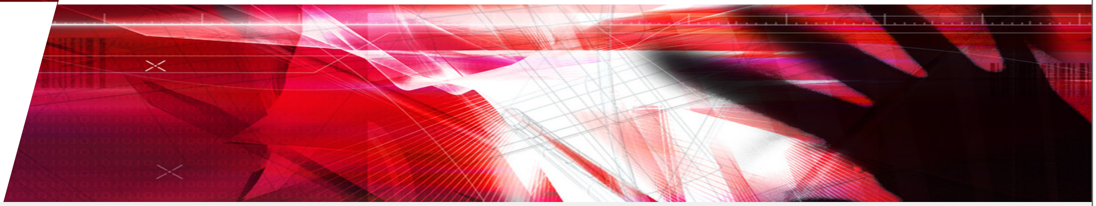
N°5 / Spring 2007

The Security Newsletter



In this issue

Published quarterly by:
Thomson Corporate Research
Part of the
Technology Division

Technical Editor:
Eric Diehl

Editor:
Nicholas de Wolff

Contributors:
Bertrand Chupeau
Olivier Courtay
Marc Eluard
Oliver Heen
Marc Joye
Mohamed Karroumi
Yves Maetz
Christophe Vincent
Çetin Koç (OSU)

SBU Technology Head:
Jean-Charles Hourcade

e-mail:
security.newsletter@thomson.net
Web:
www.thomson.net

This issue is attack oriented. We analyze two recent exploits. One exploit is a scientific masterpiece. The second one, although extremely clever, is not scientifically complex.

In 2006, a group of researchers designed a new type of attack. Modern processors use specific architecture to optimize execution time. By observing the execution time of OpenSSL on a remote computer, the researchers successfully extracted private RSA keys! One of the authors, Çetin KOÇ, agreed to explain the methodology behind the attack and forecast some future challenges.

The second exploit got extensive media coverage and raised the question: "is AACS already dead?" Hackers designed software that successfully unprotected digital copies of HD DVD and Blu-ray titles. We detail the attacks in this issue, and explore some solutions.

Although of different complexity, both attacks have one issue in common: the attacks targeted theoretically secure schemes. They attack the implementation of these schemes. Effective countermeasures will require careful design of new implementations.

In 1996, Paul KOCHER disclosed the first side-channel attack, highlighting the importance of secure implementation. Çetin Koç reminds us in this issue that new attacks will inevitably continue to shake the world of security. The AACS hack highlights that writing secure software requires a high level of expertise. Writing secure software is far more complex than just writing good software. It requires an extensive, up to date knowledge of the techniques used by attackers.

We find ourselves in limbo at present, as more and more systems rely less and less on tamper resistant hardware, while tamper resistant software remains in its infancy. Evaluation of the robustness of software to attacks is far from reliable. Both topics should attract more interest from academic researchers.

E. DIEHL, Technical Editor

THOMSON SECURITY LABORATORIES

Be our guest: Çetin Koç

INTERVIEW By M. JOYE

Çetin, some months ago, a variety of media outlets reported a new security flaw. How did you come up with the idea of branch prediction analysis?

Since the first papers on cache attacks were published, I have led my students at Oregon State University to implement and realize/simulate these attacks. Most of these students were M.S. students, who have less time to work on serious problems. Then, my PhD student Onur Acıiçmez was interested in this work. After having simulated cache attacks and other timing attacks on remote servers, we realized how important they can be. Sometime during 2004, we thought about other processor-bound side channels, and I even wrote a proposal and sent it to a certain global company; our proposal was ignored. Such is life! We had been collaborating with Werner Schindler on cache attacks, and we then started working with Jean-Pierre Seifert who was working for Intel at the time. Our collaborative efforts produced the branch prediction attack (CT-RSA 2007) and simple branch prediction attack (ACM Asia CCS 2007) papers.

The security arena is a perpetual battleground with implementers and attackers in unending conflict. Do you foresee new types of side-channel attacks?

Potentially, every unit within a commodity processor has a side channel, some more apparent than others. The branch prediction is the source of largest timing penalty in a processor; other side channels may not be that obvious. The power of a spy process in a server, constantly measuring the timing information from such side channels, cannot be ignored.

You are one of the leading experts in cryptographic engineering. This is a difficult research area as it is highly interdisciplinary. What are some of the most significant advances in the last ten years?

We have seen side-channel research develop from infancy into a mature field. Nearly a hundred researchers were involved in this work.

