;login:
DECEMBER 2014 VOL. 39, NO. 6
ELECTRONIC SUPPLEMENT

Conference Reports

WOOT '14: 8th USENIX Workshop on Offensive Technologies

August 19, 2014, San Diego, CA

Summarized by Luke Deshotels, Rik Farrow, Grant Ho, and Rijnard van Tonder

Invited Presentation

Summarized by Luke Deshotels (alecdeshotels@gmail.com)

Practical Kleptography

Matthew Green, Johns Hopkins University

Dr. Matthew Green gave an invited presentation related to his paper titled "On the Practical Exploitability of Dual EC in TLS Implementations," which was presented at the main USENIX Security conference. This talk discussed kleptography in general and how the NSA may have abused flaws in random number generators it designed and promoted.

The first section of the talk defined kleptography as "the study of stealing cryptographic secrets securely and subliminally." It also presented the Signals Intelligence Enabling Project, which suggests that the NSA planned to sabotage cryptographic designs so the designs would subtly leak secret values.

The second section discussed in detail methods to design a kleptographic system. Dr. Green suggested that randomness is the most vulnerable point in a cryptographic system and proposed several ways to sabotage a random number generator. The method he expanded upon used predetermined parameters in an elliptic curve-based random number generator. If an organization could get others to use this system with the parameters and design they chose, they could use this system to steal cryptographic secrets.

Finally, Dr. Green also discussed the methods the NSA used to standardize and exploit this random number generator. He illustrated the standardization process and discussed obstacles the NSA had to overcome such as suspicions from watchdog organizations. He also showed quantitative results his group found when exploiting this system and discussed the future of kleptography.

Rik Farrow asked whether a fingerprint was present to allow Dr. Green's team to identify which TLS library they were exploiting for the quantitative results. Dr. Green said that the RSA library contained a fingerprint. Another audience member asked whether the NSA could have implemented these vulnerabilities accidentally. Dr. Green suggested that the NSA probably knew what they were doing. Finally, Felix Lindner (Recurity Labs) asked about hardware-based kleptography, and Dr. Green agreed that a variety of attacks are feasible.

Browsers and InterWebs

Summarized by Grant Ho (grantho14@cs.stanford.edu)

Clickjacking Revisited: A Perceptual View of UI Security

Devdatta Akhawe, Warren He, Zhiwei Li, Reza Moazzezi, and Dawn Song, University of California, Berkeley

Warren He presented his work on new forms of clickjacking attacks; this was joint work with some of his fellow researchers at UC Berkeley. Their team frames clickjacking as fundamentally an attack on a user's perception; all five of their new attacks work by manipulating or diverting a user's attention from security UI events that would otherwise alert users of the clickjacking attack. He argued that their perceptual attacks defeat many existing clickjacking defenses, including the new W3C UI defense, which requires cross-origin elements to be visible for a short amount of time before a UI event can interact with the element. However, their attack model does not include breaking X-Frame-Options because X-Frame-Options cannot be used for third-party mashup applications, such as the Facebook Like Button, that are embedded on many popular Web sites.

For their first attack, He exploits a user's corrective reflexes through a "pointer destabilization attack." When a user is about to click on a desired element, the clickjacking attack hides the real mouse pointer and displays a fake pointer image at a fixed offset away from the element the user wants to click. Naturally, the user moves his/her mouse to correct the visible, but fake pointer's location; this also causes the hidden, real mouse pointer to move the same, fixed offset to where the fraudulent click element (e.g., Facebook Like Button) has been placed by the attacker. To distract the user from noticing the appearance of the Like Button that results from the W3C visibility defense, the attack also creates several moving objects that suddenly appear on the screen whenever the real mouse hovers over the Like Button; the presence of these new, flashy elements diverts the user's attention so that the clickjacking occurs unnoticed. Their remaining four attacks abuse users' perception and motor adaptation abilities in similar ways. For example, their second attack trains a user to click on one part of the screen while focusing on another area of the screen; eventually the click area of the screen is replaced by fraudulent UI elements, but the user is unaware because he/she is focusing on a different part of the screen. An example of one of their motor-adaptation attacks is a game that trains a user to repeatedly perform a sequence of UI events (e.g., clicking on a series of moving asteroids in a game); after a short period of time playing the benign game (and learning this motor/click pattern), fraudulent UI elements are injected with the game elements along the trained click path. Because the user has trained this motor sequence and is focusing on the game, an attacker can harvest a large amount of fraudulent clicks without the user noticing.

