;login:
THE MAGAZINE OF USENIX & SAGE
November 2001 • Volume 26 • Number 7

Special Focus Issue: Security
Guest Editor: Rik Farrow

inside:

CONFERENCE REPORTS
10th USENIX Security Symposium

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

conference reports

This issue's reports are on the 10th USENIX Security Symposium

Our thanks to the Summarizers:

Takeaki Chijiiwa
Sameh Elnikety
Kevin Fu
Rachel Greenstadt
Yong Guan
Anca Ivan
George M. Jones
Stefan Kelm
David Richard Larochelle
Ross Oliver
Evan Sarmiento
Cole Tucker
Mike Vernal
Sam Weiler

10th USENIX Security Symposium

Washington, DC
August 13–17, 2001

KEYNOTE

Web-Enabled Gadgets: Can We Trust Them?

Richard M. Smith, CTO, The Privacy Foundation

Summarized by George M. Jones

Richard Smith started off by saying that what he primarily does is "cause problems," mostly for companies that have not thought through the security implications of products that they have released. They often "discover unintended consequences that companies don't like to talk about." The three main areas they consider are security, privacy, and control.

Richard M. Smith

He stated that "consumers care more about the security of cell phones than about Web servers" because cell phones are personal devices with which consumers have immediate connections. Application developers and companies are more concerned with functionality than security. Products such as consumer devices based on real-time operating systems tend to have lower concerns for security.

Smith said that DirecTV was the first consumer device that got his interest about privacy issues. It had a phone jack. What information was it sending back? Later, a call to customer service on a different issue revealed that the customer service people were able to send commands (via satellite?) to turn his TV on. Is this a good thing? Still another time the company apparently chose to "advertise" new services by causing the TV to tune to a soft-porn channel which he had not subscribed to or selected.

Earlier this year, just before the Super Bowl, the company downloaded a program to all DirecTV boxes. The goal was to disable black market devices used to pirate programming. It succeeded. But what if they had made a mistake? What if they had disabled service for legitimate customers? Who, in fact, owns the boxes? DirecTV clearly did not own the black-market devices. Did the company's actions constitute "hacking"? Did the terms of service allow them to reprogram the legitimate boxes?

It turns out that DirecTV was not sending back "Nielsen" information, just a lot of information about the temperature inside the box. Their competitor Tivo does send in "Nielsen" info. You have to explicitly opt out by calling customer service.

We're entering a brave new world of connected devices. A company called Sports Barn sold a strap-on device that monitored your daily exercise...and then uploaded it via phone to their Web site to create a "personal profile" (which, of course, would never be used for marketing or other) purposes. One could have gotten the same effect by uploading to a PC without disclosing personal information, and there are inexpensive stand-alone devices available at sports shops that do similar things. But to maintain a record you might have to (gasp) write things on paper: the price of privacy. Oh, the company just went out of business. Many formerly happy customers now have worthless devices. Similar things could never happen with subscription software licensing, could it? Software companies never go out of business, get bought out, refocus on newer products, or have turnover or loss of support staff.

Ever considered plugging your picture frames into the phone? Kodak wants you to so that you can "register" your digital pictures. Of course, you'll pay a recurring subscription fee to do so. And they'll never share your private pictures with anyone either, their servers never get hacked, and all their employees are intimately familiar with their information security policies and actively make it their top priority each day to follow them.

Want free wireless Internet access? See the Global Access Wireless Database at *http://www.shmoo.com/gawd/*. Want to see what your neighbors and coworkers are doing? 802.11 is your friend. At least one non-USENIX conference person was observed using the USENIX wireless network at the symposium.

Convergence is a good thing, right? Fewer devices, more functionality, lower cost, but do you really want someone using the cell phone API in your combo phone/palm pilot to run a program that (1) turns off the speaker, (2) places a call, and (3) turns on the microphone? Your phone is now a bugging device, in addition to a tool for pinpointing your location at all times. Personally, I'll stick with dumb one-way pagers and only turn my phone on when I want to make a call (and announce my location).

Do you ever store personal/low-sensitivity data and business/high-sensitivity information on, say, a palm pilot, a laptop computer, or a home computer connected to public networks? Mudge and Kingpin of @stake pointed out in a later talk (dressed in bathrobes to protest their 9 a.m. speaking slot) that PalmOS has serious security problems. Cable modem providers do not generally provide security/firewall services. Laptops are routinely stolen (the laptop that was being used as the gateway/router for the conference terminal room disappeared overnight and one of the terminal-room attendants stopped someone who attempted to walk out with its replacement in broad daylight). Information security management issues that were once handled by trained security professionals in controlled, centralized environments are now the problem of your grandmother, Joe Six-pack, and your CEO.

Do you ever speed in a rental car? In the Q&A session Rik Farrow noted that at least one person has been fined by the rental car company for doing so. The company installed GPS devices in all it's cars that enable it to track down stolen cars...and tell how fast you go. Any guesses how long it will be before law enforcement and insurance companies push for legislation requiring such devices in all new cars?

Steve Bellovin noted that during the talk there had been multiple nmaps of the wireless net and an ongoing battle for address of the default router (Dug Song of dsniff fame was in the room).

And lastly, true confessions — Smith admitted that he has not turned on WEP on his own wireless net at home.

And the beat goes on...

Invited Talks

A Maze of Twisty Little Statutes, All Alike: The Electronic Communications Privacy Act of 1986 (and Its Application to Network Service Providers)

Mark Eckenwiler, U.S. Department of Justice

Summarized by Cole Tucker

The Electronic Communications Privacy Act of 1986 has a reputation for complexity. Mark Eckenwiler gave an expressive overview of the law, primarily from the viewpoint of a system administrator/provider. Basically, the act covers the relationship between, providers and customers, and providers and the government. It tries to allow for communication privacy while keeping in mind that online records are the key to prosecuting network criminals.

The law distinguishes four types of environments, based on whether the data is content or transactional in nature and whether it's being intercepted in real time or after it's been stored. The class that receives the most protection, real-time content, has a very basic rule for the normal user: don't get or look at it. For the government, the rule is nearly as simple: don't get it without a wiretap order. Providers aren't supposed to look at it unless they're in the process of protecting their rights and property. So if you're a regular user, don't run an unauthorized sniffer. If you're representing a provider, under Eckenwiler's interpretation, feel free to run an IDS or even a keystroke logger in real time; you can be proactive in defending yourself. Other exceptions are made for publicly accessible systems, such as IRC, or if all parties consent, say in a system that has a banner stating that use implies consent to monitoring. As a provider, if you have a legitimate need to monitor, there's no reason to worry.

The second class of data consists of transactional records being intercepted in real time. For providers and users the rules remain nearly the same: hands off for the latter and have a good reason for the former. The standards have been lowered for the government, so this information is essentially "less private." For access to this data, the government simply needs a court order. Examples of data that fall under this are addresses attached to incoming emails and information on where users are connecting from and whether they are online.

Next comes stored content. Eckenwiler referred to this section as "Dichotomies 'R' Us"; basically, each situation has different rules that apply, with way too many to generalize here.

Finally, there are stored transactional records. Users, hands off. Providers are

allowed to reveal this information to anyone they like, except for the government. In respect to the government, there are two classes of data: basic user data and non-contact info. Basic user data (things like name, address, and phone number) is accessible with a subpoena, and thus not strongly protected. Everything else requires a 2703(d) warrant to access, but providers can be sent a court order requiring they hold on to the data for a specified amount of time, usually in expectation of a warrant being served in the near future.

Loaning Your Soul to the Devil: Influencing Policy without Selling Out

Matt Blaze, AT&T Labs-Research

Summarized by George M. Jones

Matt Blaze commented on the public debate over cryptology that's taken place over the past 10 years or so. He included amusing stories of "hacker tourism," including nine cryptography experts all independently trying to score "cool" points by stealing stationery from secret congressional briefing rooms and NOT opening a red folder marked "TOP SECRET: President's Daily National Security Briefing" when left alone in a conference room in the old executive office building.

What can a scientist/techie contribute to the public policy debate? His main advice is "stick to what you know" (science/technology). "You are listened to because people believe you have objectivity.The basic purpose of science and engineering is to expand understanding of reality/truth, with no compromises." You are not there to comment on philosophy, politics, or constitutional law.

He gave an amazingly insightful list of the contrasting values of science and politics):

- Science is interested in finding truth. Politics is about balancing interests.
- In science, people are rewarded for new discoveries. Disruptiveness is considered good. In politics, people are rewarded for making other people happy. Disruptiveness is considered bad.
- In science, uncompromising people are admired; in politics, uncompromising people are considered fools.
- In science, "honesty" means admitting mistakes; in politics, it means keeping promises.
- In science, challenging someone shows interest; in politics, a challenge is an attack.
- In science there is no "dress code"; in politics, even suits can be considered "casual" (and thus cause you not to be taken seriously).

The policy options range from discouraging/forbidding its use; allowing limited strength crypto; allowing use of strong, modern cryptographic methods; and encouraging use. In the last few years the US has moved mostly from the first to the third stage.

Matt Blaze

The tone of the debate has also changed and includes more actual dialogue. We no longer have one side yelling, "You're a bunch of long-haired hippies," and the other yelling, "You're a bunch of jack-booted thugs." Now it's just "you're a bunch of hippies" vs. "you're a bunch of thugs." See, for instance, "Thou shalt use skipjack/clipper" vs. the process for selecting AES.

"Washington, D.C. is another planet, a closed system." "Much of what happens here is for show." "Any meeting with a policy maker involves a little conspiracy to make each other feel important." "Meetings with congressional staffers always end with the question 'What do you suggest we do?'" Stick to what you know.

Cops Are from Mars, Sysadmins Are from Pluto: Dealing with Law Enforcement

Tom Perrine, San Diego Supercomputer Center

Summarized by Ross Oliver

Tom Perrine described some of his experiences with law enforcement people and discussed his recommendations for other sysadmins who may need to interact with law enforcement.

Like system administration, law enforcement is a culture as well as an occupation, with its own lingo, inside jokes, etc. There are also many different law enforcement agencies: federal, state, city, county, military, and customs. Even schools and universities often have their own police force.

Throughout the talk, Perrine emphasized the importance of trust in individuals rather than organizations. Just as in any large organization, there are "clueful" and not "clueful" members, and building personal relationships is key. Also realize that the goals and priorities of law enforcement may be different from yours.

Because they are "agents of the government," law enforcement officers have many legal constraints on their actions that may not apply to private citizens. Sysadmins can take advantage of "ISP exemptions" in the law to take "any steps necessary to protect the communications system."

