design and effective deployment, fraud was rumored to have occurred in Indian elections. The study found two serious vulnerabilities: (1) installing a dishonest display board by replacing the LED component; (2) designing a device that modified the votes while in storage on the EEPROM. As a result of these findings, Halderman and his colleagues fell out of favor with the Indian Election Commission and with local law enforcement. After Hari Prasad, one of the Indian collaborators, promoted these findings on television, he was detained by the police. The Commission finally accepted the need for change, but on a subsequent trip to India Halderman was barred from entering the country for 24 hours. Stalling for as long as possible to avoid Halderman's deportation, the Election Commission was able to speak on his behalf and get him into the country as their guest. The Commission, now prototyping a paper trail add-on, is seen as a model for developing democracies.

Lessons learned included the power of being technically correct, the importance of effective communication with the public, and the dire threat to democracy posed by insecure electronic voting systems. Halderman concluded by charging the audience to continue to change the world through computer security.

Poster Session

First set of posters summarized by Michael Z. Lee (mzlee@cs.utexas.edu)

IMD Shield: Securing Implantable Medical Devices

Shyamnath Gollakota and Haitham Al Hassanieh, Massachusetts Institute of Technology; Benjamin Ransford, University of Massachusetts Amherst; Dina Katabi, Massachusetts Institute of Technology; Kevin Fu, University of Massachusetts Amherst

Benjamin Ransford (ransford@cs.umass.edu) presented this work. The goal is to counter a set of attacks on implantable medical devices (IMDs) published in 2008. The primary issue is that some devices are susceptible to passive and active attacks. However, invasive surgery to retroactively fix these issues is expensive and carries risk, so the authors sought another solution. Their proposal is a wearable device, called the IMD Shield, that uses friendly jamming to block messages to and from an IMD. This device blocks incoming active attacks as well as outgoing messages. The radio configuration employs two antennas, which allows them to simultaneously receive the sensitive signal from the IMD and jam the signal so that eavesdroppers cannot decode it. The IMD Shield's random jamming signal works like a one-time pad; it is the only device that is able to decode the new signal.

Using GPUs for OS Kernel Security

Weibin Sun and Robert Ricci, University of Utah

Security can be computationally expensive, but some operations can be parallelized and would benefit greatly from using the computational power of a GPU. Weibin Sun (wbsun@cs.utah.edu) presented KGPU, a kernel driver that leverages the GPU to offload expensive but easily parallelized operations such as encryption and AV signature matching. Because the current interface to GPUs is through a proprietary driver, they use a helper program to translate between KGPU requests and CUDA calls. Although this requires extra memory copying from kernel to user space, it seems to provide a nice speedup.

The Art of War Applied to Intrusion Detection in Wireless Ad-Hoc Networks

Stefan Stafrace and Bogdan Vrusias, University of Surrey

When working with intrusion detection systems in wireless ad hoc networks, the efficient use of resources is key, because nodes in ad hoc networks are resource-constrained. In a traditional network, you're able to deploy intrusion detection systems in strategic choke-points, but in wireless ad hoc networks this is not possible, due to the use of the shared medium and node churn. Stefan Stafrace (s.stafrace@surrey.ac.uk) suggests applying risk-based military principles to efficiently detect intrusions in wireless ad hoc networks. The authors offered a case study in which systematic route patrols were conducted by squads of agents to detect a sinkhole attack. The results show that high detection precision can be obtained while also conserving resources and limiting the data packet loss due to the attack.

A Digital Forensics System Using a Virtual Machine Monitor Integrated with an ID Management Mechanism

Manabu Hirano and Hiromu Ogawa, Toyota National College of Technology, Japan; Takeshi Okuda, Nara Institute of Science and Technology (NAIST), Japan; Eiji Kawai, National Institute of Information and Communications Technology (NICT), Japan; Youki Kadobayashi and Suguru Yamaguchi, Nara Institute of Science and Technology (NAIST), Japan

Manabu Hirano (hirano@toyota-ct.ac.jp) presented this poster. When performing digital forensics, one can run into the problem of unattributed data tampering, which can lead to false accusations and other bad outcomes. The authors propose a system called BitVisor, a hypervisor-based solution that employs user ID management to securely record who is accessing and modifying data. The end result is that the VMM is able to securely store the ID outside the reach of the guest operating system, translating actions in the guest OS with a helper program installed by the hypervisor. They use TPM and SecVisor-like (Cylab/CMU) properties to guarantee that an attacker cannot statically replace the VMM or tamper with its data dynamically during runtime.

Automated Model-based Security Management of Web Services

Rajat Mehrotra and Qian Chen, Mississippi State University; Abhishek Dubey, Institute for Software Integrated Systems, Vanderbilt University; Sherif Abdelwahed, Mississippi State University; Krisa Rowland, US Army Engineer Research and Development Center

Rajat Mehrotra (rm651@msstate.edu) presented an autonomic performance and security management framework for Web services. The goal is to integrate system control, optimization, and security analysis into a common model-based framework. It enables distributed Web services to efficiently adapt to varying load requirements and identify and mitigate potential security incidents. In modeling the behavior of a system, the authors wish to efficiently estimate system behavior and make adjustments as necessary. Using various inputs from security, performance, network, and system measurements, they can differentiate between different safe and unsafe system scenarios.

NotiSense: An Urban Sensing Notification System to Improve Bystander Privacy

Rob Smits, Sarah Pidcock, Ian Goldberg, and Urs Hengartner, University of Waterloo

Although crowd-sourcing data collection using mobile devices is de-anonymizing for the participant, bystanders should be notified so that they can preserve their privacy while protecting the identity of the data collector. Sarah Pidcock (snpidcoc@cs.uwaterloo.ca) presented NotiSense, a service to help notify such bystanders. The authors accomplish this by collecting enough information about the data collector, hashing and filtering the locations, and then having the collector's mobile device rebroadcast information. Bystanders in the area can see these broadcasts, check whether they're affected, and notify users. In a field test, they find that it is effective enough to cover a reasonable area around a data collector.

Secure Computation with Neural Networks

Brittany Harris and Jiamin Chen, University of Virginia

Brittany Harris (bjh3ev@virginia.edu) and Jiamin Chen (cjmyezi@gmail.com) presented this work. Oblivious computation can be used to jointly compute values while preserving each party's privacy. This work applies Yao's Garbled Circuit to enable the joint computation of the weights of a neural net.

