usenix ;login:

DECEMBER 2012 VOL. 37, NO. 6
ELECTRONIC SUPPLEMENT

# Conference Reports

## 7th USENIX Workshop on Hot Topics in Security (HotSec '12)

Bellevue, WA
August 7, 2012

### Opening Remarks

*Summarized by Rik Farrow (rik@usenix.org)*

Patrick Traynor explained the format he and the program committee had chosen for this year's HotSec: each presentation would last only 15 minutes, leaving 45 minutes at the end of each session for a panel discussion. Patrick then explained that he would be leaving early because of an event long in the making (the birth of a child) meant he needed to leave early. I later learned that the event was successful, adding a baby girl to Patrick's family.

Note that not all scribes covered the discussions as well as they might have. The session was recorded, however, so you can experience the complete discussion session if you wish.

### Maturing Malware

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

#### *GANGRENE: Exploring the Mortality of Flash Memory*

Robert Templeman, Indiana University Bloomington and Naval Surface Warfare Center, Crane Division; Apu Kapadia, Indiana University Bloomington

Robert Templeman pointed out that while flash memory has many desirable properties, it suffers from limited write endurance. Various strategies exist to overcome this weakness, such as wear-leveling and error correction codes. Unfortunately, Templeman argued, flash's write endurance is still highly susceptible to attack.

Templeman discussed how the lifetime of a flash device is dependent largely on the way it is used and that manufacturer endurance estimates fail to take into account malicious activity. Instead, manufacturers generally test flash memory under what they consider to be normal usage conditions. To prove his point, Templeman introduced a simple attack based on rapidly writing to memory, and empirically demonstrated an order of magnitude increase in flash wear over the manufacturer estimates.

When asked about how best to defend against flash attacks, Templeman responded that hardware solutions are not feasible; instead, the defense should be software-based, with the software spanning multiple layers of the operating system and the flash controller.

#### *NoisyKey: Tolerating Keyloggers via Keystrokes Hiding*

Stefano Ortolani, Vrije Universiteit, Amsterdam; Bruno Crispo, University of Trento, Trento

Stefano Ortolani described how current defenses against keyloggers are based on either prevention or detection and, despite our best efforts, are not sufficient to address the threat completely. He proposed an alternative approach based on the idea that keyloggers are unavoidable and, as a result, a system should learn to tolerate their presence.

Ortolani's idea is to hide user keystrokes in a noisy channel such that the original keystrokes can only be recovered by a legitimate application and not by the keylogger. His approach is complicated by the unpredictability of user activity and the need for robustness. Ortolani concluded that his approach can be adopted with a limited impact on the user.

Kevin Fu asked how new non-keyboard devices might affect the current threat landscape. Ortolani answered that new interfaces provide more and varied opportunities for attackers to exploit and that this is an area worth exploring.

#### *Potentia Est Scientia: Security and Privacy Implications of Energy-Proportional Computing*

Shane S. Clark, Benjamin Ransford, and Kevin Fu, University of Massachusetts, Amherst

Shane Clark argued that as computers become more energy-proportional (i.e., power consumption scales closely with workload) they will begin to leak more fine-grained information about their current state. He demonstrated how a clever attacker could exploit this trend to violate a user's privacy. Specifically, he detailed one attack whereby an attacker can use power readings obtained external to the computer to determine what Web sites a user is browsing.

Clark also pointed out how whole-system power analysis can be beneficial for malware detection on embedded devices. He suggested that external detection is preferable, especially for medical devices, given the tight hardware constraints and relative difficulty incorporating malware detection directly into the device. Traditional techniques are further complicated by the number and variety of already deployed embedded devices.

While several audience members questioned the feasibility of the discussed power analysis attack, Clark stressed that malware detection is the much more promising application for whole-system power analysis..

## Maturing Malware

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

### *Impeding Automated Malware Analysis with Environment-Sensitive Malware*

Chengyu Song, Paul Royal, and Wenke Lee, Georgia Institute of Technology

Adopting an adversarial viewpoint can be instructive. Chengyu Song presented a suite of techniques that make malware less amenable to automated analysis. Their technique has already appeared in some form in the Flashback malware targeting Mac OS X; the authors sought to identify and refine the trend.

