conference reports

THANKS TO OUR SUMMARIZERS

16th USENIX Security Symposium

Boston, MA
August 6–10, 2007

KEYNOTE ADDRESS

- ***How the iPod Shuffled the World as We Know It***
  Steven Levy, Senior Editor and Columnist, Newsweek
  Summarized by Kevin Butler (butler@cse.psu.edu)

Steven Levy used the shuffle feature of the Apple iPod as a metaphor for the digital age. Newspapers used to be the medium for getting news in one physical bundle, but today people can "shuffle" between news sites. Similarly, thanks to the Internet, shoppers can shuffle between stores, no longer constrained by what is in a mall. Levy postulated that even Apple did not realize the transformative nature of this facet of the iPod when it was first introduced. Despite the naysayers who considered Apple to be out of its depth dealing with consumer electronics, and despite the inauspicious timing of the product announcement, just a few weeks after 9/11, the iPod has become extremely popular, particularly since Apple stopped the strategy of tying the device only to the Mac and opened it up to the Windows market.

Levy said the random shuffle feature of the iPod didn't seem particularly random, with different people reporting to him how their devices seemed to play favorites. Although some would criticize the iPod for creating a "nation of zombies," Levy pointed to similar criticism mounted at the Sony Walkman when it was created. Competitors to the iPod may have better features (e.g., Microsoft's Zune can "squirt" songs between devices through their wireless interfaces), but the models are often botched by DRM that hampers the user experience in unacceptable ways. For example, the Zune will only allow playing of a song squirted to it three times or for three days, whichever is less.

Levy talked about the iPhone, described by Steve Jobs as the best iPod ever. In contrast to the low-key launch of the iPod, the iPhone's launch was a spectacle, and Jobs has been quoted as expecting it to sell 10 million units by the end of 2008. The iPhone, however, will not be as significant as the iPod, reasoned Levy, as the iPod was a once-in-a-lifetime device that symbolizes the digital age and represented our connection to our music. No later device will give the visceral charge users got from their iPods.

Bill Cheswick pointed out that nonrandomness in random generators was well known at AT&T Labs, given Ken Thompson's generator. The predecessor to

the iPod was developed at DEC research, which was subsequently bought by HP, but HP co-branded iPods with Apple, and even Carly Fiorina probably never realized that HP had the patents for many of these digital players. Niels Provos asked about convergence and interoperability between devices, and Levy answered that companies are too invested in rights management, preventing seamless interoperation. Consumers want a revolution, but given how much money they pay to companies such as cable and telephone for service, there will likely not be any revolutionary changes until high-speed Internet connectivity brings streaming content directly to end users.

WWW SECURITY

Summarized by Patrick Traynor (traynor@cse.psu.edu)

- ***SIF: Enforcing Confidentiality and Integrity in Web Applications***

  Stephen Chong, K. Vikram, and Andrew C. Myers, Cornell University

According to a recently released report by Symantec, Web applications account for more than two-thirds of all Internet vulnerabilities. These vulnerabilities are largely caused by inappropriate information flows within applications. To address this problem, Stephen Chong and his co-authors proposed the construction of the Servlet Information Flow (SIF) framework. Built using the Java Information Flow (Jif) language, SIF ensures at compile time that Web applications provide and respect confidentiality and integrity constraints. More specifically, SIF ensures that expressive policies dictating the flow of information can be enforced before, during, and after it is handled by an application.

Although the compile-time checks of the Jif programming language provide a number of benefits, they are insufficient to support highly dynamic operations. For instance, a user may want to specify policy during runtime. To address this concern, the authors significantly extended the language such that applications can dynamically delegate authority to "principals." By allowing an application to dynamically constrain the privilege associated with each specific instance, SIF provides significantly increased assurance over previous application frameworks. To demonstrate the utility of SIF, the authors created two sample applications. The first, a cross-domain information sharing tool, uses a mail-like interface. Information is subjected to a mandatory review before crossing domains. The second application, an online calendar, employs dynamic policies to limit a user's view of events. As situations evolve, users can appropriately change their information flow policies.

A number of attendees inquired about the difficulty associated with programming in Jif. Others were curious about information flow policy false positives generated during the compilation process. Whereas the first is certainly nontrivial, Stephen told the audience that the second was largely not a problem. Finally, in terms of performance, SIF causes no noticeable degradation to the system; however, a full set of microbenchmarks was not created.

- ***Combating Click Fraud via Premium Clicks***

  Ari Juels, RSA Laboratories; Sid Stamm, Indiana University, Bloomington; Markus Jakobsson, Indiana University, Bloomington, and RavenWhite Inc.

Advertisers have long had problems determining whether publishers actually deliver content to a targeted audience or simply claim their fees without doing the agreed work. As advertisers move significant portions of their efforts to the Internet, such fraudulent behavior is increasingly affecting the industry. Specifically, click fraud, or dishonestly reporting the frequency with which consumers are viewing advertising, has become the best-known vehicle of advertisement embezzlement. Unchecked, such behavior could not only be used to dishonestly gain revenue but also to maliciously drain the advertising budget of rival companies.

Recognizing the magnitude of this problem, Ari Juels and his co-authors propose a change to the online advertisement revenue model. Instead of attempting to filter out "bad" clicks, this group of researchers proposes that "good" clicks be used as the basis of advertising reimbursement. Such a technique need not be the result of a change from the "pay-per-click" to a "pay-per-click-creating-a-sale" model. Clients instead receive tokens in response to performing beneficial operations, such as making purchases at an online store. As clients with tokens visit a Web site, their visit is billed at a higher rate because they are more likely to be a real customer (as opposed to a bot performing millions of clicks). Although such a solution certainly would work for individual domains, Ari discussed how cross-domain use of tokens and user privacy (whether the token contains a list of items purchased or a single "premium" bit) presents challenges for future work.

One of the participants asked about how tokens fundamentally differed from cookies, which can also be used to track user behavior across Web sites if set by third parties. Another member of the audience questioned the benefit to the consumer of willfully providing such tokens to every site they visit. Ari responded that in addition to reducing the impact of fraud on prices, customers might be offered discounts to provide such information.

- ***SpyProxy: Execution-based Detection of Malicious Web Content***

  Alexander Moshchuk, Tanya Bragin, Damien Deville, Steven D. Gribble, and Henry M. Levy, University of Washington

Malicious Web content is one of the most serious security threats facing systems. As the escalating arms race between adversaries and providers shrinks the window between exploit and patching, new solutions must be provided to protect commodity computing devices. In response, Alex Moshchuk and his co-authors have created SpyProxy, a fil-

tering mechanism sitting between the Web and a user's browser. SpyProxy uses a virtual machine (VM) to sandbox and pre-executes incoming traffic in order to determine whether or not it is malicious. Should such traffic be benign, it is allowed to pass to the user. Malicious code, however, is not allowed past the sandbox.

The determination of whether incoming traffic is malicious relies upon three simple but effective signals. Because normal Web traffic should not cause new processes to spawn, suspicious files to be created, and the registry to be modified, the presence of any such change is used as the indicator of maliciousness. To test their hypothesis, the authors gathered 100 sites known to host malicious content (browser exploits and spontaneous downloads) and ran them through SpyProxy. Whereas commercial services such as McAfee SiteAdvisor were only able to block 80% of these pages from loading, the SpyProxy approach prevented all 100 pages from infecting the client's browser. In spite of its success, Alex noted that there were a number of challenges facing SpyProxy. For instance, a naive implementation introduces significant delays to the rendering of benign pages in the user's browser. Accordingly, incremental rendering was implemented to make the presence of a virtual machine transparent. All 100 pages analyzed by this work also contained largely static content. However, a significant portion of dynamic content, such as advertisements, can be made deterministic to the system through the use of caching.

One attendee wanted to know how encrypted traffic over SSL and streaming content would be handled. Alex acknowledged that both issues presented challenges and would be investigated in future work. Other audience members were curious about the false-positive rate observed during the group's experiments. Alex noted that false positives were extremely low (4 out of 2000 known safe pages) and were caused by the download of browser plug-ins. Such problems could be minimized by installing the most popular plug-ins into the VMs.

INVITED TALK

■ *The Human Factor in Online Fraud*

Markus Jakobsson, Indiana University

Summarized by Adrienne Felt (felt@virginia.edu)

Markus Jakobsson spoke about the importance of considering both human and technical factors when designing for security. He said that the perfect technical solution is not enough and that people cannot be treated like machines. Realistic experiments need to be conducted. Public education is required to successfully combat online fraud, with an emphasis on understanding why certain things are dangerous; for example, people who know not to click on links in email may instead copy and paste the link into their browser, because they do not understand the reason for the warning.

Jakobsson cited improper configuration, neglect, and deceit as three major human factors that lead to tricked and compromised users. Improper configuration includes issues such as weak passwords on access points, and neglect occurs when trusted companies use bad practices. "Spear phishing" is a highly sophisticated form of deceit that combines traditional phishing methods with data mining. He called these "context-aware attacks," in which a phishing email could use the correct bank or mail service name for a given user. These powerful attacks are spurred by the increasing amount of information available in public databases and social networking sites. Alternately, they may take advantage of a technical security hole such as the use of CSS to access a browser's history.

He promoted user experiments as a way to gauge the effectiveness of fraud prevention techniques and predict future trends. He said that they should not be done in the lab, because users have artificially heightened awareness when they know they are being tested. As an example of commonly misunderstood user behavior, he explained that users tend to look for negative signs of danger but not be alarmed by the lack of positive signs of security. This renders commonly used positive reinforcement techniques, such as small lock logos, ineffectual in many cases. Nonlab experiments are difficult to do because of ethical concerns and the lack of debriefing, but Jakobsson asserted that they can be done.

To address the need for public education, Jakobsson said that users need to be taught a basic understanding of security principles. He suggested the avoidance of specific examples (which often cannot be sufficiently generalized) or bullet lists. He runs a Web site, www.securitycartoon.com, that provides security-themed cartoons that are intended to teach accessible security lessons. The site http://www.human-factor.org contains his annotated talk slides and links to specific studies he mentioned during his talk.

PRIVACY

Summarized by Kevin Butler (butler@cse.psu.edu)

■ *Language Identification of Encrypted VoIP Traffic: Alejandro y Roberto or Alice and Bob?*

Charles V. Wright, Lucas Ballard, Fabian Monrose, and Gerald M. Masson, Johns Hopkins University

Charles Wright presented his work on determining information about anonymous calls made through VoIP networks. Although anonymous VoIP calls encrypt the data associated with voice communication, it is possible to infer who is speaking. This "perfect storm" of factors includes the use of variable bit-rate (VBR) codecs and length-preserving stream ciphers; these are good for compressing information but allow information leakage because of the different-sized packets found with VBR, which carry information about the original waveform. Additionally, by look-

ing at a spectrogram, it is easy to see different energy levels associated with frequencies.

By looking at speech data from a corpus of 21 different languages, Wright and his co-authors found that each language had its own fingerprint that could be compared against. By using *n*-gram probability distributions and chi-squared tests, the group was able to determine the language a caller was using with up to 40% accuracy. Arabic, the most difficult language tested, was detected with 30% accuracy after three tests. With binary classification, determining whether the caller was speaking English vs. Farsi was 98% correct, whereas determining Czech vs. Hungarian (the most difficult binary classification) was about 86% correct. Wright discussed countermeasures, which involved the need to decouple packet sizes from contents. With extra padding, the accuracy of determining the language can be reduced at the cost of extra overhead, making this less beneficial for VBR schemes. For a quick fix, a user may as well use constant bit rate (CBR) schemes and use the padding for improved sound quality; this will mean that all packets are large, however. There is a tension between efficiency and privacy, and an ongoing research question is how much more information is really leaked.

Nikita Borosov (UIUC) asked whether use of multilayer classifiers that had individual strengths was possible. Wright said that it was and that even a decision-tree approach could be used. Perry Metzger asked how many speakers were in the original corpus and, when doing extraction, whether elements of the original data set were recognized. Wright replied that between 70 and 100 speakers were in the corpus and that cross-validation was done, leaving one user out for statistics generation and testing against that user for validation. It was noted that patterns exist in these languages. Another question was about non-native speakers; Wright conceded that they may throw off the distributions. Jeff Donnelley (LBNL) pointed out that the tradeoff of bandwidth versus privacy was discouraging, and he wondered whether there are other parameters such as jitter or latency that could be traded off. Wright said this hadn't been thought of but could be nice. Lucas Ballard (Johns Hopkins) mentioned that jitter will not affect accuracy of this method. Paul van Oorschot (Carleton) asked about dialects within a language. Wright wasn't sure how it would play out but said it would be fun to look at.

