5th USENIX Workshop on Hot Topics in Security (HotSec '10)

August 10, 2010
Washington, DC

CLOUD AND WEB

Summarized by Katherine Gibson (gibsonk@seas.upenn.edu)

- ***Visual Security Policy for the Web***
  Terri Oda and Anil Somayaji, Carleton Computer Security Laboratory

Terri Oda presented ViSP, a visual security policy that builds on previous mash-up work, that she and her co-author hope will address the numerous and diverse ways in which Web site security can fail. Oda pointed out that approximately 83% of Web sites will have a security vulnerability in their lifetime, and that two-thirds have one right now. As an example, a user posting a comment on a Web site may inject code into their comment that would change the login box on that page such that if a user typed in their username and password, this information could then be exploited. As another example, advertisers may want to edit the content of the page on which their ads are displayed, perhaps negatively changing reviews of a competing product. What ViSP aims to do is to prevent attacks like these by isolating elements on a page.

ViSP is based on four tags: a box tag, which defines a region of interest; a channel tag, which is placed within a box and defines a communication channel from another box; a multibox tag, which indicates that all sub-elements should be automatically boxed; and a structure tag, which is necessary for layout but does not have any security properties. The ViSP system can be thought of as "drawing boxes" around volatile content on Web pages, not only to prevent malicious code from affecting other parts of the page, but also to prevent vulnerable areas of the site, such as logins, from being modified without authorization. As Oda succinctly put it, you "don't want sharks in your sandbox." ViSP currently has some limitations—it has no support for isolating elements without a visual representation, and it has no way to specify partial access between boxes, among others—but Oda and her co-author have released it as a JavaScript-based Firefox 3 add-on which seems intuitive to use. Additionally, the visual element of ViSP seems much more in tune with how Web designers think and is much easier to comprehend and implement, while still protecting against a wide array of attacks.

During the discussion, Lucas Ballard (Google) asked how ViSP fits in with CSS, HTML, and JavaScript. In the beta version of ViSP, it goes on after all the other components, but Oda hopes that the final version will be integrated. Ballard also asked how ViSP deals with scripts that lack a visual presence. Oda stated that there is nothing to do about those at the moment, but that a lot of non-visual scripts are tied to a visual element, giving more support for the idea that designers' minds work visually. Collin Jackson (CMU) asked how ViSP could prevent an attacker from "pushing" the boxes off the page. Oda said that ViSP would need to fix the box location to prevent this kind of attack. Finally, Adam Aviv (University of Pennsylvania) asked how ViSP assures the user that it's using the appropriate security policy, and Oda responded that you don't, but that even without ViSP most users will assume Web pages are inherently okay.

- ***Cybercasing the Joint: On the Privacy Implications of Geo-Tagging***
  Gerald Friedland, International Computer Science Institute; Robin Sommer, International Computer Science Institute and Lawrence Berkeley National Laboratory

According to Gerald Friedland, geo-tagging is cool, generates revenue, and helps to organize pictures and videos: there are over 3 million geo-tagged YouTube videos and over 180 million geo-tagged photos uploaded to Flickr. Unfortunately, people are unaware of geo-tagging, possibly

a consequence of most technology requiring the user to opt out of their content being geo-tagged, rather than opting in. Following the example of the site pleaserobme.com, which is drawing attention to the privacy lost by Twitter users when they update on their mobile away from home, the authors performed a few case studies to determine how difficult it is to use geo-tagging to determine information that most people think of as private.

For the first case study, using Twitter and tweeted pictures (which upload your location if you update via a geo-tagging-capable device, such as an iPhone), the authors "stalked" a celebrity, determining where he lived, where he walked his dog, and where his kids went to school. They then took on the For Sale section of Craigslist, using the geo-tagged pictures of the sellers' items to determine the location of the seller, which could be done to such accuracy that the address could be determined. Finally, they used a few fine-tuned search parameters to find the home locations of YouTube users who were on vacation, looking for those who had uploaded videos more than 1,000 km away from their likely home location within the past week. All of these "cybercasing" scenarios were successful for various reasons: many users don't realize that they are releasing geo-tagged information, that fast and easy-to-use APIs can pull out and search through the geo-tagged data, and that the default for geo-tagging is high-precision and enabled. In addition, services such as Google Maps allow geo-tagging information (e.g., longitude and latitude) to be easily correlated with street addresses.