Malware analysts rely on their ability to capture malware samples and run them in a controlled environment. Some malware tries to detect such analysis and appear innocuous, but when analysis tools improve, the arms race continues. What if it were fundamentally intractable for analysts to pick apart malware samples? By separating malicious duties and judiciously using modern cryptography, malware can achieve this dreadful goal.

Under the authors' model, malware seeking a foothold on a host uses a mélange of system properties to derive a host-specific encryption key that "binds" the malware to one machine. An analyst with a transplanted malware sample on a different host would therefore derive a different key. Song called this mechanism host-identity-based encryption (HIE). The malware interpreter uses its HIE key to authenticate itself to a command-and-control server, which sends back an encrypted payload with a randomized instruction set (which the authors call instruction-set localization, or ISL).

Song described several possible countermeasures that would fail primarily for privacy reasons: nobody wants to send an entire infected host to an antivirus vendor. Analyzing malware directly on end hosts is intractable for similar reasons. An "allergy attack" approach that changed system parameters for each OS process would presumably break legitimate software.

### *Software Diversity: Security, Entropy, and Game Theory*

Saran Neti and Anil Somayaji, Carleton University; Michael E. Locasto, University of Calgary

Saran Neti tested the audience's memory of information theory and game theory with a talk about quantifying software diversity, a much touted but seldom understood goal of many software ecosystems. Neti and his co-authors formalized software diversity using techniques from other fields, then modeled defense through diversity as a game to explore the space of software-diversity strategies.

The authors model a software ecosystem as a bipartite graph with n hosts, m (catastrophic) vulnerabilities, and k vulnerabilities per host, all with distributions drawn from public OS and browser vulnerability data. Seen as a collection of probability distributions, the graph of a software ecosystem has a Renyi entropy (a generalization of Shannon entropy) that the authors argue is a useful measure of software diversity. Neti showed an intuitive plot from the paper to illustrate the lacuna between measured and "ideal" diversity.

With these formalizations in place, Neti proposed game-theoretic models to analyze defense-through-diversity strategies. In particular, "dispersion" games favor maximal outcome diversity (i.e., greater software diversity). The paper cheekily but accurately considers switching among software equivalent to switching among vulnerabilities; doing so has a cost that game theory can help analyze. Neti proposed a real-world "capture the diversity" variant of "capture the flag" wherein defenders could choose one of two functionally equivalent services to keep running.

### *When Good Services Go Wild: Reassembling Web Services for Unintended Purposes*

Feng Lu, Jiaqi Zhang, and Stefan Savage, University of California, San Diego

Feng Lu and his co-authors consider a world in which cloud providers, rather than users, inadvertently pay the cost of unorthodox service mashups. In a discussion that presented familiar services in a new light, Lu described how his group assembled a usable Web proxy out of Google and Facebook APIs.

Modern Web services often need to incorporate content from other Web services. The authors used content-fetching and HTML-parsing mechanisms to build a general-purpose Web proxy that could be used for anonymous Web browsing, bypassing content restrictions, or denial-of-service attacks. The fact that they succeeded highlights the difficult policy decisions service providers face when designing publicly available APIs.

### *Discussion ("Maturing Malware" Session)*

The session chair, Kevin Butler, noted the common thread of software diversity in all three talks, and Will Enck noted the difficulty of managing a diverse software ecosystem. Neti drew a parallel between software diversity and financial portfolio diversity, arguing that diversifying increases the variance of the expected payout, which may be desirable in both contexts. Song pointed out that address-space layout randomization (ASLR) could be considered a successful small-scale diversity measure. Lu asserted that diversity in Web services opens up the door to unintended uses.

Asked to weigh in on where malware trends are heading, all three speakers presented different but not conflicting opinions. Song predicted that targeted malware like Stuxnet would continue to become more sophisticated, but so would run-of-the mill malware benefiting organized crime; he advocated increasing the use of sandboxing in particular. From his vulnerability-modeling perspective, Neti suggested that treating everything as compromised would yield better defensive design decisions; he further suggested that existing software repositories might improve by "intelligently" offering alternative software to encourage end-host diversity. Lu expressed his hope that open-source code and documentation would reduce the incidence of malware.