Perrine recommends that sysadmins become familiar with applicable laws (both federal and state) before the need to apply them arises. Advice of qualified legal counsel is strongly recommended. Also, make sure your organization's policies are suitable, and adhere to them during any investigation.

READING BETWEEN THE LINES: LESSONS FROM THE SDMI CHALLENGE

Summarized by Rachel Greenstadt

Scott A. Craver, Min Wu, and Bede Liu, Princeton University; Adam Stubblefield, Ben Swartzlander, and Dan S. Wallach, Rice University; Drew Dean and Edward W. Felten, Princeton University

Program Chair Dan Wallach introduced this talk as being a long time in the making and mentioned that he was pleased to have it here. However, he stressed that this first section would be a normal, boring technical talk. THEN there would be a panel discussion where policy questions would be allowed. Matt Blaze asked when the subpoenas would be served; however, despite the large mass of press and lawyers that joined the USENIX attendees, there was no last-minute withdrawal of the talk this time, and no FBI agents came to cart Scott Craver away as he gave his talk.

Craver began by describing the challenge, which took place during three weeks in September and October of 2000. SDMI (Secure Data Music Initiative) invited "hackers," otherwise known as the general public, to crack six of their proposed technologies labeled A through F. There were four watermarking technologies and two others. SDMI offered a cash prize for the successful defeat of one of their technologies, but this required the winners to sign a Non-Disclosure Agreement, so the Felten group decided to forego the prize in favor of publishing their findings.

SDMI is an organization, an initiative, and the technology for that initiative. At the time of the challenge, that technology was watermarking and related technologies. The watermarks (technologies A, B, C, and F) were composed of a robust and a fragile component, the robust part of which would survive altered music. Through a missing watermark in the fragile component, such as if it had been mp3 compressed, an SDMI device could perhaps determine if a CD track was ever an mp3 in the past, perhaps illegally downloaded. The other technologies (D and E) were used to sign tables of contents, supposedly to control the propagation of CDs with mixed tracks.

For the watermarking technologies there were three samples given: (1) a sound clip without a watermark, (2) the same sample with a watermark, and (3) a different sound clip with a watermark. The challenge was to remove the watermark from the third sound clip. SDMI provided no actual embedders or detectors. There was an online oracle to which you could submit a sound clip and get a response. There was no description of the algorithms used, and no details or reasons were given when an oracle rejected a clip. The challenge lasted only three weeks and the oracle had a turnaround time in hours. As such, adaptive oracle attacks, which would be possible if the system were deployed, were not feasible.

There were several approaches used against the marks: (1) brute force attacks not specific to the algorithm used and which mostly consisted of adding noise and filtering, (2) slight brute force attacks loosely based on supposed details of the algorithms, and (3) full-blown reverse engineering.

For technologies B and C, the group noticed that there was a narrow band signal added to the clip. By the slightly brute method of filtering at the frequency and adding narrow-band noise they were able to foil the oracle.

In their analysis of technology A, the group noticed a slight warping in the time domain as though the signal was slowly advancing or decreasing. They determined that this phase shifting was pre-processing and not the actual watermark, since the oracle did not admit the sample when the distortion was removed. However, removing this distortion in technology F was able to make that watermark undetectable (quick, somebody call the FBI).

Another approach to defeating technology A would have been to try reinstating the fragile component. However, there was no way to test this type of attack using the oracle.

The group noticed a ripple in the frequency domain, which led them to believe that technology A used some sort of echo hiding technique consisting of deliberate but inaudible echoes, which meant that there was a signal which was delayed and then added back into the music. They tried a filtering approach to reduce the audibility of the echo sufficient to remove the watermark. Wanting to discover more, they decided to do a patent search figuring correctly that this was a proprietary algorithm with a patent. They found a patent belonging to Aris corporation which became Verance, one of the SDMI companies. This made them feel like they were on the right track. They also discovered that it was a simple echo every fiftieth of a second and that a delayed version was added or subtracted every fifteenth interval. To further analyze the signal they used the auto-kepstral technique for echo hiding, combining techniques to estimate the echo. They've come up with better echo hiding detection software subsequent to the challenge. Scott demonstrated a program that was color coded to detect the echo.

For technologies D and E, SDMI presented table-of-contents files for 100 CDs and signature tracks. The challenge was to create a new table of contents and successfully forge a signature for it. For technology D they found that all the energy was concentrated in a small frequency band of 80 frequency bins which only actually used a 16-bit signal

repeated five times with constant shuffling. Since there were only 16 bits of output, a user should be able to acquire many authenticators, as there were two hash collisions among the CDs given. However, it was difficult to get further than this analysis because the oracle for D didn't work; it would always return "invalid" regardless of input. Technology

Q & A: Scott Craver & Dan Wallach

E, however, didn't have any data to analyze at all. You could submit a mail saying you'd try mixing this track and that track, and you'd get a reply saying that you couldn't do that.

The speaker concluded by saying that many claimed that this was a system to "keep honest people honest." However, though the Felten group felt that the system was too complex for that, they wouldn't claim any type of strong security. The systems require trusted clients in a hostile environment, but if deployed they would be broken quickly. No special EECS knowledge is needed and there are no dirty secrets. Anyone with reasonable expertise could do this. Watermarking can be useful but not in this situation. The weakness is in the overall concept, not the specific technology. One main lesson learned is that security through obscurity STILL doesn't work. This is particularly the case for secret algorithms which are patented and therefore public.

Peter Honeyman asked about the possibility of a secure watermark. Scott replied that he personally thinks that watermarking won't work for actively enforcing a usage policy since doing this provides all targets an oracle that they can use. He is pessimistic about the use of watermarking for copyright control. He clarified that they broke, according to the oracle, technologies A, B, C, and F, but that D and E had no valid responses. He also clarified that only technology A used echo hiding, and that though they don't know what the criteria for the oracle was, it appeared to make a decision based on detectability and quality. He explained that some areas where watermarking might prove useful is in fragile watermarks which provide tamper evidence in digital photographs and in preventing duplication of currency. These technologies have a different threat model. Someone asked about copy protected CDs; Scott replied that that was a completely different approach done entirely at the hardware level. People wondered why honest people would not want a complex copy protection scheme; Scott answered that complex schemes have higher rates of failure and higher cost. Someone asked how this was relevant to detecting steganographic information and Scott answered that they were basically the same and that the information about echo detection would be useful.

Panel Discussion on SDMI/DMCA

Moderator: Dan Wallach, Rice University; Panelists: Edward W. Felten, Princeton University; Cindy Cohn, EFF; and Peter Jaszi, American University College of Law

The three panelists spoke about the legal and social questions surrounding the SDMI/DMCA issue. Dan Wallach mentioned that if there were any representatives from the record company, the panel would love to have someone from the other side come speak; he doubted, however, that they would be here.

Peter Jaszi then presented a detailed description of the Digital Millennium Copyright Act (DMCA), section 1201. He explained the difference between the DMCA and copyright law. Copyright law has been developed and refined over a few hundred years and maintains a delicate balance between owners and users' privileges. To that end it has been relatively successful. It is important to understand that the DMCA is not copyright law but, rather, a supplement to copyright law or para-copyright legislation. As such it has the potential to override the copyright default protections which had been carefully laid out over time. He sought to explain these overrides and mentioned that the risk the DMCA poses to the fundamental copyright system isn't news and wasn't news when it was passed. As a result some limitations to the DMCA were built in, but most of these exceptions are not very functional.

Prof. Peter Jaszi

The fundamental commandment of the DMCA is "Thou shall not circumvent for access." The fact that it was access and not use was a compromise intended to limit the DMCA. However, it limited the legislation less than some imagined it would since there is a great deal of confusion between access and use. There are also secondary prohibitions concerning making goods and services which can be used for circumvention available. Section 1201(b)(1) can be interpreted broadly, and it was under this provision that the threats from SDMI to the authors of the paper were made.

Section 1201(c) presents a fair-use exception in wonderful ringing language, however, it is completely irrelevant since it references fair use as a defense of copyright and the DMCA is not copyright.

The law enforcement exception is actually sweeping and robust; it applies to all the provisions of the act. The reverse engineering exception is not half bad; it refers to the whole range of prohibitions although it is still narrower in scope than the protections under copyright law. Sections 1201(g) and (j) present limited exceptions for encryption research and security testing which are uncertain in scope. Section 1201(h) presents a small but robust exception to allow adults to circumvent in order to frustrate a minor's attempt to achieve privacy in a Web environment. Section 1201(i) allows ordinary people to protect their privacy, but it is only a conduct exception; you need to make your own tools and not distribute them. There is less to all these limitations and exceptions than meets the eye.

There are risks posed by this legislation to the traditional balance of interest in copyright law, which calls for a pushback against legislative excess. To this end Jaszi is forming a new access coalition. They have a Web site at *http://www.ipclinic.org*.

Cindy Cohn from the EFF said that Peter had already said everything about section 1201 but stressed that the EFF was "pushback central" and explained ways in which people could get involved in this effort. The EFF has been involved in this issue even before the cases involving *2600 Magazine*, Felten and the USENIX presentation of the SDMI paper, and the California trade secrets case.

Thomas Greene from the *Register* wondered why the mainstream press hasn't realized their stake in this and what it implies about freedom of the press. Cindy replied that they were getting increased press support with the Skylarov arrest; speaking speculatively, she also mentioned that the mainstream press is owned by content holders. In addition, most press organizations wish to be seen as nonpartisan and objective.

Someone asked about dual use technologies, such as echo detection. The DMCA takes this into account but the language doesn't give much comfort. There are a series of criteria which will give you liability.

There was a question about potential connections between the lawsuit and the Skylarov case. Cindy answered that in strictly legal terms there was no overlap.

Someone asked what to do in Felten's situation, what lessons had they learned? Felten responded that they learned a great deal responding to the threats regarding the paper. He said to talk to people who've been there and keep in mind your goals and values.

Someone brought up the question of whether a person would be at risk for summarizing the session. Cindy said that the letter they received only pertained to the particular paper, but because the paper can be published, prosecuting for summarizing would be hard. Peter suggested that if you were going to synthesize the talk (uh...this is starting to sound disturbingly familiar...) and discuss strengths and weaknesses, theoretically you could be in trouble. Especially if you implement something based on the presentation. Felten mentioned that the fact that this question has no simple answer is telling.