This allows two parties to jointly train a neural net using data from both parties, without exposing their private training data or intermediate weight results to the other party. Alice first computes the weights using her training data directly, and then Alice and Bob execute a garbled circuit protocol where the inputs are Alice's learned weights and Bob's training data, to obtain the final weights without revealing either the intermediate results or training data.

SPATor: Improving Tor Bridges with Single Packet Authorization

Rob Smits, Divam Jain, Sarah Pidcock, Ian Goldberg, and Urs Hengartner, University of Waterloo

Tor is used for anonymity but is susceptible to some kinds of attacks. Rob Smits (rdfsmits@cs.uwaterloo.ca) and his colleagues are addressing an attack on Tor clients who have opted to become Tor bridges. The adversary assumes that the correct IP address for his victim is contained in one of the bridge descriptors. He can perform aliveness checks on the Tor bridges he has collected and then take an intersection of bridge IP addresses that were detected as online to de-anonymize this Tor client. The authors propose that, as clients receive bridge descriptors, an additional time-limited key be included. From this, clients derive a ConnectionTag—a 32-bit field, encoded in the initial sequence number and TCP timestamp of the initial SYN packet. If it does not validate, the Tor Bridge can drop the request before revealing aliveness.

Vulnerabilities in Google Chrome Extensions

Nicholas Carlini, Adrienne Felt, Prateek Saxena, and David Wagner, University of California, Berkeley

Adrienne Felt (apf@cs.berkeley.edu) presented this poster. Chrome allows users to install extensions that run with elevated browser privileges. Bugs in extensions can leak privileges to malicious Web sites or active network attackers. To help mitigate this, Chrome's extension platform includes several security features. However, in analyzing the top 50 Chrome App Store extensions and 50 randomly selected extensions, the authors found that 42 have vulnerabilities. For example, the Google Voice Chrome extension automatically searches for strings that look like phone numbers and converts them into links that, upon click, will make a call. Thus, a malicious site can use JavaScript to click pay-per-call numbers. Isolated worlds successfully reduce the number of vulnerabilities that a malicious Web attacker can leverage, but there are numerous bugs that active HTTP modification can attack. In general, privilege separation is not effective, because developers circumvent privilege separation, either intentionally or accidentally.

Unifying Data Policies across the Server and Client

Jonathan Burket, Jenny Cha, Austin DeVinney, Casey Mihaloew, Yuchen Zhou, and David Evans, University of Virginia

Web applications currently take a decentralized and ad hoc approach to security. Austin DeVinney (adevinney@radford.edu) and Yuchen Zhou (yz8ra@virginia.edu) presented a unifying framework applied to specific security policies once and then automatically enforced throughout the application. On the server side, they provide GuardRails, an additional layer on top of Ruby on Rails which allows an author to specify certain server security properties that are automatically enforced throughout the application. In addition, they modified the Chromium browser to interpret the generated attributes and enforce policies that protect private content from untrusted scripts running in the browser.

Improved XSS Protection for Web Browsers

Riccardo Pelizzi and R. Sekar, Stony Brook University

Riccardo Pelizzi (rpelizzi@cs.stonybrook.edu) presented this poster. Chrome and IE have implemented detection for reflected cross-site scripting (XSS) attacks from GET and POST parameters. However, the two browsers do not detect partial XSS injection attacks that take advantage of existing scripts already in the Web page. The authors found that 8% of the Web sites surveyed are vulnerable to this type of attack. Their approach to this problem is to improve filtering by parsing the input from GET and POST requests into parameters, and to use approximate substring matching to cover a wider range of Web application sanitization logic. Their implementation and evaluation shows that they cover more cases than Chrome's own filtering with an acceptable overhead as compared to Chrome.

Challenges in Deployment and Ongoing Management of Identity Management Systems

Pooya Jaferian, University of British Columbia; Kirstie Hawkey, Dalhousie University; Konstantin Beznosov, University of British Columbia

Pooya Jaferian (pooya@ece.ubc.ca) presented this poster, which tries to answer the question, how do people (corporations) do ID management? Their preliminary results from collecting and analyzing support logs show that, overwhelmingly, issues arise during installation. The process of troubleshooting is a close collaboration among consultants, support staff, and users employing a variety of content and debug methods such as interactive debugging, screen shots, and calls through many rounds of communication over a variety of channels.

An Arithmetic Operation Implementation Strategy for Privacy-Aware Role-Based Access Control

Yoonjeong Kim, Hyun-Hea Na, and Ji-Youn Lee, Seoul Women's University; Eunjee Song, Baylor University

Role -based access control (RBAC) is a model that effectively limits security vulnerability by controlling access to a specific role. However, the model is incomplete—the intent of a user is equally important when trying to enforce least privilege. Yoonjeong Kim (yjkim@swu.ac.kr) presented this work whose goal is to add arithmetic operations to allow easier specification of purpose, obligation, and conditions of access. To this end, the authors use XPath and XML specification to port Java applications to allow for arithmetic operations.

Second set of posters summarized by Christian Rossow (christian.rossow@gmail.com)

AdSentry: Comprehensive and Flexible Confinement of JavaScript-based Advertisements

Xinshu Dong, National University of Singapore; Minh Tran, North Carolina State University; Zhenkai Liang, National University of Singapore; Xuxian Jiang, North Carolina State University

Xinshu Dong presented AdSentry, a framework to reliably execute JavaScript Web advertisements. The system is based on a shadow JavaScript engine that is used as a sandbox to run untrusted ads in parallel to the normal JavaScript execution. The sandbox monitors accesses, and access control policies help to mitigate the insecurity of malicious JavaScript code. AdSentry was implemented as a prototype for Mozilla Firefox.

The Socialbot Network: When Bots Socialize for Fame and Money

Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu, University of British Columbia

Many users of social networks make their personal data private and only accessible to their friends. Yazan Boshmaf (boshmaf@ece.ubc.ca) presented work in which the authors created more than 100 artificial Facebook accounts and analyzed how users reacted to friendship requests from these accounts. More than a third of the friend requests were accepted. As a consequence, attackers were able to gain significantly more personal data about other users than is accessible via public profiles.

DETER Testbed: New Capabilities for Cyber Security Researchers

Terry Benzel, John Wroclawski, Bob Braden, Jennifer Chen, Young Cho, Ted Faber, Greg Finn, John Hickey, Jelena Mirkovic, Cliff Neuman, Mike Ryan, Arun Viswanathan, Alefiya Hussain, and Stephen Schwab, USC Information Sciences Institute (ISI); Brett Wilson, Cobham, Inc.; Anthony Joseph and Keith Sklower, University of California, Berkeley

DETERlab, the DETER Testbed, is an Emulab-based cluster testbed for cyber-security research which has been operated for many years by the DETER group. This poster showed new capabilities of the testbed, particularly that DETER is able to scale up to emulating an entire Internet infrastructure, including an autonomous-systems network.