## Making Decisions

*Summarized by David Barrera (dbarrera@ccsl.carleton.ca)*

### *How to Ask for Permission*

Adrienne Porter Felt, Serge Egelman, Matthew Finifter, Devdatta Akhawe, and David Wagner, University of California, Berkeley

Adrienne began her talk by explaining how application permissions, both on desktops and smartphones, are difficult to design. Related work has found that 95% of surveyed Windows participants blindly click through UAC dialogues, and 42% of Android users don't know there is a permission screen at app installation time. Adrienne explained that when users see warnings or permissions, they can either choose to optimize or satisfice (a combination of "satisfy" and "suffice"). When optimizing, the user reads the question or prompt, retrieves memories, and decides how to proceed. When satisficing, users don't understand the question so they rely on heuristics to decide on an answer.

Next, Adrienne presented four guiding principles for good warning/permission screen design. Designers should conserve user attention and avoid repeating permission screens. This can be done by automatically granting a permission, but allowing the user to revert the setting if necessary. This works well with low-severity permissions. Another suggestion is to use trusted UI elements, such that the only way to obtain privileged data such as a picture from a camera is by pressing an OS-provided UI element. This strategy works well for user-initiated actions. Runtime consent applies to all permissions but should be used sparingly since it is habit forming and interruptive. Finally, install-time warnings work for permissions that can be granted in advance, with the caveat that they often act like EULAs.

Adrienne presented a flowchart on how to decide which kind of prompt or permission to use depending on several factors: can the action be undone? is it an annoyance? does a user initiate it? etc. The authors applied the flowchart to 76 platform-independent permissions and found that 80% can be granted automatically or with a trusted UI.

### *PeerSec: Towards Peer Production and Crowdsourcing for Enhanced Security*

Zheng Dong and L. Jean Camp, Indiana University

Dong gave a talk outlining the necessary requirements to use peer production and crowdsourcing for security effectively. Dong motivated the work by explaining that today, there is little incentive for people to apply patches or fight spam, since having a clean computer doesn't have an impact on the amount of spam that they receive. Peer production and crowdsourcing, however, have been an effective way to generate goods and services (witness Wikipedia), and this work proposes that they could also serve as useful tools in designing and implementing security systems.

Dong listed some of the guidelines that should be considered when designing a crowdsourced security system. Some include making community reputation visible to all users in the community and limiting the size and number of communities. Also it's important to define the parameters of acceptable behavior clearly and how those parameters impact an individual's reputation.

Dong went on to explain the notion "budget-based access control." Here, this type of access control is used to reduce the insider threat by assigning a "risk budget" and later deducting points from the budget depending on the number of accesses to a resource, and the associated cost for accessing that resource. Dong explained that by introducing crowdsourcing to this scheme, certain types of insiders can be more easily identified.

Although some of the ideas in this work are preliminary, the main ideas on how to build a crowdsourced security system have been outlined, and should be tested for validation.

### *Context-centric Security*

Mohit Tiwari, Prashanth Mohan, Andrew Osheroff, and Hilfi Alkaff, University of California, Berkeley; Elaine Shi, University of Maryland, College Park; Eric Love, Dawn Song, and Krste Asanović, University of California, Berkeley

Tiwari began his presentation by arguing that security experts believe that context is very important to help users make decisions. Contexts are lightweight real-life events such as a birthday party or a hallway meeting. The current app-centric security model is problematic for users, and data-centric security is too difficult to configure.

Tiwari and his co-authors present the new notion of "bubbles," which are digital counterparts to real-world contexts. Users configure bubbles based on an event or context, and for each bubble add different people and applications. Tiwari explained that in reality users would have a small number of bubbles, so setting them up wouldn't be very difficult; the more difficult and important challenge would be allowing the user to transition between bubbles.

The bubbles idea would require developers to change the way they design apps. Developers would have to think in terms of these contexts, and while many apps already fit inside bubbles, others might require substantial changes. Tiwari and his co-authors have implemented the user-level portion of the bubbles application, including creating and modifying bubbles, but have yet to try a full implementation on real users.

### *Discussion ("Making Decisions" Session)*

If the user is generally the weak point in a security system, Will Enck asked, what approaches should we be taking in terms of evaluating some of these proposals? Adrienne responded that purpose-built prototypes can be used to run small-scale pilot studies or focus groups. Not all user studies have to be large-scale. Will Enck asked about the number of bubbles that a user would have, and Tiwari responded that a field study is probably needed to get a realistic estimate. Dong added that in the case of insider threat, usability is also a concern.