But, ever so quietly, the design and implementation of embedded software and hardware for cryptography is also advancing, powered by the security needs of mobile devices. Cryptographic engineering is an overlapped research and development field of electrical engineering, computer science, and mathematics. We need to continue to work in the same collaborative and interdisciplinary way to design for better security.

What are the most active research topics in your expertise area? What is the most challenging problem to tackle?

Side-channel analysis for commodity processors, and hardware and software countermeasures to circumvent such attacks will be important for a while. Innovative arithmetic representations and algorithms will continue to be also important, so will the design of true random number generators. A unified approach for designing secure execution platforms in order to support the commercial world (DRM systems and such) is something we are very much in need of. I consider this the greatest challenge.

In the News

CNC recommendations

French CNC (Centre National du Cinéma) and ALPA (Association pour la Lutte contre le Piratage Audiovisuel) recently published a guideline of good practices. This excellent guide suggests ten simple rules that may drastically reduce piracy risk in the AV production chain. An English version of this guide is available at http://www.cnc.fr/Site/Template/T8.aspx?SELECTID=2531&ID=1661&t=1

E. DIEHL

Are RFID payment cards weak?

Radio Frequency Identification (RFID) technology is used in many areas, including contact-less payment cards. Usability requirements of payment cards are very strong: including easy handling of the card and short transaction time.

A secure contact-less system should protect communication between card and reader, and data stored into the card. Unfortunately, current implementations do not protect communication; securing it would lead to unacceptable transaction delays. Thus, confidential and personal information are transmitted in plain text over an insecure channel. It is quick but unacceptably insecure. It is easy to eavesdrop on exchanged data and use it either to create a copy of the original card or to collect personal information.

Security is always a tradeoff [16]. Usability is sometimes more important than security. The ease with which one may duplicate magnetic stripes [17] or the English e-passport [18] are other illustrations. The problem is not exclusive to RFID technologies.

What is the weakest element? The technology, the card provider, the user or usablity? RFID technology can be trusted when employing proper cryptography.

M. ELUARD, Y. MAETZ

Help yourself

Many consumers complain about the limitations introduced by Copy Control for CD. To allow consumers to make private copies, EMI provides a very efficient tutorial. It explains step-by-step how to use the analog hole to make an "analog" digital copy of the protected CD [14]. Obviously, the copy is unprotected.

E. DIEHL

Viruses spread through GPS

This past February, the Dutch company TomTom™ had to admit that Trojan viruses infected some of their GO910 satellite navigation systems [20]. When the device connected to a host PC, the Trojans attempted to infect it. A security-savvy customer was alerted by his anti-virus software, when connecting his brand new TomTom GO910 to his computer. Unfortunately, TomTom did not take this infection as seriously as they should; first, they minimized the potential effects of the viruses, then they provided insufficient recommendations to fix the problem.

Unfortunately, this is not the first case of a virus affecting (or should that be "infecting"?) CE products. Last year, McDonald's® recalled 10,000 mp3 players infested with Trojan horses which stole passwords [21]. Meanwhile, Apple was shipping 1% of their iPods™ with malware [22]. In all these cases, the devices were contaminated during the manufacturing process.

Customers can no longer expect CE devices to be virus-proof. Indeed, Kapersky™ is starting to market anti-virus solutions for mobile phones. Cars, refrigerators, light bulbs; which device will be the next one on the list?

Y. MAETZ, C. VINCENT

Google Desktop Search vulnerability

Google Desktop Search (GDS) is a popular information retrieval tool. The key to its success is a mix of simplicity and tight integration with the browsing environment. Indeed, the GDS home page is incorporated within the www.google.com home page. This last feature may generate security problems in the case of malicious code injection: it opens a path from the Internet domain to the local GDS domain. In other words, it provides a way for an Internet attacker to browse the victim's data.

Researchers at www.watchfire.com discovered the vulnerability at the end of last year. On January 4, 2007, a full report was sent to Google. Google quietly released a patch on February 1, 2007. This patch was distributed via the automatic update feature of GDS. On February 21, 2007, the vulnerability was publicly acknowledged.

The attack exploits a combination of three factors: a vulnerability in google.com called XSS(*), a structural vulnerability in the GDS page protection mechanism, and the tight link between google.com and GDS pages. No factor is sufficient by itself, but the genius association of the three adds up to a powerful attack.

Via XSS, the attacker injects javascript code within a google.com page browsed by the victim. This code collects the cryptographic key that protects the local GDS page.