To test the efficacy of their attacks, He recruited over 500 real users on Amazon Mechanical Turk to play his team's videogame, which conducted clickjacking attacks during gameplay. Ultimately, they found that their attacks had between 20–99% success rates. Several clarifying questions were asked about the evaluation procedure and metrics used in He's analysis. He's work used four survey questions at the end of the game to assess whether participants did not notice that they accidentally clicked on a Like Button during the game; users who completed the game and did not click on a fraudulent Like Button were tallied to see how many users the clickjacking attack was ineffective against (users who clicked on a Like Button during the game, but did not notice or intentionally click on a Like Button were considered "victims" for which the clickjacking attacks succeeded).

Some members of the audience were curious why the number of "failed" clickjacking victims in the survey did not include participants who clicked the Like Button, but who answered that they had seen the Like Button and intentionally clicked on it; He responded that this choice was similar to previous clickjacking work (Huang et al. USENIX Security '12) and that it was hard to assess what these users' motivations were and whether the clickjacking should be deemed successful since the users ultimately clicked on the fraudulent Like Button. Concluding their work, He noted that their paper discusses several defenses in detail and emphasized that this work was just the beginning of perception-based attacks on user interfaces, and future work will investigate this area more deeply.

Tick Tock: Building Browser Red Pills from Timing Side Channels

Grant Ho and Dan Boneh, Stanford University; Lucas Ballard and Niels Provos, Google

Grant Ho presented his work on how to build browser-based red pills: JavaScript code that can detect whether it's rendered in a VM or real machine; this was joint work with Dan Boneh at Stanford and members of Google Safe Browsing. A "red pill" is a piece of code that can detect if its execution environment is a VM or real machine. Prior work has extensively discussed how to build native red pills (red pills that execute directly on the operating system), but Ho's work is the first to examine how to build browser red pills: red pills embedded on Web sites that execute within the browser sandbox.

While native red pills can leverage low-level operations such as examining registers and program counters for anomalous values or checking the TLB size, browser red pills are unable to reuse these techniques because of limitations imposed by the JavaScript language and browser sandbox. Ho's work overcomes these obstacles by using performance/timing side channels to detect the presence of a VM. Specifically, they construct browser red pills by executing two JavaScript operations on a visitor's machine; one operation, the "baseline operation," gauges the background load and base performance of the visitor's machine, and the other operation, the "differential operation," measures the performance of the visitor's machine. The red pill then computes the ratio of the differential operation's execution time to the baseline operation's execution time and checks whether this ratio falls within an expected range of a real machine's time; their work uses a ratio for their red pill timing measurement in order to make the red pills robust across various hardware configurations and background activity that Internet users might have.

Testing 12 JavaScript red pills on modern versions of Chrome, Firefox, and IE, Ho noted that his team had multiple red pills that could successfully distinguish between a bare-metal machine and a VM, regardless of the choice of browser, operating system (Windows 7 or 8), and whether the VM used binary translation or hardware-assisted virtualization. Looking at potential defenses, Ho concluded that the easiest defense for honeypots might be to detect red pill code on Web sites, rather than trying to distort timing measurements in an attempt to trick the red pill into revealing its malicious payload.

During questions, one researcher was unsure whether the baseline time measurement resulted from a priori knowledge of a visitor's specific machine configuration/background load or whether it came from executing a special operation on the visitor's machine. Ho clarified that the baseline timing measurement came from a JavaScript operation, the "baseline operation," that is executed on the visitor's machine for each red pill. Karl Kurchner asked whether Ho tried constructing JavaScript red pills using cryptographic operations (in particular, using the Stanford JavaScript Crypto Library, which another one of Dan Boneh's students developed); since this was just the preliminary and exploratory step, Ho replied that only more common-use operations like rendering graphics and writing to local storage were tested, but future work might consider more computationally intensive operations like crypto. Finally, Luke Deshotels asked why Ho's team concluded that having a honeypot exit early from expensive JavaScript operations could be easily defeated by an attacker; during his discussion of defenses, Ho remarked that one (weak) defense a honeypot might try is exiting early from expensive operations in order to trick a red pill into believing that the honeypot has very fast performance, and is therefore a real user/machine. Ho sketched the following attack scenario that would defeat this honeypot behavior: one successful browser red pill was rendering complex graphics on the HTML5 canvas. In addition to recording the red pill timing measurement, a malicious Web site could fetch a specific pixel's RGB value from the final canvas; if this pixel value doesn't match its expected value, which an attacker can compute on his own, then the attacker can guess that the visiting machine is a honeypot and not a real user.