An Analysis of Chinese Search Engine Filtering

Tao Zhu, Independent Researcher; Christopher Bronk and Dan S. Wallach, Rice University

Tao Zhu (zhutao777@gmail.com) presented work that analyzes the extent to which Chinese search engines censor search results for specific keyword groups. The authors found that pornographic terms and names of politically important persons are commonly filtered. A long-term analysis shows temporal changes in the censorship, presumably caused by extending filter blacklists. The authors also observed that some search engines maintain whitelists of presumably safe Web sites for specific search keywords.

More Efficient Secure Computation on Smartphones

Sang Koo, Yan Huang, Peter Chapman, and David Evans, University of Virginia

Yan Huang (yhuang@virginia.edu) presented this work on efficient and privacy-conforming data calculations on smartphones. The authors explored a protocol to find common contacts between two mobile phone users without sharing any contacts. Efficiency is gained by using a garbled circuit framework. The authors also show other privacy-preserving use cases for their framework, e.g., to determine geographical proximity between two mobile devices.

Understanding Attacks

Summarized by Robert Walls (rjwalls@cs.umass.edu)

SMS of Death: From Analyzing to Attacking Mobile Phones on a Large Scale

Collin Mulliner, Nico Golde, and Jean-Pierre Seifert, Technische Universität Berlin and Deutsche Telekom Laboratories

Collin Mulliner began his talk by pointing out that previous work on mobile phone security has largely neglected the more common feature phone in favor of smartphones. In fact, feature phones still dominate the market, with some estimating that only 16% of mobiles are smartphones. For this work, Collin set out to test the security of feature phones by looking at the SMS implementations across a variety of different phones. Since it is infeasible to test all models, he focused on

phones from the most popular manufacturers. Collin claimed that due to the reuse of phone platforms, a bug found on one phone model is likely to translate to all other models that share the same platform. Unfortunately, it is very difficult to actually analyze feature phones, because the many platforms are all closed source. Collin's solution was to look outside of the phone and perform his analysis using his own custom GSM network and fuzz-based testing.

Collin then covered the results of his SMS fuzz testing for a select set of phones. For most phones, the found bugs crashed the phone, causing it to disconnect from the network and reboot. Many of the bugs can be triggered without direct interaction by the user of the target phone: merely receiving the message will cause the crash. Interestingly, some of the bugs caused the phone to crash before it could send an acknowledgment to the provider. Collin suggested that this behavior could be used to amplify the attack's effect, because the provider will repeatedly retransmit the attack message. Collin went on to discuss a number of possible large-scale attacks, including targeting all of the customers of a specific provider or manufacturer. He noted that existing bulk SMS operators can provide the necessary SMS throughputs to make such attacks possible. Finally, Collin discussed a few possible countermeasures, including patching the firmware and filtering SMS messages. However, he pointed out that both techniques are poorly suited to addressing this problem.

Rik Farrow suggested that it might be possible for a specially crafted SMS attack to modify and effectively gain control of the phone. Collin remarked that they saw at least one bug that could possibly be used this way, but such attacks are infeasible; it is a tremendous amount of effort to exploit a single phone, and even then only that particular model would be affected. Dan Farmer wondered if there was a way to fingerprint phones to identify the specific model. Collin replied that there are some methods that rely on MMS implementations, but they found it was only possible with a small number of providers. Collin concluded the Q&A by showing a video demonstration of his attacks crashing a number of mobile phones.

Q: Exploit Hardening Made Easy

Edward J. Schwartz, Thanassis Avgerinos, and David Brumley, Carnegie Mellon University

Modern OS defenses are designed to make exploiting binaries more difficult. Edward Schwartz questioned the true effectiveness of these defenses. In his talk he focused on hardening exploits against two common defenses: data execution prevention (DEP) and address space layout randomization (ASLR). DEP prevents memory from being both writable and executable at the same time, thereby preventing an exploit's shellcode from executing. DEP can be bypassed by using return oriented programming (ROP), which utilizes instructions, or gadgets, that are already present in the target binary. ASLR seemingly makes ROP difficult to use by randomizing the location of those instructions; however, modern ASLR implementations actually leave small amounts of code unrandomized in memory. Edward's solution to evade OS defenses is called Q. Q searches the unrandomized program image to automatically build the gadgets needed for ROP, arrange gadget types such that they implement the desired computation, and assign compatible gadgets to the arrangement.

Edward then discussed how Q can automatically modify existing exploits to bypass DEP and ASLR. Q uses trace-based analysis to identify the execution path of the exploit. Using the resulting path constraints along with a set of exploit constraints, Q can automatically create a modified exploit that is unaffected by DEP and ASLR. Edward demonstrated this in a video showing Q hardening an exploit which was then successfully used on a machine with DEP and ASLR enabled. Edward went on to explain how Q was able to successfully harden a number of real exploits for both Windows and Linux. Further, he claimed that Q is able to create ROP payloads for most programs that are larger than 100 KB. He also discussed a number of limitations of Q. First, it currently only uses single path analysis, and this prevents Q from finding certain exploits. Second, Q's gadgets are not Turing-complete. Third, Q does not support conditional gadgets. Edward concluded by saying that even small amounts of unrandomized code makes DEP and ASLR completely ineffective.

John Grizzle (Illinois) asked if control flow integrity would prevent these types of attacks. Edward replied that it would. They chose to investigate DEP and ASLR because they know they are not perfect and they wanted to gauge how good they actually are. Dave Melski (GrammaTech) pointed out that a lot of vulnerabilities will place constraints on the type of inputs, e.g., no null bytes. He wondered how Q handled this. Edward responded that this is partially addressed by the path constraints; if a payload violated path constraints, Q would not find an exploit. For generating payloads, the user can specify the type of bytes that are allowed. Joe Werther (MIT) asked about the prevalence of non-ASLR images in modern operating systems. Edward responded that they did not have widespread statistics, but there is a report referenced in their paper which claims that many popular software packages have at least one module that is not marked as randomized. Finally, Karl Koscher (U. Washington) suggested that you could come up with a subset of gadgets to locate libc and subsequently use all of the gadgets provided by libc. Edward

agreed that this would be interesting and said there is another paper, "Surgically Returning to Randomized libc," which discusses locating libc, but for a different application.