Via a page link, additional "sticky code" is embedded into each GDS request. This code retrieves the dangerous part of the attack (stored on any Internet host controlled by the attacker).

The dangerous part of the attack may launch any executable file already on the victim host, downgrading its local GDS security.

Versions 5.0.0701.30540 and above of GDS are secure. To check your version, right click on the GDS icon tray, and select "About".

(*) XSS stands for cross-site scripting. Many popular web enabled applications suffered from XSS including, quite recently, Adobe Acrobat Reader.

O. HEEN

AACS Under Fire

After CSS (the DVD copy protection system) was broken, a new copy protection system, called AACS, was designed for new high definition formats. AACS was publicly released in April 2005 and was adopted as a standard protection scheme for HD DVD and Blu-ray. Blu-ray adds another protection layer named BD+.

On December 26, 2006, a hacker under the alias "Muslix64" posted software on the Doom9 Internet forum [19]. The software, named BackupHDDVD, and its source code decrypt AACS. BackupHDDVD requires a cryptographic secret key to make an unprotected copy of an HD DVD disc. Muslix64 announced also that he could obtain this key but without providing more details about his method. BackupHDDVD is an implementation of the publicly available AACS specifications.

Muslix64 claimed, on January 13, 2007, that he found the needed secret key, with little effort, in main memory while playing discs using a legal HD DVD software player. The secret key was not obtained due to an AACS weakness but rather through a specific (careless?) implementation of AACS.

What was the weakness exploited by Muslix64? A security chain is only as strong as its weakest link. Here, the weakest link was the HD DVD player software running on the PC. Any encrypted movie has to be decrypted. Thus, the descrambling keys must be present at some point in the PC memory. It is very difficult to prevent an attacker from accessing this memory, Even though he still has to search for a l6 byte AES key in a large memory bank.

Muslix64 does not disclose how he recovered the key. He probably dumped the software memory and tested every

16 consecutive bytes of the memory as a candidate for the title key. This is a memory brute force attack. This is different from classical brute force attacks that try all possible keys (2^{128} possibilities in the case of AACS). In this consecutive test, the number of possibilities is drastically reduced. The remaining difficulty is how to know when the right key is found. The descrambled data have to follow MPEG syntax. Thus, any trial that does not generate MPEG compliant data can be discarded.

Once the key is recovered, the BackupHDDVD tool permits retrieval of a clear HD DVD disc image. It's not difficult to write a tool like BackupHDDVD because it is merely an implementation of the publicly available AACS specifications. Security of the system does not rely on the secrecy of the algorithm but rather on the secrecy of the keys. That was the philosophy adopted by AACS designers. AACS uses several cryptographic keys. Each entity (player, drive, disc) holds some of them. The security relies on the combination of all these keys.

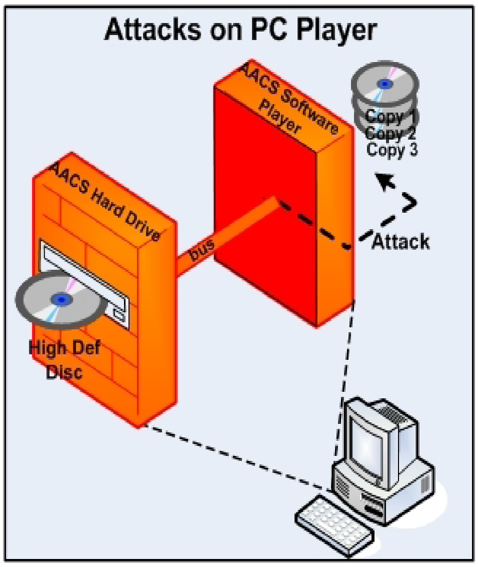
Muslix64 was the first hacker. Many attackers followed and considerably enhanced the attack and its usability. The main enhancement is the work of Arnezami. On Sunday 11 February 2007, he managed to extract a "processing key" (a master key in AACS) from an HD DVD player application. He says that this processing key can be used to decrypt all existing HD DVD and Blu-ray discs. This attack is more powerful than previous ones. Since then, automatic key retrieval tools have been published.

The speed of the attackers to create these exploits is alarming. They needed just a few days to extract the keys and design automatic tools. Thus, we may conjecture that the keys where not properly protected in at least one commercial implementation of AACS. Furthermore, they did not even have to reverse engineer the software.