■ ***Devices That Tell on You: Privacy Trends in Consumer Ubiquitous Computing***

T. Scott Saponas, Jonathan Lester, Carl Hartung, Sameer Agarwal, and Tadayoshi Kohno, University of Washington

Scott Saponas said that privacy was important, but people are often unaware of how much information they are leaking . For example, most people have no idea that their employers can read their online information. Similarly, loyalty cards allow grocers to get information about what we buy, but that information is also being used for other things, such as in lawsuits and for divorce cases. New technology in particular has unknown privacy issues, but there is plenty of potential information to be mined from us.

Saponas and his group looked at the Nike + iPod sport kit, the Microsoft Zune, and the Slingbox to determine what information was being leaked. The Nike kit lets the user see their speed and whether workout goals are being met. GPS is not used and, according to Nike, movement cannot be tracked. However, each device has a unique identifier that is often left in the shoe. Saponas et al. wired sensors around their campus and built a Google Maps application that allowed them to determine the whereabouts of people with the device in their shoes. The Microsoft Zune also had privacy issues, since user Bob can "squirt" content to user Alice. Although Alice can block Bob, the blocking feature is only based on the MAC identifier, so after Bob is blocked, he can turn off his Zune, spoof his MAC identifier, and send more unwanted content. The majority of their investigation, however, was into the Slingbox, a streaming-media server that uses VBR encoding and transmits the difference between frames rather than the frame itself. Saponas et al. grabbed 26 movies and captured the encrypted packets they streamed from the Slingbox to gain a fingerprint of the movie. By querying a 10-minute window, they were able to identify a movie with 62% accuracy; looking at 40 minutes allowed identification of the content with 77% accuracy. All the streaming was done in a lab environment, so it is unclear how well it would work over longer distances or over the Internet, but this information could be a useful way of determining whether movies are being transferred online.

One questioner asked about the range that was possible with the Nike + iPod kit. Saponas answered that detection could be done from about 40 feet away, and the shoe broadcasts about every second. By contrast to an RFID, the shoe contains an active sensor. Paul van Oorschot (Carleton) asked about the first two attacks presented and weak versus strong identifiers—is this privacy versus denial of service? Saponas relied that this may be a tradeoff between privacy and performance, and sometimes privacy vs. user experience (e.g., the sensor in a shoe could change its identifier every time, but it becomes difficult to differentiate the user from the recipient, making tracking progress problematic). Marcus DeShaun asked why the ID device in the Nike shoe was so strong. Saponas answered that it is persistent and tied to the device from the time of manufacture. Paul van Oorschot asked about the confusion matrix presented in the talk. Saponas replied that this is useful for giving hints about correlation.

■ ***Web-Based Inference Detection***

Jessica Staddon and Philippe Golle, Palo Alto Research Center; Bryce Zimny, University of Waterloo

Philippe Golle discussed the inference problem, giving the example that medical databases combined with voter registration databases can expose lots of information about a person. For example, the military set up a Web site to

make a public archive, but it did not realize that documents existed on the site that, according to experts, showed previously undisclosed ways to make nuclear bombs. AOL published a series of half a million anonymous queries, but it was easy to correlate these by the anonymous user ID used. As a result, it was possible to determine the entire personal life of someone through these queries, and the *New York Times* was able to track an individual user down by looking at this information. Inference detection is a general method of identifying potentially sensitive knowledge that may be derived when new information is combined with reference information. The closest related work is manual expert review of information to redact, as is performed in industries such as healthcare and litigation; however, these processes are expensive and error-prone.

Golle suggested using the Web to proactively detect inferences. The Web provides an up-to-date corpus of public information. It can act as a proxy for reference knowledge and allows for efficient automatic inference detection. By combining inferences and extracting keywords from documents, queries may be issued to a search engine and results determined. For example, most pages that reference the terms *Saudi, magnate*, and *sibling* also contain *Bin Laden*, so if these are found in a document, with a high probability Osama bin Laden is the inferred subject of this document. The algorithm presented was data-intensive but simple and effective, with no formal representation or NLP required. The Web is imperfect for reference knowledge, as the less famous people are, the less information appears about them on the Web. Also, whereas inferences may be detected by co-occurrences, it does not explain some of them (e.g., Madonna is correlated with "gay": although she herself is not gay, she has a large gay audience). These techniques could be used to find the identity of anonymous bloggers and also to find sensitive inferences between keywords and topic. For example, the words "transmit" and "infected" often inferred "HIV"; "transmit" and "mucous" together were an inference for "herpes." A potential use of this scheme is for redaction of documents to be declassified. There are billions of pages of data that need to be redacted but the rules are extremely long and complex for what to release; this line of inference of sensitivity could be very valuable as a declassification mechanism. Should there be a privacy firewall for redacted information? Individuals are aware of privacy needs but are not good at protecting their privacy, and this trend will likely only continue as information gets mined more and more while sources of information leakage continue to accrue.

One questioner asked whether redacting certain information may lead to an observer seeing that this information is gone and wondering what is being hidden. Golle replied that this was a good question and an area of research. Another questioner said that this was just inference detection but not control. Corman Deley (Microsoft Research) pointed out that the three keywords for Osama were found because he is famous and the information about him is well known; what about regular people? Golle answered that some information is more easily inferable than other information. Only the top three results were looked at before the USENIX deadline, but more terms were also looked at and indices created for them, and this yielded valuable new information. Additionally, companies may create internal indices and use these for redaction to not let people know that they are being looked at. In response to the question of what to do about this inference and whether it would be better to drown the signal emitted with high-quality disinformation (a potential anti-Google), Golle said that this was one way to go and that some companies specialize in removing a user's Web presence.

INVITED TALK

Windows Vista Content Protection

Peter Gutmann, University of Auckland, New Zealand
Summarized by Stefan Kelm (stefan.kelm@secorvo.de)

After being introduced as a "self-proclaimed hippy," Peter began one of his jam-packed, highly entertaining, high-speed talks about the new content protection "features" as introduced by Windows Vista. It all started with a posting made by Peter in late 2006 to the crypto list, which soon ended up in discussions on slashdot as well. The posting was about "A Cost Analysis of Windows Vista Content Protection" and slightly stirred up organizations such as Microsoft.

To summarize the contents of both the paper and his talk: Windows Vista's content protection mechanism introduces an SSL-like end-to-end encryption of all content paths within the PC. The software is protected by OS mechanisms such that, in theory, if there's any break at all the content quality gets degraded by the system. In effect, it looks like an attempt to turn a general-purpose PC into a sealed audio/video jukebox.

Peter described at great length the many problems with this approach. He mainly did so by reading the available specifications and interpreting them into general language.

The main issue is that general functionality gets disabled by Vista once anything looks suspicious. Because this affects not only commercial HD content blocking but the user's own content as well, this could lead to "premium silence." A key issue is that of driver problems, especially when using signed drivers. Already, Microsoft had (for no apparent reason) revoked a driver signature, which effectively led to denial-of-service via driver revocation.

Peter mentioned a number of other odd things in the spec, such as the so-called tilt bits ("you have to be insane to actually implement this stuff"), which need to be set by any device that detects anything unusual, whatever that means. Of course, this all leads to increased hardware and software costs as well as additional CPU consumption, espe-

cially because of all the (128-bit AES) encryption going on. Laptops, for example, cannot enter power-saving mode when content protection is active. However, one might argue about Peter's claim that Windows Vista is thus causing global warming . . .

In his closing thoughts he tried to answer the question, "Why did they do it?" The intent, Peter said, was not to protect HD content. In fact, if there was any threat model at all it was pretty badly done. The content protection "features" were likely being added because of requirements driven by lawyers.

One attendee claimed that what Peter described might be regarded as generally good since Microsoft is "raising the bar." Peter disagreed, stating that, rather than raising the bar, the company is annoying consumers, encouraging them to buy cheap PCs in order to view their own content. Another attendee asked for advice on how to build his own home cinema. Peter's recommendation was to not go near a Windows PC, but buy one of those cheap Asian-made media players using component video instead of HDMI and the like. Finally, Peter replied to another question that this'll never stop commercial pirates, since they "can just walk around it."

AUTHENTICATION

- ***Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks***

  Saar Drimer and Steven J. Murdoch, Computer Laboratory, University of Cambridge

  Awarded Best Student Paper!

  Summarized by Patrick Traynor (traynor@cse.psu.edu)

The average American consumer is unfamiliar with "Chip & Pin" readers, but customers throughout the rest of the world recognize these portable credit card readers as the product of a concerted effort to thwart fraud. Instead of allowing a waiter or waitress to make arbitrary charges to a card, t smart cards combined with PIN technology strives to force customers to become active participants in all transactions. Although potentially promising, such systems suffer a number of significant vulnerabilities. Most critically, Chip & Pin systems provide no means for a customer to verify device authenticity or correctness.

Saar Drimer demonstrated this point in as direct a manner as possible: by converting a seemingly normal Chip & Pin system into a Tetris-playing handheld device. Having demonstrated the ease with which such a device could be subverted and reprogrammed, Saar discussed a more critical threat. Specifically, card and PIN information can potentially be transmitted to a remote location and used by an adversary. Current systems attempt to bound such interactions by setting an upper bound on transaction length, but the allotted time of such interactions is sufficiently large to allow remote use of a card. To prevent such an attack from happening, the authors presented the first implementation of distance bounding protocol. Using the Hancke-Kuhn protocol, the authors sought to bound the latency between challenge and response to within a reasonable physical distance between card and reader. Because signals cannot travel faster than the speed of light, such a protocol can drastically reduce the area in which an adversary could launch such an attack (from around the world to within a room).

A number of attendees wondered whether the fraud caused by such attacks could more easily be tracked if customers were simply to retain all of their receipts. Saar agreed that such an approach would certainly make recognizing such incidents easier, but that in practice it would be difficult to enforce such actions.

- ***Human-Seeded Attacks and Exploiting Hot-Spots in Graphical Passwords***

  Julie Thorpe and P.C. van Oorschot, Carleton University

The security of passwords seems to be constantly in peril. Because most users are unable to remember strings that are both long and pseudo-random, accounts are "protected" by short and/or dictionary-word passwords. In response to this problem, new schemes including graphical passwords have been created. Passwords are represented as sets of "points of interest" selected by the user. By entering the correct points in the correct order, a client can authenticate to the system.

Although such systems are in many ways more usable than traditional password schemes, previous research did not investigate the security of this new approach. Specifically, because humans are known to pick distinctively nonrandom passwords, it is highly likely that their selection of points in an image will be similarly nonrandom. The authors performed two tests. In the first, 43 university students were asked to create click-based graphical passwords for 17 different images. In the second, 223 users created passwords on one of two images. These experiments were used to determine whether an adversary-seeded set of points could be used to reduce the password dictionary size. When using all of the user points from the first set of tests, the password program was able to guess 20% and 36%, respectively, of the passwords for each of the two images used in the larger study. Moreover, a significant number of those passwords (11% and 8%) were recovered in only three attempts. Julie then discussed the use of techniques from computer vision that automatically recover points of interest and thus do not require intervention by the adversary. This approach, while able to recover some passwords, will likely require additional tuning to increase its success rate.

One attendee wondered whether preventing users from clicking obvious hotspots via filtering would increase security. Julie agreed that such an approach would be possible, but potentially at the cost of usability. Others discussed the

discrepancies between lab and field tests and wondered whether the incentive of the lab participants was sufficient to create good passwords. Audience members were interested in whether the people picking good text passwords were also those picking more robust click-based passwords.

■ ***Halting Password Puzzles: Hard-to-break Encryption from Human-memorable Keys***

Xavier Boyen, Voltage Security, Inc.

A number of techniques have been used to dampen attacks on passwords. For instance, attacks on passwords that are only accessible through an online interface are easily detectable and often subjected to "three strikes" lockdown policies. Offline attacks are slowed by forcing adversaries to perform an expensive cryptographic operation a large number of times. The difficulty with this approach, however, is that this workload is client invariant. For instance, a user logging into an account from a low-power device (cell phone) will have to perform the same expensive operations. Moreover, a fixed number of iterations of a protocol fails to take into account advances in technology. Whereas 1,000 HMAC operations may be expensive today, technology may advance sufficiently in the coming years to reduce this burden on the attacker.

