4th USENIX Workshop on Hot Topics in Security (HotSec '09)

Montreal, Canada
August 11, 2009

SOCIAL FACTORS AND MINIMIZING TRUST

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

- ***Using Social Factors in Digital Rights Management***
  Bader Ali and Muthucumaru Maheswaran, McGill University

Bader Ali began by summarizing the current anti-piracy efforts and their weaknesses. Such efforts include the digital locking of software and hardware (DRM), legal measures (lawsuits), and reducing the availability of pirated content (content poisoning). Any anti-piracy efforts need to be considered from the perspective of all stakeholders: both the content publishers and the end users. For example, DRM fails both because it is vulnerable to hacking and because it hinders the goals of the end user.

Bader Ali continued by pointing out that part of the prevalence of piracy is due to lack of social stigma associated with pirating content or obtaining pirated content. The idea behind this project, therefore, is to leverage economic incentives and social pressure between friends to cope with digital content piracy. More specifically, the project concept is to have content publishers deliver digital content to local distributors in online social networks (OSNs).

Users form groups in OSNs. Content publishers deliver digital content to local distributors, who then sell the content to users in their groups. End users benefit because they are able to acquire content from local distributors at a reduced price. Local distributors benefit because they receive a percentage of the profit from content sales in their group. Distributors benefit because they are able to monitor the circulation of watermarked content and grade distribution groups based on their piracy rates; content publishers can then refuse to deliver content to groups with high piracy rates. The desired end result would be the reduction of piracy due to social pressure from peers in one's group, since the distributor and the other group members are punished for any content that is leaked from that group.

One audience member asked why this approach is better than having the content publishers watermark every end user's content. Watermarking for every user requires overhead, as does tracking and punishing every pirating user. This system proposes moving the punishment for piracy out of the legal realm and into the social realm—in short, by bringing anti-piracy norms into mainstream society.

- ***FaceTrust: Assessing the Credibility of Online Personas via Social Networks***
  Michael Sirivianos, Duke University; Kyungbaek Kim, University of California, Irvine; Xiaowei Yang, Duke University

Michael Sirivianos presented this workshop paper on producing credible assertions via online social networks (OSNs). The problem addressed by this work is how to gauge the truth of statements made by online personas. For example, when browsing the Web one might not know whether or not to trust that a product reviewer on Amazon is actually a doctor, as he claims he is. Other problem areas include dating Web sites, Craigslist, eBay transactions, OSN introductions, and age-based access controls.

The authors propose supporting relaxed credentials, where an assertion made by a user is bound to the probability that the assertion is true. A user posts his assertions to his profile on his OSN, where his friends can tag them as verified or rejected. The challenge here is that friends can collude and lie together; therefore, the system assigns credibility values to taggers. The authors use the Advogato trust metric [Levien et al., Security '98] and employ taggers' credibility ratings to assign a final credibility score to a user's assertion. Assertion-credibility pairs can be provided to others as a signed value produced by the credential system. If a user wants to provide a credential without revealing his or her identity, the system can use idemix (http://www.zurich.ibm.com/security/idemix/).

Someone asked how the system protects against a generally credible group of friends who lie about one thing—for example, high school students who lie about their age. This is mitigated by the trustworthiness that is assigned based on assertion type, and not based on the aggregate score; however, this situation is still problematic. Might this system foster a false sense of security? The system is only meant to produce relaxed credentials, and thus cannot be completely trusted. Another audience member suggested that users post false assertions as red herrings to help protect their privacy on the OSN. The speaker agreed with this idea and stated that posting red herrings would also help verify tagger credibility.

- ***How to Print a Secret***
  Aleks Essex, University of Ottawa; Jeremy Clark and Urs Hengartner, University of Waterloo; Carlisle Adams, University of Ottawa

Aleks Essex presented this workshop paper on how to print a human-readable secret without allowing the participating printers to reconstruct the secret. An example of this technology is cryptographic paper-based voting, where the user marks a ballot with a particular pen to reveal confirmation codes. The work involved three components: oblivious transfer, visual cryptography, and invisible ink.