Some countermeasures may mitigate these sorts of attacks:

- Secure coding, sometimes called code obfuscation, makes software more confusing and harder to interpret. This requires skilled developers in the secure coding field. However, code obfuscation does not assure absolute security. It must be regularly updated.

- Trusted Platform Module (TPM): more and more computers embed one TPM. This hardware module verifies the integrity of the operating system, and of the applications. Thus, the computer could prove to the HD DVD hard drive that a compliant application is running. Nevertheless, this does not prevent bad implementations from being written.



- Traitor tracing: traitor tracing is not an active protection methodology, but does manage to deter some attackers. To get clear HD DVD content, some cryptographic keys are needed. Some of these keys are associated to the player. Knowing these keys, AACS LA may revoke them. A revoked drive cannot decrypt new HD DVD discs.

- BD+, the additional protection layer of Blu-ray, uses a Security Virtual Machine (SVM). SVM may insert a player dependent invisible watermark.

All these countermeasures would make the system more robust. Nevertheless, each one has its own limitations. For example, TPM may not prevent hardware attacks, and traitor tracing may be sensitive to collusion attacks.

The attack described is not a class attack of AACS. It concerns weak implementations of decryption software. Consequently, player manufacturers should better protect secret keys in their memory.

O. COURTAY, M. KARROUMI

Fingerprinting Camcorders

Following their recent publications on digital camera identification [10], Jessica Fridrich and Mo Chen (State University of New York at Binghamton) presented an extension of their technique for identification of camcorders [5], at the last Electronic Imaging symposium. They claim to be able to determine whether two video clips came from the same camcorder or whether two differently transcoded versions ofthe same movie came from the same camcorder.

The same approach as for digital camera identification is used: the intrinsic noise pattern of the sensor, otherwise known as photo response non-uniformity (PRNU), is a robust fingerprint of the individual devices, unique to each imaging sensor. The PRNU captures the varying sensitivity of individual pixels to light, due to inhomogeneity and impurities in silicon wafers and imperfection due to the sensor manufacturing process. Despite the much smaller spatial resolution in video than present in still images, as well as the high compression rates, it was possible to adapt the technique developed for digital cameras, by taking advantage of the time resolution. The detection method involves (1) estimating the PRNUs $\hat{K}$ and $\hat{K}$, from two different incoming video clips, (2) removing the non sensor specific artifacts such as blockiness effects due to compression, and (3) deciding whether both clips were taken by the same camcorder, by detecting the presence of a pronounced peak in the normalized cross-correlation surface between $\hat{K}_A$ and $\hat{K}_B$

Experimental results proved that only 40 seconds of video is required for a reliable decision from clips encoded as low as 450 kb/sec. Good results are still obtained at even lower bit-rates and with decreased spatial resolution, at the expense of longer video clips: at 264×352 pixel resolution and 150 kb/sec., 10 minutes of video is needed.

Such a forensic technology appears very promising for fighting in-theater motion picture piracy, as an additional tool to traitor tracing or anti-camcorder techniques. This identification method is robust to compression and resizing, but more complex (e.g., geometric) video post-processing could hamper it, however. Another question is how such measurements could be accepted as a piece of evidence in court, when prosecuting a content pirate.

B. CHUPEAU

Side-Channel Attacks Against OpenSSL

In the last issue of our newsletter, we announced a new security flaw that could affect the security of Internet exchanges. By exploiting *branch prediction* capabilities that are present in most modern computer architectures, O. Acıiçmez, Ç.K. Koç, and J.-P. Seifert developed a new technique to collect almost all the secret key bits from a *single* execution of a cryptographic algorithm. We discuss below the details of their attack.

Side-channel attacks

The new attack belongs to the wider class of *side-channel attacks*. Side-channel attacks were introduced to the cryptographic community by P. Kocher in 1996 [8]. Contrary to the common, black-box cryptography context where an attacker has only access to the inputs and outputs of an algorithm, a side-channel attacker can get further information by *monitoring* the execution of the algorithm. Typical side channels include timing [8], power consumption [9], or electromagnetic radiation [7][12].

Initially restricted to dedicated cryptographic tokens like smart cards, side-channel attacks were later shown to be applicable against *remote servers*. In [4], D. Brumley and D. Boneh describe a successful attack against a remote server running an OpenSSL RSA implementation. More recently, side-channel attacks gained much attention by considering *cross-process* information leakage through memory caches (see for example [11] and the references therein).