Someone suggested widespread civil disobedience as the only way to effect change. Cindy responded that she never advises people to break the law. Though she feels that if the law is out of step with what people believe their rights are, the law should be changed. Peter added that copyright law has functioned well based on shared social investment. Like the tax code, it works not because it is policed but because there is a high degree of collective buy in to its premises. The most corrosive thing about the DMCA is that the basic assumptions it makes about people are dark and pessimistic. We need to question those assumptions and what flows from them.

Someone asked for some insight into why the industry wouldn't want this research since it would allow them to build better protection schemes. Felten responded that to us the question is, is this technology weak? We didn't make it weak, and we think it should be fixed. The industry's concern is not whether the technology is strong or weak so much as whether people believe it is strong or weak. They think that if the public reaches a consensus that the technology is strong, that will be enough. Many of us find this hard to understand.

[More information and photographs can be found at *http://www.usenix.org/events/sec01/index.html*

Changes in Deployment of Cryptography, and Possible Causes

Eric Murray, SecureDesign

Summarized by Takeaki Chijiiwa

A survey of cryptography deployment was conducted last year (2000) by Eric Murray, and a similar survey was conducted in 2001 to measure changes in the deployment of SSL (Secure Socket Layer) and TLS (Transport Layer Security) Web servers.

The results of the 2000 survey showed 10,381 unique hostname and port number combinations compared to 12,630 in 2001. Detailed results are available at *http://www.lne.com/usenix01*.

There were several noteworthy changes between the results from the surveys in 2000 and 2001:

What got better?

- A 14% increase, 5% decrease, and 8% decrease among servers categorized as Strong, Medium, and Weak, respectively.

- The number of servers supporting 1024-bit key size increased by 10% while a decrease of 8% was seen for support of less than 512-bit key size.
- The protocol adoption saw a shift from SSL v2 (3% decrease) toward TLS (5% increase).

What got worse?

- The number of expired certificates increased from 3.1% to 3.7%.
- Self-signed certificates increased from 0.8% to 2.0%.

The results presented raised many questions from the audience.

Question: Why do you think there was an increase in the number of self-signed certificates?

Answer: This may be due to people playing around with OpenSSL, or the survey may have picked up servers used for internal use. Furthermore, the increase in the number of expired certificates may have been a result of study error and/or the inclusion of abandoned Web sites.

Question: Did you retest the servers from last year's survey?

Answer: No. This was a new list and, therefore, a completely new survey.

Question: Is the raw data available?

Answer: You can email ericm@lne.com for private requests.

Question: Which browsers do you use for personal use?

Answer: Linux and Netscape.

Reversing the Panopticon

Deborah Natsios, cartome.org; John Young, cryptome.org

Summarized by Mike Vernal

Deborah Natsios described the mission of cartome.org and cryptome.org as an attempt to reverse the one-way flow of information controlled by the national surveillance state. Based upon the assumption that information is power, Natsios likened the work of cartome.org and cryptome.org to that of Ariadne in the myth of Theseus and the Minotaur. By reversing the flow of information, cartome.org and cryptome.org hope to empower those who may be caught in the labyrinth of the security state, much as Ariadne empowered Theseus with a trail of silk thread through the labyrinth of Crete.

John Young continued by explaining that cryptome.org welcomed the submission of proprietary or classified documents and trade secrets from any nation or corporation. Young described a few such documents and the unfavorable responses they had received. The British government objected to one document and attempted to have cryptome.org's Internet service provider shut the site down. Another document prompted diplomatic requests from the Japanese government for its removal. All attempts to shut the site down have thus far been rebuffed, but Young imagines that someone will eventually be successful.

Other information cryptome.org has received and published include proofs that American corporations used US intelligence to stay ahead of foreign competitors, the names of over 8,000 CIA informants, and, currently, the programs and keys associated with Russian programmer Dmitri Skylarov's crack of Adobe's E-book system, for which he was arrested in July.

An audience member asked what types of material cryptome.org would not publish. Young explained that cryptome.org is open to any kind of publication, but they have refused to publish child pornography documents and information related to biological warfare. They also feel that personal prerogative takes precedence over the public's right to know, so they will remove personal information and documents if requested by the person in question. They also reminded the audience that they do not verify the authenticity of the information they publish – they leave that to the interested reader.

Young repeatedly stressed what he believed to be the transitory nature of cryptome.org. He assured the audience that at some point cryptome.org will either be silenced or it will simply mature away from the cutting edge. When that finally happens, Young is confident that someone else will emerge at the vanguard of the quest to reverse the Panopticon state.

Designs against Traffic Analysis

Paul Syverson, U.S. Naval Research Laboratory

Summarized by Yong Guan

Paul Syverson used a pseudonym, "Peter Honeyman," on his talk, a joke which pervaded the rest of the conference.

Although the encryption of network packets ensures privacy of the payload in a public network, packet headers identify recipients, packet routes can be tracked, and volume and timing signatures are exposed. Since encryption does not hide routing information, public networks are vulnerable to traffic analysis.

Traffic analysis can reveal, for example, who is searching a public database, what Web sites are surfed, which agencies or companies are collaborating, where your email correspondents are, what supplies/quantities you are ordering and from whom, and so forth.

Knowing traffic properties can help an adversary decide where to spend resources for decryption and penetration. Therefore, it is important to develop countermeasures to prevent traffic analysis.

The security goal of traffic-analysis-resistant systems is to hide one or more of the following:

- Sender activity: that a site is sending anything
- Receiver activity: that a site is receiving anything
- Sender content: that a sender sent specific content
- Receiver content: that a receiver received specific content
- Source-destination linking: that a particular source is sending to a particular destination
- Channel linking: identifying the endpoints of a channel

Some systems were described:

Dining Cryptographers (DC) – networks, in which each participant shares secret coin flips with other pairs and announces the parity of the flips the participant has seen to all other participants and the receiver.

Chaum mixes – a network of mix nodes, in which messages are wrapped in multiple layers of public-key encryption by the sender, one for each node in a route. Most widely used anonymous communication systems use the Chaum mix method.


Paul Syverson

There are two kinds of routes for the messages: mix cascade, where all messages from any source move through a fixed-order "cascade" of mixes, and random route, where the route of any message is selected at random by the sender from the available mixes.

Remailers, mainly used for email anonymity, employ rerouting of an email through a sequence of multiple mail remailers before the email reaches the recipient, so that the true origin of the email can be hidden.

Anonymizer and SafeWeb provide fast, anonymous, interactive communication services. They are essentially Web proxies that filter out the identifying headers and source addresses from Web browsers' requests. Instead of the user's true identity (e.g., IP address), a Web server can only learn the identity of the Web proxy. Both offer encrypted links to their proxy (SSL or SSH). Anonymizer is a single point of failure, whereas SafeWeb is a double point of failure. SafeWeb offers additional protection from censorship.

Crowds aims at protecting users' Web-browsing anonymity. Like Onion Routing, the Crowds protocol uses a series of cooperating proxies (called jondo) to maintain anonymity within the group. Unlike Onion Routing, the sender does not determine the whole path. Instead, the path is chosen randomly on a hop-by-hop basis. At each hop a decision is made whether to submit the request directly to the end server or to forward it to another randomly chosen member according to forwarding probability. The expected path length is controlled by the forwarding probability. Cycles are allowed on the path. The receiver is known to any intermediate node on the route. Once a path out of a crowd is chosen, it is used for all the anonymous communication from the sender to the receiver within a 24-hour period. Crowds does not have a single point of failure and is a more lightweight crypto than mix-based systems. However, Crowds has limitations: all users must run Perl code, users have to have long-running high-speed Internet connections, an entirely new network graph is needed for a new or reconnecting Crowd member, connection anonymity is dependent on data anonymity, and responder protection is weak.

Onion Routing provides anonymous Internet connection services. The Onion Routing network operates on top of existing TCP/IP networks such as the Internet. It builds a rerouting path within a network of onion routers, which in turn are similar to real-time Chaum mixes. In Onion Routing, the data packet is broken into fixed-size cells, and each cell is encrypted multiple times (once for each onion router on the path). Thus, a recursively layered data structure called an onion is constructed. An onion is the packet transmitted along the rerouting path. The fixed size of an onion limits a route to a maximum of 11 nodes in the current implementation. Onions can be tunneled to produce arbitrary length routes.

Onion Routing I (Proof-of-concept) uses a network of five Onion Routing nodes operating at the Naval Research Laboratory. It forces a fixed length (five hops, i.e., five intermediate onion routers) for all routes.

Onion Routing II can support a network of up to 50 core onion routers. For each rerouting path through an Onion Routing network, each hop is chosen at random. The rerouting path may contain cycles, although only cycles with one or more intermediate nodes are allowed.

Freedom Network also aims at providing anonymity for Web browsing. From the user's point of view, Freedom is very similar to Onion Routing. Freedom consists of a set of nodes (called Anonymous Internet Proxy) which run on top of the existing Internet infrastructure. To communicate with a Web server, the user first selects a series of nodes to form a rerouting path and then uses this path to forward the requests to its destination. The Freedom Route Creation Protocol allows the sender to randomly choose the path, but the path length is fixed to be three. The Freedom client-user interface does not allow the user to specify a path-containing cycle. The Freedom client must either have all the intermediate nodes in the path chosen or choose a preferred first node and last node, and the intermediary nodes are picked at random.

For more information, visit *http://www.onion-router.net* and *http://www.syverson.org*.

Question: Who manages the onion routers? Are they managed independently?

Answer: Yes. The onion routers can be distributed anywhere and be managed by different groups.

Question: Do you believe that, the longer the path, the safer the anonymous communication system?

Answer: I am not sure.

Countering SYN Flood Denial-of-Service (DoS) Attacks

Ross Oliver, Tech Mavens

Summarized by David Richard Larochelle

SYN flood attacks are a nasty DoS attack. The attacker sends a SYN packet but does not complete the three-way handshake. This is hard to defend against because SYN packets are part of normal traffic, and unlike ping attacks you can't firewall them. Since SYN packets are small, the attack can be done with limited bandwidth. Finally, the attacks are difficult to trace because source IP addresses can be faked. Ross Oliver stressed that it's up to you to defend yourself (law enforcement is unable to deal with attacks as they occur, if they can deal with them at all) and suggested that firewalls employing SYN flood defenses are the best way of doing this.

He reviewed four such products: PIX by Cisco, Firewall-1 by Checkpoint, Netscreen 100 by Netscreen, and AppSafe (previously called AppSwitch) by TopLayer. To test these products, he placed a Web server behind the firewall and used a machine with a script which called wget repeatedly to request Web pages to represent the legitimate client traffic. An attacking machine threw SYN packets with forged source addresses at the Web server.