Oblivious transfer is used as an alternative to a trusted dealer, and would be used to blindly divide a ballot print job between entities. Visual cryptography is used so that the end user can re-assemble the secret—the voting confirmation codes. This work also involves developing an invisible ink that color-matches the non-reactive ink that occludes the confirmation code.

An audience member asked what prevented the second printing entity from printing a ballot, revealing its secrets, and then reprinting the ballot with modifications? There are new, cheap ways to incorporate items into paper and then perform an authenticity check with a scanner. Another audience member asked if the invisible ink was indistinguishable from the visible ink under a microscope, since the authors had specified that it is indistinguishable with a black light. The authors had not yet explored that possibility.

NETWORKS AND SOFTWARE

Summarized by Akshay Dua (dakshay@gmail.com)

- ***MitiBox: Camouflage and Deception for Network Scan Mitigation***
  Erwan Le Malécot, Kyushu University and Institute of Systems, Information Technologies and Nanotechnologies

Erwan Le Malécot presented a new approach for network scan mitigation using MitiBox, a camouflage and deception system. He argued that with the growth of the Internet it has become increasingly profitable for attackers to compromise network-connected devices. Attackers use automated tools that can quickly scan large portions of the network and discover potentially interesting targets (devices with open services). This unwanted network scanning activity now accounts for a significant portion of the traffic, and Le Malécot wants to find an effective method for system administrators to deal with it.

Le Malécot claims that little seems to be being done to fight unsolicited network scans. The predominant approach is to rely on network intrusion detection systems, but using them for accurate and early detection is problematic. Le Malécot proposed a new direction which focuses on reducing the pertinence of information that is "leaked" by a network in response to scanning probes. This can be done by making the network behave uniformly, that is, the network responds in an indistinguishable fashion no matter what traffic it receives. To do so, the system proposed by the authors implements the following processing: (1) drop all malformed traffic, and (2) either drop or reply to traffic with equal probability, replies being forged as necessary. Concurrently, observed traffic sources are assigned a trust level based on their initial behavior. This level is then dynamically updated over time.

One person pointed out that any botnet member could initially behave in a way that makes it trusted and then perform the scanning activity. Le Malécot replied that the trusted status lasts only for a single connection and is therefore temporary. Another person stated that botnets have cheap resources and so they could use multiple resources to gain the system's trust. Le Malécot replied that an attacker with extensive resources at its disposal could indeed bypass certain mechanisms of the system but not all (e.g., he would still need to differentiate between forged replies and replies from authentic hosts).

- ***SPAN: A Unified Framework and Toolkit for Querying Heterogeneous Access Policies***
  Swati Gupta, Indian Institute of Technology, Delhi; Kristen LeFevre and Atul Prakash, University of Michigan, Ann Arbor

Swati Gupta presented SPAN, a system that can unify multiple heterogeneous security policies and allow them to be queried later on. Swati pointed out that when security policies from different domains interact with each other frequently, policy loopholes get created even when each individual policy is configured correctly. For example, the SSH service is configured to run on port 22, but the firewall is not made aware of that fact. Swati also mentioned that most tools today don't deal with multiple heterogeneous security policies. Thus, policy unification is left to the system administrator, who then does it in an ad hoc fashion. SPAN helps to alleviate configuration issues by automatically unifying policies from different domains: e.g., Firewall, SSH, NFS.

SPAN takes as input multiple native security policies, parses them, and stores them in an internal format based on Binary Decision Diagrams. These decision diagrams can handle ranges and make them more suitable to policies

with large domains such as firewalls. The unified policies can then be queried using an SQL-like language that also includes special statements, called "constraints," to model policy boundaries.

Someone asked, "Can you feed configuration files as is?" If SPAN supports the application, then its configuration file can be input as is. Can others in the community get involved? Swati was happy to work with them and figure out which other policies to include. Can SPAN scale to SELinux policy files? The current version of SPAN is written in Python and designed for functionality rather than speed. They will look into scalability in the future.

- ***Pre-Patched Software***
  Jianing Guo, Jun Yuan, and Rob Johnson, Stony Brook University

Jianing Guo spoke about patching software before it is released rather than after. She pointed out that patches were slow and error-prone, exposing the user to "zero-day exploits." On the other hand, including runtime checks in software involved high overhead and resulted in compatibility issues.

