usenix ;login:

DECEMBER 2012 VOL. 37, NO. 6
ELECTRONIC SUPPLEMENT

Conference Reports

6th USENIX Workshop on Offensive Technologies (WOOT '12)

Bellevue, WA
August 6–7, 2012

Keynote Address

Summarized by Alexandru Totolici (totolici@cs.ubc.ca)

DEFCON Behind the Scene

Jeff Moss, Black Hat, ICANN CSO

Jeff Moss took the audience on a behind-the-scenes look at cybersecurity policy and practice, starting from the origins of DEFCON, and continuing on to present day Washington, DC policy-making as ICANN's chief security officer. Two decades ago, when the first DEFCON was taking shape, part of the goal was to get as many computer security-focused individuals in one place in order to encourage the open exchange of knowledge, at a time when there were no better ways to do so. To that end, DEFCON was the first open hacker conference, eschewing the invitation-only model of similar contemporary events. And, it turned out, not only security professionals and hackers were (and still are) interested in this kind of knowledge exchange, but governments were as well: intelligence and counter-intelligence agents have been spotted at DEFCON from the very early days and continue to appear year after year.

Cybersecurity is one of the hottest topics in US government circles currently, being addressed in various forms in more than 78 bills, and receiving considerable funding. The primary areas of focus are information sharing (vulnerability disclosure and exploit sharing), breach notification (which, among other things, would allow the creation of actuarial tables, of high interest for insurers), and voluntary industry best practices. There is still some confusion, however, about what cybersecurity really is: decisions regarding which community it should belong to (military versus intelligence) or whether it is an entirely new battleground rather than a vector in existing ones are not without difficulty. At the moment, the military has decided to set up USCYBERCOM to oversee cyberspace-related activities, although in many instances other government agencies may be brought in to provide assistance.

Moss went on to describe some of the international problems surrounding cyberspace policy and Internet governance, and how different countries think global control should be addressed. Whereas Western democracies are interested only in cybercrime restrictions, other countries also want the broad ability to filter out content in ways that are largely incompatible with free speech. There has also been a push to delegate the United Nations as one of the possible bodies that would take over the Internet reins from ICANN, although this means closing out industry players altogether and allowing governments to regulate a space they have not always been very comfortable (or capable) with.

In closing, Moss encouraged subject matter experts from academia and industry to involve themselves as much as possible in the policy-making process. As with the first DEFCON, there is a lot of benefit in opening up conversation between the various interested parties, and it is perhaps in everyone's best interest to ensure that this conversation is integrated into the process of policy-making.

Smartphone Insecurity

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

Abusing Notification Services on Smartphones for Phishing and Spamming

Zhi Xu and Sencun Zhu, Pennsylvania State University

Zhi Xu began by pointing out that on Android, Windows Phone, Blackberry, and iOS, developers can create custom notifications such as pop-ups, status bar alerts, and icons. Each platform provides streamlined APIs to customize each notification type. For example, app developers can choose a trigger event (such as "the screen has been turned on") that launches the notification. Developers can also choose the text, image, and subviews used in the notification. Finally, developers can choose the action or operation that occurs after the user clicks on the notification.

Xu noted that one of the main concerns for notifications is the lack of authentication. More specifically, on some platforms, any app can display a notification using the Facebook icon, but that app does not officially need to be approved by Facebook or even the Facebook app. The end result is that fabricating a message that says: "You have 1 unread message in Facebook" (and making it look real) is extremely simple on smartphones. Xu showed an attack that uses a specially crafted login screen that looks like the Facebook login screen and can steal credentials. After credentials have been recorded, the phishing app can even launch the real Facebook app.

Interestingly, Xu stressed that iOS does not allow anonymous notifications and is therefore not vulnerable, but jailbroken devices can display anonymous notifications. iOS does not allow the developer to change the icon displayed in the notification, meaning that it must be the same as the app icon, which in turn raises the bar for attackers. The authors suggest that other mobile OSes should implement similar defenses. Xu also discussed using a "SecureView," which displays a login page along with a user-chosen image to help users detect when the login page is not being displayed by the legitimate application.

During questions, someone suggested that the authors run a follow-up study to see how many users fall for these phishing notifications. Xu said that members of his lab had no problem entering their credentials in the fake login screens he created.

Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks

Ralf-Philipp Weinmann, University of Luxembourg

▶ *Awarded Best Paper!*

Weinmann started by explaining that spoofing a fake Global system for Mobile (GSM) base station and network operator is relatively easy and inexpensive because the cellular network is built on the premise that the infrastructure is trusted. Further, there is no mutual authentication between base stations and phones, meaning that phones will connect to the base station with the strongest signal. To make matters worse, phones using newer (and more secure) 4G specifications can be tricked into falling back to older specs.

Weinmann then mentioned there are only three cellular baseband (the device's radio component) makers: Qualcomm, Marvell, and Intel. Weinmann walked the audience through the different software and hardware layers in the baseband and pointed out where finding bugs and vulnerabilities would be likely. Layers 1 and 2 are regarded as relatively secure. Layer 3 (made up of the connection management, mobility management, and radio resource components) has several places where variable-length messages can be injected and used for overflows.

Using IDA Pro, Weinmann reverse-engineered the baseband files and found several types of bugs in all basebands analyzed: unchecked memory functions like memcpy(), use after free, uninitialized variables, integer overflows, and memory leaks. One particular issue was found in a Qualcomm baseband where a fixed 16 bytes value was encoded as a variable-length value, so it could be overflowed. Weinmann did note, however, that exploiting one of these bugs to do something interesting in the OS is a different story.

The speaker showed a video of an actual attack that caused a remote iPhone to answer an incoming call without user interaction. The attack was executed with a $1500 software radio running OpenBTS station, which allows injection of raw layer 3 messages to phones that connect to it.

All the bugs and exploits discussed in the paper have been reported to vendors and have been fixed.

Security Analysis of Smartphone Point-of-Sale Systems

WesLee Frisby, Benjamin Moench, Benjamin Recht, and Thomas Ristenpart, University of Wisconsin-Madison

Frisby began by explaining that there has been increased interest in audio-jack magnetic stripe readers (AMSRs). These devices allow smartphone users to swipe payment cards and transmit data to a payment processing company such as Square or Intuit.

Frisby's talk focused on attacks against the UniMag II device (used by Intuit GoPayment), but the paper discusses other attacks on other AMSRs. Frisby discussed how a malicious app could mount an attack against the UniMag II without OS compromise (or user-enabled root access). The attacks involved anything from recovering the embedded secret key to disabling it permanently and preventing further payments from going through.

The UniMag II includes a TI microchip, which has a rich development API. The authors built a custom Android app that uses the UniMag II SDK. The development API allows the retrieval of swipe data and reading/writing of settings. Most commands require no authentication, meaning that any app can interact with the reader.

Frisby's team identified a length-checking vulnerability in the getsettings() API. Sending bad data to this function caused the AMSR to reject further swipes permanently. They were also able to recover 124 bits of the 128-bit secret key and, subsequently, trivially brute-force the remaining 4 bits. Intuit has since fixed the bugs with the UniMag II reader.

During questions, someone asked how easy it is to fuzz these AMSRs. Frisby answered that it is a time-consuming task because the round trip time (RTT) for each command is about two seconds. Frisby also said that six or seven AMSRs were broken during their experiments.

Second Keynote Address

Summarized by Alexandru Totolici (totolici@cs.ubc.ca)

iOS and the Rising Cost of Reliable Exploitation

Dionysus Blazakis, TrapBit

Apple's iOS is one of the most successful current mobile platforms, in use on tens of millions of devices worldwide. As with many other proprietary platforms, an active effort is made by the vendor to lock down the operating system in order to prevent access by unauthorized applications, exfiltration of media managed via digital rights enforcement, etc. Blazakis provided an overview of the technical security features iOS has featured over time, and the ways in which they have been circumvented.

Early versions of iOS had very primitive protection mechanisms for the time (2007), lacking a non-executable heap, address-space randomizations, code signing, or trusted boot. The primary goal for circumventing Apple's device security is "jailbreaking," a process through which mandatory code signing is removed and users are given the ability to run any application they want. Blazakis went on to describe some of the security features in iOS, and the notable attacks against them.