The End Is Nigh: Generic Solving of Text-Based CAPTCHAs

Elie Bursztein, Google; Jonathan Aigrain, Stanford University; Angelika Moscicki, Google; John C. Mitchell, Stanford University

Elie Bursztein presented his work on a novel machine-learning approach to breaking all major text-based CAPTCHAs used in practice; this was joint work with researchers at Google and Stanford. Existing CAPTCHA techniques consist of two steps: first, the CAPTCHA image is segmented into individual character images; next, these individual characters are fed to a machine-learning algorithm; the resulting labels are then reconstructed as the CAPTCHA solution. This popular approach stems from several excellent, existing machine-learning algorithms that can recognize individual characters; the main challenge (and strength of a CAPTCHA) lies in the attack's ability to segment a CAPTCHA into individual characters. Bursztein's novel contribution is an attack that breaks CAPTCHAs in a single machine learning step, without the need to segment CAPTCHAs into individual characters.

At a high level, Bursztein's technique discovers all possible ways to cut and segment a CAPTCHA, after which an ensemble machine-learning algorithm scores all possible cuts and chooses the final letters based on the maximum score; reinforcement learning is used to correct the algorithm when it outputs an erroneous letter during training. Their team gathered a corpus of CAPTCHAs from several prominent sites such as reCAPTCHA, Baidu, and Yahoo! for testing. Running the full algorithm that processes all possible cuts was too slow for practical usage, so Bursztein's team developed a few heuristics, such as ignoring steep cuts or nearly duplicate cuts, to decrease the number of segments analyzed; for some companies' CAPTCHAs, these heuristics led to roughly eight times fewer segments (60 times faster) at the cost of approximately 20% accuracy. Overall, their technique proved to be effective at breaking all text CAPTCHAs in their corpus with accuracy rates ranging from 5.3% (Yahoo) to 55.2% (Baidu), all of which are above the standard 1% accuracy rate for which CAPTCHA attacks are considered successful.

Concluding on a bleak note, Bursztein noted that his team could not come up with modifications to defend text CAPTCHAs against this attack. But answering a question from the audience about what the future of CAPTCHA security should look like, Bursztein advocated for a "risk-based analysis." The core idea in "risk-based analysis" is that there are heavier CAPTCHAs, such as game-based and more interactive CAPTCHAs, but these heavier CAPTCHAs are not usable enough to be universally deployed. However, by assessing the likelihood of whether a machine registering an account is a human or abuse-bot, companies might serve heavier, harder CAPTCHAs when sufficient suspicions are present. Cynthia Irvine (Naval Postgraduate Institute) then inquired about the privacy implications of risk-based analysis; Bursztein replied that in the case of Google, there were no privacy concerns because their heuristics do not use or collect personally identifying information about users.

Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks

Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz, Ruhr-University Bochum

Christian Rossow presented his work on exploiting TCP-based protocols for DDoS attacks; this work was joint research with former colleagues at Ruhr-University Bochum. While prior work has examined the use of UDP protocols for DDoS attacks, it was unknown whether TCP protocols could be exploited and amplified enough for a DDoS attack. In a canonical UDP-based DDoS attack, an attacker spoofs a small number of packets with the victim's IP to a host using a UDP protocol; the host then replies with an amplified number of packets to the victim's IP as a result of the protocol. With respect to TCP, random sequence numbers should theoretically prevent an attacker from easily spoofing a victim's IP, and single-packet responses in the three-way handshake (either SYN/ACK or RST response) should mitigate a large amplification. But, in fact, Rossow's work shows that thousands of hosts on the Internet can be exploited through TCP for DDoS amplification attacks as a result of misconfigurations or non-standard responses to a single TCP SYN packet.

For each TCP-based protocol (e.g., FTP, HTTP, Telnet, etc.), Rossow's group sent a single SYN packet to 20 million random hosts on the IPv4 address space, but they did not complete the TCP handshake with any ACK replies. From this initial scan, they determined that many hosts could be exploited for amplification attacks because of three kinds of erroneous TCP behavior: repeatedly responding to the initial SYN with multiple SYN/ACK packets, transmitting payload data along with the SYN/ACK (even though the TCP handshake hadn't completed), or sending multiple RST packets to refuse the connection. Based on this data from their initial 20 million hosts, Rossow's group then performed an Internet-wide scan (once again only sending a single SYN packet) and discovered that over five million hosts responded with an amplification factor of 20 or greater, and over five thousand hosts responded with an amplification factor of 2,500 or greater. While their follow-up experiments showed that hosts who responded with multiple SYN/ACK packets could not be exploited for amplification DDoS attacks, their real-world simulation experiments indicated that PSH and RST amplification hosts could be exploited to launch practical DDoS attacks.

Digging deeper, Rossow presented an analysis of the different types of hosts/devices that his group identified as amplifiers (hosts that could be exploited for amplification). While amplifiers that were exploited through NetBIOS had a wide diversity of underlying devices, the majority of FTP, Telnet, and SSH amplifiers were either routers or embedded systems. Fielding a question from the audience about contacting the vendors who made devices vulnerable to amplification exploitation, Rossow said that his group had reached out to several of the major router companies whose devices are vulnerable; but he noted that it naturally takes time for these vendors to fix and update currently deployed, vulnerable devices. Rik Farrow wondered whether nmap was used for OS fingerprinting. Rossow answered that they combined nmap with code they developed themselves, as nmap was limited in identifying many of the devices with problems.