Jianing's solution was to "pre-patch" software by including latent runtime checks that are enabled selectively in the future. She argued that a significant benefit of this approach is that it provides an immediate response to discovered vulnerabilities without the user incurring any visible overhead until the vulnerability is discovered. Jianing presented a Memsafe prototype that included latent checks for bounds violations. A performance evaluation of the prototype indicated a 10% increase in execution time with all checks off (the default case), a 33% increase with one check on, and a twelvefold increase with all checks on.

One person asked Jianing how she differentiated her work from Valkyrie. Valkyrie used program binaries without any knowledge of the program and was very resource intensive. How can a user discover which particular check to turn on? One way would be to turn all checks on and then see which one fails; another way would be for the developer to work with the user to figure this out. Wouldn't this encourage vendors to write sloppy code? The checks were there only to help increase the quality of code. Does this method catch all bugs left over after static analysis of the program code? Their method guarantees memory safety and not type safety, but they haven't written a proof for that yet.

MOBILE AND THE USER

Summarized by Michael Sirivianos michael.sirivianos@gmail.com)

- ***Authentication Technologies for the Blind or Visually Impaired***
  Nitesh Saxena, Polytechnic Institute of New York University; James H. Watt, University of Connecticut

Since security often relies on users taking relatively difficult actions, choosing hard-to-guess passwords or timely installation of security patches, a disabled person may not be able to appropriately deal with security tasks. Attackers have actually taken advantage of the vulnerability of visually impaired persons by attacking JAWS, software for screen reading.

Rob Johnson presented this work covering current directions on user authentication for the blind and visually impaired. The first is an observation-resilient user authentication method that relies on a challenge transmitted over an audio headset and on the user performing mod 10 computation. In summary, the method encrypts a PIN with an audio challenge modulo 10 from a terminal, e.g., an ATM. The method itself has some open research issues, particularly the possibility of eavesdropping on the audio channel and the user-friendliness of mod 10 computations. Next, Johnson presented strong password management using a mobile phone. Under this family of methods, a user logs in with his cell, the terminal sends a challenge, and the cell responds by vibrating the response to the accelometer of the terminal. An open issue is to investigate the secrecy properties of vibration channel.

With respect to secure device pairing, the talk focused on the Fake-Audio attack. Blind users may be disadvantaged under such attacks because they will not be able to see the attacker. An observation is that Button-Enabled Device Authentication could protect the user because it replaces sound with vibration. The Seeing Is Believing method is also not appropriate, because the visually impaired would have trouble aiming a camera. Any pairing method that relies only on sound is susceptible to the Fake-Audio attack. The talk concluded that the only appropriate solutions are the vibration-button and vibration-vibration pairing methods.

Someone asked how realistic the audio impersonation attacks are. Rob replied that Nitesh Saxena (one of the authors) and his team are currently investigating the practicality of those attacks. Since the visually impaired usually have acute hearing, they may be able to detect the Fake-Audio attack. Could MadLibs alleviate this? Srdjan Čapkun commented that Fake-Audio attacks target devices as well as humans.

- ***Towards Trustworthy Participatory Sensing***
  Akshay Dua, Nirupama Bulusu, and Wu-chang Feng, Portland State University; Wen Hu, CSIRO ICT Centre, Australia

Akshay Dua presented his work on trustworthy participatory sensing. Traditional sensor networks have a high hardware cost of deployment. On the other hand, participatory sensing leverages user devices as sensors. For example, GPS sensors in cars can assist with predicting or detecting congestion. However, the very openness of participatory sensing makes them open to abuse. Privacy is also a concern, as users may transmit sensitive information. The first part of Dua's talk focused on abuses with respect to content integrity; how to ensure that a reported event is not a fabrication. Previous solutions to this problem employed

reputation and incentive mechanisms. Dua argued that their proposed trusted sensing peripheral (TSP) is a more appropriate solution, since any data that originates from the TSP is considered trusted.