In the discussion, Lucas Ballard (Google) asked Friedland if he had any thoughts on whether changing the APIs would have an effect. Friedland responded that although the high precision currently used is unnecessary, inferring can still provide a lot of information, so it is also necessary to educate users. Stuart Schechter (MS Research) questioned whether geo-tagging really provided better information than what criminals currently have access to. Bill Cheswick (AT&T Labs) half-jokingly suggested changing the location of geo-tagging information to that of the nearest police station. Finally, Adam Aviv (University of Pennsylvania) asked if there were any trends, given the widespread use of iPhones (which have geotagging on images and mobile posts turned on by default), as well as wondering whether useful information could be lost in the noise created by this ubiquitousness. Friedland responded that geo-tagging will only get more common, and that despite the massive amounts of geo-tagging data available, with the use of APIs it is surprisingly easy and fast to sift through photos and tweets.

- ***On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing***
  Marten van Dijk and Ari Juels, RSA Laboratories

Marten van Dijk's talk posits that cryptography alone is not enough to enforce privacy when dealing with cloud computing services, even when one takes into account such powerful tools as fully homomorphic encryption. Instead, van Dijk suggests the use of a nested hierarchy of three classes. The first class is private single-client computing, in which a single client's data is given to the cloud in an encrypted form such that when the cloud performs the requested function, it does not have access to the data. The second class is private multi-client computing, in which multiple clients that do not necessarily trust each other give the cloud their encrypted data, the cloud performs the requested function over all of the data, and the results are given back to the appropriate clients, without the cloud having access to the unencrypted data. The third class is private stateful multi-client computing, which differs from the second class only in that the access control policies are dependent on the full history of data a specific client has sent to the cloud. Essentially, in all of these classes, the cloud can't see unencrypted information from the clients, and the clients can't collude with the cloud to see other clients' information.

Ian Goldberg (University of Waterloo) posited that two-party obfuscation algorithms may be possible. Next, Adrian Perrig (CMU) brought an upcoming paper on how to use cloud computing securely (by Rosario Gennaro, Craig Gentry, and Bryan Parno, titled "Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers," to be presented at Crypto 2010) to van Dijk's attention. Finally, Lucas Ballard (Google) wanted to know whether the given schemes would work if a user was worried about entropy rather than cryptography. Van Dijk replied that although he was not sure, his intuition told him that it would be very difficult.

SYSTEMS AND DEFENSES

Summarized by Rik Farrow (rik@usenix.org)

- ***Popularity Is Everything: A New Approach to Protecting Passwords from Statistical-Guessing Attacks***
  Stuart Schechter and Cormac Herley, Microsoft Research; Michael Mitzenmacher, Harvard University

Stuart Schechter began by quipping that in high school, popularity is everything. He next outlined several threats against passwords by using statistical guessing: first, the password file itself being compromised, using the RockYou loss of 30 million passwords as an example; second, online dictionary attacks using the variant of statistical guessing (most popular first), and using bots in a botnet to attempt guesses to avoid lockout; third, attacks against sites that require special characters in passwords, trying simple substitutions, like "$" for "s." Schechter pointed out that many sites use restrictions on password choices, winding up with passwords like "Pa$$word1" and "blink182" (a name of a popular band that includes numbers).