Infrastructure Insights

Summarized by Rijnard van Tonder (rvantond@andrew.cmu.edu)

IPv6 Security: Attacks and Countermeasures in a Nutshell

Johanna Ullrich, Katharina Krombholz, Heidelinde Hobel, Adrian Dabrowski, and Edgar Weippl, SBA Research

Johanna Ullrich presented a systemization of attacks and countermeasures identified in the IPv6 protocol. She started off by mentioning the limitation of IPv4, which provides only 32 bits for network addresses, and hence the need for IPv6. Johanna said that IPv6 has been investigated by diverse communities—for example, academia and the corporate world—and that attacks over IPv6 come in different flavors from various sources, which makes it challenging to form a holistic idea of the type of attacks and countermeasures present in IPv6. Johanna delivered a brief background to IPv6, referring to the lack of broadcast addresses in IPv6 and simplified header format, in addition to the increased address space offered by IPv6.

Various sources of attacks were collected and described with a common language. Vulnerabilities are systemized by means of six attributes, describing their action, object, target, result, origin, and type. Johanna explained that an ARP spoof attack, for example, had a "spoof" action, of type "modification"; further details of vulnerability classification can be found in the paper. Johanna said that she believes the major future challenges of IPv6 will be address assignment and structures, securing the local network, and reconnaissance. For address assignment, she provided the example of an attacker who can still track the addresses of nodes by sniffing a unique identifier in DHCPv6. Furthermore, securing the local network is a challenge, since features such as Secure Neighbor Discovery (SeND) exist but are not deployed in practice. Lastly, Johanna talked about the inability of attackers to discover unknown nodes through brute force, due to the vast IPv6 address space. Nevertheless, reconnaissance methods are still possible through other means, such as DNS queries. Johanna closed by saying that IPv6 is not less secure than IPv4, but not exactly more secure either.

Julien Vanegue (Bloomberg) asked how IPv6 addresses multicast security, and how nodes are prevented from multicasting. Johanna responded that there is essentially nothing stopping a node from doing that. Sergey Bratus (Dartmouth College), the session chair, commented that nodes using IPv6, on the one hand, aren't supposed to talk to their neighbors (only the router), but on the other hand, there remains the issue of neighbor and router discovery. He asked whether we are going to see new ARP neighbor discovery issues. Johanna responded that she doesn't think so for the moment, and that the state of this issue will remain the same in the new protocol.

Through the Looking-Glass, and What Eve Found There

Luca Bruno, Mariano Graziano, Davide Balzarotti, Aurélien Francillon, Eurecom

Aurélien Francillon introduced the notion of "looking-glass" Web application software, tools used by network operation centers (NOC) that are useful for connectivity troubleshooting. He explained that the software is deployed on a public facing Web server, whereby an administrator on the network can interact with the router. Furthermore, the software is open source, with public lists available of looking glass software. With the goal of evaluating the security impact of this software, Aurélien disclosed that the source code revealed poor design, and contained a number of misconfigurations as well as flaws that could allow network attacks such as cross-site scripting.

Further investigation with Google Dorks revealed that looking-glass configuration files contained readable cleartext usernames and passwords, accessible over the Internet. In total, 28 configuration files were found to be exposed during the study. Moreover, Aurélien pointed out that they discovered cases of exposed SSH private keys in two instances. A vulnerability was also found in the fastping utility that allows for remote memory corruption, and was assigned CVE-2014-3931. Aurélien said that command injection attacks could also be performed on routers via HTTP, and mentioned that a shell could be obtained on a BGP router in this manner. Aurélien summarized by saying that these attacks have simple countermeasures but are not being addressed properly, and expressed concern about the number of critical systems that currently remain vulnerable on the Internet.

Julien Vanegue (Bloomberg) asked whether the researchers attempted to access the router using the exposed private key. Aurélien responded that they chose not to incriminate themselves in some way by doing so. Felix Lindner (Recurity Labs) remarked in jest that vulnerable BGP routers should be utilized to steal bitcoins, in the interest of obtaining metrics of their exposure.