Cloaking Malware with the Trusted Platform Module

Alan M. Dunn, Owen S. Hofmann, Brent Waters, and Emmett Witchel, The University of Texas at Austin

Trusted computing aims to provide a secure environment for computation. It attempts to accomplish this by creating a hardware root of trust, most commonly using a trusted platform module (TPM). Interestingly, Alan Dunn argues that the same security properties provided by a TPM can be used to provide a hardware cloak for malware. Alan said that malware can, for example, use TPMs to store secret keys, prevent monitoring by security analysts, and ensure that only unmodified malware is executed. More concretely, the TPM can be used with special processor instructions to provide secure execution via a non-analyzable late launch environment that is separate from system software on the platform. To do this, the malware writers must first make sure that sensitive computations are separated and encrypted such that they can only be decrypted by the TPM within the late launch environment. This is accomplished through use of TPM binding keys and remote attestation. When a remote malware distribution platform is satisfied that the conditions are met, it returns the encrypted payload for execution on the compromised host. Alan and his colleagues implemented three examples of malware using the TPM.

Alan described some possible defenses against TPM malware. The first defense is whitelisting late launch binaries. This defense is largely satisfying; however, it requires a hypervisor, which may be troublesome for home users to install. Additionally, it might be difficult to maintain the whitelist. The second defense is manufacturer cooperation, in which the manufacturer breaks TPM security guarantees to allow a security analyst to impersonate a legitimate TPM. The last defense is based on physical compromise of a TPM. However, the industry has incentives to fix existing physical attacks in order to maintain meaningful TPM security guarantees. Alan argues that strengthening TPMs against physical attacks actually makes TPM malware more resilient.

Bryan Parno (Microsoft Research) questioned whether an analyst would really have a problem analyzing this type of malware, given that they have sufficient resources to physically compromise the TPM. Alan questioned Bryan's assertion that the TPM's physical protections are only there to protect against low-capability attackers like common laptop thieves. Alan then suggested that there might be a range of adversaries between laptop thieves and the NSA. Another attendee expanded on Bryan's question by asking if the industry actually has any incentives to fix TPM's physical compromises. Alan conceded that this claim was closer to an opinion than a fact.

Invited Talk

The (Decentralized) SSL Observatory

Peter Eckersley, Senior Staff Technologist for the Electronic Frontier Foundation, and Jesse Burns, Founding Partner, iSEC Partners

Summarized by Italo Dacosta (idacosta@gatech.edu)

SSL/TLS is the most popular cryptographic system. It allows establishment of a secure communication channel between a client and a server by relying on X.509 certificates signed by a certificate authority (CA). SSL/TLS robustness is as good as its ability to authenticate the other party. However, as has been shown recently, there are several problems with the CA trust model. Certifying identities on the Internet is a hard job with odd incentives. CAs often make mistakes resulting in vulnerabilities, there is (circumstantial) evidence of governments compelling CAs to sign rogue certificates, and there are a great number of CAs, all equally trusted. In addition, the X.509 standard has a history of implementation vulnerabilities, and its extreme flexibility and generality have created a large number of disparate certificates.

The goal of the SSL Observatory is to investigate the problems associated with CAs, the types of certificates they are signing, and the size of the PKIX (public X.509) attack surface. In 2010, the SSL Observatory collected all available X.509 certificates on the Internet by scanning the IPv4 address space (3 billion IANA-allocated addresses) for port 443/TCP. They found 16.2 million IP addresses listening on port 443, 11.3 million SSL handshakes, and 4.3+ million valid certificate chains, with only 1.5+ million distinct certificates (leaves). The results are publicly available for anyone interested in analyzing them. The approach used with IPv4 will not work with IPv6, due to its larger address space, so new approaches will be required once IPv6 is fully deployed. The new version of this project is the Decentralized SSL Observatory, a browser extension that will allow the SSL Observatory to collect certificates from different network viewpoints. This approach is important because most attacks against SSL/TLS are only visible in the network path between the victim's client and the server (i.e., localized attacks).

Eckersley and Burns then explained their findings. First, the results confirmed that there are a lot of CAs on the Internet: 1,482 CAs trustable by Microsoft or Mozilla from 651 organizations. Second, CAs are located in approximately 52 countries, which means exposure to many jurisdictions. Third, several vulnerabilities were detected: around 30,000 servers were using broken keys or valid certificates with generic

names (e.g., localhost). Fourth, several problems associated with certificate revocation were found—for example, a large number of revoked certificates (~1.96 million revocations), the lack of revocation support (683 certificates without revocation information), and lack of a clear reason for revocation. Fifth, they found several configuration errors: violations of Extended Validation (EV) rules, CA certificates with keys from expired certificates, 512 and 1024 EV certificates, and certificates with huge list of names. They concluded that the attack surface includes not only CAs and target server but also the DNS infrastructure and anywhere in the network path between the client and the server.

Next, they discussed some proposed solutions to the SSL/TLS problems. (1) the consensus measurement approach (e.g., Perspectives and Convergence.io) attempts to get certificate information from different network vantage points to detect any anomaly; a disadvantage of this approach is the possibility of false positives. (2) More vigilant auditing, such as the SSL Observatory project, could be done. (3) The DNSSEC+DANE solution uses the existing relationship with the domain registrar to get the certificates for a Web site without requiring a CA, but this solution requires DNSSEC to be fully deployed. Also, DNSSEC+DANE defends against attacks to CAs but does not protect the rest of the attack surface. (4) Certificate pinning could be done via HTTPS headers: "whoever used to be domain.com should stay domain.com." This idea is simpler than DNSSEC and provides better security if implemented correctly (it protects the whole attack surface except for the first request). Eckersley described use of a private CA per domain in parallel to PKIX to cross-sign pinned certificates. However, X.509 certificates do not support cross-signatures. He suggested possibly using a second leaf certificate signed by the pinned "private CA" key or using an X.509 extension with a cross-signature.

A member of the audience commented on client certificates as another possible solution to the attacks against SSL/TLS and noted that federated login could help to deploy client certificates. The presenters responded that while the use of client certificates prevents the theft of authentication credentials, many other attacks are possible and additional solutions are still required. To the question of how pinned certificates are revoked, Eckersley commented that he has several ideas in mind, such as adding a timestamp, but more discussion is needed. Someone suggested having a hierarchical structure like DNS, where CAs are limited to sign certificates for particular domains. The speakers agreed and noted that an X.509 extension, name constraints, allows such functionality. The problem is that this extension is not widely supported, and it introduces some operational problems. Another person added that limiting the scope of CAs is also against their financial model and therefore will be difficult to implement. Adam Langley (Google) talked about the many problems with current certificate revocation mechanisms (e.g., performance and privacy issues) and asserted that new approaches are required. Stephen Kent (PKIX WG chair) commented that PKIX is not in itself bad—it is the way it is implemented in browsers. Eckersley noted the semantic problem caused by the increasing number of top-level domains. Finally, Burns mentioned that more transparency is needed regarding the sub-CA information of each root CA.