Their solution is to limit the number of people sharing a given password. Rather than storing passwords, which itself is dangerous, they use four truncated hash tables with count-min structure, similar to a counting Bloom table. When their software gets a password, they hash it, truncate

the hash, look it up in four versions of the hash table, and increment all matching hash buckets. Collisions are actually desirable, as they want false positives to exist so that the lookup cannot be used as an oracle. They can use these hash tables to inform a user that the password they chose is "too popular," thus limiting the fraction of users with the same password. Popular strategies for choosing poor passwords, like including the user name, still must be applied.

Paul van Oorschot (Carleton University) pointed out that up to 100 people, the limit chosen in this work, can still share popular passwords such as "password1". Schechter countered by saying that the worst that can happen is that the attacker can compromise 100 accounts, when Microsoft is protecting many millions. Bill Cheswick (AT&T Labs) asked if they had experimented with usability, and Schechter said that they have asked permission from RockYou for their (already public) data set to seed their hashes, but have not done this yet. Someone wondered about using this for much smaller sites, to which Schechter suggested that small sites could pool their hash tables.

- ***Moving from Logical Sharing of Guest OS to Physical Sharing of Deduplication on Virtual Machine***
  Kuniyasu Suzaki, Toshiki Yagi, Kengo Iijima, Nguyen Anh Quynh, and Cyrille Artho, National Institute of Advanced Industrial Science and Technology; Yoshihito Watanebe, Alpha Systems Inc.

Kuniyasu Suzaki gave an interesting talk about replacing logical sharing of shared objects, such as DLLs and shared libraries, with deduplication of physical memory. Using DLLs and shared libraries can itself be risky, as libraries get searched for during dynamic linking. The week after this paper was presented, exploits against this very feature in Windows versions were made public. Suzaki suggested static linking of files as a defense. He also mentioned an attack against files using ELF format, by changing the Global Offset Table to point to locations of the attacker's choosing. Again, static linking solves this problem. Static linking also solves issues like dependency or DLL hell, and problems with mismatched libraries after package updates.

The disadvantage of using statically linked binary files is that they are much larger in size and require more memory when executed than programs that use shared libraries. The authors' solution is to use memory deduplication. Suzaki pointed out that this is already done in virtual machines, such as VMWare ESX, Satori, and Differential Engine. Their implementation used their own program, statifier, on binaries, and KSM (Kernel Samepage Mapping) for memory. Binary file sizes increased 40 times on average, but less physical memory was required. Also, booting is faster as dynamic reallocation overhead is eliminated. In conclusion, Suzaki said that self-contained binaries strengthened OS security by preventing some attacks, as well as avoiding dependency hell.

Nathan Taylor (University of British Columbia) wondered if the suggested use case, in the cloud or IaaS, was correct, and Suzaki said that their experiments suggest that this is the best use. Taylor then asked if this requires an extra layer in the VM, and Suzaki replied that there is no extra layer, just an extra module. There is a weakness in their approach, one that Suzaki explained during the Rump Session, in that an adversary could detect whether a particular memory image had already been loaded.

- ***Embedded Firmware Diversity for Smart Electric Meters***
  Stephen McLaughlin, Dmitry Podkuiko, Adam Delozier, Sergei Miadzvezhanka, and Patrick McDaniel, Penn State University

Stephen McLaughlin explained that this research began as a penetration test that showed that an attack that works once works everywhere. Smart meters include both limited local processing and a wireless interconnect. A smart meter can report an outage to your house, but can also be used to disconnect your power.

McLaughlin described three security concerns: fraud, that is, hacking meters to reduce the cost of electricity; privacy, as detailed load profiles can be used to infer a lot about the inhabitants of a house; and blackout exploitation, where an attack cuts off power to one or many houses.

Smart meters, so far, are almost a perfect monoculture, with identical hardware and firmware. The current meters use simple processors, with 8-bit registers, no protected mode, and no segments or MMU. Their solution is to use software diversity by encrypting return addresses, using a simple XOR and three different keys. In this scenario, failed attacks will have the side effect of causing the firmware to fail or misbehave. Stephen McLaughlin concluded by asking for suggestions, such as reducing TCB code that needs diversification. He pointed out that 10 million smart meters have already been deployed, with a planned replacement time of 30 years (and a 10-year MTBF).