Relevant URL: http://s3.eurecom.fr/lg

Green Lights Forever: Analyzing the Security of Traffic Infrastructure

Branden Ghena, William Beyer, Allen Hillaker, Jonathan Pevarnek, and J. Alex Halderman, University of Michigan

Branden Ghena started by pointing out the ubiquity of traffic lights in the world, but noted that no one has a clear indication of whether their operation is secure. From this standpoint, an investigation was conducted to determine whether this fundamental piece of infrastructure is as secure as it ought to be. Branden prefaced their results by saying that the issues they discovered were due to both road agencies and software vendors. Branden gave an overview of the components comprising a network of 100 traffic lights in an urban setting in the US. Such components include, among others, a traffic controller, a malfunction management unit (MMU), and strong 20 dBm antennas operating on two communication bands at 900 mhz and 5.8 ghz, enabling traffic lights to communicate on the network.

Branden reported that their findings found the communication bands of the wireless network to lack basic security properties. For example, although WPA could be enabled, the wireless network communication was unencrypted on both bands. Another glaring issue is the use of default username and passwords, with vendor configuration allowing authentication to the network with default credentials. Moreover, Branden indicated that there are security concerns surrounding the traffic controller, which exposes a debug port and FTP services that could be abused by attackers. Branden then explained that they were able to attack the traffic light system in a controlled exercise whereby they connected to the network and issued commands (obtained from reverse engineering), thereby changing the state of traffic lights. This ability implies that a number of attacks are possible, such as denial-of-service, traffic congestion, and individual control over traffic lights. Branden closed by saying that both the road agency and vendors should ensure basic security practices, such as network encryption and enforcing secure credentials.

Warren Kamar noted that the MMU stops an attacker from making all traffic lights green. He asked whether an attacker could bypass this safeguard by turning the lights green in quick succession. Branden responded that they had indeed tried this approach, but found that the MMU checks the direction and timing of light switches. Luke Deshotels (North Caroline State University) asked what would stop an attacker from physically accessing the traffic light box and simply installing their own malicious MMU. Branden said that physical access would indeed allow attackers to do so, or they could simply unplug the MMU. However, such an attacker may risk being observed by cameras that surveil the box. Cynthia Irvine (Naval Postgraduate School) questioned how the security of traffic light networks could be improved, since there is little economic incentive to do so. Branden responded that despite this limitation, they hope to provide simple steps so that the road agency at least starts to think about potential issues. Branden also said that the hardware for attacking the traffic light network can be affordably found on eBay, in response to Warren Kamari's question on where he might acquire such hardware.

Zippier ZMap: Internet-Wide Scanning at 10 Gbps

David Adrian, Zakir Durumeric, Gulshan Singh, and J. Alex Halderman, University of Michigan

David Adrian introduced Zippier ZMap, which allows Internet-wide scanning at an increased speed of 10 Gbps. David reflected that a year ago, the ZMap port scanner was capable of scanning at only 1 Gbps, yielding a 45-minute scan time for all of the IPv4 address space. At the outset of the investigation to improve scan time, however, it was found that the increased 10 Gbps uplink did not immediately translate into a 10x speedup in scanning. David explained that a number of ZMap optimizations had to be made in order to achieve this goal, including parallelization of address generation, efficient blacklisting of addresses, and achieving low overhead processing when sending packets.

Because network operators can choose to opt-out of being scanned, ZMap requires an efficient blacklisting mechanism. David described a tree data structure that models the IPv4 address space, allowing efficient lookups. With the parallelization of address generation and blacklisting optimization, ZMap could still only achieve 2 Gbps during scanning. The final optimization was to bypass kernel system calls when probe packets are sent, and to communicate these actions directly to the NIC hardware. With all the optimizations, ZMap was able to achieve an Internet-wide scan of the IPv4 address space in only 4 minutes 29 seconds, at approximately 95% of the 10 Gbps line speed. However, it was also found that the number of scan results at this speed decreased by 37% compared to the 1 Gbps rate. David closed by saying that the improved speed nevertheless carries the benefits of allowing a more accurate snapshot due to short scan time, faster multi-packet scanning, and large-scale detection of network vulnerabilities and exploitation.

Felix Lindner (Recurity Labs) suggested that the ZMap team look at their PortBunny work in the interest of explaining the drop in scan results at higher speed, and mentioned that there are many things that could influence this phenomenon—one way to find out would be to measure which part of the network is dropping packets. He then asked whether the team considered removing randomization during scanning to identify such bottlenecks. David responded that they did not attempt such an approach. Karl Koscher asked whether sharding a scan across multiple machines could improve speeds. David said that such sharding was not really done, since the goal of ZMap is to do it on one machine, and that no performance impact was seen due to running the scan on only one machine. Tobias Fiebies asked whether there are other ways of generating addresses, with something like maximal LSFR (linear shift feedback register). David agreed that there are other ways, but that they liked the benefits of the randomization property.