The TSP uses a Trusted Platform Module (TPM) which offers platform attestation, i.e., sensors process the data as expected. It also offers data attestation in the form of origin authentication and verifiable data integrity. The user assigns tasks to the peripheral, and the TSP periodically responds with attested readings. The security problems that the TSP has addressed are: (1) data poisoning, since sensed data are signed by a TPM; (2) spoofing, since a burned-in private key makes it impossible to fake the origin of data; (3) collusion; and (4) the Sybil attack, since the embedded private key makes it impossible to separate identity from the device itself. There are still ways that the system could be attacked: (1) fake events—e.g., a lit candle to fake high temperature; (2) damaged sensors; and (3) effective attacks on the trusted module.

The second part of the talk focused on content protection. One possible solution would be for the sensor to encrypt the data for each individual consumer of the sensor data. Dua argued that broadcast encryption (BE) is more suitable. The TSP can also assist in content protection by providing tamper-resistant key storage, and in the future the TSP may also be able to perform BE. Their group has designed and implemented BE on Nokia N800. BE on the N800 takes only a few seconds; they also think BE with symmetric cryptography will improve performance.

Srdjan Čăpkun wondered whether there are scenarios where the attacker can fake events. Dua replied that if they use reputations and a peer-review system they may be able to detect event fakers. Is the Flec OS, which was used in this study, open source? It is not, and to acquire it one has to contact its authors.

- ***Implicit Authentication for Mobile Devices***
  Markus Jakobsson, Elaine Shi, Philippe Golle, and Richard Chow, Palo Alto Research Center

Philippe Golle listed current trends in authentication and mentioned that we are now reaching the limits of password authentication, resulting in two-factor authentication becoming more commonplace. At the same time, there has been substantial growth in the number of mobile Internet devices, which are now used to access personal, financial, and medical data. Philippe and his team performed a user study that showed that device passwords are weak; 50% of users mistype the passwords, and most users report that typing passwords on a mobile device is much harder than on standard keyboards. Users consider having to enter passwords as a more significant annoyance than the small screens of their devices.

Golle noted that proxy solutions and single sign-on solutions do not solve the problem, because they do not identify the user but only the device. Consequently, they do not account for the case of theft. Graphical passwords may have higher entropy and be more memorable but have not caught on. Biometrics have been hampered by high error rates. Golle presented their proposed solution: implicit authentication (IA). IA relies on authenticating users based on their habits and the usage patterns of their devices. For example, if a user arrives at work in the morning, the GPS says that he is in the usual location, he gets a call from his spouse and a message from his boss, he may not need to authenticate his cell phone to the bank again. Their initial steps on implicit authentication consider the following types of data: (a) location and co-location; (b) application usage; (c) biometric measurements; and (d) contextual data such as time of day, calendar entries.

The system computes an authentication score on the device. Scores computed on the device protect user privacy but do not defend against theft or corruption of device. Alternatively, the authentication score can be computed by the carrier, which is more secure but raises privacy concerns. To evaluate their insights, Philippe and his team built a Java prototype that runs on BlackBerry and Android.

Rob Johnson noted that it seems an attacker could game the system by going through a user's address book. Philippe responded that the authentication score may not just rely on phonebook/call history but also GPS, and may also decrease every time the user looks at his call history. Fabian Monrose noted that if someone snatches a phone and uses it immediately, in the absence of biometrics there are very few things this system can do. Tadayoshi Kohno wondered whether their technique can be adopted by Google and others for Web-based authentication. Philippe replied that they collect more dynamic and detailed data as mobile devices, and network providers have a lot more data than a simple user Web access profile. Eric Goldman asked whether a user can still authenticate when the authentication score decreases during the day. There is always the option to log in with a password, which also supercharges the authentication score.

SECURE SYSTEMS AND APPLICATIONS

Summarized by Akshay Dua (dakshay@gmail.com)

- ***Garm: Cross Application Data Provenance and Policy Enforcement***
  Brian Demsky, University of California, Irvine

Brian Demsky presented Garm, a system that can prevent accidental disclosure of arbitrary data and track its history even when used across multiple applications. Brian designed Garm to seamlessly support legacy applications and current data use patterns, such as protecting data even if copied to USB drives.

Garm consists of a dynamic binary-rewriter that instruments binaries under its control to track the provenance (i.e., history) of the application's state during its execution. Further, Garm implements an intermediate layer between

the application and the OS to enforce policies for, and possibly encrypt, each byte of data that is read or written by that application. A remote policy server along with a Trusted Platform Module (TPM) on the user's machine are responsible for making sure that Garm and any associated policies have not been tampered with.