Someone asked if it was possible to attack back up the chain, starting with meters. McLaughlin said that the utility servers are Windows systems, but better defended and more isolated (they communicate only to gateway servers, which collect data from smart meters). Ulfar Erlingsson (Google) suggested looking again at software-based enforcement policies: "You may be assuming some hardware support is needed, but it is not." McLaughlin repeated the need for a supervisor mode for an inline reference monitor. Erlingsson replied that software-based techniques can be used to protect the reference mode, so they could use software fault isolation.

NETWORKS AND COMMUNICATIONS

Summarized by Femi Olumofin (fgolumof@cs.uwaterloo.ca)

- ***Evading Cellular Data Monitoring with Human Movement Networks***
  Adam J. Aviv, Micah Sherr, Matt Blaze, and Jonathan M. Smith, University of Pennsylvania

Adam Aviv began by describing HumaNet (Human-to-Human Mobile Ad Hoc Network), which is a network of humans and smartphones for providing unmonitored, completely decentralized, and out-of-band communication. Unlike cellular networks, which are centrally administered and prone to monitoring and censorship attacks, HumaNet avoids centralized controls by routing messages over mobile phones, at a cost of added delays to message delivery.

The design of HumaNet assumes that the movement patterns of mobile users are regular. It leverages the return-to-home principle, which assumes that a person is likely to return to places they visited in the past. The three main concepts behind the HumaNet protocol idea are that messages are not being duplicated as they travel though the network, messages are addressed to the recipient's likely future locations, and all local routing decisions are based on the movement history of the current carrier of the message. Message routes are refined with a local timeout and a global timeout to ensure that a message does not stay too long in the network. There might be some minor flooding when a message is close to the intended recipient (last mile flooding). The protocol makes local routing decisions by considering the profile of the mobile user's movement histories and ensures the sender's anonymity. They construct a user's movement history by clustering the GPS coordinates of geographical points she has frequented in the past.

They performed an evaluation of HumaNet using trace-driven simulation on a 20-day cabspotting dataset of 536 cabs in San Francisco. In comparison to similar routing protocols, such as epidemic flooding and probabilistic flooding, HumaNet requires a fixed number of messages for successful delivery, the same as for the random walk routing protocol. In terms of message latency, 76% of all messages are delivered within one day. In terms of successful delivery, 85% of messages are delivered for HumaNet, compared to the 76.3%, 60.3%, and 28.7%, respectively, for epidemic, probabilistic epidemic, and random walk.

Aviv also identified some challenges to overcome. First, the HumaNet protocol provides best-effort routing, which raised the question of how much reliability is needed for successful message delivery. Second, HumaNet routing is subject to the same set of attacks for peer-to-peer systems. Third, HumaNet requires periodic broadcast of a mobile phone's location information. Fourth, there arises the question of whether HumaNet can simultaneously provide both sender and receiver anonymity resistant to surveillance from the cellular service. They identified a k-anonymity scheme that resists Sybil attacks as a possible solution for preventing eavesdropping attacks on messages within the network. Some of the discussion questions included the feasibility of, and the number of resources required for, a successful attack against HumaNet, and what would need to be accomplished to motivate people to participate in HumaNet.

Prateek Mittal (University of Illinois at Urbana-Champaign) expressed concern that HumaNet might be weak against interception and routing attacks on anonymity. For example, a malicious user might go around town collecting people's location information and trying to map messages to sender or hijack messages meant for specific receivers. Rik Farrow commented on the need for sender's deniability; otherwise no one will use the system. Revealing a message and the intended destination is not sufficient for the users to be able to deny that they originated a particular message. Some form of encryption might help. Aniket Kate (University of Waterloo) raised some concerns with message secrecy and DoS attack vulnerability. Another attendee noted that clustering based on the mobile user's location is not enough; there needs to be some element of timing in the clustering process. For example, it might not be okay to route messages to the home of working people during the day, because they are likely to be at work.