Relevant URLs: https://zmap.io, https://github.com/zmap

Embedded and Hardware Security

Summarized by Rijnard van Tonder (rvantond@andrew.cmu.edu)

Automated Reverse Engineering Using Lego®

Georg Chalupar and Stefan Peherstorfer, University of Applied Sciences Upper Austria; Erik Poll and Joeri de Ruiter, Radboud University Nijmegen

Joeri de Ruiter presented a technique for performing automated reverse engineering of a USB Internet banking device by using Lego. Specifically, automated learning techniques are employed to learn the state machine of the e.dentifier2 device, which performs banking authentication according to the EMV-CAP implementation. Joeri explained that they had previously used manual analysis to find flaws in similar devices, but sought to automate the process. The L* Mealy algorithm, implemented in the LearnLib package, was used to infer the Mealy state machine associated with the device. Joeri mentioned that such automated learning techniques are useful for other applications such as fuzz testing and model checking as well.

Joeri explained that, with the aid of a Lego robot comprising three motors, and controlled by a Raspberry Pi, they were able to perform button presses on the e.dientifier2 device. This action allowed them to drive input to the device, yielding the state machine. Joeri said that they had experienced some challenges—at times, they would see non-deterministic behavior because buttons were not always pressed correctly. It also happened that buttons could get stuck on the device after numerous presses. Nevertheless, the state machine could be successfully extracted and was converted to models which served as further input for the CADP model checker. By doing so, they could verify that a new version of the e.dentifier2 device did not exhibit a flaw discovered manually in an older version.

Joe Grand asked how well the Lego pieces coped with the strain of pushing device buttons for extended periods of times. Joeri replied that they performed surprisingly well, with no sign of wear during their testing.

Are Your Passwords Safe: Energy-Efficient Bcrypt Cracking with Low-Cost Parallel Hardware

Katja Malvoni, University of Zagreb; Solar Designer, Openwall; Josip Knezovic, University of Zagreb

Katja Malvoni introduced her hardware-cracking device by saying that airport security initially gave her mixed reactions about it, but upon hearing that it does password cracking didn't want to have any further involvement and allowed her to proceed on her journey. Katja maintains that although Bcrypt is a very resilient password hashing scheme, it is possible to derive an energy-efficient solution for cracking hashes on dedicated hardware. The implementation consists of the Epiphany manycore accelerator, which computes bcrypt hashes, and an ARM CPU running the John the Ripper password cracker.

Katja proceeded with a live demo of the cracking software, showing a speed of approximately 20,000 cracks per second. Comparing the Epiphany to x86 Intel i7 CPUs, it was found that while performance stayed the same, the Epiphany could achieve a 47x increase in energy efficiency in terms of cracks per second per watt. This is due to the lower power consumption of two watts that the Epiphany uses. Katja said that their solution compares favorably in terms of performance and cost to traditional CPU and GPU password cracking schemes, while remaining lower in terms of power consumption. Katja closed by saying that they are looking at future work in FPGA optimization, introducing more FPGAs, and reducing communication overhead.

An audience member asked how many gates are on each FPGA, and Katja responded that they didn't perform that estimate. Sergey Bratus (Dartmouth College) asked whether Katja could comment on what this work means in economic terms to a vendor who might need to perform password recovery. Katja said that one could have financial gain when password cracking is run for long periods of time, up to months. Over such a long period, their method of performing password cracking would yield the same performance as traditional CPUs, with much lower power consumption.

Printed Circuit Board Deconstruction Techniques

Joe Grand, Grand Idea Studio, Inc.

Joe Grand presented a number of techniques that can be used to deconstruct circuit boards for reverse engineering purposes. He said that the main goal for a reverse engineer would be to deconstruct PCBs layer-by-layer, which would reveal enough information to derive the original schematic and design of the board. He explained that mileage varies from method to method, and depends on the number of PCB layers, thickness, and their integration. Joe then proceeded to elaborate on the various techniques that he used to deconstruct a PCB in three areas: solder mask removal, delayering, and imaging.

For solder mask removal, Joe stated that sandpaper proved to be the cheapest and easiest method. While more automated methods like sand blasting could have great results, it requires some finesse to operate; an error during operation could damage a PCB, while this was less likely to occur with sandpaper. Joe continued by mentioning that they also used chemicals and lasers with success. For delayering, Joe found that sandpaper again worked well, as did Dremel tools. The latter, however, needed more fine-grained control than sandpaper. CNC milling and surface grinding was also effective, but Joe said that one can't always rely on a smooth planar surface. Therefore, such accurate methods can, in the worst-case, damage the board. Non-destructive imaging included 2D x-ray techniques, which Joe found to be underwhelming, particularly because the x-ray didn't reveal individual layers. Computerized tomography delivered more satisfying results, revealing separate layers on a PCB board.