The Cisco PIX used a threshold technique which allowed a set number of incomplete connections and dropped additional SYN packets. The tests showed no significant improvement over no firewall. The Firewall-1 fared slightly better. It lets SYN packets reach the Web server and then sends an ACK packet to the Web server to complete the three-way handshake. Under a SYN flood attack, the Web server will then have a bunch of completed connections instead of half-open ones. Firewall-1 protected up to 500 SYNs per second but with degraded response time. The Web server returned to normal 3–10 minutes after the attack ceased.


Ross Oliver

Netscreen and AppSafe had the best results. If these firewalls detect a SYN flood attack, they proxy the incoming connections and only send the Web server the SYN and ACK packets if the handshake is completed by the client. Netscreen detects SYN floods by looking at the number of incomplete connections. It protected up to 14,000 SYNs/sec with acceptable response times and continued to function at higher SYN rates but with increasing delays. The server responded normally immediately after the attack.

AppSafe used a more elaborate approach. It determined whether to proxy a connection request based on the source IP address. SYN packets from IP addresses which had recently behaved legitimately were let through to the Web server immediately. Only connections from previously unseen or malicious IP addresses were proxied. AppSafe was effective up to 22,000 SYNs/sec, which was the most traffic that the attacking machine could produce in this test. However, it was pointed out that, in the test, the client machine used only one IP. This technique may not work as well in a situation in which there are new connections from previously unseen clients.

How much protection you need depends on what type of attack you expect. An attacker with a Cable or DSL connection can produce 200 SYNs/sec. An attacker with a T1 can produce 2,343 SYNs/sec. According to the paper "Inferring Internet Denial-of-Service Activity" presented the previous day, 46% of DoS attacks involved more than 500 SYNs/sec but only 2.4% were above 14,000 SYNs/sec. This level can be handled with a single firewall. Multiple or distributed attacks may require multiple parallel firewalls. Because of the wide range of performance between devices, Oliver stressed the importance of testing and advised testing the devices yourself if possible.

Real Stateful TCP Packet Filtering with IP Filter

Guido van Rooij, Eindhoven University of Technology

Summarized by Evan Sarmiento

Old firewall implementations used to filter TCP sessions using addresses and ports only, creating an interesting problem. The administrator would have to guess the source port of the packet in order to filter it correctly. In order to solve this, a new trend in firewalls is to introduce stateful packet filtering. Stateful packet filters remember and only allow through addresses and ports of connections that are currently set up.

Even before Guido van Rooij's work, IP Filter did have stateful packet filtering, but it was implemented in the wrong way. IP Filter does take sequence, ACK, and window values into account, but it makes the wrong assumption that packets seen by the filter host will also be seen by the final destination. This assumption caused IP Filter to drop packets in certain situations. The new state engine for IP Filter encompasses the following goals:

- Conclusions made by the engine must be provable.
- All kinds of TCP behavior must be taken into account.
- The number of blocked packets must be minimized.
- Blocking of packets must never lead to hanging connections.
- Opportunities for abuse should be made as small as possible.

The new state engine includes 20 bytes per state entry and about 40 lines of C code without loops; thus, the performance overhead is minimal.

However, even the new state engine is not always successful, even though it is a great improvement. Occasionally, blocked FIN and ACK packets cause problems in the state timeout handling for TCP half-closed sessions. IP Filter drops packets coming from a few Windows NT workstations for a strange and as yet unknown reason.

Guido then outlined some future additions to IP Filter. He would like to be able to fix fragment handling, add support for sessions entering the state table after establishment, and check validity of a session if a packet comes in from the middle of the connection.

REFEREED PAPERS

SESSION: DENIAL OF SERVICE

Summarized by Stefan Kelm

Using Client Puzzles to Protect TLS

Drew Dean, Xerox PARC; Adam Stubblefield, Rice University

Adam Stubblefield presented their work on a DoS protection technique, namely, the use of client puzzles within the TLS protocol. Even though client puzzles have been supposed to be a solution to DoS attacks, Stubblefield pointed out the lack of actual implementations. The choice of TLS as the protocol to protect against DoS seems obvious, but TLS is subject to DoS attacks because of the computing-expensive cryptographic operations performed at both the client and the server side. A server, however, has to perform the more expensive RSA decrypt operations during the session handshake; thus any small number of clients could easily overload a TLS server by flooding the server with TLS handshake messages. The goal of this work is to prevent this using cheap methods.

The idea of TLS-based cryptographic puzzles is to first let the client do the work, and subsequently the server. If the server is under a heavy load it sends a so-called "puzzle request" to the client. The client, in turn, has to compute a number of operations which it then uses to send a "puzzle solution" back to the server. Thus, the server will not need to continue the TLS handshake unless the client has proven its intent to really open a TLS connection.

Client puzzles are surprisingly easy to implement on both the client and the server side. Stubblefield used modified OpenSSL and mod_ssl source code to test the implementation. The implementation uses a metric which tracks unfinished RSA decrypt requests in order to decide whether or not the server is assumed to be under attack. Since adding more latency to the TLS protocol was not a goal of this work, the server only sends a puzzle request back to the client if it really has to. This is implemented by using variable thresholds.

The author concluded that they are able to protect against certain denial-of-service attacks at not much cost and with a good user experience. Moreover, the proposed solution can be implemented using already existing code.

For more information, contact *astubble@rice.edu*.

Inferring Internet Denial-of-Service Activity

David Moore, CAIDA; Geoffrey M. Voelker and Stefan Savage, University of California, San Diego

This paper, awarded the best paper award, tried to answer the question of how prevalent denial-of-service attacks in the Internet currently are. The authors ran a test over a period of three weeks, trying to come up with an estimate of worldwide DoS activity.

David Moore presented the so-called "backscatter analysis" as their key idea and outlined the basic technique: since attackers normally use spoofed source IP addresses, the "real owners" of those IP addresses regularly receive response packets from the systems being attacked (Moore called these "unsolicited responses"). By monitoring these unsolicited responses one is able to detect different kinds of DoS attacks. Furthermore, by observing a huge number of different IP addresses over a longer period of time, sampling the results can provide an overview of attacks going on.

Moore presented some interesting results and displayed a number of figures and tables showing the number of attacks, the attacks over time, the attack characterization, the attack duration distribution, and the attack rate distribution. Moore's team observed a number of minor DoS attacks (described as "personal vendettas") as well as some victims under repeated attack. Classifying the victims by TLD showed countries like Romania and Brazil being attacked far more often than most other TLDs. The presenter's hypothesis was that either those countries host ISPs that attack each other, or there simply are more hackers located in Romania and Brazil (this was later denied by someone in the audience stating that Romania has really nice people).

In conclusion, the authors observed some very large DoS attacks, though most attacks seem to be short in duration. Another result showed the majority of attacks being TCP based. To clarify, this technique is not good at distinguishing between DoS and DDoS

attacks since it is not good at distinguishing between attackers.

During the Q&A session, one question was on why the analysis showed no attacks on the .mil domain. The response given was that either .mil is not under attack (unlikely) or that backscatter packets are being filtered.

For more information, contact *dmoore@caida.org*, or see *http://www.caida.org/outreach/papers/backscatter/*.

MULTOPS: A Data-Structure for Bandwidth Attack Detection

Thomer M. Gil, Vrije Universiteit/MIT; Massimiliano Poletto, MIT

Thomer Gil proposed a heuristic as well as a new data structure to be used by routers and similar network devices to detect (and possibly eliminate) denial-of-service attacks. Most DoS attacks show disproportional packet rates with a huge number of packets being sent to the victim and only very few packets being sent by the victim in response. The new data structure, called MULTOPS (Multi-Level Tree for Online Packet Statistics), monitors certain Internet traffic characteristics and is able to drop packets based on either the source or the destination address.

The main implementation challenges with MULTOPS have been and still are the precise identification of malicious addresses, athe achievement of a small memory footprint, and a low overhead on forwarding "real" traffic as opposed to DoS-based traffic. MULTOPS is implemented as a memory-efficient tree of nodes which contains packet-rate statistics and which dynamically grows and shrinks with the traffic being observed. At the current implementation, packets are dropped based on either a variable packet rate or a ratio. Since it usually is impossible to identify an attacker (because of IP spoofing), packets can be dropped based on the victim's IP, too.

The authors succeeded in simplifying memory management and the mechanism that keeps track of packets. Gil pointed out that their solution is successfully being used by a network company. They are currently trying to focus on the behavior of different TCP implementations as well as protocols other than TCP.

Someone brought up the question of differentiating DoS traffic from traffic that normally shows disproportional packet flows, e.g., video traffic. The reply suggested the possibility of building some kind of knowledge base. A lively discussion on random class A addresses within MULTOPS subsequently arose but was taken offline.

For more information, contact *thomer @lcs.mit.edu*.

SESSION: HARDWARE

Summarized by Anca Ivan

Data Remanence in Semiconductor Devices

Peter Gutmann, IBM T.J. Watson Research Center

Peter Gutmann explained the dangers of deleting data in semiconductors. Everyone knows that deleting data from magnetic media is very hard, but not too many realize that the same problem exists for semiconductors, especially since there are so many ways of building semiconductors, each with its own set of problems and solutions. After giving a short background introduction in semiconductors and circuits (n-type, p-type, SRAM, DRAM), Peter described some of the most important issues:

- Electromigration: because of high current densities, metal atoms are moved in the opposite direction of the normal current flow. The consequence is that the operating properties of the device are strongly altered.
- Hot carriers: during operation, the device heats up and its characteristics change considerably.
- Ionic contamination: this is no longer an issue and its effects are no longer significant.
- Radiation-induced charging: it freezes the circuit into a certain state.

The first phenomenon enables attackers to recover partial information from special-purpose devices (e.g., cryptographic smartcards). The next two can be used to recover data deleted from memory. In order to avoid long- and short-term data retention from semiconductors (a DES key was recovered in the '80s), researchers developed a series of solutions that use various semiconductor forensic techniques, including the following two:

- Short-term retention: probably the safest way to defend against it is not to keep the same values in the same memory cells for too long (maximum a few minutes).
- Long-term retention: in 1996, some researchers proposed periodically flipping the stored bits. In this way, no cell holds the same bit value for long enough to "remember" it.

In the end, Peter talked about how all the problems cited above extend to flash memory. For example, random generators can generate strings of 1s when the pool is empty, or information can be leaked into adjacent cells into shared circuitry.