- ***Challenges in Access Right Assignment for Secure Home Networks***
  Tiffany Hyun-Jin Kim, Lujo Bauer, James Newsome, and Adrian Perrig, Carnegie Mellon University; Jesse Walker, Intel Research

Tiffany Hyun-Jin Kim began this talk by outlining a vision of future smart homes, enabled by a number of technology trends such as user interfaces (UIs) for "everything," network communication, digital media, smartphones, smart meters and grids, and wireless medical devices. One central security and privacy challenge in smart homes is access control management for non-expert homeowners. Poor access control management could result not only in a privacy breach for an individual or family but in direct physical harm as well.

Kim subsequently discussed some of the challenges that make smart home access control management a unique and particularly difficult task. These include diversity of visitors, complexity and diversity of devices and resources, low sophistication of administrators, and social context in which a user might not want to reveal distrust for a visitor, but the user's distrust will become visible though the home access control policy (distrust revelation problem). Kim noted that some of these challenges might have appeared in some other contexts; however, a smart home environment presents a unique combination of these challenges.

Kim described a user study that forms the basis of their preliminary policy assignment. The study interviewed 20 people (8 males and 12 females). Participants' ages ranged from 20 to 60. The interview instructions asked participants to first list eight people who visit their homes on a semi-regular basis or who are potential future home visitors. The

participants were then asked to imagine what electronics and appliances would likely be in their future homes, and to define access policy for the identified devices.

They made three observations from the interview user study. First, they were able to validate some of the challenges anticipated for the smart home's access control management, such as the users being non-expert administrators; the complexity of home environments in terms of the number, diversity, interface support, and data stored on each device; the diversity of visitors; and concerns about distrust revelation. Second, they found three types of access policies (different from the current two-dimensional allow-or-deny policies), which are sufficient for defining desired policies: presence, logging, and asking for permission. The presence policy only grants access when the user is inside the home; the logging policy maintains detailed audit logs; and the asking-for-permission policy contacts the owner when a visitor attempts to use a resource. The three policies were used to derive two others: a combination of two or of all three of the policies (hybrid policies) and the always-deny policy. Third, they found four fixed groups of access-control rights to visitors, based on the duration of relationship and level of trust. These groups are: full control (grants complete control and full access to all devices for owners, close relatives, and household members); restricted control (grants full access to resources excluding entertainment and security systems for teenagers in the family); partial control (grants full access to sharable devices, such as a home telephone, for trusted friends); and minimal control (grants restrictive access to some devices for casual visitors).

Kim highlighted two areas of further research. The first is to conduct a full evaluation of the access policies and rights with a larger set of participants. The second is to work on the identified open problems of access control management for smart homes, such as dealing with multiple administrators in the home.

Hugo Straumann (Swisscom) identified the inconvenience of an unsophisticated smart home user always having to authenticate to a smart home device before changing system settings. Nathan Taylor from the University of British Columbia emphasized the place of an emergency override during a catastrophe. For example, in the event of fire, the babysitter might not know the code to open the front door. A related issue is how to activate the emergency override. Kim commented that the smoke detector coming on could be a way of telling when the emergency override should become active. Adam Drew (Qualcomm) commented on the importance of keeping things simple. One of the last things a working homeowner would like to do is to fiddle with home access control systems after dealing with access control at work. Since the device sits in your home, why can you not simply trust it? Kim said that the interview reveals that people restrict their definition of access-control rights to fixed groups of four, which is quite manageable.