Rik Farrow (USENIX) asked which features of the PCB board layer would be of interest to a reverse engineer. Joe responded that there were many things, and that it depended on the board design and potential attack. He stated that everything could be observed electrically—at a high level, one can see which components are connected, which might, for example, be useful for attacking firmware. Felix Lindner (Recurity Labs) asked whether they could achieve distance measurements with the laser and CNC methods. Joe responded that they could adjust the laser to compensate for nonpolarity, but not other methods. An audience member asked whether they had tried a chemical solution for performing PCB delayering. Joe replied that they hadn't tried that and said that chemical solutions should typically be avoided where possible due to the danger of damaging the PCB. Another audience member asked whether they had tried using a belt sander. Joe replied that they had, but that he was not as confident using those machines.

Lowering the USB Fuzzing Barrier by Transparent Two-Way Emulation

Rijnard van Tonder and Herman Engelbrecht, Stellenbosch University

Rijnard van Tonder started by reviewing previous methods of USB fuzzing, which includes both software and hardware approaches. He pointed out that pure software methods may use

QEMU emulation, and hardware methods include USB analyzers or other bespoke testing hardware. He noted, however, that these approaches all carry various limitations, including hardware devices that are not tailored for fuzzing purposes. Rijnard went on to describe the Facedancer tool, developed originally by Travis Goodspeed, which enables USB peripheral or host emulation. He then introduced the Transparent Two-Way Emulation (TTWE) technique, which enables man-in-the-middle fuzzing of USB data by using two Facedancers.

Rijnard explained that by placing one Facedancer in peripheral emulation mode and another in host emulation mode, one is able to relay data between a real host and USB peripheral. For example, a USB request would be received from a real host on a Facedancer that was emulating a peripheral. This request would then pass through a mediating computer, and be replayed to a real peripheral by a second Facedancer emulating a USB host. Rijnard added that the technique was not specific to the Facedancer, but to any device that operated like it, which requires a USB microcontroller and USB serial adapter. Endpoint hijacking, which corrects the mapping of hardcoded USB peripheral endpoints, is employed by the TTWE framework so that USB data can be relayed. Rijnard closed by sharing fuzzing results, which revealed memory corruption bugs in USB printer and WiFi adapter drivers, as well as describing the notion of USB devices as seed files for fuzzing.

Cynthia Irvine (Naval Postgraduate School) asked how a USB hub affects the operation of the TTWE framework. Rijnard replied that a USB hub would function like any normal USB peripheral, but that additional software in the emulation drivers might be required to handle extra USB devices that were attached to the USB hub. Sergey Bratus (Dartmouth College) commented that endpoint hijacking is analogous to network address translation (NAT), reinforcing that "everything is a network."

Relevant URL: https://github.com/rvantonder/ttwe-proto

Security Analysis

Summarized by Luke Deshotels (alecdeshotels@gmail.com)

Attacking the Linux PRNG on Android: Weaknesses in Seeding of Entropic Pools and Low Boot-Time Entropy

David Kaplan, Sagi Kedmi, Roee Hay, and Avi Dayan, IBM Security Systems

David Kaplan and Roee Hay demonstrated a method for reconstructing the internal state of the Linux Pseudo Random Number Generator on Android (LPRNG). This allows them to predict canary values and perform buffer overflows. This attack has been mitigated by recent versions of Android, but the majority of devices are still vulnerable. This paper builds on prior work, but it is the first end-to-end attack based on weaknesses in the LPRNG.

The attack uses malware or other methods to get a random number from the LPRNG. This is referred to as the leak value. Since the LPRNG has very low entropy during boot up, its internal state can be determined given a leak. This attack was demonstrated during the presentation with a Samsung Galaxy S4 Android device.

Felix Lindner suggested that a buffer used during boot up should have prevented the attack. The authors suggested that the mitigation mentioned might only be present on newer devices.

Security Impact of High Resolution Smartphone Cameras

Tobias Fiebig, Jan Krissler, and Ronny Hänsch, Berlin University of Technology

Tobias Fiebig discussed the potential threat of high resolution smartphone cameras being used for keylogging and capturing a user's fingerprints. Fiebig et al. took images of a user's face with the front-facing smartphone camera. These images have a sufficiently high resolution to observe a reflection of the device's screen in the user's eyeball or sunglasses. From this reflection in conjunction with prior knowledge on the keyboard layout, they could determine which keys a user pressed.

They also used the rear cameras of smartphones to photograph a user's hand while handling the device. In these photographs the user fingerprints are clear enough to create forgeries of them. These can be used to plant false fingerprints in a crime scene or to bypass biometric authentication systems. Tobias emphasized that powerful agencies such as the NSA could use techniques like these to steal credentials and biometric data from high profile targets.