In response, Xavier Boyen argued that the number of iterations used by the transformation algorithm should not be fixed. Instead, the number of iterations "t" should be selected by the user and become part of the secret itself. An adversary with the correct password would not know whether it was correct without generating the results of all possible iterations. To execute this protocol, the user simply enters his or her password when prompted and then presses the start button. When the counter reaches the user's secret value of "t," the user presses the start button again. The transformed value of the password is then checked to determine correctness.

Several attendees noted that such a scheme was similar to those that use two passwords and asked why "t" was not simply appended or prepended to the password itself. Xavier noted that this approach allowed users to gauge their own security needs and tailor performance to their expected needs. Another member of the audience asked whether the selection of "t" could be included in password strength analysis applications. Although such an approach seemed possible to Xavier, he noted that he did not consider such a use in his work.

INVITED TALK

■ ***How to Obtain and Assert Composable Security***

Ran Canetti, IBM Research

Summarized by Kevin Butler (butler@cse.psu.edu)

Bill Aiello introduced Ran Canetti to the audience as one of the co-authors of HMAC, a contributor to IETF efforts such as IKE v2, TLS, and other protocols, and a co-chair of the multicast and cryptography groups within the IETF, but whose work has also garnered fundamental cryptographic results. Canetti has been instrumental in helping to formalize definitions of privacy and confidentiality.

Canetti began with a description of the millionaires problem, where two millionaires want to figure out which one lost more money in a stock crash without telling each other how much they lost. Each has a private input, which is then securely evaluated. This is an example of a cryptographic task, where two or more parties want to perform some joint computation while guaranteeing "security" versus "adversarial behavior." This has many applications such as secure communication, secure storage, e-commerce, and database security. Basic cryptographic building blocks have been established, such as key exchange, contract signing and fair exchange, coin tossing, zero knowledge, commitment, oblivious transfer, and secret sharing. Many cryptographic protocols have been developed over the years for obtaining authenticated and secure communication, with both general solutions and more efficient constructions for specific problems.

What does "security" mean? Some of the concerns include correctness of local outputs, and distributional and unpredictability guarantees. For example, in a gambling scenario, one wants to ensure that nobody knows the output beforehand, tallies are performed correctly, input is independent, and there is unbiased randomness in the output. Other desirable traits include the secrecy of local data and inputs, privacy, fairness, accountability, and availability against denial of service attacks. Rigorously capturing an intuitive notion of security is tricky. Security can only hold against computationally bounded adversaries and only in the probabilistic sense. There are unexpected interdependencies between security requirements and computational assumptions made (e.g., the difficulty of finding discrete logarithms in a finite field), and situations such as secrecy depend on correctness while correctness may depend on secrecy. Canetti's goal was to describe a paradigm for formulating definitions of security in a way that guarantees it in any given environment.

Canetti introduced a simple, insecure protocol combination, where each protocol was secure when run alone but when the protocols were run together they became completely insecure, because they used joint secret information in an uncoordinated way. This is a problem with the Needham-Schroeder-Lowe protocol, as, after authentication, the key N is XORed with the message it protects.

Now an attacker in the middle between A and B can intercept the message C, if it knows something about parties A and B. Thus, a message C = N XOR M, where M is the text to be delivered. Now if an attacker in the middle knows the message is either "buy" or "sell," that attacker can capture C and use B as an oracle to recover the key (because if

B sells, the attacker finds that C'=(N XOR "sell") XOR "sell" = N) to determine whether it has the right key. The weakness only comes into play in conjunction with another protocol that gives the adversary two possible candidates for the key; consequently, there is a need to explicitly incorporate the encryption protocol in the analysis of the key exchange protocol. Canetti also gave examples of the insufficiency of stand-alone security by showing the malleability of commitment protocols and weaknesses in the original zero-knowledge protocols, demonstrated through an auction example. He reiterated the need for a general framework for representing security concerns and requirements from protocols.

Universally composable (UC) security is a framework that allows any set of concerns and requirements for any cryptographic task, such as authenticity, secrecy, and anonymity. The basic paradigm is that a protocol is secure if it can emulate the ideal operating case, where two parties hand their information to a trusted third party (a criterion expressed by Goldreich, Micali, and Wigderson). This is an intuitively attractive definition, but it is difficult to formalize. Three necessary steps are formalizing the process of protocol execution in the presence of an adversary, formalizing the ideal process for realizing functionality, and formalizing the notion of a protocol emulating ideal operation. For example, in the ideal case for the millionaires problem, parties A and B would send their input x and y to a trusted third party, who would compute $b = x > y$ and send the result to A and B. Each party is then assured that its own output is correct based on the other's input, the inputs are independent, and its own input is secret except for the function value. Examples of ideal key exchange and commitment functionality were also given. Having these in place, it is possible to start with a protocol R that uses ideal calls to functionality F and to have a protocol P that securely realizes F. The protocol R^P can then be composed with each call to F replaced by an invocation of P, and each value returned from P treated as if it came from F. This allows for security complex environments and for modular system design and analysis, as decomposition into small protocols is possible and each can be analyzed individually. The security of the system composed of these protocols can then be deduced. Already there are many known protocols for secure communication primitives that are UC-secure, such as IKE, TLS, and NSL, and some notions are equivalent to traditional ones (e.g., digital signatures and CCA-secure encryption). Multiparty computation is also UC-secure if there is an honest majority; the case is more complex if only a minority is honest.

One problem with UC security is that it applies only to instances that do not share any local state, but this occurs often in reality. UC security with joint state is an attempt to solve this problem. Canetti reflected on this work by mentioning that although components were separately analyzed and the analysis only dealt with a single session, it was still possible to assert security for the entire system, with guarantees holding within any external context for the application. He concluded by briefly discussing pros and cons of some formal methods for security analysis and presenting further research topics such as making security analysis ubiquitous, finding better protocols that guarantee UC security, and potentially applying composition to other scenarios such as program obfuscation.

POSTER SESSION

Summarized by Charles V. Wright (cvwright@jhu.edu)

The poster session Wednesday evening featured posters on a wide range of interesting topics in security, from cryptography and authentication protocols, to botnets and network security, voting systems, and the business aspects of successful security products.

Adrienne Felt from the University of Virginia presented "Defacing Facebook: A Web 2.0 Case Study." Her poster described the system used by the social networking Web site Facebook to allow users to write and run custom applications, and how a single cross-site scripting vulnerability opens up the entire site to attack.

Alok Tongaonkar and R.C. Sekar from Stony Brook presented "Fast Packet Classification Using Condition Factorization." This work focuses on finding efficient ways to match packets to rules, such as in a firewall or Berkeley packet filter implementation. Using a novel technique for building rules into automata, they can achieve polynomial size and near-optimal runtime performance.

David Botta, Rodrigo Werlinger, Andre Gagne Konstantin Beznosov, Lee Iverson, Brian Fisher, and Sidney Fels, from the University of British Columbia, presented "Towards Understanding IT Security Professionals and Their Tools." They are working on learning about and understanding the human aspect of network operations and security, and how it affects the ad hoc communication systems that evolve among admins. The Web site for this study is http://hotadmin.org/.

Andrew Blaich, Qi Liao, Aaron Striegel, and Douglas Thain, from the University of Notre Dame, presented "Lockdown: Distributed Policy Analysis and Enforcement within the Enterprise Network." This work uses a wide array of tools—agents, visualization, Linux security modules—to construct a comprehensive framework for creating, verifying, and enforcing security policies throughout a modern enterprise network.

Chaochang Chiu and Yu-Ching Tung, from Yuan Ze University, and Chi-I Hsu, from Kainan University, presented "The Study of Identifying Critical Success Factors in Predicting PKI Implementation Success: A Bayesian Classifier Approach." They studied several public key infrastructure (PKI) products funded by the Taiwanese government and evaluated several different factors to determine what makes a successful product.

Chris Nunnery, Brent ByungHoon Kang, and Vikram Sharma, from the University of North Carolina at Charlotte, and Julian Grizzard, from the Johns Hopkins University Applied Physics Laboratory, presented a poster on "Locating Zombie Nodes and Botmasters in Decentralized Peer-to-Peer Botnets." They monitored P2P-based botnets and found that zombies are characterized by searching behavior, whereas botmasters are often the ones providing files that the zombies search for.

Janne Lindqvist, from the Helsinki University of Technology and the International Computer Science Institute, presented "Privacy-Preserving WLAN Access Point Discovery." His work, now in the exploratory stage, focuses on finding a way to achieve both ease of use and security in wireless network protocols.

George I. Davida and Jeremy A. Hansen, from the University of Wisconsin—Milwaukee, presented "A Four-Component Framework for Designing and Analyzing Cryptographic Hash Algorithms." By studying each of the constituent pieces of a typical hash function in isolation, they hope to build a component framework so that if one component (for example, the compression function) is found to be insecure or too slow, it can simply be removed and another plugged in to replace it.

Manigandan Radhakrishnan and Jon A. Solworth, from the University of Illinois at Chicago, presented a poster on "KernelSecNet," a framework for building secure applications. In this system, security-related functions such as authentication, authorization, audit, cryptography, and process creation are moved out of application programs and into an application proxy or even into the OS itself, to minimize the risk of vulnerabilities in application code.

Micha Moffie and David Kaeli, from Northeastern University, and Winnie Cheng, from the Massachusetts Institute of Technology, presented "TRACKS: A Behavior Based Policy System to Detect Hidden Malware." TRACKS is an intrusion detection system that analyzes program events to detect malware and enforce security policy. It learns malware behaviors from real trojan programs, and it detects attacks using binary analysis similar to taint detection.

Peter Williams, Radu Sion, and Erez Zadok, from Stony Brook, presented "NS3: Networked Secure Searchable Storage with Privacy and Correctness." The goal of this work is to develop techniques such as personal information retrieval, providing not only confidentiality for the stored data, but also privacy of the user's searches and query correctness.

Robert Beverly and Steven Bauer, from MIT, presented "Tracefilter: A Tool for Locating Network Source Address Validation Filters." They are conducting an ongoing measurement study on the Internet to determine where network administrators place firewall rules to filter out invalid source IPs. Usually, most filtering occurs at either the first or the second outbound hop.

Ryan Gardner, Sujata Garera, and Aviel D. Rubin, from Johns Hopkins, presented "Dynamically Establishing Trust: Can It Be Done Reliably?" Their poster deals with the difficult problem of using software on an electronic voting machine to authenticate itself to a human operator.

Thomas J. Holt, Lyudmila Leslie, Joshua Soles, and Brittany Spaulding, from the University of North Carolina at Charlotte, presented "Exploring the Social and Technical Aspects of Political Conflict On-Line." This work details the cyber attacks on Estonia, launched earlier this year apparently from sources in Russia. The poster gives a timeline for the attacks and Estonia's response, with analysis of the major events of the conflict.

Zhiyao Liang and Rakesh M. Verma, from the University of Houston, presented "Authentication Considering a Dishonest Insider." In this work, they seek to extend classical work on authentication protocols such as Needham-Schroder, to allow for a more malicious insider threat.

THREATS

Summarized by Charles V. Wright (cvwright@jhu.edu)

- ***Spamscatter: Characterizing Internet Scam Hosting Infrastructure***

  David S. Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M. Voelker, University of California, San Diego

Chris Fleizach presented a study of the Web server infrastructure used by spammers and other scammers to host their malicious Web sites. In this work, the authors sought to determine (i) how scams are distributed across servers, (ii) whether multiple scams might share servers, (iii) how long scam sites stay up, and (iv) where (geographically) these sites tend to be hosted.

First, the authors had to develop a method for determining to which particular scam each spam mail or fraudulent Web site belongs. For this, the authors developed a novel "image shingling" algorithm, whereby a screenshot of the scam page is taken, then broken up into several smaller blocks. The blocks, or "shingles," are then hashed, and the similarity of two Web pages can be computed as the fraction of shingles they share. This technique cleverly sidesteps the noisy data and ambiguity in the scam URLs, spam emails, and even the HTML content of the scam pages themselves.