Even though the entire presentation scared at least one person in the audience (guess who?), Peter assured us that reality is not that gloomy. In fact, the only problem is the lack of a standard. Every time people decide to choose one implementation method, they should also choose which solutions are best for it. Answering a question, Peter told us that personal computers are not affected

by those problems but that most specialized devices, like airplane black boxes, can leak information if analyzed with very sophisticated equipment. But then again who has such equipment?

StackGhost: Hardware-Facilitated Stack Protection

Mike Frantzen, CERIAS; Mike Shuey, Purdue University

The authors presented a software solution to the return-pointer hijacking problem. The most important step in the function-call process is when the caller saves the return pointer before giving the control to the called function. Many attacks are based on changing this pointer. When the callee finishes, the return pointer dictates which function takes control next. StackGhost is a piece of software that automatically and transparently saves the return pointer and replaces it with another number. When the called function completes, StackGhost verifies the integrity of that number (catching, in this way, possible attacks) and reinstalls the correct pointer value.

The security of StackGhost depends on how it modifies the return pointer to catch attacks; the authors have tried several ways:

- Per kernel XOR with a 13-bit signed cookie: the main problem is that an attacker can find out the cookie by starting several arbitrary programs.
- Per process XOR with a 32-bit cookie: this is safer than the previous method, but more expensive.
- Encrypt/decrypt the return pointer: this method seems to be the most expensive.
- Return-address stack: this method replaces the return pointer with another number and saves the pointer into a return-address stack. However, this would impede other applications from running correctly.

After making performance measurements for all techniques, the authors noticed that the chances of StackGhost not catching an attack were 1 in 3 for XOR cookies and 1 in 2^{32} for the return-address stack. The conclusion was that StackGhost was offering protection against return-pointer overriding to all processes in the system (which might be seen as a disadvantage).

Improving DES Coprocessor Throughput for Short Operations

Mark Lindemann, IBM T.J. Watson Research Center; Sean W. Smith, Dartmouth College

While the first two talks in this session were at opposite poles (one deeply hardware and one purely software), the third one was somehow in the middle. The presenter, Sean Smith, is one the fathers of the cryptographic card developed at IBM and presently working at Dartmouth College. Everything started in a very optimistic fashion, with the usual introduction we would have expected from an IBM representative trying to sell us this device: "It is secure . . . it is fast . . . it is reliable." All the buzzwords were there. However, with the next slide this changed to "It is not as secure . . . fast . . . as we thought." For example, the specification promised the DES speed to be 20 megabytes/second when in reality a friend obtained less than two kilobytes/second in a database application. Where was the discrepancy coming from? The main intuition was that the specification gives the performance for operations on megabytes of input. The real speed is much slower if the data is shipped to the card in small chunks. The difference between specs and reality was too big not to be studied, and Lindemann decided to find out the reasons behind it. First, they built a model that simulates the database application and then tried to improve the speed by modifying the execution conditions in the following ways:

- Reducing card-host interaction: "folklore in IBM" taught them that any card-host interaction consumes too much time. Thus, they rewrote the application to minimize the number of interactions. The speed went up to 18–23 kilobytes/second; however, it was still too far from megabyte speed.
- Batching all operations into one chip operation: chip resets were too expensive. The speed became 360 kilobytes/second.
- Batching into multiple chip operations: it reduced the number of Layer 3 – Layer 2 switches. The speed changed to 30–290 kilobytes/second, still not good.
- Reducing data transfers: they did it by using an internal key-table and boosted the speed to 1,400 kilobytes/second.
- Using memory-mapped I/O: this eliminated the internal ISA bus bottleneck. The speed went up to 2,500 kilobytes/second.
- Batching operation parameters: instead of sending them as separate packets. It increased the speed to 5000 kilobytes/second. This was even more than they were expecting, but the results were incorrect. The client had asked for speed but hadn't mentioned anything about correctness. So was the problem solved?
- Not using memory-mapped I/O: to increase accuracy, they gave up on memory-mapped I/O for initialization vectors and count. Unfortunately, there was a small performance cost: the speed was now 3,000 kilobytes/second.

From the client's point of view, all of these steps showed them that the only way to maximize the performance while using the secure coprocessor was to design DES-batched API. From the designer's point of view, the conclusions

were simpler: always distrust folklore and think if and how people will use your product before designing it!

SESSION: FIREWALLS/INTRUSION DETECTION

Summarized by Stefan Kelm and Yong Guan

Architecting the Lumeta Firewall Analyzer

Avishai Wool, Lumeta

"What is your firewall doing?" Avishai Wool asked the audience at the beginning of his presentation, thereby describing the motivation to build LFA, the "Lumeta Firewall Analyzer."

Firewalls have been installed by almost all companies connected to the Internet. However, the underlying policy often is far from being good enough to actually protect the company from outside attackers. Network administrators often do not know how to set up a firewall securely, much less how to test or audit the firewall configuration. Wool pointed out that LFA is the successor of the Fang prototype system built at Bell Labs as a firewall analysis engine.

The key idea is not to probe the actual firewall in any way but to allow testing of the configuration before the firewall is deployed. The firewall's routing table and configuration files are used as input to the LFA, which parses these files and simulates the behavior of any possible packet flow combination (LFA mainly offers support for Firewall-1 and PIX). The results are presented to the user as HTML pages.

Wool concluded by giving a short demonstration. As input to the LFA, he used a short Firewall-1 policy which contained only six rules and explained why even such a short rule set might lead to problems once the firewall is deployed. During the Q&A session he emphasized that the LFA only checks packet headers, not the content, and cannot therefore detect tunneling problems.

For more information, contact *yash@acm.org*, or visit *http://www.lumeta.com/firewall.html*.

Transient Addressing for Related Processes: Improved Firewalling by Using IPv6 and Multiple Addresses per Host

Peter M. Gleitz and Steven M. Bellovin, AT&T Labs-Research

The authors proposed a method to simplify firewall decisions. By using the large address space brought by IPv6, they employed a strategy of multiple network addresses per host. That is, for each request on the client host an IPv6 address is tied to the client process. The firewall now makes access decisions based on transport layer protocol information (i.e., filtering is shifted from ports to addresses). Once approved, the firewall allows all traffic between the two peers to pass to and fro. Once the service is finished the IPv6 address is discarded. This method is called TARP (transient addressing for related processes). TARP employs two different types of addresses: (fixed) server addresses and process group addresses.

Gleitz discussed how TARP works with TCP and UDP applications and with the firewall, router, domain name server, and IPSEC. Employing TARP does not necessarily affect the routers, though TARP-aware routers can perform better. Moreover, Gleitz pointed out that no modifications to standard applications such as Telnet, SSH, FTP, Sendmail, or TFTP are necessary in order to use TARP. He also mentioned briefly some interop problems with protocols such as DNS and ICMPv6.

For more information, contact *pmgleit@netscape.net.*

Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics

Mark Handley and Vern Paxson, ACIRI; Christian Kreibich, Technische Universität München

This paper focused on the problem of network intrusion detection system (NIDS) evasion. Attackers usually can fool any NIDS by exploiting certain ambiguities in the packet flow being monitored by the NIDS, i.e., (1) the NIDS may lack complete analysis of the packet flow (e.g., no TCP stream reassembly); (2) the NIDS may lack end-system knowledge (e.g., certain application vulnerabilities); and (3) the NIDS may lack network knowledge (e.g., the topology between the NIDS and an end system).

As a solution, Paxson proposed the deployment of a "normalizer," the goal of which would be to observe all packets being sent between two network nodes (he called that a "bump-in-the-wire") and to modify ("normalize") packets that seem to be ambiguous for one reason or another. As an example the author described problems with two overlapping fragments: the normalizer would re-assemble (and re-fragment, if necessary) those packets before forwarding. Since re-assembly is a valid operation, the normalizer would, in this example, have no impact on the semantics at all.

Paxson also pointed out some of the problems with this approach, one of which is the "cold start" problem: (re-)starting the normalizer will show many valid connections already established. It is difficult to handle those connections accordingly (this is also true for the NIDS itself). The normalizer has been implemented and will be available at *www.sourceforge.net* soon.

In the Q&A session Steven Bellovin wanted to know whether normalization

would not be needed at the application layer as well. The presenter answered in the affirmative.

For more information, contact *vern@aciri.org*.

SESSION: OPERATING SYSTEMS

Summarized by Mike Vernal

Security Analysis of the Palm Operating System and Its Weaknesses against Malicious Code Threats

Kingpin and Mudge, @stake, Inc.

Kingpin and Mudge began their presentation with a bold fashion statement, appearing in matching white bathrobes. Their bathrobes aimed to underscore the fact that PDAs can undermine user privacy in a public setting. Their efforts were later rewarded with the coveted USENIX Style Award, presented by the real Peter Honeyman, of the University of Michigan.

Kingpin and Mudge

The presentation centered on the security threat posed by the recent ubiquity of Personal Digital Assistants (PDAs), and, more specifically, devices running the Palm Operating System. Palm devices increasingly are being used in security-sensitive settings such as hospitals and government agencies. While the government is now aware of the security threat posed by PDAs, the corporate world has remained generally oblivious.

The security threat of the Palm stems from the PalmOS's lack of a well-defined security framework. Specific weaknesses enumerated include the direct addressability of hardware, the lack of memory encryption, the lack of ACLs, weak obfuscation of passwords, and a backdoor debug mode that allows for the bypassing of "system lockout." Because of these and other weaknesses, the audience agreed with the assertion that developing a secure application on top of the PalmOS would be impossible.

The presentation suggested that unscrupulous users could exploit a number of weaknesses to install malicious code, including normal application installation, desktop conduits, creator ID replacement, wireless communications, and the Palm Debugger. Another threat raised by Kingpin and Mudge was the possibility of a new set of cross-pollinating viruses, which could be acquired via a Palm and propagate themselves to desktop computers via the HotSync operation, or vice versa.

With the growing popularity of PDAs, Kingpin and Mudge invoked Occam's Razor: all other factors being equal, the PDA may be the malicious user's easiest point of entry into an information network. The upcoming PalmOS 4.0 reportedly fixes some of the security concerns raised. In the interim, users should be made aware of the possible security threats and restrict or eliminate their use of sensitive data and applications on Palm devices.

Secure Data Deletion for Linux File Systems

Steven Bauer and Nissanka B. Priyantha, MIT

Steven Bauer presented an implementation of a kernel-level secure data-deletion (SDD) mechanism for the ext2 file system.

Bauer suggested that with the increasing prevalence of public kiosks, thin clients, multi-user computing clusters, and distributed file systems, users will want to ensure that when their data is deleted from these systems, it is truly and irretrievably deleted.