Trusted Boot is used to verify that every stage in the device boot process is signed and its validity is trusted. This removes the ability simply to patch the kernel and update the firmware, but bugs anywhere in this code cannot be patched without new hardware that provides a new boot ROM. Attacks at this level have focused on exploiting bugs in the Device Firmware Upgrade mode in order to disable signature checks and then deploy a custom firmware.

Sandboxing is used for fine-grained system-call filtering, and in iOS it is based on TrustedBSD. Attacks have involved either simply changing the value of the sandbox enforcement variable (in earlier versions of the OS) or using shared memory to inject a return-oriented-programming (ROP) payload into an un-sandboxed process before using the (almost locked down) ptrace system call to trigger the ROP chain.

Data Execution Prevention (DEP) disables the executable stack and heap, requiring (in iOS) that all mapped executable pages are properly signed. One exploit leveraged the fact that only code must be signed, and used a dynamic library with no executable segments to store a list of initializer functions and perform a ROP attack.

Address Space Layout Randomization (ASLR) ensures the dynamic library cache is randomized on boot, and every binary is also randomly placed in memory at runtime; ROP exploits must therefore calculate their address chains using leaked addresses in order to determine the base of the dynamic library. This was exploited using a T1 font program to both gain control of the execution flow and compute the ROP payload on the interpreter stack.

All of the exploits used to jailbreak the device could have been used for much more malicious purposes, especially as some of them could be triggered by having the user access a Web page. The related security issues have been patched, and upcoming security mechanisms are going to further increase the difficulty of exploiting the device. Blazakis observed that so-called "weird machine" attacks—exploiting code and combining bugs in multiple pieces of software, such as the aforementioned font exploit—are likely to become much more commonplace due to these additional measures, making iOS more secure and, by extension, less jailbreakable.

Improving Malicious Code

Summarized by Karl Koscher (supersat@cs.washington.edu)

Microgadgets: Size Does Matter in Turing-Complete Return-Oriented Programming

Andrei Homescu, Michael Stewart, Per Larsen, Stefan Brunthaler, and Michael Franz, University of California Irvine

A limitation of return-oriented programming (ROP) is that the target program must have enough different gadgets (small snippets of existing code that can be chained together) that the desired payload can be composed from them. The key insight in this paper (presented by Andrei Homescu) is that shorter gadgets are more common, so if a Turing-complete set of small gadgets can be found, we can maximize the chance of generating an arbitrary ROP payload by searching for these gadgets.

In their approach, one-byte x86 instructions are grouped into operation classes. These instructions, together with a ret, form the smallest useful gadgets on x86 systems. Although the authors were not able to show Turing-completeness from this set of gadgets, they were nevertheless able to demonstrate building a second-stage exploit loader on non-ASLR-protected systems, which calls mmap to unprotect a page of memory for the x86 shellcode to execute from.

With two-byte instructions, the authors were able to show Turing-completeness. Finally, they evaluated the ubiquity of their approach by scanning binaries in /usr/bin of several Linux distributions for suitable sets of gadgets under a variety of scenarios (including vs. excluding libraries, Turing-completeness vs. mmap only, etc.).

Frankenstein: Stitching Malware from Benign Binaries

Vishwath Mohan and Kevin W. Hamlen, University of Texas at Dallas

NOTE: This talk was given during the Network Attack session

Vishwath Mohan presented this paper that combines return-oriented programming (ROP) techniques with malware obfuscation. The idea is automatically and randomly to transform malicious programs into an equivalent composition of code snippets taken from benign binaries, thereby fooling antivirus scanners. This approach is similar to generating ROP exploits from gadgets inside the target program except that these gadgets do not need to jump or return to the next gadget, and they can be picked from any benign program on the target system.

In their approach, malware is expressed as "semantic blueprint," which is a set of predicates in Prolog. Gadgets are discovered by mapping instruction sequences from benign executables to possible semantics. Gadgets are then assigned with Prolog, whose output is converted into an executable by a Python script.

Despite the project being in its infancy, this approach produces binaries that are a little less than double the size of the original malware, and each variant shares very little code with the others. The authors conclude that this technique is an attractive target for further development.

Invited Talk

Summarized by Rik Farrow (rik@usenix.org)

Everything You Know About Password-Stealing Is Wrong

Cormac Herley, Microsoft

At one point in his presentation, Cormac Herley stated that his goal was to rile the people in the audience. Herley did a fairly good job of this as he explained why cybercrime doesn't pay.