The authors found that although scammers use many machines to send out spam messages, they typically use a much smaller number of Web servers to host the scam page itself. For example, one common scam used only three Web servers (one in Russia and two in China). Overall, it is more typical for the scam site to be hosted in the U.S. (over 60%), perhaps to be closer to the victims, to facilitate payment processing, or to cultivate the appearance of legitimacy. Spam relays, in contrast, are much more widely distributed around the world, with only 15% lo-

cated in the U.S. Most of the scams were not distributed across multiple servers, but a small number of sites were observed to use more than 20, and 40% of the scams were hosted on a server that they shared with at least one other scam.

A lively Q&A session followed the talk. Two early questions dealt with discrepancies between trends in the UCSD group's data and what audience members have observed in their own networks. Chris explained that spam changes constantly and can vary considerably between networks; for example, rapidly changing DNS names might be used by one group of spammers and not by another, and spam collected in other countries might differ considerably from that targeted at the U.S. Other questions concerned the nature of the spammers and scammers themselves: whether the UCSD group has made any effort to contact them personally; where the money from the scams is going; and whether the scammers could really be making much money from short-lived scams. Chris replied that, although they have not examined this aspect of spamming and scams yet, they plan to delve deeper and learn more about this "underground economy" in future work.

■ ***Exploiting Network Structure for Proactive Spam Mitigation***

Shobha Venkataraman, Carnegie Mellon University; Subhabrata Sen, Oliver Spatscheck, and Patrick Haffner, AT&T Research; Dawn Song, Carnegie Mellon University

Shoba Venkataraman presented a measurement study of the IP addresses that send mostly spam and those that send mostly legitimate mail. Based on this study, she proposed a new solution to help overloaded mail servers prioritize legitimate messages. Because existing spam-filtering techniques use content-based analysis, the filtering happens fairly late (after the mail has been accepted for delivery) and can be computationally expensive. In contrast, by filtering based on the sender's IP address, the new scheme is computationally efficient and can weed out most spam before it can even be transmitted.

To show that such a technique would be effective at stopping spam with only a minimal impact on nonspam mail, the authors examined traffic logs from a busy mail server. They calculated the "spam ratio" (i.e., the fraction of all mail from an IP that is flagged as spam) for each IP address that sent mail to its server and found that most IPs had either very low or very high ratios. Moreover, IP addresses that are present in the logs on most days tend to send a lot of legitimate mail and very little spam. Most of the spam comes from transient IP addresses, which show up only rarely in the logs, but the authors found that these spammer IPs tend to cluster together in blocks of address space and that the spam-sending blocks tend to be long-lived. (In fact, 90% of them persist for longer than 60 days.) Therefore, when the receiving mail server is very busy, it can prioritize connections from IP addresses from which it has recently received mostly nonspam mail, and it can drop connections from IPs in the recent spam blocks.

There were several thoughtful questions after Shoba's talk. One audience member asked whether spam really does tend to overload mail servers, preventing the delivery of nonspam mail (a behavior he had personally never observed); another asked about the potential effects of deploying the proposed system in different places around the world. Shoba replied that spam behavior can vary from network to network, so the blocks of IP addresses flagged as spammers might be specific to the location of the defender's network. Likewise, the intensity of spam varies, and it does in fact sometimes prevent delivery of legitimate mail on the network where the authors collected their data. There were other questions on whether the clusters of "bad" IPs identified here are correlated with hijacked address space, and how the results of the current study compare to earlier work by Feamster et al. Shoba explained that although the earlier work did not consider legitimate mail and that for the current study they didn't consider hijacked address space, the new results agree with Feamster et al.'s findings about the sources of spam traffic.

■ ***BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation***

Guofei Gu, Georgia Institute of Technology; Phillip Porras, Vinod Yegneswaran, and Martin Fong, SRI International; Wenke Lee, Georgia Institute of Technology

Guofei Gu presented a new approach for detecting botnet infections in a local area network. The system, called BotHunter and available for download on the Web at http://www.cyber-ta.org/BotHunter/, examines two-way communication patterns between hosts in the local network and those external to it, and it detects successful bot intrusions by correlating intrusion alarms on inbound traffic with scanning patterns in subsequent outbound connections. The authors defined a five-step model of botnet propagation: inbound scanning, exploit, download of bot code, communication with the botnet command & control, and outbound scanning. They use this model in BotHunter to improve the accuracy of the network intrusion detection system over what is possible when looking for individual bot-like behaviors in isolation. For example, Guofei explained that inbound infection attempts are not enough evidence of an infection to warrant raising an alarm, because zombie machines on the Internet are continuously scanning for victims, and most of their attempts are unsuccessful. However, when the IDS also sees scanning or other bot-like behavior from machines in the local network, it's much more likely that there really is a problem.

BotHunter is built on the popular open-source Snort intrusion detection system. In addition to using many of the standard Snort filters, the authors also developed two new extension modules: the Statistical sCan Anomaly Detection Engine (SCADE) for detecting inbound and outbound scanning and the Statistical payLoad Anomaly Detection Engine (SLADE), which performs *n*-gram analysis on packet payloads to detect malware. SLADE compares fa-

vorably to the earlier network intrusion detection system PAYL, offering more fine-grained *n*-gram pattern matching, better runtime performance, and improved detection rates. The authors evaluated BotHunter's detection ability in a virtual network in a lab, in a honeynet, and finally in live network settings at Georgia Tech and SRI, finding that it produced good detection rates and a low number of false positives.

The first of several questions dealt with detecting bots that do no inbound scanning, propagating instead via trojan horse, email, etc. Guofei explained that BotHunter doesn't always have to observe all five steps of infection in order to raise an alarm; the model could even be tailored to different types of spreading behavior. Another question concerned whether an attacker might try to avoid detection by going very slowly, so that BotHunter might fail to correlate an inbound scan with an outbound scan happening much later. Guofei replied that, yes, in fact all schemes based on time windowing are vulnerable to such attacks, but the defender could randomize the window size to make the attack more difficult.

INVITED TALK

Exploiting Online Games

Gary McGraw, Cigital

Summarized by T. Scott Saponas (ssaponas@cs.washington.edu)

Gary McGraw began by suggesting that among the eight million users of online multiplayer games there are bound to be many dishonest or malicious users. In this talk, McGraw focused on how and why some of these users cheat in online games. He said he chose online games because they are a bellwether of things to come.

Online games include the trinity of trouble: connectivity, complexity, and extensibility. Originally, these games only had an online component to help prevent piracy of games; however, connecting to many other users online is now an essential part of many games. Blizzard's World of Warcraft (WoW), for example, has 500,000 simultaneous users on six continents and represents a large distributed system with a fat client pushing the limits of computing systems.

One of the major reasons for cheating is that there is big money to be made. Some users do not have the time or desire to earn achievements in the game, such as collecting gold (virtual wealth) or attaining a certain level. As a result, there is a world market for virtual gold where people will pay real money through online auctions and intermediaries. McGraw cited that in some places, such as China, one can make more money playing video games and selling virtual wealth than through traditional jobs (http://youtube.com/watch?v=ho5Yxe6UVv4).

Many people cheat at WoW by manipulating game state stored locally in memory. They look through memory and find where information such as their virtual location or how much virtual gold they have is located and change this information. Modifying such state allows cheats such as teleporting and duplicating gold. These hacks get distributed on the Internet as unauthorized add-ons to the game.

Blizzard's first approach to thwarting this problem was to look all over one's computer for files related to cheating hacks. This Big Brother approach results in users getting banned when they have downloaded a cheat but not installed or used it. Blizzard also added the Warden software to WoW to watch for interference with the memory or execution of the game.

McGraw described how recently these defense mechanisms have been defeated with more sophisticated techniques attacking from the kernel or video card. From the kernel, it is possible to manipulate memory without the Warden being able to detect changes. From the video card, there have been hacks to change how the virtual 3D world is rendered to give users advantages over their enemies. For example, it is possible not to render walls or to render enemies as large orange objects.

McGraw concluded by suggesting that online games require a new model for software design and security in which distinguishing insiders from outsiders is less clear. For more information, see the book Gary McGraw has written with Greg Hoglund on this topic, *Exploiting Online Games*.

ANALYSIS

Summarized by Stefan Kelm (stefan.kelm@secorvo.de)

Integrity Checking in Cryptographic File Systems with Constant Trusted Storage

Alina Oprea and Michael K. Reiter, Carnegie Mellon University

Alina Oprea's talk on integrity protection in encrypting file systems was motivated by the many companies that nowadays outsource their data. There already are a number of different protocols and services available. However, do the users actually trust the remote servers?

Alina proposed an end-to-end security architecture in which the overall security is only maintained by the clients through the usage of so-called trusted key storage. In her work she has been looking at integrity in cryptographic file systems (CFS). These systems usually divide files into fixed-lengths blocks, with each block encrypted individually. The integrity of these file blocks needs to be protected by using a constant amount of trusted storage per file, usually at a size of only several hundred bytes.

She proposed two new integrity algorithms, RAND-EINT and COMP-EINT, which utilize two properties of many common file systems: (1) a low entropy of files, and (2) the sequentiality of file writes. After briefly discussing Merkle trees, Alina introduced the two new algorithms.

The entropy integrity algorithm (RAND-EINT) does not initially protect against replay attacks, thus counters are being introduced. The compression integrity algorithm (COMP-EINT), however, is based on Merkle trees but is much improved compared to previous approaches (including one by the author herself).

A main part of her work was to actually implement the algorithms in EncFS (which does not provide for integrity) and evaluate the algorithms in terms of performance. To do this, she described four metrics that were being used to test the performance of both algorithms and suggested which algorithm would be best for low-entropy files vs. high-entropy files, as well as read-only access vs. other access. Alina recommended implementing both new algorithms and letting the corresponding application choose which one to use based on its typical workload.

One question asked was whether the solutions proposed may be used for pipelining in network file systems. You can't do that for integrity, Alina replied, except for HMAC algorithms.

For more information see http://www.cs.cmu.edu/~alina/.

- ***Discoverer: Automatic Protocol Reverse Engineering from Network Traces***

  Weidong Cui, Microsoft Research; Jayanthkumar Kannan, University of California, Berkeley; Helen J. Wang, Microsoft Research

In this talk Weidong Cui described the development of a new tool called Discoverer. The motivation for this work was to automate the process of reverse engineering network protocols, a process that today is performed manually. The goal is to improve security techniques such as IDS/IPS, penetration testing, and generally identifying protocols. The challenges of automatically reverse-engineering these protocols are that they greatly differ from each other and that many message formats are fairly complex, consisting of text and binary fields.

Weidung described the key design goals of their tool as protocol-independence (since they did not want application-specific customization), correctness, and, primarily, automation such that no manual intervention is necessary when examining network flows. The basic idea is to infer protocol idioms, two of which he then described: the message format, usually considered as a sequence of fields, and their semantics (lengths, offsets, cookies, etc.). Moreover, most protocols use what Weidung calls format distinguisher (FD) fields to differentiate the format of subsequent message parts.

Discoverer's architecture can be divided into four phases: (1) during the tokenization phase, field boundaries are being identified; (2) during initial clustering, each message gets dissected into different tokens belonging to either the binary token class or the text token class; (3) during the recursive clustering phase, the initial clusters are being further divided, because initial clustering may be too coarse-grained; and (4) the merging phase merges clusters of the same format by using type-based (instead of byte-based) sequence alignment.

He then went on to describe the prototype they've implemented and the evaluation conducted. The following metrics were of importance during the evaluation: correctness, conciseness, and coverage. Two binary protocols (CIFS/SMB and RPC), as well as one text protocol (HTTP), were evaluated with Discoverer, with network flows having been collected in an enterprise network. Weidung concluded that more than 90% of their inferred messages actually correspond to "real" network flows (as obtained from Ethereal).

One questioner asked about the actual goal of this work. Weidung reiterated that this is all about automatically reverse-engineering network flows. On how resistant this scheme would be against an adversary garbling the messages, Weidung replied that you can't do anything at all about encryption or obfuscation of messages.

For more information, see http://research.microsoft.com/~helenw/pub.html.

- ***Towards Automatic Discovery of Deviations in Binary Implementations with Applications to Error Detection and Fingerprint Generation***

  David Brumley, Juan Caballero, Zhenkai Liang, James Newsome, and Dawn Song, Carnegie Mellon University

  Awarded Best Paper!