In 1996, Peter Gutmann of IBM demonstrated that data that had been overwritten on a magnetic disk could be recovered using advanced probing techniques. While popular lore has suggested certain government agencies may be able to recover data overwritten dozens of times, no commercial data recovery company contacted in conjunction with Bauer's research believed that it could recover data that had been overwritten more than once. As such, the SDD system as described probably only needs to overwrite data a few times.

This SDD system was designed to ensure that all flagged data is deleted, even in the event of system failure. The deletion process was designed as an asynchronous daemon to ensure that it did not interfere with normal operation and performance. Though implemented for the ext2 file system, Bauer asserts that this system should be portable to any block-oriented file system.

The ext2 implementation used the unused secure-deletion flag, settable with the chattr() function. With this mechanism, the granularity with which secure deletion can be specified ranges from an entire device to an individual file. Questions were raised as to the vulnerability of temporary files that are not flagged in a secure deletion zone. Bauer recommended that for maximum security, the entire device should be flagged for secure deletion.

SESSION: MANAGING CODE

Summarized by Sameh Elnikety

Statically Detecting Likely Buffer Overflow Vulnerabilities

David Larochelle and David Evans, University of Virginia

Buffer overflow attacks account for approximately half of all security vulnerabilities. Programs written in C are particularly susceptible to buffer overflow attacks because C allows direct pointer manipulations without any bounds checking.

Run-time approaches to mitigate the risks of buffer overflow incur performance penalties, and they turn buffer overflow attacks into denial-of-service attacks by terminating execution of the attacked processes. Static checking overcomes these problems by detecting likely vulnerabilities before deployment.

The authors developed a practical lightweight static analysis tool based on LCLint to detect a high percentage of likely buffer overflow vulnerabilities.

The tool exploits semantic comments (annotations) that describe programmer assumptions and intents. These annotations are treated as regular C comments by the compiler but are recognized as syntactic entities by LCLint. The annotations represent preconditions and postconditions for functions to determine how much memory has been allocated for buffers. LCLint uses traditional compiler data flow analyses with constraint generation and resolution. Also, LCLint uses loop heuristics to efficiently analyze many loop idioms in typical C programs.

The authors used the tool to analyze wu-ftpd, which is a popular open source FTP server, and part of BIND, which is a set of domain-name tools and libraries that is considered the reference implementation of DNS. Running LCLint is similar to running a compiler. For wu-ftpd, it took less than one minute for LCLint to analyze all 17,000 lines of unmodified wu-ftpd source code. This resulted in 243 warnings that showed known and unknown buffer overflow vulnerabilities.

LCLint source code and binaries are available from *http://lclint.cs.virginia.edu*.

FormatGuard: Automatic Protection from printf Format String Vulnerabilities

Crispin Cowan, Matt Barringer, Steve Beattie, Greg Kroah-Hartman, WireX Communications, Inc.; Mike Frantzen, Purdue University; and Jamie Lokier, CERN

In June 2000, a major new class of vulnerabilities called format bugs was discovered when a vulnerability in WU-FTP appeared that looked almost like a buffer overflow but was not. It is unsafe to allow potentially hostile input to be passed directly as the format string for calls to printf-like functions. The danger is that the inclusion of % directives, especially %n, in the format string coupled with the lack of any effective type or argument counting in C's varargs facility allows the attacker to induce unexpected behavior in programs.

The authors developed FormatGuard, a small patch to glibc. It provides general protection against format bugs using particular properties of the GNU CPP macro-handling mechanism to extract the count of actual arguments to printf statements. This is then passed to a safe printf wrapper. The wrapper parses the format string to determine how many arguments to expect, and if the format string calls for more arguments than the actual number of arguments, it raises an intrusion alert and kills the process.

FormatGuard fails to protect against format bugs under several circumstances. For example, if the program uses a function pointer that has the address of printf, then it evades the macro expansion.

FormatGuard is incorporated in WireX's Immunix Linux distribution and server products. It is available as a GPL'd patch to glibc at *http://immunix.org*.

Detecting Format String Vulnerabilities with Type Qualifiers

Umesh Shankar, Kunal Talwar, Jeffrey S. Foster, and David Wagner, University of California, Berkeley

Systems written in C are difficult to secure, given C's tendency to sacrifice safety for efficiency. Format string vulnerabilities can occur when user input is used as a format specifier. One of the most common cases is when the program uses printf with one argument: a user-supplied string assuming that the string does not contain any % directive. The authors presented a tool (cqual) that automatically detects format string bugs at compile time using type-theoretic analysis techniques. With this static analysis, vulnerabilities can be proactively identified and fixed before the code is deployed.

Cqual builds an annotated Abstract Syntax Tree (AST). Then, it traverses the AST to generate a system of type constraints, which is solved online. Warnings are produced whenever an inconsistent constraint is generated. Cqual presents the results of tainting analysis to the programmer using Program Analysis Mode for Emacs (PAM). PAM is a GUI that is designed to add hyperlinks and color mark-ups to the preprocessed text of the program. The interface shows the taint flow path to help programmers determine how a variable becomes tainted.

The configuration files makes cqual usable without modifying the source code. The authors analyzed four security-sensitive benchmark programs with the same standard prelude file and no

direct changes to the applications' source code. Typically a few application-specific entries were added to the prelude file to improve accuracy in the presence of wrappers around library functions. Cqual reliably finds all known bugs for the benchmark programs. It also reports few false positives. Cqual is fast; it usually takes less than a minute.

Cqual is available at *http://bane.cs.berkeley.edu/cqual*.

SESSION: AUTHORIZATION

Summarized by Rachel Greenstadt

Capability File Names: Separating Authorization from User Management in an Internet File System

Jude T. Regan, consultant; Christian D. Jensen, Trinity College

On the Internet there is no reliable way to establish an identity. Flexible user-user collaboration outside of an administered system so that people could create ad hoc work groups and remove arbitrary limitations to information sharing is the authors' goal.

Such a system should be globally accessible, easy to use, and require as little intervention by system administrators as possible. This system should integrate with existing systems and applications. It should have fine granularity so that users would not have to use complicated export mechanisms to share files.

The authors used the concept of a capability, a token conveying specified access rights to a named object in order to make the identity of the object and the access rights inseparable. They embedded the capability in something every system knows – the file name.

The authors concluded that the system was safe from interception and modification Attackers could forge the client part but not the server part of the file names. Service could be interrupted, but protecting against this is impossible without complete control of the network. Performance was evaluated in comparison to NFS. Most of the overhead was in the open statement. Reads were slightly slower and writes were much slower, but they felt this could be alleviated by implementing symmetric writes.

Kerberized Credential Translation: A Solution to Web Access Control

Olga Kornievskaia, Peter Honeyman, Bill Doster, and Kevin Coffman, CITI, University of Michigan

There are two different authentication mechanisms: those used for services such as login, AFS, and mail, for which Kerberos is popular, and public key-based mechanisms such as SSL, which is used to establish secure connections on the Web. These systems need to be able to work together to satisfy a request.

The authors propose to achieve the best of both worlds by leveraging Kerberos to solve PKI key management. This will use existing infrastructures which allow strong authentication on the Web with SSL and which provide access to Kerberized back-end services. They propose a system to provide interoperability between PKI and Kerberos. Their system consists of (1) a Certificate Authority (CA), KX509, which creates short-lived certificates, (2) a Web server which acts like a proxy for users by requesting services from Kerberized back-end services and (3) a Kerberized Credential Translator, which translates public-key credentials to Kerberos. They created a prototype of their system called WebAFS using AFS as an example Kerberized service.

Dos and Don'ts of Client Authentication on the Web

Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster, MIT

[This paper received the Best Student Paper Award]

Kevin gave a very amusing presentation which illustrated the gap between security theory and practice. He described a variety of Web sites that used insecure client authentication schemes and presented hints on how to avoid their mistakes.

Client authentication seems like a solved problem, but many sites continue to come up with homebrew schemes which just don't quite get it right. Out of the 27 Web sites the cookie eaters group examined, they weakened the security on two sites, were able to mint authenticators on eight, and on one site were able to obtain the secret key. Some of these sites were high profile, such as the *Wall Street Journal* (wsj.com), Sprint PCS (sprintpcs.com), and FatBrain (fatbrain.com).

In most cases, the mistakes made in these sites were simple. By simply looking at their cookie files the authors could query Web servers and look at headers, responses, and create sample authenticators. Except for Sprint, these attacks involved no eavesdropping at all. The schemes were not even strong against what the authors termed the "interrogative adversary." This adversary has no special access, but it adaptively queries a Web server a reasonable number of times. It just sits there and connects to port 80; it cannot defeat SSL client authentication, HTTP basic, or digest authentication. The best such an adversary can do against a pass-

Kevin Fu

word sent in the clear is a dictionary attack. However, some homebrew cookie schemes are vulnerable.

In the case of the *Wall Street Journal*, a site with half a million paid subscribers who can track their stocks and buy articles, the authors found that the makers of the site had misused cryptography and created an authenticator weaker than a plaintext password.

Some hints provided for client authentication were: limit the lifetime of authenticators since browsers cannot be trusted to expire cookies; expiration dates must be cryptographically signed (this was another problem with *WSJ*). Authenticators should be unforgeable, and cookies should not be modifiable by the user. There should be no bypassing of password authentication. Digital signatures are great, but you should not allow the things you sign to be ambiguous. For example, the concatenation of "Alice, 21-Apr" and "Alice2, 1-Apr" is the same. Delimiters can help solve this problem. He presented a simple scheme for building an authenticator which would work against the interrogative adversary.

In summary, there are many broken schemes out there, even in popular Web sites. There are even more juicy details in the authors' technical report. Cookie schemes are limited; live with it or move on. You can join the authors by donating your cookies for analysis at *http://cookies.lcs.mit.edu*.

SESSION: KEY MANAGEMENT

Summarized by Sameh Elnikety

SC-CFS: Smartcard Secured Cryptographic File System

Naomaru Itoi, CITI, University of Michigan

Storing information securely is one of the most important applications of computer systems. Secure storage protects the secrecy, authenticity, and integrity of the information. SC-CFS implements a secure file system and is based on Matt Blaze's Cryptographic File System for UNIX (CFS). SC-CFS uses a smartcard to generate a key for each file rather than for each directory. The per-file key encryption counters the password-guessing attack and minimizes both the damage caused by physical attack, compromised media, and bug exploitation.