Dealing with Malware and Bots

Summarized by Lakshmanan Nataraj (lakshmanan_nataraj@umail.ucsb.edu)

Detecting Malware Domains at the Upper DNS Hierarchy

Manos Antonakakis, Damballa Inc. and Georgia Institute of Technology; Roberto Perdisci, University of Georgia; Wenke Lee, Georgia Institute of Technology; Nikolaos Vasiloglou II, Damballa Inc.; David Dagon, Georgia Institute of Technology

Manos Antonakakis said that Internet Protocol (IP) address-based blocking techniques can no longer keep up with the number of IP addresses the command and control (C&C) servers use. Also, there is a time delay between the day a malware is actually released in the wild and the day security researchers analyze that malware. Furthermore, the daily DNS lookup signal for malware-related domain names is different from that of normal Web sites. Hence, the authors propose a system, called Kopis, that statistically models the DNS lookup signal by utilizing the data in the upper part of the DNS hierarchy and builds an early warning system to detect malicious domain names. It leverages the fact that since DNS is a distributed hierarchical database, there must be a place in the DNS hierarchy that enables one to have global visibility from the point of view of who is looking up the domain names. Based on this observation, the system detects malicious domain names.

An interesting and important point to be noted here is that the system does not need a malware binary to detect malware domain names. The system can analyze large volumes of DNS messages at AuthNS or TLD servers. It also introduces an alternative IP reputation classification signal for DNS due to which botnets can be identified several weeks before the malware is actually found.

The basic building block of the system is an authoritative domain name tuple with two components: the resource record, which is a mapping from the domain name to its IP address, and the requester. Features such as requester diversity, requester profile, and resolved-IPs reputation are used. The requester diversity feature identifies whether the

machines that query a given domain name are localized or globally distributed. Using this feature, the authors show that malicious domain names are more widespread than benign domain names. The requester profile feature allows one to see whether a certain IP has historically had lookups for some specific malicious domain names. This enables one to know if a network has been well protected or not. Using these features, the authors perform a long-term evaluation where they show their system can reliably detect malicious domain names with a low false-positive rate. Manos concluded the talk with some case studies on some of the botnets their system had discovered. Kopis can be incorporated as an early warning system that can detect malicious domain names well before the malware reaches a network.

Marc Eisenbarth (HP) asked if their system would work from lower tiers in the DNS hierarchy. Manos answered that their system works as long as you have enough visibility.

BotMagnifier: Locating Spambots on the Internet

Gianluca Stringhini, University of California, Santa Barbara; Thorsten Holz, Ruhr-University Bochum; Brett Stone-Gross, Christopher Kruegel, and Giovanni Vigna, University of California, Santa Barbara

Gianluca began by noting that 85% of worldwide spam is through botnets and it is important to locate those spambots responsible for sending most of the spam. A simple approach to track spambots would be set up spam traps with fake email IDs and use the spam received in these fake IDs to track the bots. However, spam traps suffer from some limitations, since only a subset of spambots which are trapped using the spam trap can be detected. Also, the implementation of spam traps may not be easy, since some spambots operate only in certain countries and send spam only within those countries. Based on the premise that bots within a botnet share similarities, the authors propose a system called BotMagnifier, which observes a portion of a botnet and identifies more bots belonging to it.

The system builds on two inputs. The first is a set of IP addresses of known spam bots called seed pools. These are the ones that participate in a specific spam campaign (emails with similar subject) and are obtained by setting spam traps. The second is a log of both benign and malicious email transactions called a transactions log. This log was obtained from a Spamhaus mirror. The system is operated periodically where, at every instant, a set of seed pools (minimum of 1000 IPs) are supplied as input, and at the end of each observation period (typically a day), the IP addresses of bots in the magnified pool and the botnet name are generated as output. The system considers an IP address as behaving similarly to bots in a seed pool if three conditions are satisfied: that address has sent emails to at least a finite number of destinations in the target set, that it has never sent an email to a destination outside the target set, and that it has contacted at least one destination in the characterizing set.

They validated their approach by studying the Cutwail botnet, for which there was direct data available about the IP addresses of the infected machines. The C&C servers that were analyzed accounted for 30% of the botnet, and the validation experiment was run for 18 days. During this period, the spam campaigns were identified using the spam trap, and the seed and magnified pools were generated. Most of the original IP addresses were identified, indicating a good detection rate. Finally, the system ran for a period of four months, during which it tracked close to 2 million IP addresses. Of these, nearly half were from the magnified pools and the rest were seed pools. In an experiment where they use network logs to identify spam bots, the authors showed that their system is data-stream independent .

Vern Paxson (UCB) wondered about the rules in the paper that a bot should have sent a message to an IP address that is not in the seed pool. Paxson asked why that should be the case when a bot can send to a unique destination not in the seed pool. Gianluca answered that this was because the current system did not support that case; in the in future they would make it more general. Jelena Mirkovic (USC/ISI) asked if they had tried dropping that criterion, and Gianluca said that they had not.

Jackstraws: Picking Command and Control Connections from Bot Traffic

Gregoire Jacob, University of California, Santa Barbara; Ralf Hund, Ruhr-University Bochum; Christopher Kruegel, University of California, Santa Barbara; Thorsten Holz, Ruhr-University Bochum

Gregoire Jacob began the talk by presenting a system, called Jackstraws, that will identify command and control connections from bot traffic. Existing techniques for detecting botnets are either host-based (traditional malware detection, signature generation, behavioral monitoring) or network-based (IP blacklists of C&C severs). However, both these techniques are difficult to automate. This is because these techniques require clean C&C logs of system calls or traffic. But getting these logs is difficult, since the traffic could be encrypted. Furthermore, not all the traffic in a bot is associated with C&C activity.

In order to address these issues and identify C&C traffic in bot traffic, the authors propose a system called Jackstraws. The basic rationale behind the system is that C&C traffic results in observable activity at the host such as system modifications, critical information accesses, etc. Hence, the authors combine the network traces with the host-based activity, i.e., they use both a host-based model (system call graphs with data dependencies) and network-related links

(every graph associated with a network connection). Another observation is that similar commands will result in similar core activities even if the bots are different. These similarities can be learned using machine learning to identify and generalize C&C-related host activity. This is done using graph mining over known connections and then clustering these graphs to identify similar activities. These graphs are then merged into a template and template matching is carried out to detect C&C activity over unknown connections.