Roy Maxion asked about the cognitive load associated with bubbles, and what could be done to focus on user-centered design. Tiwari assured Maxion that there is a long tail of bubbles that are infrequently used, and a few bubbles that are used every day, so he doesn't think that cognitive load would be an issue. Adrienne said that user studies are important in security, but unfortunately too many studies focus on students, and the peer review process doesn't always identify bad studies vs. good studies.

Franzi Roesner asked, given all the work that has been done in the permission space, do we now have all the tools to design new systems? Adrienne responded that one area that is problematic is where data goes when it leaves the phone. Tiwari said that they are still thinking about how to evaluate their proposal.

Roy Maxion asked what it is about the permission problem that makes it intractable? Why haven't we been able to solve it and what kind of progress has been made? Adrienne replied that there is a lack of communication between people who run the studies and people who implement the systems. Tiwari said that the space is a moving target, and as time passes, new functionality has new requirements. Maxion quickly interrupted to say that "moving target" was the oldest excuse in the book.

Will Enck asked whether we should build a system that allows failures for a small number of users. Adrienne said that permission systems have helped, but there will always be users who disregard them and intentionally make mistakes. We're currently at a stage where users don't know what kind of risk they are taking.

## Aiding the User

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

### *The Benefits of Understanding Passwords*

Markus Jakobsson, PayPal; Mayank Dhiman, PEC University of Technology

Although he "usually tries to avoid" using passwords to authenticate users, Markus Jakobsson believes we are stuck with them and had better try to understand how to assess their quality. In this work, the authors evaluated corpora of passwords with a custom parser and associate the quality of a password with its likelihood under the observed distribution.

Jakobsson repeatedly stressed that because people invent passwords and must remember them, they employ a set of simple cognitive rules, including "L33T" numeric substitution, concatenation of words, and misspellings. The authors built a parser to identify occurrences of these common rules. They found that the popularity of certain rules varied across different password corpora.

The benefit of their technique, Jakobsson explained, is a more meaningful measure of strength for user-supplied passwords. Instead of forcing users to follow abstruse and arbitrary-seeming lexical rules, a system could evaluate a password by how well it diverges from the distribution of rules in a training set. Jakobsson pointed out that their metrics can also gauge how forgettable a password is, or the likelihood that a given account compromise was due to a password guess.

### *Functional Privacy or Why Cookies Are Better with Milk*

Robert J. Walls, Shane S. Clark, and Brian Neil Levine, University of Massachusetts, Amherst

Robert Walls urged the audience to aim for "functional privacy"—the maximum amount of privacy achievable with no reduction in functionality—when designing user-facing privacy systems. As a working example of functional privacy, Walls described Milk, the authors' Chrome browser extension that restricts the spread of information via tracking cookies.

The authors observed that on today's rich Web sites, users must often choose between privacy and functionality. Existing browser extensions such as NoScript or AdBlock apply a heavy-handed approach that can break Web pages in confusing ways. Milk, on the other hand, allows sites to display ads and set tracking cookies, but it tags cookies with the domain name of the page that included them, thereby making the same user on multiple sites look like different users to a tracker. Walls suggested that analogous approaches might work to limit tracking via traveler tolling or cellular IMEI numbers.

Walls evaluated several other privacy principles through the lens of functional privacy, observing that informed consent (via reams of privacy policies) and the infamous "Do Not Track" header fall short. Walls offered a URL for the Milk extension: http://forensics.umass.edu/milk.php.

### *Discussion ("Aiding the User" Session)*

The session chair, Ian Goldberg, remarked that Jakobsson and Walls focused on different problem domains, with the adversary for Walls being an "extremely well-funded" advertising network. Walls noted that his goal was to refine a privacy principle and provide a proof-of-concept that resulted in a net improvement of privacy. Jakobsson noted that, since passwords are so widespread, different sites need to worry different amounts about password strength; small sites may not be targets. He added that organizational trust in passwords varies from site to site.

The audience's questions were primarily technical ones for the authors. Shane Clark, Alexei Czeskis, and Jelena Mirkovic pointed out, and Jakobsson agreed, that applying Jakobsson's proposed strength metrics would result in a corpus unlike the training sets. Jakobsson added that weak passwords are correlated with other weak "security hygiene," and advocated for additional scrutiny for those users. Several users asked Walls about Milk's handling of login cookies that must work across sites (e.g., logging in to example.com using a Google account); he noted that it treats login cookies specially but needs not whitelist them. Before the authors implemented this special handling, certain sites broke in a spectacular fashion.