When an encrypted file is updated, a new key is generated for that file and the file is re-encrypted for increased security. SC-CFS employs the same authentication mechanism as CFS, using an encrypted signature containing both a random number and a predefined sequence. A signature is stored in each directory. When a user starts to access a directory, SC-CFS gets the user key and decrypts the signature to recover the predefined sequence. If the sequence is not recovered, SC-CFS denies the user access to the directory.

SC-CFS is more secure than CFS because the master key is a random number instead of a password. This prevents dictionary attacks. Also, the user master key is not exposed to the host, and a stolen file key would reveal only one file and then only until that file is updated and consequently re-encrypted with a new file key.

The author implemented SC-CFS as an extension to CFS, then evaluated the performance of SC-CFS in comparison with CFS and a local Linux file system (ext2) using the Andrew Benchmark test. The results show that the performance of the system is not yet satisfactory because smartcard access is the bottleneck of SC-CFS. SC-CFS works as efficiently as ext2 and CFS when it does not access a smartcard. However, SC-CFS is significantly slower than CFS when it accesses a smartcard because the smartcard generates a key in 0.31 seconds.

Secure Distribution of Events in Content-Based Publish Subscribe Systems

Lukasz Opyrchal and Atul Prakash, University of Michigan

Some Internet applications, such as wireless delivery services and inter-enterprise supply-chain management applications, require high scalability as well as strict security guarantees. The content-based publish subscribe paradigm is one of the messaging technologies that facilitate building more scalable and flexible distributed systems. In the publish subscribe model, publishers publish messages and send them to subscribers via brokers. Each broker manages a large number of subscribers. The broker encrypts every message and broadcasts it to subscribers. The broker needs to guarantee the confidentiality of the messages so that only a specific group of subscribers can read the message.

Each subscriber has an individual symmetric pair key shared only with its broker. A naïve way to achieve this secure end-point delivery is for the broker to encrypt each message with a new key. Then, the broker sends the new key securely to each subscriber in the target group, by encrypting the new key with the symmetric key shared between the broker and the subscriber. The number of encryptions limits the broker throughput and system scalability. For the naïve approach, the number of encryptions is the same as the group size.

The authors presented four caching strategies to reduce the number of required encryptions. Simple cache assumes that many messages will go to the same subset of subscribers. Simple cache creates a separate key for each group and caches it. Build-up cache is based on the observation that many groups are subsets of other larger groups. Build-up cache uses a heuristic

to select some groups to cover the target group. Clustered cache uses a much smaller cache size by dividing the subscribers into clusters. Then, it uses the simple-cache method to send a message to the target subgroup in each cluster. Clustered-popular cache maintains both a simple cache and a clustered cache. When a new message arrives, clustered-popular cache searches for the target group in the simple cache. If the group is not found it uses the clustered cache to send the message to the appropriate subgroup in each cluster.

The authors analyzed the four caching strategies to find the average number of required encryptions and ran a number of simulations to confirm the theoretical results. They found that clustering the subscribers can substantially reduce the number of encryptions, which can be further reduced by adding a simple cache to clustered cache. Build-up cache, however, has little effect on the number of required encryptions.

A Method for Fast Revocation of Public Key Certificates and Security Capabilities

Dan Boneh, Stanford University; Xuhua Ding and Gene Tsudik, University of California, Irvine; Chi Ming Wong, Stanford University

The authors presented a new approach to fast certificate revocation using an online semi-trusted mediator (SEM). Suppose an organization has a Public Key Infrastructure that allows users to encrypt and decrypt messages and to digitally sign the messages. If an adversary compromises the private key of a user, then the organization needs to immediately prevent the adversary from signing or decrypting any message.

The overall architecture of the system is made up of three components. First, the central Certificate Authority (CA) generates a public key and a private key for each user. The private key consists of two parts. The CA gives the first part only to the user, and the other part only to the SEM. Second, the SEM responds to user requests with short tokens. The tokens reveal no information to other users. Third, the user contacts the SEM in case he wants to generate a digital signature or to decrypt a message. The system uses the MRSA encryption technique, which is similar to RSA, in a way that is transparent to peer users. The encryption process is identical to standard RSA. For the decryption process, the SEM does part of the decryption and the user does the remaining part. Both the SEM and the user must perform their share to decrypt a message. Digital signatures are generated in a similar way to performing decryption.

The authors implemented the system using OpenSSL and provided a client API and server daemons. The performance measurements showed that signature and encryption times are essentially unchanged from the user's perspective. The authors also implemented a plug-in for Eudora that enables users to sign their emails using the SEM. This approach achieves immediate revocation of public key certificates and security capabilities for medium-size organizations rather than the global Internet.


Gene Tsudik

The implementation of the system is available at:
http://sconce.ics.uci.edu/sucses.

The SEM Eudora plug-in is available at:
http://crypto.stanford.edu/semmail.

SESSION: MATH ATTACKS!

Summarized by Kevin Fu

PDM: A New Strong Password-Based Protocol

Charlie Kaufman, Iris Associates; Radia Perlman, Sun Microsystems Laboratories

A bright and cheery Radia Perlman talked about Password-Derived Moduli (PDM), a protocol useful for both mutual authentication and securely downloading credentials. PDM's notable features and improvements over existing protocols include unencumberance by patents, better overall server performance, and better performance when not storing password-equivalent data on the server.

Despite the promise of smartcards, passwords are still important for authentication. Demonstrating this importance, Perlman cited her own habit of misplacing any hardware token given to her. However, she can remember a password.

PDM deterministically generates a prime from a user's password and salt such as the username. To generate a prime, the user Alice fills out chunks of the right size with the hash of ("Alice," password, constant). PDM then searches for a safe "Sophie Germain" prime (p). A prime is Sophie Germain if (p-1)/2 is also a prime. PDM then uses this prime as the modulus in Diffie-Hellman exchanges.

PDM is potentially fast on a server and tolerably slow on a client. Although 512-bit Diffie-Hellman moduli are within the realm of breakability, a dictionary attack against PDM requires a Diffie-Hellman exponentiation per password guess. This places a lot of computational burden on an adversary. Using 512-bit moduli instead of 1024-bit moduli improves performance on the server by a factor of six.

PDM strives not to leak information and avoids timing attacks by properly order-

ing cryptographic operations. PDM can also avoid storing password-equivalent data on the server. If the server is compromised, the user's password can remain safe. Other protocols avoid password equivalence by having extra Diffie-Hellman exchanges.

Deriving a 512-bit prime from a password is computationally expensive. Ten seconds on a reasonably modern machine is not uncommon. However, there are simple improvements. Perlman's son improved the client performance by a factor of three by using a sieve instead of division. If a user provides a hint in addition to the password, the generation of the prime can finish in a fraction of a second. The hint could be the first few bits of the prime, easily encoded as a single character to remember.

Then came questions. Asked about the distribution of primes derived from passwords, Perlman answered that the primes are uniformly distributed in the range of possible primes. For all possible passwords, this is uniformly distributed.

Asked why PDM depends on a strong Sophie Germain prime, Radia explained that the base 2 is then guaranteed to be a generator if the prime is also congruent to 3 mod 8. If 2 were not a generator, then 2 would generate a smaller subgroup – reducing security.

Detecting Steganographic Content on the Internet

Niels Provos, CITI, University of Michigan

Because Slashdot had just discussed a "theoretical" system to detect steganographic content on the Internet, Niels decided it was time to discuss a system already doing this. Instead of talking about methods to defend against statistical steganalysis, Niels talked about his software to find hidden messages in JPEG files.

The popular press claims that terrorists like Osama bin Laden use steganography. Of course, this is totally unsubstantiated. Hence, Niels sought answers to three questions:

- How to automatically detect steganographic content
- How to find a source of images with potentially steganographic content
- How to determine whether an image contains hidden content

Steganography is the art and science of hiding the fact that communication is happening. In modern steganography, one should only be able to detect the presence of hidden information by knowing a secret key. The goal of an adversary is to detect steganography, not necessarily to recover the message. One must select a cover medium to embed a hidden message. Bits are changed to embed a message. The original cover medium is then destroyed.


Niels Provos

There are many systems to hide messages in images: JSteg, JPHide, and Niels' Outguess. All of these systems cause different distortions in images. Niels wrote the "stegdetect" program to detect images modified by JSteg, JPHide, and Outguess. The program gives a notion of how likely it is that an image contains hidden content.

On a 1200MHz Pentium III, stegbreak processes 15,000 words/sec for JPHide, 47,000 words/sec for Outguess, and 112,000 words/sec for JSteg. Because a single fast machine can only process so much, Niels wrote the "disconcert" program to mount a distributed dictionary attack.

Niels has sorted through over 2 million JPEG images from eBay. Although 17,000 images came up positive, no genuine steganography was found. There is as yet no final conclusion on whether the underworld uses steganography in this way. The popular press will have to continue with unsubstantiated claims.

Asked if one can determine the quality metric used to create a JPEG, Niels said this is possible but will not reveal whether there is steganographic content because modifications of DCT coefficients do not modify quality of images much.

Another person asked for advice on how to hide messages while minimizing distortion. Niels explained that hiding just one bit is easy. Otherwise it is important to realign the statistical properties of the image after embedding a message.

One audience member suggested that terrorists might use homebrew steganographic software. In such a case, will the same statistical tests help detect hidden messages? Niels said that with certain generic assumptions, maybe. One would need to know the statistical signature common to the software.

Another audience member asked if Niels has searched for JPEGs on sites other than eBay. Niels responded that he has only considered eBay because the popular press mentioned auctions as the perfect venue. So far the press seems to be fantasizing.

Finally, a participant asked if the number of false positives fit any hypothesis. Niels answered no. The images vary in quality and size. So, from the beginning, many images are mischaracterized by the statistical tests. Niels did run his software against a test set though. It correctly detected the hidden messages.

For more information, see *http://www.citi.umich.edu/u/provos/* or *http://www.outguess.org/*.

TIMING ANALYSIS OF KEYSTROKES AND TIMING ATTACKS ON SSH

Dawn Xiaodong Song, David Wagner, and Xuqing Tian, University of California, Berkeley

Dawn Song explained how two traffic analysis vulnerabilities in the SSH protocol can leak damaging amounts of information. By eavesdropping on an SSH session, Song demonstrated the ease of recovering confidential data such as root passwords typed over an SSH connection. Song's group then built the Herbivore attacker system, which tries to learn users' passwords by monitoring SSH sessions. Herbivore can speed up brute force password searches by a factor of 50.