Gregoire then focused on a more detailed explanation of the above basic steps. The system was evaluated on a malware dataset of around 37,000 malware samples comprising over 700 families. After further processing, over 400 templates were generated. They tested these over labeled connections, for which they got a detection rate of close to 80% with a very low false-positive rate but a rather high false-negative rate. Gregoire mentioned that the high false negatives were due to some incomplete graphs. The system was then tested with over 66,000 unknown connections, out of which over 9,000 connections were identified. Among these, over 190 connections were new and not covered by any network signatures. Gregoire concluded by saying that they proposed an automatic system to separate C&C traffic from noise traffic. Their system, which is protocol agnostic, could give more information to analysts and also uncover new malware families that were not present in training.

Rik Farrow was curious why the system picks up families that were not included in the training set. Gregoire answered that, on the network side, the C&C may have a completely different protocol, but that is not the case on the host side. The botnets use the same system calls, in most cases. Also, new botnets are usually created by reusing parts of codes from old botnets. Hence, these behaviors can all be captured from a given template. Christian Kreibich (ICSI) asked how these malware samples were executed in a sandbox. Gregoire answered that the samples were executed using Anubis for four minutes to make sure that they were establishing the connections to the C&C server.

Panel

SSL/TLS Certificates: Threat or Menace?

Moderator: Eric Rescorla, Skype

Panelists: Adam Langley, Google; Brian Smith, Mozilla; Stephen Schultze, Princeton University; Steve Kent, BBN Technologies

Summarized by Nick Jones (najones@cs.princeton.edu)

Each panelist spoke briefly before taking questions from the audience. Schultze argued that there are many fundamental problems with the existing CA model, including that too many people are trusted, trust can be delegated almost infinitely, and accountability is difficult to achieve. He said it's a fundamental problem that users' perception of security is very different from the security they actually have. Additionally, the expansion of the existing Web architecture onto mobile devices exacerbates this problem, by having even fewer UI indicators and a much longer patch cycle.

Adam Langley from Google discussed the mindset of browser vendors who are considering the implementation of new features. At Google, when considering deployment of new features, they consider the possible security gain multiplied by the number of users affected. Specifically, in the case of CA controls for Android, Langley argued that only a small number of users would take advantage of the feature and that it didn't make sense to spend significant development resources implementing it. Langley then discussed several up-and-coming technologies for increased security, such as HTTP strict transport security (HSTS), blocking mixed scripting, and DNSSEC signed certificates. He said that features like strict transport security take priority over fixing the certificate model, because they pose a larger risk and are easier to fix.

Brian Smith from Mozilla's Firefox team pointed out that, from a browser vendor's perspective, there are several requirements which must be met before a new security feature can be deployed: new features must not confuse users, must be fast, and must not be prone to misconfiguration by server admins. He acknowledged that many security-enhancing features cannot meet these requirements, and he endorsed Firefox's extension architecture as a model for testing new security features. Smith discussed DANE, one technique Mozilla is considering implementing for increased certificate security. DANE is currently an IETF draft standard, which proposes using DNSSEC to associate certificates with domain names.

Steve Kent of BBN Technologies advocated the "Mao Zedong approach to PKI," arguing that the fundamental requirement of any CA is to establish and maintain an accurate binding of public key to identity attributes. Kent favors a model with lots of CAs, with a focus on organizational and proprietary CAs. In his model, a proprietary CA serves applications tied to the name space for which the CA is authoritative. Similarly, an organizational CA would serve entities associated with that organization.

During the Q&A, one person asked if the panel would be happy if they lived in an ideal world where all of the technical infrastructure problems were solved. Schultze said that even with the technical problems fully solved, there are still real-world security problems, such as typo squatting, which

have to be addressed. Kent said that solving the technical problems is a good first step, but not the entire solution.

Another person asked how DANE can be enforced outside the US. Kent responded that below the DNS root, there are lots of country TLDs. Thus, if a user goes to a URL containing a country TLD, then that TLD will be part of the DNSSEC hierarchy.

Nick Weaver (ICSI) asked about situations in which people choose not to run SSL. He wanted to know if there were any ways to enforce integrity over HTTP without encryption. Langley responded that it is technically possible to do so, but that industry doesn't think anyone would use it in practice.

Diana Smetters (Google) asked about building user interfaces that convey the right security message to users. Specifically, she asked how users should deal with expired and misconfigured certificates, and how normal users should understand what those warnings mean. Schultze responded that user desensitization comes from users seeing too many errors. He argued that browsers should just fail whenever they see a misconfigured cert, because that would force site administrators to be more proactive about fixing these errors. Langley responded that one of the attractive aspects of DNSSEC is its hierarchical delegation, which could reduce the number of errors users see.

One person asked about "trust agility," specifically regarding a Firefox plugin where users decide via consensus whether to trust a certificate. Langley responded that the consensus model places too much burden on users, and that normal users shouldn't be expected to think. Smith responded that users shouldn't have to choose which notaries they trust, because that can devolve into the same problem as choosing which CAs to trust.

Someone asked about browser warnings, and why the browser might not warn a user if Bank of America was using a certificate issued by a Romanian CA. Langley responded that no matter how big the warning, user design studies show that users will bypass them.

Privacy- and Freedom-Enhancing Technologies

Summarized by Ben Ransford (ransford@cs.umass.edu)

Telex: Anticensorship in the Network Infrastructure

Eric Wustrow and Scott Wolchok, The University of Michigan; Ian Goldberg, University of Waterloo; J. Alex Halderman, The University of Michigan

Eric Wustrow presented Telex, a system designed to circumvent blacklisting censors by steganographically hiding requests to prohibited sites in requests to permitted sites. Two observations drove the design of Telex: first, oppressive governments tend to favor IP address blacklists; second, those governments often do not control all intermediate routers. The authors propose inserting Telex stations at intermediate ISPs that are not under censorship. These stations inspect TLS handshake traffic looking for encrypted requests that Telex clients have placed in the TLS nonce field. Upon finding such a request, the station proxies it on the client's behalf and injects responses back into the return traffic.