- ***Scalable Anonymous Communication with Provable Security***
  Prateek Mittal and Nikita Borisov, University of Illinois at Urbana-Champaign; Carmela Troncoso and Alfredo Rial, ESAT/COSIC, IBBT-K.U.Leuven

Prateek Mittal began by identifying the requirement for Tor clients to maintain a global view of the network before they can construct circuits for anonymous communication as one of the main problems hindering the scalability of the network. An approach that requires clients to have a partial view of the network is desirable. A number of peer-to-peer approaches have been proposed, including Morphmix, ShadowWalker, Salsa, AP3, NISAN, and Torsk. However, all of these approaches are complex, require structured topologies, and are prone to attacks because they only provide heuristic security.

Mittal proposed two alternative solutions to this problem. The first solution is a peer-to-peer scheme based on reciprocal neighbor policy where the appearance of a peer node in the fingertables of other peer nodes is reciprocal. With this policy (also known as tit-for-tat policy), if a malicious peer A de-lists an honest peer B from its fingertable, then the honest peer B also de-lists the malicious peer A from its fingertable. A client constructs a route for anonymous communication using a random walk. The client first establishes a circuit with one of its random neighbors X. Next, the client queries X's fingertable for one of X's neighbors Y and then extends the circuit to Y, through X. This process is repeated to construct a circuit of any length. They also proved that this policy allows for better random sampling of Tor's node and substantially reduces the probability of route capture attacks. They also proposed some mechanisms for securing this scheme for both structured and unstructured topologies.

Their second solution is a client-server architecture called PIR-Tor. PIR-Tor leverages private information retrieval (PIR) to overcome the need for clients to know the IP addresses of all available Tor relays. In this architecture, such addresses will only need to be stored on some of Tor's central servers (e.g., directory servers). A Tor client intending to establish a circuit would need to query a few of these Tor central servers a fixed number of times to retrieve relays. Currently, the default number of relays needed to establish a Tor circuit is three. Using PIR minimizes the bandwidth needed to privately retrieve relays and prevents malicious central servers from knowing which particular set of relays the user has chosen for circuit construction. They also described how they overcame some of Tor's restrictions with respect to choosing relays. For example, Tor requires the first relay, called the guard node, to be a stable relay with a proven record of availability. In addition, this relay should be fixed for a particular client. Since PIR provides an effective means to trade off bandwidth for computation, they proved that the computation is still practical on modern commodity

hardware. They argued that deploying PIR-Tor will enhance Tor's scalability by an order of magnitude.

Someone raised some concerns about scalability when central servers have to sign and re-sign relay information after every change in the Tor network. Mittal commented that Tor has the notion of directory servers/authorities providing blind signatures on relay information and that the Tor network carries some overhead by having clients update their global view of the system every 30 minutes. Another person raised a concern that PIR-Tor does introduce some restrictions on client circuit construction. His reasoning was that the global view of the Tor network that users normally have is now being outsourced to some central servers. Someone responded by pointing out that most users would actually prefer PIR-Tor, since they will require less bandwidth to download the relays they need to construct their circuit. Besides, the subset of users who prefer to have a global view of the system can still download the entire database of relays from the central servers. In addition, the central servers may be required to send some metadata (e.g., exit policy) on available relays to the user before the user sends any query.

CATCHING MALWARE

Summarized by Quan Jia (qjia@gmu.edu)

- ***Retroactive Detection of Malware with Applications to Mobile Platforms***
  Markus Jakobsson and Karl-Anders Johansson, FatSkunk Inc

Markus Jakobsson opened by showing a market forecast for smartphones. He argued that smartphones' surging popularity has resulted in an accelerated rise in the incidence of mobile malware. Meanwhile, newly emerged mobile malware is becoming faster, stealthier, and smarter. However, the constraint on the battery power of smartphones is preventing the use of sophisticated antivirus software. Thus, approaches different from traditional malware detection methodologies should be adopted to ensure security.

From the consumer's point of view, usability and convenience are always primary concerns. In the case of security incidents, Jakobsson suggested that the ability to revert to a previous healthy state by clicking on the "Undo" button is often desired. This goal inspired the design of their retroactive malware detection mechanism. Before presenting the technical details of their solution, Jakobsson provided three key principles for malware detection: malware must be active to block detection; malware needs to stay in RAM to be active; malware is faster than flash and radio.