The SSH protocol has largely replaced insecure Telnet. Ideally SSH should withstand attacks by eavesdroppers. Alas, SSH leaks information about the approximate length of data. Moreover, each key press generates a separate packet. The length can indicate when a user is about to enter a password during an established SSH session. By watching the inter-keystroke events, an eavesdropper can make educated guesses about passwords and other confidential information.

The most startling example is that of the su command typed over an SSH session, which results in a very recognizable traffic signature. Simply by looking at the lengths of requests and responses, an eavesdropper can detect the transmission of a password. Song noted that su disables echo mode. The resulting asymmetric traffic indicates that a password will follow.

Once an eavesdropper knows that a sequence of packets corresponds to a password, the inter-keystroke timings can reveal characteristics of the password. Herbivore looks at the frequency distribution of a given character pair. For instance, one may type vo with alternating hands while typing vb with the same hand. The latency between each keypress is distinguishing. For randomly chosen passwords, inter-keystroke timings leak about 1.2 bits per character.

One countermeasure against this attack would be to hide inter-keystroke timings by using a constant packet rate in active traffic.

Next, a slew of people raced to the microphone. One person asked whether taking many samples of a single user would reduce the password search space even more. Song responded that this technique has diminishing returns.

Asked about the effect this work has on passwords typed over a wireless network, Song reported that her group did not test real users' passwords. Each test subject used an assigned password. All the test subjects were touch typists.

When one audience member asked why not set TCPNODELAY right before typing passwords, another audience member said that is already the case.

Song also explained that randomly inserting a delay in traffic will not help much. An eavesdropper can obtain your typing of passwords many times to filter out the randomization.

WORKS IN PROGRESS

Summarized by Sam Weiler and David Richard Larochelle

USING THE FLUHRER, MANTIN, AND SHAMIR ATTACK TO BREAK WEP

Adam Stubblefield, Rice University; John Ioannidis and Avi Rubin, AT&T Research

The authors implemented a recently published attack against WEP, the link-layer security protocol for 802.11 networks. Exploiting WEP's improper use of RC4 initialization vectors, they recovered a 128-bit key from a production network using a passive attack. For assorted legal and moral reasons, they're not planning to release the code, but others are developing similar tools.

For more information, visit *http://www.cs.rice.edu/~astubble/wep/*.

SRMAIL – THE SECURE REMAILER

Cory Cohen, CERT

SRMail allows groups of people who may not share common crypto methods to communicate. It can generate encrypted form letters and convert between encryption formats when used as a remailer. SRMail will be used at CERT to allow several people to masquerade as CERT and generate documents signed with CERT's keys without requiring them to have direct access to those keys.

VOMIT – VOICE OVER MISCONFIGURED INTERNET TELEPHONES

Niels Provos, CITI, University of Michigan

Vomit converts a Cisco IP phone conversation into a wave file, allowing users to play a call directly from the network or from a tcpdump output file. Vomit can also insert wave files into ongoing telephone conversations. Provos suggested that Vomit can be used as a network debugging tool, a speaker phone, and so on.

For more information, visit *http://www.monkey.org/~provos/vomit/*.

VILLAIN-TO-VICTIM (V2V) PROTOCOLS, A NEW THREAT

Matthias Bauer, Institut für Informatik

Bauer amused us with several ways to transport or temporarily store data on correctly configured machines without the consent of the owner (i.e., in Web guest books, in ICMP-echo-request datagrams sent over connections with long RTTs, or in SMTP messages sent via open relays to domains that refuse to accept the messages for several days). In addition to providing an unreliable backup medium, these methods can be used to build an unobservable channel. He proposes that these theft-of-service attacks should be called "villain-to-

victim" computing because some of the engineering problems of P2P can be solved by V2V protocols.

For more information, visit *http://www1.informatik.unierlangen.de/~bauer/new/v2v.html* .

Detecting Manipulated Remote Call Streams

Jonathon Giffin, Bart Miller and Somesh Jha, University of Wisconsin

In a distributed grid computing environment, remotely executing processes send call requests back to the originating machine. A hostile user may manipulate these streams of calls. This technique statically analyzes the process's binary code at dispatch time and generates a model of all possible call sequences. As calls come back during execution, they're checked against the model, which detects some types of manipulation.

A Quantitative Analysis of Anonymous Communications

Yong Guan, Xinwen Fu, Riccardo Bettati, and Wei Zhao, Texas A&M University

This probabilistic analysis of rerouting systems found that longer paths don't necessarily provide better protection against sender identification. They also found that path complexity doesn't have a significant impact on the probability of identifying a sender. Additionally, the ease of identifying a sender increases as the number of compromised nodes in the system increases, but that growth is sublinear.

For more information, visit *http://netcamo.cs.tamu.edu/*.

Distributed Authorization with Hardware Tokens

Stefan Wieseckel and Matthias Bauer, Friedrich-Alexander-University Erlangen-Nuernberg

The authors have written a PAM module for user authentication to workstations based on RSA credentials stored on a Dallas Semiconductor Java-iButton. They use the KeyNote policy engine to make authorization decisions, which allows for complex trust relationships and delegation of authority. They do not presently address user or token revocation.

For more information, visit *http://www.wieseckel.de/ibutton_smartcard.html* .

Moving from Detection to Recovery and Analysis

George Dunlap, University of Michigan

Dunlap proposed a mechanism of rollback and selective replay of network events to aid in intrusion analysis and recovery. Being able to answer questions like "What if this packet had not been delivered?" or "What if this TCP session hadn't happened?" should facilitate debugging, forensic analysis, and intrusion detection signature development.

A Cryptanalysis of the High-Bandwidth Digital Content Protection (HDCP) System

Rob Johnson, Dawn Song, and David Wagner, University of California at Berkeley; Ian Goldberg, Zero Knowledge Systems; and Scott Crosby, Carnegie Mellon University.

HDCP is a proposed identity-based cryptosystem for use over the Digital Visual Interface bus, a consumer video bus already in widespread use. The authors found serious design flaws in HDCP which allow one to eavesdrop on HDCP communications, clone HDCP devices, and build an HDCP-compliant device that cannot be disabled via HDCP's Key Revocation facilities. Because of the DMCA mess (see page 7, the summary of "Reading Between the Lines: Lessons from the SDMI Challenge," particularly the question regarding whether a person would be at risk for summarizing the session), they aren't releasing the full details of their cryptanalysis.

Trust, Servers, and Clients

Sean Smith, Dartmouth University

WebALPS extends an SSL connection into a tamper-resistant coprocessor. By using the coprocessor as a trusted third party, sensitive information is protected from rogue server operators. Credit card information, for example, can be sent from the coprocessor via encrypted email to a merchant with the web hosting provider never having access to it.

Additionally, Smith described how SSL connections can be spoofed and presented an impressive demo in which Java script and DHTML were used to spoof the URL, the SSL warning windows, the SSL icon, and the certificate information.

For more information, visit *http://www.cs.dartmouth.edu/~pkilab.*

Source Router Approach to DDoS Defense

Jelena Mirkovic and Peter Reiher, University of California, Los Angeles

The authors propose a system to prevent a network from participating in a DDoS attack. Located at the source network router, the system watches for a drop-off in reverse traffic from a particular destination with heavy outgoing traffic. It then throttles all traffic to that destination while attempting to identify attacking flows and machines. The system is similar to MULTOPS, but its source side only, and its traffic models don't depend on packet ratios.

For more information, visit *http://fmg-www.cs.ucla.edu/ddos*.

SAVE: Source Address Validity Enforcement Protocol

Jun Li, Jelena Mirkovic, Mengqiu Wang, Peter Reiher, and Lixia Zhang, University of California, Los Angeles

SAVE is a new protocol for building incoming address tables at routers, even in the face of asymmetric routes. Those tables can be used to filter out packets with spoofed IP source addresses, build multicast trees, debug network problems, etc. To build the tables, SAVE sends valid source address information downstream along the paths used for delivery.

For more information, visit *http://fmg-www.cs.ucla.edu/adas/*.

Code Red, the Second Coming — From Whence Diurnal Cycles

Colleen Shannon and David Moore, CAIDA

Using the same system presented in the Denial of Service session on Wednesday morning, CAIDA analyzed the second round of Code Red. They observed that many of the infected hosts were using dynamic addressing, suggesting that the owners were not intentionally running IIS. The data also showed a clear diurnal pattern – one-third to one-half of infected machines were being turned on and off daily – again suggesting that these machines were not running production Web servers.

For more information, visit *http://www.caida.org/analysis/security/code-red/*.

Fast-Track Session Establishment for TLS

Hovav Shacham and Dan Boneh, Stanford University

The authors describe a new, "fast-track" handshake mechanism for TLS. A fast-track client caches a server's public parameters and certain client-server negotiated parameters in the course of an initial, enabling handshake; these need not be present on subsequent handshakes. The new mechanism reduces both network traffic and flows and requires no additional server state. The bandwidth savings are particularly relevant to wireless devices.

For more information, visit *http://crypto.stanford.edu/*.

Electromagnetic Attacks on Chip Cards

Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi, IBM Research

Chip cards and other devices leak substantially more information through electromagnetic emanations than through other side-channels such as power consumption and timing analysis. Additionally, the countermeasures for the other side-channel attacks are often insufficient to protect from electromagnetic attacks. Because of the sensitive nature of this work, the authors are working with interested parties to secure vulnerable devices prior to disclosing complete details.

For more information, visit *http://www.research.ibm.com/intsec*.

Password Authentication

Philippe Golle, Stanford University

Philippe Golle proposed a scheme for authenticating to a large number of Web sites with different passwords, while requiring the client to remember only a single master password. The scheme can be adapted to master passwords as short as 40 bits and can resist coalitions of up to three Web sites.

For more information, visit *http://crypto.stanford.edu/~pgolle*.

A Traffic Capture and Analysis Framework

Josh Gentry, Southwest Cyberport

Josh Gentry presented some Perl tools for collecting network statistics. The capture engine uses libpcap to collect traffic, does some pattern matching and analysis, and stores the results in Perl hashes. The command line client can query that data locally or over the network.

For more information, visit *http://www.systemstability.org/*.

Open Source Implementation of 802.1x

Arunesh Mishra, Maryland Information and Systems Security Lab, University of Maryland

Lib1x is an open source implementation of 802.1x, a port-based authentication mechanism for wireless networks that's intended to be an alternative to 802.11 WEP (see the WiP by Adam Stubblefield et al., above, for details on why an alternative is needed). Contributions are welcomed.

For more information, visit *http://www.missl.cs.umd.edu/1x/*.

[Photographs of the Symposium can be found at *http://www.usenix.org/events/sec01/index.html*]

Will the real Peter Honeyman please stand up!