To a censor shallowly inspecting the client's traffic, the client appears to be connecting only to the permitted site, the path to which contains the Telex station. To the prohibited site, the client's request appears to come from the Telex station. Each client needs Telex client software to run, which would pose a problem for online-only software distribution, but Wustrow optimistically described a system of out-of-band channels (e.g., USB flash drives) through which the software could be passed. Although Telex is not ready for general use—the only "permitted" site is currently a single server at Michigan—Wustrow reported that several of the paper's authors had been using the system full-time for months. He closed with several open questions about how to deploy Telex on the open Internet. Telex software is available at http://telex.cc.

Matthew Green (Johns Hopkins) asked whether the presence of Telex stations in a country provides incentive for censoring nations to attack it. Wustrow responded that the additional motivation provided by Telex was minimal and noted that the US has funded proxy services for oppressed users since 2003. Zack Weinberg asked how Telex would work under routing asymmetry, in which traffic follows one path to a destination and a different path back. Wustrow remarked that putting Telex stations sufficiently close to a desirable prohibited site could probably ensure that the station had access to traffic in both directions; he also suggested that multiple Telex stations on different paths could communicate out of band.

PIR-Tor: Scalable Anonymous Communication Using Private Information Retrieval

Prateek Mittal, University of Illinois at Urbana-Champaign; Femi Olumofin, University of Waterloo; Carmela Troncoso, K.U.Leuven/IBBT; Nikita Borisov, University of Illinois at Urbana-Champaign; Ian Goldberg, University of Waterloo

Prateek Mittal described PIR-Tor, a modification of Tor to improve its scalability. When Tor clients join the onion-routing network, they contact a Tor directory server and download a full list of thousands of potential relays through which the client can route traffic. From the full list, the client selects only three relays. Clients currently download

the full list in order to prevent malicious directory servers from directing Tor clients to chosen compromised relays. Mittal cited a study showing that directory-listing traffic will soon exceed data traffic on Tor. Noticing that clients use at most 18 middle and exit relays per three hours of Tor use, the authors developed PIR-Tor, a modification of Tor that uses private information retrieval (PIR) techniques to allow clients to fetch a subset of available relays without revealing to the directory server which ones were fetched. In PIR-Tor, a client chooses three candidate guard nodes (initial relays) from the list of directory servers, downloads a small amount of signed meta-information from each, and performs 18 PIR queries to choose its relays.

Crucially, none of these relays learns which relays the client attempted to select, so a malicious relay cannot simply give the client a list of servers with which it colludes. Mittal showed some plots to demonstrate that PIR-Tor exchanges one to two orders of magnitude less directory data with directory servers, arguing for its scalability over the current implementation of Tor.

Matthew Green asked whether PIR-Tor would ever become part of the Tor codebase, to which Mittal replied that PIR-Tor is open source. Roger Dingledine expressed a concern that the authors had not taken all the subtleties of Tor security into account; Mittal directed him to the paper for more information.

The Phantom Tollbooth: Privacy-Preserving Electronic Toll Collection in the Presence of Driver Collusion

Sarah Meiklejohn, Keaton Mowery, Stephen Checkoway, and Hovav Shacham, University of California, San Diego

Sarah Meiklejohn presented Milo, a protocol for privacy-preserving transit payments that, unlike previous systems, enables fine-grained road pricing without revealing to drivers the locations at which their presence is recorded. Meiklejohn gave an overview of previous approaches. In both vPriv and PrETP, presented at USENIX Security in 2009 and 2010, respectively, drivers upload logs of their driving activity to a central authority; the tolling authority performs audits to keep drivers honest. Meiklejohn pointed out a flaw in the previously proposed approach to auditing, wherein the tolling authority sends drivers photos of their cars in certain places and asks them to pay for having been there: drivers can learn the locations of traffic cameras and cheat en masse to avoid them.

Milo, the authors' modified version of PrETP, uses several cryptographic primitives to maintain driver privacy, driver honesty, and audit secrecy. A driver records location/time pairs and forms Pedersen commitments to segment prices. The driver sends the price commitments to the tolling authority along with zero-knowledge proofs of their correctness (as in PrETP) and additionally sends a blind identity-based encryption (IBE) of the commitment's opening. In an audit, the tolling authority's parent organization sends an IBE request to the driver, who responds without learning which segments were audited. Meiklejohn presented performance measurements from an implementation on both Intel and ARM architectures and showed that the time required to audit a driver's activity scales linearly with the number of segments driven.

Matthew Green asked whether audits could be made faster by performing encryptions on the car in advance; Meiklejohn said that it could. Diana Smetters asked how drivers could be given any confidence about their privacy; Meiklejohn acknowledged that that was a fundamental question perhaps more easily addressed in European nations, where citizens trust their governments to take privacy seriously.

Invited Talk

Pico: No More Passwords!

Frank Stajano, University of Cambridge

Summary by Ed Gould (summary@left.wing.org)

Frank Stajano presented a design for a system that would eliminate the need for and use of passwords in interactive authentication protocols. He began by reminding us that, in the past, passwords worked acceptably well. We had only one or two to remember, and 8-character passwords were beyond the scope of brute-force attacks. This is no longer true, and has not been for quite some time. See http://www.fastword.me for some related work.

Users have been told that passwords must have many properties (unguessable, un-brute-forcible, all different, memorizable and memorized), but the intersection of all these requirements yields the null set. Thus, passwords are both unusable and insecure—the worst of both worlds.

The goals of Pico, the author's design for a no-password system, include

- no more passwords, pass-phrases, PINs, etc.;
- scalable to thousands of verifiers;
- no less secure than passwords;
- increased usability (no searching or typing, continuous authentication);
- increased security (no guessing, phishing, eyelogging, etc.).

Two non-goals were mentioned as well:

- zero cost;
- backwards compatibility.

Pico includes a device that is somewhat like a smartphone (although it may be very much smaller), with a few buttons and a display. It has a radio communication facility as well as a camera. It can be shaped like a key fob, a watch, a MP3 player, or jewelry, for example. Importantly, it is a dedicated device, not something running on a multi-purpose device.

The authentication process using Pico involves the Pico device capturing a visual image from the verifier app to which one is authenticating, and a multi-step confirmation of identity, which is different the first time, when the user "pairs" with the app. The app is able to repeatedly communicate with Pico to ensure continued authentication.

Mechanisms for disabling the use of a Pico when it's not in the possession of its proper owner, as well as mechanisms to recover from loss or damage to the Pico, were described as well. Several possible ways to avoid being coerced into using one's Pico were described. A method using "Picosiblings" was described to enable the Pico to operate at all.