Under these guidelines, he described the main steps of the proposed malware detection process. First, all programs are swapped out from RAM, while malware may refuse to swap, so that it can remain active. Then, the "free" RAM will be overwritten by pseudorandom content generated by an external verifier. Similarly, active malware will again refuse to be replaced. At last, the keyed digest of all RAM will be computed and compared at the external verifier. In addition, the verifier times each step of this process. If an abnormal timing variance occurs at any phase or a digest mismatch arises in the end, a malware alert will be triggered.

Jakobsson emphasized that detecting latency is essential to defeat malware's attempts to fool the external verifier. As far as performance is concerned, experimental results produced by a prototype system showed the ability to finish each detection process within three seconds.

Paul van Oorschot (Carleton University) asked which device decides the correctness of the digest generated and what would be the follow-up action in case of incorrect response. Jakobsson replied that the external verifier always makes the decision. When digest conflict occurs, the entire RAM would be flushed before all programs are swapped back. This cleans up the active malware. Adam Drew (Qualcomm) asked how kernel-affecting malware, for example a rootkit, could be detected. Jakobsson responded that the entire operating system is swapped out during the detection process so that a rootkit can also be exposed. Angelos Stavrou (George Mason University) asked whether event-driven malware could bypass such detection. Jakobsson said that malware of this kind makes no difference, in that it needs to be active in RAM to listen for its trigger. Finally, someone asked what measures are employed to counter phone cloning. The SIM card of each phone is used to mark its unique identification.

- ***Scalable Web Object Inspection and Malfease Collection***
  Charalampos Andrianakis, Paul Seymer, and Angelos Stavrou, Center for Secure Information Systems, George Mason University

At the very beginning of his talk, Angelos Stavrou indicated that the goal of this work is to collect URLs where malware originates. This goal is achieved by constructing a honeynet with the proposed framework that does automatic malware analysis. To build a system for this purpose, full virtualization techniques—for example, VMware ESX and Xen—are inefficient in that they are expensive and thus not scalable. Therefore, Stavrou and his team opted to design their architecture using WINE combined with lightweight virtualization. To further describe their framework, Stavrou explained that they used OpenVZ for building isolated containers. Each container is installed with a Debian Linux operating system and a modified version of WINE. An unpatched instance of IE running within a container is responsible for visiting supplied URLs and executing downloaded objects. The customized WINE installation has a built-in memory allocator that is able to detect NOP sleds. By this means, the URLs that are spreading heap-spray exploits will be identified and logged.

Stavrou then presented an experimental evaluation of their system, showing that heap-spray-based exploits can be successfully detected as expected. What's worth mentioning is that the system not only could identify known exploits but also is able to catch many zero-day exploits. Meanwhile, im-

pressive data was shown to prove the superior scalability of lightweight to full virtualization. As for the limitation of this work, Stavrou said that the current framework could only enforce heap-spray memory detection. Other exploit detection mechanisms need to be integrated with the system in the future so as to enrich its functionality.

Carlton Davis (École Polytechnique de Montréal) asked whether the framework could detect malware carried by file droppers. Stavrou answered yes and reiterated the premise that the malware should be heap-spray based. Someone asked why IE was chosen. IE is the most popular target for attacks. Were static IP addresses used for the clients, and how did the malware server react? They had the resource of an entire C class IP pool and used dynamic IP addresses for each client. This protected their clients from being remembered by a malware server. Wietse Venema (IBM Research) asked if different OSes were used to run each individual exploit. Starvou responded that WINE in different containers was configured to mimic different versions of Windows. Because of this, they were able to observe some malware adjusting their behavior to adapt to such change. The last question was about the source of new malware URLs. Stavrou said they used Google's safe URL on the one hand and extracted URLs from GMU network users on the other.