Luke Deshotels (North Carolina State University) asked about indicator lights and shutter sounds that would indicate when the camera is recording. Tobias—as confirmed by Felix (FX) Lindner—replied that in order to minimize costs many phones do not implement such hardware defenses. Additionally, these features may be distracting or annoying for a user. Tobias also stated that many of these indicator lights are controlled by software and can be bypassed by malware, as demonstrated in the USENIX Security paper "iSeeYou: Disabling the MacBook Webcam Indicator LED."

Inaudible Sound as a Covert Channel in Mobile Devices

Luke Deshotels, North Carolina State University

Luke Deshotels presented his findings regarding ultrasonic sound and mobile devices. The presentation was largely limited to discussion of mobile device speakers and microphones, but the paper also discusses sound-based attacks using vibrators and accelerometers. The second slide in his presentation captured the gist of his findings. He successfully allowed wireless communication between two mobile devices using ultrasonic sound, which most humans cannot hear. This is a concern for users interested in their privacy. Even if security features restrict an application's access to radio networks such as WiFi, 4G, or Bluetooth, the application could leak data by playing ultrasonic sound with the speakers.

A combination of unexpected events and Luke's presentation style made the talk quite amusing. While Luke was explaining the function of a dog whistle, a high pitched alarm went off in the back of the room. At first everyone assumed it was part of the show. Luke assured the crowd that this was not expected, and a distressed audience member quickly ran out of the room with a shrieking phone. Jokes during the talk, and humorous questions also kept everyone's attention.

Luke's results suggest that ultrasonic sound on mobile devices allows for several practical attacks. Transmissions from one mobile device to another can survive distances of 100 feet or more. Luke also found that a mobile device in a pocket could still clearly send ultrasonic signals to another device 20 feet away. Specific attacks and solutions were covered in the slides. The most practical solution seemed to be a case with a physical barrier to block high frequency sound, but allow lower frequencies through.

A member of the audience asked about records in logcat left after producing tones on the Android devices used in the experiments. Luke had not observed these logs during experimentation, but one of the solutions he had proposed suggested doing something similar to this to detect suspicious activity. Another person commented that he was doing similar work, but could not increase the sampling rate of the microphone beyond 44,100 Hz. Luke agreed that he encountered the same limit, but could not suggest a way to increase this sample rate. Someone jokingly suggested that playing music while typing sensitive information might prevent this attack. Luke replied that background noise is not always sufficient to disrupt the ultrasonic transmissions because the noise may be of the wrong frequencies. Finally, Cynthia Irvine (Naval Postgraduate School) stated that she could hear sounds at the frequencies used in Luke's experiments until she turned 30 and suggested that Luke investigate theoretical limits of sound and other evaluation methods. Luke admitted that a few adults with exceptional hearing might be able to detect the frequencies used in his experiments. Luke also stated that her suggestions would make excellent future work, but time constraints prevented him from implementing them in this paper.

An Experience Report on Extracting and Viewing Memory Events via Wireshark

Sarah Laing, Michael E. Locasto, and John Aycock, University of Calgary

Michael Locasto presented a tool that allows viewing of memory events in Wireshark. The memory events are converted into network packets and exported to Wireshark. Treating memory activity like a network trace is a useful abstraction that allows utilization of many network monitoring tools.

To emphasize the complexity of memory events, Locasto displayed a graph representing the memory activity of an unnamed process. After a few guesses he revealed the process to be bin/true. This is an incredibly simple process that always returns true, but his graph was incredibly complex and intimidating. This tool significantly simplifies memory activity analysis.

This talk generated a lot of discussion. Felix Lindner (Recurity Labs) asked what would happen if the memory event packets were actually sent over a network and if they could be used to control other devices. Locasto replied that this sounded feasible. Rik Farrow wondered what Locasto classified as a memory event. In this project they considered reads and writes to certain areas of memory to be memory events. Sergey Bratus (Dartmouth College) emphasized that abusing memory automata has led to better memory processing features. Julien Vanegue discussed three applications for this tool including intrusion detection, reverse engineering of protocols, and vulnerability detection. Lastly, Cynthia Irvine (Naval Postgraduate School) asked whether this tool would be further discussed in future workshops. The speaker said that this was possible.

Closing Remarks

Summarized by Rik Farrow (rik@usenix.org)

Co-chair Felix (FX) Lindner answered his own question, why we do offensive research, by saying that, besides being fun, understanding offense is the number one requisite for defense. Lindner said that we really need classifications of offensive technology so that other disciplines can reference them, including the legal processes. He then displayed a slide, originally showing a nuclear-powered naval carrier group, with modified captions turning it into "Cyberwar Carrier Groups" that included "Nimitz-class exploits frameworks," with this caption pointing to the nuclear carrier. Lindner said that this example actually makes more sense than our current terminology, but if your research gets classified as "carrier-grade exploits," expect the men in black suits to come calling.