Would it be important to track anything other than the sources of input that created the data, such as time? It would be easy to track the time as well. Is such fine-grained control worthwhile? Brian highlighted the advantage of fine-grained control with an email example where different emails in the inbox could potentially have a wide variety of different policies (as selected by the senders). Coarse-grained access control, on the other hand, would cause all policies to apply to all email in the inbox. He also mentioned that there was an opportunity here to optimize, since blocks of bytes would have the same provenance.

- ***Convergence of Desktop and Web Applications on a Multi-Service OS***
  Helen J. Wang, Microsoft Research, Redmond; Alexander Moshchuk, University of Washington, Seattle; Alan Bush, Microsoft Corporation

Alexander Moshchuk spoke about ServiceOS, a new operating system that treats applications as services, thus enabling convergence of Web and desktop applications. Alexander pointed out that the PC sharing model has changed from a multi-user model to a single-user, multi-application model. However, although browsers can support multiple applications, they were not designed to be operating systems for rich applications. The major differences are that browsers do not provide reliable cross-application protection, they have many vulnerabilities, they do not really manage resources such as CPU or network, and they do not provide Web applications with APIs to access devices like cameras on the system.

Alexander also argued that browsers have the right principal model, where the principal is the application rather than the user. Systems where the user is the principal are plagued with malware that can misuse the privileges of the current user. ServiceOS incorporates the best of both worlds: it leverages the principal model used in browsers and combines it with features from a traditional OS. ServiceOS models each application as a service consisting of a chain of content and content renderers (e.g., a movie is rendered by a Python movie player, which in turn is rendered by Jython, which is rendered by the JVM). Each entity in the chain can have different owners and the owner of the head of the chain is the principal. The unit of protection in ServiceOS is the principal and the unit of failure containment and resource allocation is an instance of the principal.

Had they looked into Mobile Agent Systems (MAS), which dealt with many similar issues? They had not looked into that line of research in detail but were confident that their vision of a cache-only Web-centric device was different from the MAS that was being referred to. Another person was concerned about backward compatibility. Alexander said that it was an issue, but suggested that one could start with device classes where it's easy to be backward-compatible and then move on to harder ones. What if the owner wants a user to use a particular renderer for some content? That's a challenging issue that requires carefully defining the precedence between users, the OS, and content providers in terms of who specifies the mapping from content to its renderer.

- ***System Configuration as a Privilege***
  Glenn Wurster and Paul C. van Oorschot, Carleton University

Glenn Wurster talked about creating a system configuration privilege that would be separate from the traditional root. A separate privilege could be used to prevent stealthy configuration changes and to restrict the abilities of installers. Further, he pointed out that systems are normally used for doing work and that configuration is seldom changed.

Glenn emphasized that most install procedures require the user to become superuser first, thus granting the installer unrestricted access to the file system. He mentioned that existing file-system protection mechanisms (e.g., Discretionary or Mandatory Access Control) seem to assume that the system is in a steady state—applications are not being installed or removed. His personal observation is that existing protection mechanisms can handle installs but were not designed for them. Glenn's approach is to create a new configuration privilege and assign it to a single configuration daemon. The privilege cannot be obtained or granted to any other entity. Installers can then interact with the configuration daemon to request configuration changes. However, since it is hard to differentiate between applications that are installers and those that aren't, all applications need to use the configuration daemon to make configuration changes. The configuration daemon rejects or performs configuration changes based on user input, the presence of a specific USB key, file-system state, or other criteria. This approach helps limit dangerous configuration changes regardless of whether or not the application is an installer.

What is the difference between a traditional install and one using this system? Glenn responded that the installers' requests for changes can still be rejected by the configuration daemon (e.g., if the specific USB key is not inserted). Another person suggested an alternative that would involve installing on an overlay file system, verifying the changes, and then applying it to the real file system. Glenn pointed out that for this to work one must know which applications are installers beforehand, but that can be difficult to figure out. Someone else asked how the system deals with trojans or good applications that change the configuration in an unwanted manner. Glenn replied that their system treated the two the same.