The final talk of this session was on detecting differences in how multiple implementations handle the same protocol. Zhenkai Liang began by explaining that many different implementations usually exist for the same protocol but that two implementations often do not interpret the same input alike, often as a result of implementation errors or because the implementer chose to implement only a subset of the protocol.

In their work, the authors propose a new approach to detecting those deviations for error detection or fingerprint generation. This approach is based on behavior-related deviations instead of checking minor output details as provided by any application. The main problem they face is the question, "How do I find input in order to demonstrate that two implementations behave differently?"

The key concept of their work is use of symbolic formulas. Their approach is an iterative one, consisting of three phases: (1) during the formula extraction phase, x86 instructions of a particular implementation are first transformed into an intermediate language, which in turn is transformed into symbolic formulas; (2) the deviation detection phase constructs queries from those formulas which are sent to application servers using different inputs; (3) finally, the validation phase checks whether or not two different implementations really reach different protocol states.

Zhenkai then discussed their prototype implementation. They evaluated their approach on NTP and HTTP, testing Ntpd, NetTime, Apache, Miniweb, and Savant. He concluded that their prototype already is pretty fast in detecting deviations, but that they need to improve the prototype in order to be able to explore different program paths that might lead to the very same protocol state.

One attendee wanted to know how this approach would be superior to static analysis of applications. Zhenkai answered that, with static analysis, one might not know where the program will jump to.

For more information see http://www.andrew.cmu.edu/user/liangzk/.

INVITED TALK

Computer Security in a Large Enterprise

Jerry Brady, Morgan Stanley

Summarized by T. Scott Saponas (ssaponas@cs.washington.edu)

Jerry Brady's invited talk discussed the present challenges for managing computer security at a multinational financial organization. He began by explaining that the security priorities of a large organization vary by location. In some countries, a company is concerned about employee safety or physical theft. However, in the United States, his company is primarily concerned about electronic security.

Electronic security is made challenging by a diversity of technology platforms and processes, as well as global operational requirements such as "no off hours" and limited patch windows. Brady said that upgrading a firewall with an important patch can take as long as a year because of the planning required to identify what will be impacted by the change and the downtime.

Brady identified risk management as the main complicating factor in computer security. Ultimately, risk management has to be the main priority of an organization such as his. However, it is often at odds with security. For example, when there is a complex interdependency of applications and platforms, sometimes the least risky option is not to make an update or change to the system.

Security is a careful balancing act of risk, cost, regulatory obligations, priorities, and risk tolerance. Different business units within a firm have different needs and expect several options. As a result, security must be tiered and responsibility has to be distributed. One challenge of distributed responsibility is that it is not always clear who accepts the risk: the application, the business unit, or the entire company's brand.

Five years ago, exchanges among banks or law enforcement agencies about security incidents or intelligence about computer security threats did not happen or were not useful. Brady said that now when fraud or attacks happen everyone shares knowledge with everyone else so that everyone can fight these attacks.

Brady concluded his talk by discussing the challenge of finding and training good security staff. He said that the security field needs academia's support. The IT culture must be changed. He said we need to weave risk and security into the technology curriculum.

PANEL

Cellular Network Security

Panelists: Ron Buskey, Motorola; John Larson, Sprint Labs; Simon Mizikovsky, Alcatel-Lucent, Wireless Emergency Response Team; Hao Chen, University of California, Davis; Thomas La Porta, The Pennsylvania State University, Bell Labs Fellow; Patrick Traynor, The Pennsylvania State University

Summarized by William Enck (enck@cse.psu.edu)

The recent surge of academic interest in cellular network security has caught the attention of academic and industrial researchers alike. USENIX Security organizers put together a panel to bring together industry and academia to expose the community to the concerns and issues as seen by both sides. Radu Sion introduced the panelists and prompted each to give a short presentation.

Ron Buskey began addressing questions he was sent by the panel committee. Will telecom networks independent of the Internet continue to exist in 25 years? Will providers (wired or wireless) move their services to the general "public infrastructure" of the Internet, or will networks dedicated to voice traffic and its specific requirements remain? Ron said that technology has changed so fast that it is hard to foresee the future. However, the trend has been to "bring the Internet to the user." This means there will be more equipment closer to users, and we must be very careful of the security of these devices. With migration toward an IMS (IP Multimedia Subsystem) core, how can members of the security community become more active in the security of telecommunications networks? Ron believes a major issue will be the widespread adoption of SIP-based services. Specifically, the ability to authenticate and integrity must be clean and easy to manage. Finally, billing has always been, and will remain, important.

Patrick Traynor looked to the audience and boldly claimed that *they* are the next big threat to telecommunications security. The design of future systems will play an important role. Current designs rely heavily on traditional abstraction. For example, devices simply assume the network is there. However, in the cellular network, each action requires a significant amount of work, and a device not sensitive to the network reaction will unintentionally (or intentionally) cause disruption.

Patrick divided future problematic software into three classes: ported, buggy, and malicious. Ported software assumes the cellular network acts as a packet-switched net-

work. However, applications that have long periods of silence (e.g., Skype, IM, and SSH) require costly setup messages for every transaction. Buggy software will, for example, contain timing problems, and refresh connections too late, resulting in many superfluous connection setups. Finally, malicious software will act like buggy software, but will do so on purpose. Patrick concluded by stating that there are over two billion cellular users, whereas the Internet is estimated at only one billion. More sophisticated devices are beginning to penetrate the cellular user base, and systems design must proceed with care, or else there will be serious impacts on the core of the network.

John Larson focused his presentation on WiMAX security and 4G networks. There are many well-known wireless threats, including malware, intrusion, eavesdropping, DoS attacks, and rogue base stations. There are also less-known attacks such as protocol fuzzing that result in DoS or intrusion. Future networks must be sensitive to all of these issues. He would like to see a comprehensive security architecture, for which work is currently in progress. The architecture needs encryption at various layers; however, unnecessary redundancy is bad, as it will hurt performance. Devices need to be made more secure, and the network must have the ability to quarantine malicious devices. John also believes DoS is a crucial issue that is not really dealt with. His lab was able to knock over all the equipment it has tested. They need help from the research community to fix this problem in protocols and applications.

Thomas La Porta presented an overview of the evolution of 2G to 3G. Traditionally, the telecommunications networks were closed. Who knows how SS7 (an out-of-band telephone signaling protocol) works? It is hard to do damage to something if you don't know how it functions. However, this situation is changing. The first step in opening up the cellular network has been to convert the core to IP. The purpose is to attach services that are not directly attached to the Internet. This will result in many new avenues into the network. As the network evolves, a common IP core will be connected to an ANSI-41 core (older cellular network), a UMTS core (3G cellular network), and IP access. The network accessed by a client does not necessarily "own" that client, and it must rely on the owning network for authentication. Hence, a network is only as strong as the weakest network it is connected to!

Thomas foresees both single infrastructure and cross-infrastructure attacks that will result from modifying message data, modifying services logic, and denial of service. We have already seen with the Greece incident that such attacks can compromise provider equipment. Again, a network is only as strong as its weakest connected network. Problems will get worse, and we need to be proactive about finding problems and fixing them.

Simon Mizikovsky presented an overview of the 3GPP and 3GPP2 architecture evolution. The trend has been to push functionality toward the edges. This will result in an intelligent, concentrated base station. He foresees new base station routers (BSRs) with IP interfaces showing up in a bunch of different configurations (e.g., hanging on a wall, in a basement, or on a telephone pole). The BSR will be the gateway to an IP core and will handle security connections with client devices and will act as foreign agents in mobile IP. By moving functionality toward the user premises, the user must now be viewed as adversarial. How do we protect you from your neighbor? As the network transitions into this form, we need to build components so that they can be trusted, and ensure that when one component falls, the rest of the network remains.

Hao Chen reiterated previous concerns by asking, "Are cell phones friends or foes?" As more complex services and phones become available, there is a greater chance of malfeasance by end users. Traditionally, phones have been viewed as dumb terminals and not a significant threat. However, this is no longer the case. Hao reviewed a number of attacks that allow a user to circumvent billing and fairness constraints. He concluded that the network should no longer consider the cell phone a friend.

After the panelists completed their presentations, Radu kicked off the questions by asking why none of the panelists mentioned anything about privacy. Ron responded, agreeing that it is a problem. Devices are becoming more like PCs, which means they contain more private information. How to simultaneously protect a phone from a malicious network and the network from a malicious phone is a hard problem.

Conversation then moved toward equipment security. Radu asked whether network devices placed in town are more than simple PCs. John confirmed that many are PCs. From the audience, Perry Metzger noted that in New York City, components are stored in poorly locked rooms. Simon responded that such instances will go away because the equipment of today will go away. John added that new equipment is rolling out fast; however, physical access problems will remain.

In the next line of questioning, Radu asked the panelists whether there is concern with natural disasters and DoS. Thomas replied that there has been significant work and research looking at graceful degradation for overload. However, one of the biggest concerns is that a lot of equipment is outdoors, and components such as towers can be blown down. Simon commented that as the network becomes flatter, and network functionality moves toward the edge, with the number of nodes increasing, survivability increases as well. A flat network is more resilient. John added that systems are getting smaller and more portable. This makes it easier to rapidly replace infrastructure.

With all this talk of impending collapse of connecting networks, an audience member asked where everyone had been in the past 20 years. Have the telcos not been preparing for this? Are they preparing to protect the end user

from threats? John responded that the answer is defense in depth. Many measures need to be put in place to help the situation. Some parts have been deployed, but the solution is not there yet, and a better security architecture is required. When asked about lessons learned, Ron responded that the situation is similar to laptops and desktops five years ago, when users began installing applications from random origins. However, the situation is better for the telcos, because they do not have to deal with a legacy OS. However, the hackers have had 20+ years to learn better techniques. Protecting in both directions at the same time will be difficult.

Conversation changed as another audience member wondered how easy it would be to spoof caller ID/NAI in new networks. Simon responded that such data is passed in the clear in IP networks; however, 4G devices do not have unique identifiers other than their MAC addresses. John added that devices will have x.509 certificates; however, a challenge here is binding device and user authentication.

Radu then asked what skills students need that are not being taught and that the telcos would like to have. Ron responded that students need to have the ability to look at use cases and figure out attacks. Students can look at and understand requirements, but they are not good at looking for problems that don't follow protocols. Also, there is a lack of software people who know what is going on below in the hardware. The software/hardware boundary is where many attacks are occurring. John echoed Ron's comment and added that there is a need for much better software delivery systems. We should not be seeing buffer overflows in these devices. It is very difficult to find people with both security and telecom experience. There is a need for people who can build security-resiliant-protocols.

From the audience, Luke St. Clair asked whether there are proposed solutions for securing endpoint devices and whether any proposals consider secure hardware. Simon responded that some work has gone toward provably secure protocols. As for secure hardware, it depends on how much you want to spend for the platform. Hao commented that there is always the "our phone" solution, where users cannot install software. John mentioned that as the industry moves toward more open APIs, user access will increase. Today's EVDO PC cards do not allow the user to gain access to lower levels, but this access will be available in 4G cards. Thomas added that the network designer can never assume end devices are secure. You must assume it will misbehave. You must protect yourself. Patrick commented that defense in depth is always a good model. Thomas, referencing an earlier audience comment, replied that all locks are doing is buying you time. If you have a strong lock on your apartment, a burglar will pass by you and go to your neighbor; however, eventually that lock will be broken, just as people will break end devices.

Perry Metzger inquired about SS7. He said it is as vulnerable as you can imagine and the only thing standing in the way is "will." People do not know how it works, for now, but this will not always be the case. Thomas replied that it is too expensive to retrofit SS7 networks. If you want to keep the upper parts of SS7, you need to replace the lower levels. Even MAPSEC isn't that secure. Patrick prompted the industry panelists to discuss how widely deployed MAPSEC is. Simon responded that the standards were defined and that it was deployed in two networks, only to be removed because it was determined to be a performance bottleneck.

Next, an audience member asked how he can get involved in testing cellular security without going to jail. Patrick responded that it is getting easier as things move away from SS7 and toward IMS. It is possible to set up an IMS/SIP network for yourself. John added that he has a active security group looking to recruit good people.