Herley began by lambasting estimates of the cost of cybercrime in the US by prominent federal officials, such as Robert Mueller (FBI director) and Keith Alexander (NSA director), of hundreds of billions of dollars. Herley pointed out that while getting money is easy, keeping it is not. And losses to consumers, even small businesses, are limited by regulation to $50 as long as the consumer has not been involved in the fraud. This latter point, proving that you have not transferred your own assets or given your credentials to a confederate for this purpose, is what makes getting your money back difficult. But Herley only made this point later.

As his first example, Herley described how various forms of the Nigerian email scheme often fail. The banking industry has centuries of experience dealing with fraud, and will actually pay attention if someone starts an electronic transfer of large sums of money to offshore locations. Instead, the attacker finds mules, people who will process the money for the attacker, "keeping" 10% for themselves. The trouble with this is that the mule is eminently traceable, while the attacker uses some form of anonymous money transfer, such as Western Union or Virtual Gold. In the end, the money comes from the mule, not the initial victim.

Niels Provos asked about small businesses recovering from attacks on their bank accounts, and Herley explained that establishing that someone who works for the small business hadn't been involved in the theft just takes longer. Rik Farrow pointed out that some people, such as Max Butler (subject of Kevin Poulsen's book Kingpin), had certainly made a lot of money through the theft of credentials. Herley was not aware of Max Butler, but postponed answering the question until later when he discussed credit card theft. Matt Blaze said that if fraud cost him $50 plus a lot of hassle, it would still be important to him. Herley responded with a hearty, "Preach it, brother! Broken windows have a depressing effect on economic vitality." Matt than wondered whether a better conversion rate would translate to less crime. Herley pointed out that lowering the barriers to conversion would attract more criminals.

Herley then used the example of the Cymru ;login: article (December 2006) as how the overabundance of stolen credentials has driven down prices. I found this an interesting insight, while I had considered that the large numbers of stolen credit cards and bank account information was evidence of the huge amount of theft, Herley pointed out that this is stronger evidence of just how difficult it is to monetize these thefts. Herley said that instead of Symantec's estimate of $5.3 billion in credit card theft, losses of $41 million were closer to reality.

Herley used the analysis of the Alaskan (Klondike) Gold Rush as an example: 100,000 people head for the Klondike, but only 300 people make more than $5000. While there was $50 million in gold extracted, more than $100 million was spent by gold rush prospectors buying equipment in Seattle.

The talk ended with more lively debates. Someone noted that UK banks don't give back money, but Herley countered that the UK has regulations very similar to the US. By this point, I had searched for and found those UK regulations, so I couldn't argue with Herley. William Simpson mentioned that even as the real money in the Klondike Gold Rush was made in selling shovels to prospectors, the real money in phishing and spamming is made by those selling the tools to do so. Herley wondered aloud why he would give the tools to people who will act as his competition. McCoy's PharmaLeak talk at Security the next day actually puts this into perspective, where the people selling pharming franchises actually do make a lot of money, while the people setting up Web sites and doing the spamming rarely make money—the long tail of the Klondike all over again.

Network Attack

Summarized by Karl Koscher (supersat@cs.washington.edu)

Under New Management: Practical Attacks on SNMPv3

Nigel Lawrence and Patrick Traynor, Georgia Institute of Technology

This paper, presented by Nigel Lawrence, looks at vulnerabilities in the ubiquitous SNMPv3 protocol used to manage networked devices. In particular, two attacks are described: one in which per-device keying is defeated such that a compromise of one device can compromise the confidentiality of all devices, and one in which requests can be redirected to other devices, potentially spoofing the response.

Per-device keying is supported by querying a device's snmpEngineID to generate the device's unique key. Unfortunately, this parameter is not authenticated. If one device's key gets compromised, an attacker can perform a man-in-the-middle attack to change the reported snmpEngineID to the compromised device's and thus decrypt a request sent to any device.

In the second attack, a man-in-the-middle uses network tricks (such as DHCP spoofing) to direct an SNMP request to the wrong device. Because SNMP discovery packets are unauthenticated, the SNMP manager cannot tell it is talking to the wrong device. Thus, an attacker may substitute a response from another "helper" device in place of a response from the actual target device.