Frank pointed out that there are some passwords, e.g., file decryption keys, that do not fit the user-ID/password model, and thus are not addressed by Pico. He also mentioned that optimizing for backwards compatibility may be necessary to get to a critical mass, and quoted Roger Needham: "Optimization is the process of taking something that works and replacing it with something that almost works, but costs less." You can find the paper related to this talk at http://www.cl.cam.ac.uk/~fms27/.

Alan Sherman (UMBC) asked if Frank would comment on the resistance to man-in-the-middle (MITM) attacks. Frank explained that there is little leverage for a MITM attack after the initial pairing. If there were a MITM present during pairing, it might be able to fool Pico. However, it would have to be present for all future interactions as well, or Pico would notice its absence. The multi-channel protocol (camera and radio) makes it harder to do a MITM. If the visual part is hard to use, it is more vulnerable. Carson Gaspar pointed out that nested authentication will be critical for things like command-line administration, allowing for context-valid credentials.

Applied Cryptography

Summarized by Adam Bates (amb@cs.uoregon.edu)

Differential Privacy Under Fire

Andreas Haeberlen, Benjamin C. Pierce, and Arjun Narayan, University of Pennsylvania

Andreas Haeberlen presented work on eliminating covert channels in differential privacy systems. Andreas described several attacks on existing differential privacy implementations (PINQ & Airavat) and presented Fuzz, a new system that addresses these problems. Many companies would like to share their potentially useful data without violating the privacy of the subjects of that data. Anonymizing that data has been shown to be an insufficient defense in the face of adversaries with outside information. Differential privacy solves this problem through a querying mechanism that adds noise to results and the concept of privacy budgets, a method of limiting the number of answerable queries.

Although several attacks are detailed in the paper, Haeberlen focused on one timing attack for the purposes of the presentation. In a properly implemented system where the database is being remotely accessed, an attacker can only observe the query's response, completion time, and her remaining privacy budget. While existing systems secure the query response, it is possible for an adversary to design a query that leaks data through its completion time or privacy budget deduction.

Fuzz is both a programming language and a runtime environment that closes all three of these channels. It employs static program analysis to determine query cost without relying on the database as an input. Predictable transactions ensure that all microqueries take the same time to execute. The runtime environment isolates microqueries, preempts microqueries to execute timeouts, and returns a default value in the case that a microquery cannot complete. The overhead of these defenses is minimal aside from the padding that is imposed by predictable transactions. With ample knowledge of the database, this can be parameterized to reduce this effect. Fuzz is available at http://privacy.cis.upenn.edu/.

Ian Goldberg asked if the system would be susceptible to network-probe timing attacks. Haeberlen responded that the computer was fully busy during processing, and that the machine could be configured to not respond to probes. Ben Fuller drew attention to the overhead imposed by predictable transactions in a large enough database; Haeberlen agreed that the defense comes at a price. Another attendee pointed out that it is necessary to know the machine's hardware configuration to properly set the timeout, and he asked for a clarification regarding the early termination attack.

Outsourcing the Decryption of ABE Ciphertexts

Matthew Green and Susan Hohenberger, Johns Hopkins University; Brent Waters, University of Texas at Austin

Matthew Green presented this work on expediting the decryption of attribute-based encryption (ABE) ciphertexts through outsourced computation. ABE extends identity-based encryption by allowing data to be encrypted to a set of attributes. This is of use in data-sharing environments where

records can only be shared with certain groups of people. ABE requires the creation of a ciphertext policy that can grow complex based on the number of attributes. However, ABE's use on mobile devices is limited, due to the rapid growth in ciphertext size and decryption time as the size of the attribute policy increases.

Green presented new versions of Ciphertext-Policy and Key-Policy ABE that allow for outsourcing this decryption to an untrusted cloud service, avoiding the need to share a private key. These new versions introduce a transformation key that is sent to the cloud to perform partial decryption. The secret key is still required to recover the plaintext, so the cloud is not part of the trust model. The performance of this new ABE allows for practical use scenarios on devices with limited computational power. Often, decryption of ciphertexts on a more powerful machine remains an easy task. This new system was evaluated in the wild with an Amazon EC2 proxy. In one test with a complex attribute policy, decryption time was reduced from 17.3 seconds to less than 1.2 seconds. The partial decryption also reduces the size of the plaintext, reducing the cost of transmission. Green identified smart cards and trusted code base reduction as other possible applications of this new system.

Diana Smetters asked Green to elaborate on the key-sharing scheme in his model, pointing out that revocation is difficult. Green explained that every user received their own transform key and that the cloud proxy can act as a reference monitor. Bryan Parno asked if this scheme could be thought of as a regular proxy encryption scheme. Green replied that they are very similar and that both schemes are selectively secure.

Faster Secure Two-Party Computation Using Garbled Circuits

Yan Huang and David Evans, University of Virginia; Jonathan Katz, University of Maryland; Lior Malka, Intel

Yan Huang presented this work on an efficient garbled circuit used for two-party environments. Garbled circuits are a method of making privacy-performing computations; the circuit generator encodes the plain wire signal of 0's and 1's with data-independent nonces, encrypts a truth table, and sends it to the circuit evaluator, who can decrypt one and only one entry in the truth table. The circuit outputs a table of values, only one of which the circuit evaluator will be able to decrypt. Traditionally, garbled circuit execution is slow and scales poorly. Huang presents a new method of garbled circuit generation that is scalable and faster, as well as a library of pre-compiled circuits.

Fairplay, a popular system for secure function evaluation, is impractical for larger circuits, due to speed and memory constraints. This work demonstrates significant improvement through pipelining the circuit creation process—gates are evaluated as they are generated, dramatically improving memory and time efficiency without sacrificing security guarantees. The system is evaluated benchmarking the hamming distance, edit distance, and AES performance problems against previous implementations. Hamming distance experienced a speed-up of several orders of magnitude, and an AES s-box was implemented with a 30% improvement in the number of non-free gates.

Huang concluded that the pipelining technique, along with circuit-level optimization, allowed for garbled circuits to scale to large problem size. This framework and Android app demos are available at MightBeEvil.com. Ian Goldberg commented that he loved this work and hopes to see a trend of people realizing that garbled circuits can be efficiently implemented. He asked if this work can be applied to multi-party problems. Huang responded that much of what was learned in this work can be applied to the multi-party scenario. Diana Smetters inquired about the slow-down of Huang's circuits compared to a native run. Huang replied that it was still several orders of magnitude slower but that this could be a worthwhile cost in security-critical situations.

Invited Talk

The Cloud-y Future of Security Technologies

Adam O'Donnell, Co-founder & Director, Cloud Engineering Immunet

No report is available for this session.