INVITED TALK

Mobile Malware

Mikko Hypponen, F-Secure Corp.
Summarized by T. Scott Saponas (ssaponas@cs.washington.edu)

Mikko Hypponen opened his talk by saying, "With new mobile platforms we get new viral vectors." He said that many analysts originally projected that the first viruses for mobile platforms would be on Windows Mobile and be spread by email, because it would be very easy to take existing Windows viruses spread by email and retarget them for the mobile platform.

However, the first mobile viruses were on the Symbian platform and used a Bluetooth vector to spread. Hypponen confirmed the common view that mobile viruses are indeed real and are really spreading. There are more than 370 mobile phone viruses so far and tens of thousands of infections worldwide. One operator with more than nine million customers claims almost 5% of their MMS traffic is infected. Another large operator says 8,000 infected devices have sent more than 450,000 messages, with one mobile device sending 3,500 messages. Hypponen demonstrated some attacks during his presentation.

Hypponen described the prerequisites for a mobile malware outbreak: have enough functionality for malware to work, have enough connectivity for malware to spread, and have enough targets for the platform to become an interesting target. The first mobile virus appeared in 2004 and since then 370 new viruses, worms, trojans, and spy tools have been discovered. Hypponen said that so far they have not seen mobile rootkits, worms that do not need user interaction for spreading, mobile botnets, or any large-scale profit-oriented malware (professionals).

Currently, most malware is for Symbian, and no malware exists for Windows Mobile. Most are trojans (297), with some viruses (58) and a little spyware (9). These trojans

break phones so that they do not work, break a subset of services, cause monetary loss by sending viruses, steal users' private information, or delete email and other important information. This contrasts with desktop viruses, which in the past two years have not been focused on breaking machines because there is no money in breaking machines.

We know that some mobile malware has come from Norway, Spain, Brazil, and many countries in Southeast Asia, Hypponen said. However, the source of many viruses is still unknown. Most infections by this malware also occur in Europe or Southeast Asia. He also explained that Symbian has likely been targeted so much because it is common, SDKs are available, and much existing viral code exists on the Internet. Most malware is just a modification of an existing piece of malware.

Recently the first examples of mobile spyware have been seen. Mobile spyware can record text messages, call information, voice recordings, and even physical location. In fact, there are even vendors on the Internet who sell this software or phones already modified with this software. Although only Symbian-signed applications can have access to the functionality needed for spyware, some spyware has passed the Symbian signing process. For example, one piece of spyware passed as a "system backup" tool.

Hypponen concluded by saying that in the future we can expect to see more for-profit mobile malware and mobile botnets. F-Secure sells a mobile antivirus and firewall product that helps protect against many of these threats.

LOW LEVEL

OSLO: Improving the Security of Trusted Computing

Bernhard Kauer, Technische Universität Dresden
Summarized by Sarah Diesburg (diesburg@cs.fsu.edu)

Bernhard Kauer began by stating that the goal is to have a secure system with only a minimal trusted computing base (TCB). Kauer then explains that he has found bugs in trusted computing systems based on a static root of trust, such as Microsoft's Bitlocker. He gave a general overview of a Trusted Platform Module (TPM), described the PCR register (which is a 160-bit-wide register holding a SHA-1 hash), and discussed what makes a trust chain unfakeable.

During his security analysis, he examined three trusted bootloaders (LILO Darthmouth, GRUB IBM Japan, and GRUB Bochum) and found that all three have bugs that break the trust chain. He then talked about current attacks, including TPM resets and BIOS attacks. For example, in a certain BIOS attack all one has to do is flip just a single bit in the BIOS image to get a TPM with fresh PCR values.

Kauer then discussed using the dynamic root of trust feature instead, because it (1) shortens the trust change, (2) can minimize the TCB of applications, and (3) is less vulnerable to TPM and BIOS attacks. He then introduced the Open Secure LOader (OSLO), which features the first publicly available secure loader that is based on AMD's skinit instruction. It includes its own TPM v1.2 driver and is available at http://tudos.org/~kauer/oslo/. He explained that OSLO works by initializing the TPM, stopping other processes, executing skinit, hashing every module into a PCR, and starting the first module. Kauer also stated that the code size and binary size of OSLO are much smaller than those of BIOS and GRUB.

Kauer's future work includes incorporating memory type detection, DMA protection, and a port of OSLO to the Intel LT platform. He would also like to search for other attack points of Trusted Computing Systems. In conclusion, Kauer stated that OSLO is one step to a minimal TCB.

One questioner asked what he meant by removing the BIOS from the TCB. Kauer explained and also stated that it is an open problem to ensure that ACPI tables are secure. Another questioner asked why OSLO is so much faster than Trusted GRUB if they both incorporate SHA-1. Kauer invited the questioner to look at the code.

Secretly Monopolizing the CPU Without Superuser Privileges

Dan Tsafrir, The Hebrew University of Jerusalem and IBM T.J. Watson Research Center; Yoav Etsion and Dror G. Feitelson, The Hebrew University of Jerusalem

Dan Tsafrir introduced an attack dubbed "cheat" that can be easily launched by a nonprivileged user in order to get any desirable percentage of CPU cycles in a secretive manner. He explained that this means users can arrange things such that their application would get, for example, 95% of the CPU cycles, regardless of any other programs that may be running, and despite any OS fairness considerations. Further, Tsafrir explained that the cheating program would appear as consuming 0% CPU in system-monitoring tools such as top and ps, making the exploit very hard to detect. Tsafrir noted that arbitrary programs can be turned into cheaters through binary instrumentation or other means.

Tsafrir identified two unrelated mechanisms that independently make systems vulnerable to cheating. The first is time-keeping. Specifically, the OS measures time in "tick" units, where each tick is a few milliseconds. The way it works is that the OS wakes up every tick and charges the currently running program for using the CPU during the last tick. This happens even if the program actually used a small fraction of it. This sampling approach is fairly accurate if applications "play by the rules" (the more the program runs, the bigger the chances are of being billed), but Tsafrir showed how a program can "cheat" this mechanism by systematically sleeping when the OS wakes up. As a consequence, the OS erroneously thinks the cheating program consumes no CPU cycles at all, which grants it a very high priority, thereby allowing it to monop-

olize the CPU. Tsafrir stated that all examined "ticking" OSes were found vulnerable to the attack to some degree, including Linux, FreeBSD, Solaris, and Windows XP.

In contrast to the tick-based time-keeping mechanism that is used by OSes since the 1960s, the second exploitable mechanism identified by Tsafrir is much more recent: scheduling for multimedia. Tsafrir noted that, in an attempt to better support the ever-increasing CPU-intensive multimedia component within the desktop workload, some modern systems (Linux, FreeBSD, and Windows XP) have shifted to prioritizing processes based on their sleep frequency rather than duration. Tsafrir concluded that this major departure from the traditional general-purpose scheduler design plays straight into the hands of cheaters, which can easily emulate CPU-usage patterns that multimedia applications exhibit. To the question of how, then, OSes will be able to provide adequate services for multimedia applications, Tsafrir responded that an OS can favor one program over another only if it is reasonably sure that the former is much more important to the user, and that for this purpose, his research group has done a lot of work to explicitly track interactions with users (through the appropriate device driver) and to leverage this information for improved multimedia scheduling.

■ ***Memory Performance Attacks: Denial of Memory Service in Multi-Core Systems***

Thomas Moscibroda and Onur Mutlu, Microsoft Research

Onur Mutlu began by introducing a new class of Denial of Service (DoS) attacks prevalent on multicore chips in which the cores share the same DRAM memory system. He explained that as different threads or processes execute on different cores, those threads or processes can interfere with other memory access requests. He noted that most scheduling policies are not even thread aware and are very thread "unfair." Mutlu then introduced the concept of a Memory Performance Hog (MPH), which is a thread that exploits unfairness in the DRAM controller to deny memory service (for long periods) to threads co-scheduled on the same chip.

Mutlu then demonstrated the problem by walking through a memory scenario caused by running the popular benchmarking program *stream*. Stream has a very high memory row-buffer locality. Unfortunately, because many memory systems implement a First-Ready First-Come First-Serve (FR-FCFS) scheduling algorithm, stream's requests will be serviced before other, older memory requests, to take advantage of the current row buffer. Mutlu says that this unfair memory request servicing can severely degrade other threads' performance and that it is easy to write an MPH.

DRAM fairness was then discussed. A DRAM system is fair if it slows down each thread equally. Mutlu defines the goal of a fair scheduling policy as equalizing DRAM slowdown *i* for all threads *i*. Two fair memory scheduling algorithms were discussed. A hardware implementation must keep track of experienced latency and estimate ideal latency. After the presentation, Mutlu commented that this problem is probably even more severe in hyperthreading. He also noted that the stream benchmarking example is not even the worst case. An attacker could create much worse cases.

INVITED TALK

■ ***Computer Security and Voting***

David Dill, Stanford University

Summarized by Adrienne Felt (felt@virginia.edu)

David Dill spoke about the history and current state of trustworthy electronic voting in the United States. Dill became involved in the politics of electronic voting in 2002–2003 when he wrote the "Resolution on Electronic Voting" petitioning for user-verifiable voting. He challenged the audience to question the current system of blindly trusting voting machines and to contribute to the campaign for a voting audit trail.

The resolution calls for a voting trail that can be authorized by the user and then is indelible. The goal is not to discourage people from voting, and there is no proof of wide-scale voting theft. However, Dill argued that the absence of known vulnerabilities is insufficient; instead, it is the responsibility of the government and voting machine manufacturers to provide evidence that the election was correctly carried out. He said that the core security problem is the threat of internal attacks. Regardless of how well employees are screened, Dill asserted that it is morally wrong to force voters to trust strangers with their votes. Even if the technical issues are perfect, how can voters understand and trust this? Even if the hardware and software are secure, how do you verify that the correct equipment is in the box? Voting must be auditable and audited.

In August 2003, few people agreed with his position. Since then, however, public opinion has begun to change. Big blunders in security design were uncovered. Manual auditing is catching on, with 13 states now practicing this. The House of Representatives will soon be considering the Voter Confidence and Increased Accessibility Act. Dill credited the computer security community for this and thanked his colleagues who have taken the time to learn about e-voting and talk to politicians and the press.

Despite the progress that has been made, Dill stressed that their work is not yet complete. It is still necessary to emphasize that patching external attacks is insufficient. They need to go district to district and ensure that the election officials understand the relevant issues. There is currently no good legal recourse for the worst-case scenario of a broken election. The general public doesn't understand that security is a continuous spectrum and a continuing problem. There remains a need for members of the computer security community to join the trustworthy voting campaign; even if the current problems are fixed, new attacks

and attackers will develop in the future. For those who are interested in participating, more information is available at VerifiedVoting.org.

His talk generated numerous questions. Two audience members asked whether paper trails were more reliable than electronic ones, and Dill responded that paper-based voting also needs to be subjected to more scrutiny but that voter registration and verification are separate issues. There was also a discussion about how secure but complex technologies can lead to errors, and Dill emphasized the importance of good human-factor design for both voters and election volunteers. When asked about the plausibility and potential effectiveness of increasing election centralization and uniformity with a constitutional amendment, Dill replied that the federal government can only regulate the national elections and not the hundreds of local elections. Additionally, the federal government is slow and counties and states are easier to influence. The bottom-up approach is part of what makes the trustworthy e-voting campaign so resource-intensive.

OBFUSCATION

Binary Obfuscation Using Signals

Igor V. Popov, Saumya K. Debray, and Gregory R. Andrews, The University of Arizona

Summarized by William Enck (enck@cse.psu.edu)

Saumya Debray began by reviewing background material. The program compilation process strips the high-level code semantics from an application. Reverse engineering and disassembly can reestablish these semantics from a binary executable. Sometimes, a program's author wants to keep high-level code from prying eyes. Binary obfuscation provides such a mechanism. However, motivations ranging from protecting intellectual property to concealing malware make binary obfuscation a double-edged sword.

There are two classes of reverse engineering: static and dynamic. As the names indicate, static analysis does not execute code. It has many advantages, including complete code coverage and simpler algorithms; however, it cannot account for self-modifying code. Dynamic analysis, in contrast, accounts for self-modifying code, but it only disassembles the executed code. Additionally, it requires significantly more complex algorithms in order to track state. Furthermore, running code can take defensive measures to hide execution from debuggers. The goal of this work is to make static reverse engineering so impractical that the adversary is forced to perform dynamic analysis, at which point runtime defenses can be implemented.

Static reverse engineering primarily relies on control flow analysis, which involves both identification of control flow instructions and inference of resulting jump locations. In short, it must find where code branches and where it ends up. Hence, the authors hypothesize that hiding control flow instructions will make static analysis impractical. Saumya proposes two techniques to hide legitimate control flow instructions. First, all such instructions are converted to "ordinary instructions" that raise a signal when executed (e.g., a segmentation fault or division by zero). A special signal handler traps the execution and first checks the code location. If the instruction corresponds to a jump, execution jumps to the correct location; otherwise, the signal is propagated to standard handlers. Second, bogus control transfer instructions are inserted at unreachable locations, further confusing the static analysis tool. Saumya explained that this binary obfuscation technique makes static analysis both provably NP-hard (when pointer aliasing is introduced) and practically expensive. Finally, he showed that the technique is a very effective deterrent against popular disassemblers and incurs only a 21% slowdown.

David Wagner pointed out that the kernel traps signals, thereby allowing the kernel to observe control flow. Saumya indicated that this is dynamic analysis, and the goal of the work was to force the adversary to use dynamic analysis; other defensive runtime mechanisms can be used to thwart such attacks. Another audience member raised concern over the location of the mapping table used by the special signal table to determine if a signal is really a jump. Saumya replied that there are multiple ways to obfuscate the mapping table, for example, smearing it across the code; however, the current implementation uses a simpler method of XORing addresses.

Active Hardware Metering for Intellectual Property Protection and Security

Yousra M. Alkabani and Farinaz Koushanfar, Rice University

Farinaz Koushanfar explained hardware piracy. Chip fabrication facilities cost billions to build and maintain. Most hardware design firms can't afford such facilities and therefore outsource the fabrication process. Unfortunately, to do so, design firms must divulge their raw intellectual property (IP) and have no way to ensure the honesty of the fabrication facility. Hardware IP piracy has been estimated to cost one billion dollars per day. The solution is active hardware metering, a process that ensures that pirated fabricated chips are unusable. A number of passive hardware metering systems have been proposed; this is the first work to consider the active variety.

Active hardware metering aims to protect a design firm from a fabrication facility wishing to make extra copies to sell for itself. Note that this is a well-funded adversary, as it owns a multi-billion-dollar facility. Farinaz proposes an approach where fabricated integrated circuits (ICs) are initially locked, and the design firm must unlock each IC before it can be used. This is done by adding additional useless states into finite state machine (FSM) logic. The resulting Boosted FSM (BFSM) is seeded by a unique value for each fabricated chip. This value is derived from fabrication variability as shown by Su et al. in ISSCC'07. With high probability, the IC starts in an inoperable state, and

only the design firm has the ability to correctly transition the BSFM into the correct operational state. To further hinder adversarial analysis and protect against brute force attacks, the BSFM may also contain black-hole states that render the IC useless. Finally, Farinaz explained that FSM logic contributes only a minimal amount to overall chip area, and converting FSMs to BFSMs has negligible impact on chip size. Further analysis shows the same for power consumption and timing delays.

Adrian Mettler inquired whether the logic containing fabrication-specific identification circuitry could be removed from the IC netlist, thereby allowing simulation to derive BFSM logic and allowing the adversary to unlock chips. Farinaz replied that hardware simulation is nontrivial. A single simulation can commonly take over six months for today's microprocessors. Furthermore, as circuit component size metrics count atoms, variation is very important. The nondeterministic portions cannot be taken out of the design. Robert Cunningham expressed concern over the scalability of incorporating active metering. Farinaz replied that much of the process is automated, and the parts that needed to be hand-designed could be automated by incorporating the technique into existing hardware design tools.

INVITED TALK

Advanced Rootkits

Greg Hoglund, HBGary

Summarized by Sarah Diesburg (diesburg@cs.fsu.edu)

Greg Hoglund's talk covered funded rootkits, types of attackers, a new attack trend of desktop exploitation, classifications of rootkits, ways in which rootkits may be installed, goals of rootkits, and, finally, how to build, package, and install a rootkit. Although Hoglund's talk was targeted at Windows operating systems, the general concepts can be applied to all other major operating systems.

A funded rootkit is subversive malware developed with a budget. Hoglund reminded us that rootkit authors put rootkits they are making through an extensive testing and development process by testing them against all known antivirus software.Types of rootkit authors include competing corporations, foreign and multinational corporations, foreign governments, intelligence services of friendly and allied countries, former intelligence officers, extremist groups, and organized crime and drug cartels. Many organizations do not report rootkit exploitation, so the extent of this problem is relatively under-reported.

Desktop exploitation was discussed as the culmination of recent rootkit trends. It involves such desktop applications as Yahoo! Toolbar Helper, Microsoft Certificate Authority Control, Crystal Reports Control, and Quicktime. A parallel was drawn between desktop exploitation attacks and Internet attacks by comparing exposed system API calls to exposed TCP ports.

Hoglund felt that the current method of classifying rootkits based on technological mechanisms is too inflexible. He described multiple rootkit types and gave examples of each. These types include tool trojans and log cleaners, permanent installations of parasitic code into existing programs, basic backdoor programs, operating system trojans, dynamic parasitic infections of existing processes, parasitic application extensions, modifications of operating system library functions, parasitic device drivers, free code, memory cloaking rootkits, rootkits that hide from DMA, rootkits for embedded systems such as cell phones, rootkits lower in hardware such as hypervisors, boot vectors, and overflow activation.

Finally, methods of building and packaging rootkits were discussed and snippets of code were shown. These pieces of codes and methods are available at www.rootkit.com.

A questioner asked how Hoglund believes we should design security solutions in the future. Hoglund suggested that the problem might be solved through trusted computing and DRM in hardware-level support. Another person asked whether there will always be a never-ending supply of undiscovered rootkit techniques, and Hoglund believes there will be until hardware stops it. He also commented that the price of rootkits is rising because demand is rising.

NETWORK SECURITY

On Attack Causality in Internet-Connected Cellular Networks

Patrick Traynor, Patrick McDaniel, and Thomas La Porta, The Pennsylvania State University

Summarized by William Enck (enck@cse.psu.edu)

Patrick Traynor began by warning of the paradox of specialization. Optimizing a subset of functionality provides benefits under normal conditions, but it causes disasters when the environment changes. A canonical example is the Tacoma bridge collapse. Although the bridge was designed to withstand strong forces, it was only a 40-mph breeze that led to its collapse. We are seeing a similar phenomenon with today's telecommunications networks. They were designed for rigid constraints and use patterns; however, many assumptions no longer hold because of the incorporation of the Internet. We have seen a number of low-bandwidth attacks against cellular networks, and Patrick claims that adding more bandwidth will not solve the problem. He believes the problems are the result of a clash in design philosophies, that is, the connection of smart and dumb networks, which differ in conceptual definitions such as traffic flow.

To further justify his claim, Patrick introduced two new low-bandwidth attacks against cellular data networks. GPRS (the data service for GSM) sets up channels whenever a client requires communication. This is an expensive

operation which potentially requires multiple paging sequences to locate a device. To alleviate setup strain, devices maintain the channel for at least five seconds. However, there are a limited number of channels. GPRS specifications allow for up to 32 concurrent flows per sector, but many equipment manufacturers use less, for performance reasons. Exhausting this virtual channel resource is straightforward. An adversary need only ping 32 devices in a sector every five seconds to ensure all channels are occupied. Patrick used mathematical analysis and simulation to show that an adversary only requires 160 kbps of bandwidth to block 97% of legitimate traffic in Manhattan, a value dwarfed by the theoretical maximum capacity of 73 Mbps. A similar attack is shown to be effective on the PRACH (packet random access channel).

So what is the problem here? For phone calls, the channel technique makes sense; the setup costs are amortized by multiple-minute phone calls. Data communication, however, does not fit this mold, and many protocols (e.g., instant messaging) require only very small messages with significant delays between network use. Throwing more bandwidth at the problem does not reduce setup latencies. Patrick showed a simple throughput equation as a function of packets, setup latency, and bandwidth. He explained that as bandwidth is taken to infinity, the throughput is entirely reliant on setup latency. Hence, although the rigid design works well for phone calls, the choice of a circuit-switched network architecture provides a fundamental limitation because the cellular network is attached to the Internet, where end points do not care about what happens inside (as in the end-to-end argument).

In the question and answer session, Niels Provos agreed with the fundamental problem of specialization and inquired how the telcos are going to fix the problem. Patrick replied that he cannot speak for them directly, but technologies such as WiMAX are promising; however, 4G is in its infancy, and it might be a while before these networks are deployed. Another audience member asked how much of the current design is dictated by battery power and if this will prevent architectural changes. Patrick agreed that power limitation of phones has a large influence, but this will change as providers expand to target laptops and more versatile devices.

■ ***Proximity Breeds Danger: Emerging Threats in Metro-area Wireless Networks***

W.Y. Chin, Institute for Infocomm Research (I^2R), Singapore; V.T. Lam, University of California, San Diego; S. Sidiroglou, Columbia University; K.G. Anagnostakis, Institute for Infocomm Research (I^2R), Singapore

Summarized by William Enck (enck@cse.psu.edu)

Periklis Akritidis began by reminding the audience of the pervasiveness of wireless networks. According to wardriving data, most access points are left unprotected. Of the remaining protected access points, most use WEP, which has been broken many times over. In metropolitan areas, access points are literally on top of one another, and wireless clients can easily see multiple SSIDs. This work proposes a new theoretical worm, called a wildfire worm, which spreads by physical proximity to wireless access points.

The wildfire worm gets its name from the way it propagates. Once infected, a wireless host listens for other SSIDs and, if one is found, it attempts to infect hosts on that network. In turn, those hosts infect other hosts on adjacent networks, and so on. There are two techniques to propagate a wildfire worm: pull and push. The pull technique requires the victim host to download malicious code (from a Web site), which may be achieved via ARP or DNS poisoning. More interesting is push propagation. To better understand the propagation potential, public wardriving data from major metropolitan areas was acquired and analyzed, suggesting that in denser areas, a well-crafted worm can infect up to 80% of wireless hosts within 20 minutes.

The existence of such an out-of-band worm allows for a twist on a number of well-known attacks. Among these are packet sniffing, ARP and DNS spoofing, and phishing. Additionally, Periklis proposes tracknets, in which an attacker "rents" wireless networked zombie hosts to track specific users as they move throughout the city. All attacks are much more practical with the use of a wildfire worm, because the adversary is within the network. Furthermore, it is harder to extinguish, because it does not rely on the Internet to communicate. Fortunately, all is not lost. Periklis proposes a number of countermeasures, which when used in conjunction will limit the effectiveness of many attacks while allowing the wireless networks themselves to remain open.

Angelos Stavrou asked whether a fast-moving car could aid propagation speed. Periklis replied that this would be similar to seeding the attack in more locations. Another audience member inquired about wildfire worms applied to access points that separate an internal wireless network from an intentionally open and public wireless network. Periklis said that if the attack can bypass the access point, it can still occur. A final audience member noted that there is something useful and attractive about having an open access point environment and inquired if there is any way to salvage it. Periklis replied that the reactive and filtering techniques described in their paper will allow networks to remain open.

■ ***On Web Browsing Privacy in Anonymized NetFlows***

S.E. Coull, Johns Hopkins University; M.P. Collins, Carnegie Mellon University; C.V. Wright and F. Monrose, Johns Hopkins University; M.K. Reiter, Carnegie Mellon University

Scott Coull began by explaining that many research areas desire real network logs. Network administrators realize the benefit of such research and would like to provide logs; however, their primary function is to protect the anonymity of their users. For example, if many employees visit monster.com, a clueful investor may steer clear of that

company's stock. A number of anonymization techniques have been proposed, but how effective are they? If made public, anonymized network logs are available for long periods of time and are subject to the scrutiny of complex inference algorithms. Previous work has revealed a number of flaws in anonymization processes; however, Scott believes there is more to learn. Ultimately, the goal is to better understand what properties are necessary for secure anonymization and to provide guidance to publishers desiring to contribute their data.

This work specifically considers anonymized NetFlow data, which consists of a time-ordered sequence of records. Each record provides information about packets in a TCP connection between a server and a client. The goal of the adversary is to find a specific Web site. The general approach is to download the front page of the target Web site and compare the many possible flows to the data set. Previous work has considered purely flow analysis; however, as Scott indicated, objects often shift between flows, resulting in an overlap between Web sites. Hence, the cumulative size of the Web page is incorporated. Analysis shows that each Web server for a given site serves a unique set of Web objects discernible in a 3D plot. Furthermore, physical servers that occupy the same space, owing to servicing the same sort of content, can be consolidated into logical servers. Now, each Web site can be identified by the presence of transactions with specific servers, and a Bayes Belief Network is used to match the target Web site to the data set. The deanonymization technique was evaluated in a number of scenarios, including two real-world scenarios with real data. The evaluation found that complex but stable sites are very detectable; simple sites have high false detection rates; and volatile sites result in low true detection rates.

An audience member inquired how easy it would be to find multiple sites matching the same model (i.e., is there deniability?), Scott replied that it is hard to tell from the small sample set of Web sites they have tested thus far. He believes that the more complex the site is, the harder it will be to find a Web site with a similar fingerprint. Another audience member posited that the more sensitive part of the problem is identification of the client IP address. How clustered is the data if you try to detect IP addresses? Scott replied that previous work shows that client IP addresses do not provide much information about users. However, other knowledge, such as knowing a specific person is always in the office at 8 a.m., will go a long way toward deanonymizing that person's traffic. Finally, Roger Dingledine asked about the reason for including the sensitive fields in the first place, and, for that matter, who is allegedly using the data? Scott replied that simply removing the fields makes the data useless. There are a number of repositories (e.g., DHS predict) that are invaluable in areas such as IDS. Fabian Monrose added that the upshot is that past threat models are too weak and this work helps better understand anonymization. The goal is not to stifle the release of anonymized data but to get more people involved to help build better frameworks.

INVITED TALK

Covering Computer Security in The New York Times

John Schwartz, The New York Times

Summarized by Sarah Diesburg (diesburg@cs.fsu.edu)

John Schwartz began by commenting on the oddity of a reporter functioning as a speaker instead of being on the other side of the podium. Schwartz went on to discuss what he sees as a general view of mainstream media: The mainstream media often get technical stories wrong, and get them wrong repeatedly, because their reporters are not technically adept, are looking for scare stories, and are trying to get the newspaper equivalent of ratings. He was there to talk about why this isn't true for those in the reporting industry that work at the top of the game, and that it truly is possible to write about these issues without hype.

Schwartz jumped right into his views of the reporting industry, his values when reporting a technical story, trends in the newspaper industry, and general advice on getting a story reported correctly. One of his mottos is "Dare to be dull." He believes it is important to cover what is most important and that it will get out to the public even if it is not the front-page story. He acknowledges that some journalists are aiming to write "sensational" pieces, and he advises us to avoid them. Instead, find those journalists who hold the old-fashioned concept of "serving the reader."

He acknowledges the substantial effect the Internet has had on the newspaper business. People expect to get free news online, and this brings new challenges to the industry.

As the floor was opened to questions, one questioner asked about those doing the fact-checking for technical articles, as it seems that many of them are not very accurate. Schwartz commented that some publications tend to push journalists to write too much and too fast. He is also stunned by what he sees as lack of attention to detail and lack of caring. His advice is to figure out who in the industry gets it right. Another questioner asked whom he should contact if he has important knowledge. Schwartz said that local reporters are very accessible, and an offer of help can go a long way. In response to a question about how it seems reporters always want to give the other side of the story equal weight, even if one side of the story is obviously more accurate, Schwartz said that he believed that everyone affected by a story needs to be in it. Nonetheless, he added, there is a tendency in the media to equate balance with equal weight. He says many reporters don't reflexively understand the distinction, but a good reporter needs to understand this.

WORK-IN-PROGRESS REPORTS

Summarized by Adrienne Felt (felt@virginia.edu)

- ***VM-Based Malware Detection System***

  Yuhei Kawakoya, NTT Information Sharing Platform Laboratories

In this presentation, Kawakoya introduced two methods of externally monitoring the security of a virtual machine. Current antivirus software is installed in the same operating system that it is trying to protect, which gives malware a way to attack the antivirus software. However, Kawakoya's approach separates the protection mechanism from the protected operating system. In the first of the two methods, called Outside System Call Hooking, the host OS recognizes the invocation of a system call and checks the status of the guest OS. The second technique, Execution Cache Investigation, uses pattern matching to compare the cache with static virus signatures. An implementation demonstrated the effectiveness of this approach with samples collected from the wild.

- ***Controlled Reincarnation Attack to Subvert Digital Rights Management***

  F. John Krautheim and Dhananjay S. Phatak, University of Maryland, Baltimore County

This presentation explained how to use a virtual machine (VM) to circumvent digital rights management restrictions. The simplest way to use a VM to avoid copyright constraints is to store the software or media in a VM with the correct state (e.g., system time). This can be prevented by requiring communication between the product and a company server, with the server remotely maintaining the state of the license. Krautheim then outlined how virtualization technology can be used to fake the server's half of the communication to trick the product into believing it has received permission to be used. Copies of the VM are identical at startup, so the VM-server communication will be exactly the same for each instance of the VM. In a "controlled reincarnation" attack, the server's authentication can be sync'd and replayed indefinitely to provide access to the restricted content without needing to break the communication's encryption. Krautheim and Phatak are working on a defense for this attack that works by preventing the execution of restricted content in a virtual machine. The protected media would come packaged with a utility that uses timing benchmarks to detect a virtualized environment.

- ***The Performance of Public Key–based Authentication Protocols***

  Kaiqi Xiong, North Carolina State University

Kaiqui Xiong compared the performance of two authentication schemes that combine Kerberos and public key cryptography. Public-Key Cross Realm Authentication in Kerberos (PKCROSS) reduces the need for maintaining cross-realm keys so that Kerberos is easier to implement on large multirealm networks. Public Key Utilizing Tickets for Application Servers (PKTAPP) was intended to improve the scalability of PKCROSS, but Xiong showed that PKTAPP does not actually scale better than PKCROSS. He also showed that PKCROSS outperforms PKTAPP in multiple remote realms.

- ***Automatic Vulnerability Management Based on Platform Integrity***

  Megumi Nakamura, Seiji Munetoh, and Michiharu Kudo, IBM, Tokyo Research Laboratory

This presentation discussed automatic vulnerability checking, the goal of which is to simultaneously reduce the administrator workload and improve security. However, automated security tools have weaknesses and can be attacked. Nakamura et al.'s work uses a trusted platform module to store integrity and vulnerability information about the security tools so that they can be compared. When a discrepancy between the expected and actual integrity values is found, the vulnerability information is used to diagnose the problem. This places some of the responsibility for security on the hardware, which is more trustworthy than software-only approaches.

- ***Attacking the Kad Network***

  P. Wang, J. Tyra, T. Malchow, Y. Kim, N. Hopper, D. Foo Kune, and E. Chan-Tin, University of Minnesota

The authors of this work created a successful attack on the Kad network, which supports the eDonkey peer-to-peer file-sharing network. They found structural vulnerabilities that allowed them to interfere with the entire network's keyword search functionality from a small number of nodes. Their experiments showed that their attacks worked on eMule and aMule, the most popular Kad clients, and that a single user could halt 65% of Kad searches. These structural weaknesses could be exploited to target specific keyword searches or to hijack the network for DDoS attacks.

- ***Virtual Machine Introspection for Cognitive Immunity (VICI)***

  Timothy Fraser, Komoku, Inc.

Timothy Fraser presented a rootkit detection and repair system that uses virtualization and artificial intelligence to monitor the state of an operating system. In the setup, a GNU/Linux kernel is observed while running in a Xen virtual machine. The VICI system is trained to detect and undo the behavior of kernel-modifying rootkits. The artificial intelligence is based on the Brooks Subsumption architecture for autonomous robots and is expected to improve its performance over time.

- ***Polymorphic Shellcode Detection Using Emulation***

  Michalis Polychronakis, Foundation for Research & Technology—Hellas (FORTH)

Michalis Polychronakis presented a network-based behavioral analysis system designed to detect malicious shell-

code. The detector runs on a network intrusion detection system CPU that intercepts and inspects traffic. Unlike traditional malware techniques, their system does not do pattern matching for known static signatures. Instead, it executes the instruction sequences included in the traffic and analyzes the behavior of the code. The focus on behavior allows for the detection of new attacks and self-modifying polymorphic code. This approach was shown to be effective on a real-world deployment of the system. An audience member asked about false positives, and Polychronakis responded that the only two false positives were due to a bug that was subsequently fixed.

■ ***CANDID: Preventing SQL Code Injection Attacks***

Prithvi Bisht, University of Illinois, Chicago

Prithvi Bisht presented a defense against SQL injection attacks that compares the post-input query to the programmer's intended query structure. He said that a SQL injection attack, by definition, changes the structure of the involved query. The defensive system attempts to dynamically discover the structure of the programmer's intended query by evaluating the query behavior over known benign inputs. The system has been implemented in a tool, CANDID, that transforms and thereby safeguards Java Web applications.

■ ***Protecting User Files by Reducing Application Access***

William Enck, Patrick McDaniel, and Trent Jaeger, SIIS Lab, Pennsylvania State University

This presentation introduced new file access controls that prevent files from being accessed by programs that are not on a permission whitelist. PinUP, an access control overlay for Linux, extends file system protections by explicitly defining program access permissions at the inode level. This approach is intended to make file access control more intuitive and secure by saying that data should only need to be accessed by applications that the user specifies. An audience member asked whether PinUP is vulnerable to a confused deputy attack, but Enck clarified that this is not currently a problem since PinUP has not yet been implemented for binary applications.

■ ***Securing Web Browsers Against Malicious Plug-Ins***

Mike Ter Louw, University of Illinois at Chicago

Mike Ter Louw described the dangers of Web browser plug-ins and how to defend against them. To demonstrate the potential harm that could be done by a plug-in, the authors of the work created a malicious extension called BrowserSpy that can be installed without special privileges. Once installed, it has access to all the functionality of the browser. They then created a code integrity checking technique to control the plug-in installation process and defend against malicious extensions. They are also exploring behavioral analysis to monitor the actions of extensions. More information and details can be found in the work "Extensible Web Browser Security," published at the Detection of Intrusions and Malware and Vulnerability Assessment in 2007.

■ ***Detecting ISP-Injected Ads with Web Tripwires***

Charles Reis, Steven D. Gribble, and Tadayoshi Kohno, University of Washington; Nicholas C. Weaver, ICSI

Some ISPs have begun altering Web page content to add advertisements and paid links to Web sites. The authors of this work created a Javascript Web tool that allows users to determine if their Web content is being modified by ISPs. The tool is aware of its intended content and can detect when it has been changed. Their site received approximately 50,000 hits from Digg and Slashdot, leading them to identify several kinds of ad injections. They have not yet revealed the names of the ISPs that are practicing this form of advertising. The tool is available at http://vancouver.cs.washington.edu. The topic proved extremely interesting to audience members and received several questions. One audience member asked whether it was possible for them to differentiate between site changes made by a client-side adware proxy and ISPs; Reis admitted that it is not always possible, but they tried. Another asked whether ISPs could use this same method of site code injection for a beneficial service such as local tailoring, but Reis clarified that although that could be good it is not what is happening.

■ ***Leveraging Non-Volatile Memory for Advanced Storage Security***

Kevin Butler, Pennsylvania State University

Kevin Butler introduced ways to use fast-access non-volatile RAM-enhanced hybrid disk architectures to facilitate secure processes such as authenticated encryption. A secure boundary is placed between the disk interface and the operating system, with the operating system as an untrusted party. This approach is significantly more sophisticated than what is possible with traditional storage systems.

■ ***Tor***

Roger Dingledine (arma@mit.edu)

Tor is an anonymous browsing site funded by various organizations including the DoD and EFF. The third largest group of users is in China, where Tor gives users the ability to circumvent Chinese Internet censorship. Roger Dingledine challenged the audience to help plan for the possibility that China will try to block access to Tor. One suggestion is to allow a non-Chinese user to act as a relay between the Chinese user and a Tor bridge, but how can this be set up? It would be necessary to hide thousands of IPs from the Chinese government yet give them out one at a time to users. Dingledine discussed the use of scarce resources, such as time and IP addresses, to determine which IPs are distributed. He said that the project is looking for members and needs ideas from the security community.