The last results on side-channel attacks, due to O. Acıiçmez, Ç.K. Koç, and J.-P. Seifert, appear in [2], [1] and make use of *branch prediction analysis*. Applied to an implementation of OpenSSL RSA, they were able to recover almost all secret key bits from a *single* RSA signing execution

(Simple) Branch prediction analysis

Superscalar computer processors rely on deep pipelines and are able to fetch and issue multiple instructions per clock cycle. The deeper the pipelines are, the better the performances *should* be. However, *branchings* (e.g., jump instructions) may downgrade the performance, as the next instruction to be executed is not necessarily the next one in the instruction stream. More problematic are *conditional branchings* (e.g., "if-then-else" statements) because the *outcome* of the branch may only be available in later stages of the pipeline. To address these issues, chip manufacturers equipped CPUs with a *branch target buffer* (BTB) and a *branch predictor* (BP). The BTB is a buffer of *limited size* that acts as a cache for recording the target addresses of previously executed branches. The BP is an algorithm making use of tables and history registers to predict the most likely branch to be executed. In the case where the BP has selected a wrong branch (*misprediction*), the execution has to start again from the wrongly predicted branch (we note here that the penalty increases with the pipeline depth), which results in a longer execution time. This timing information is then used to collect secret information. See the appendix of this article for an example of a branch prediction attack that recovers an RSA signing key from multiple executions of the signing algorithm.

The attack presented in the appendix is easily mitigated using blinding (a.k.a. randomization) techniques. Three more sophisticated attacks are described in [2]. The most powerful attack is the one presented in [1]: it requires an (unprivileged) spy process running in parallel, which measures "locally" the execution time resulting from correct or incorrect branch predictions. Consequently, it can recover secret key bits from a *single* execution of the cryptographic algorithm under attack. Note that the use of blinding techniques in this case does not thwart the attack.

Implications

Similar to cache, the BP unit is a resource shared by different applications that may run concurrently. A simple way of preventing branch prediction analysis could be to deactivate the BP unit or to disable multi-process capabilities. Such an approach is, however, not desirable, as it greatly impacts performance.

The most powerful branch prediction attacks require a spy process that runs on the host under attack. Prohibiting the installation/execution of untrusted software may help to prevent these attacks. Blinding techniques may also be useful to prevent statistical attacks (cf. the appendix).

As was the case for the smart-card industry several years ago, software developers should change their way of programming and write code so that it is immune against branch prediction analysis (as well as against all similar attacks): *programs not only should be fast, they also need to be secure*. In the future, we may expect to see countermeasures available at the hardware level. These hardware countermeasures include the disabling of the BP unit, the use of randomized predictions, and finally unsharing, locking, or partitioning the BT buffers.

About the authors of the attack

• Ç.K. Koç is a Professor at Oregon State University (USA). He is well-known for his design of efficient algorithms and architectures for cryptography. He is also the co-founder of the Workshop on Cryptographic Hardware and Embedded Systems (CHES).

• J.-P. Seifert is a Professor at University of Innsbruck (Austria). His expertise mainly lies in the development of crypto-processors and countermeasures against various attacks. He was formerly affiliated with Intel and Infineon, two leading chip manufacturers.

• O. Acıiçmez has been a doctoral student under the supervision of Prof. Koç at Oregon State University. He successfully defended his thesis in December 2006.

Appendix

In order to illustrate the methodology behind side-channel analysis, we outline in this appendix a simplified presentation of the first attack detailed [2] (see also [6]).

Imagine that the RSA exponentiation, (m, d, N) ▯ $m^d \bmod N$, is carried out with the square-and-multiply algorithm. This binary exponentiation algorithm processes secret exponent $d = (d[k-1], d[k-2], \ldots, d[0])_2$ from the most significant bit $d[k-1]$ to the least significant bit $d[0]$.

Suppose that the attacker already knows the upper bits $d[k-1], d[k-2], \ldots, d[k-i+1]$. His goal is to recover the next unknown bit $d[k-i]$ from the knowledge of those upper bits. The attacker *guesses* that $d[k-i] = 0$ and prepares off-line two sets of 'messages', S_0 and S_1, containing respectively messages m such that the computation of $m^{(d[k-1], \ldots, d[k-i+1], \mathbf{0})} \bmod N$ causes or does not cause a wrong prediction of the BP during the i^{th} modular squaring. Next, the attacker runs the algorithm for all messages in each set and measures the running time. Let τ_0 and τ_1 denote respectively the average running time of the algorithm for messages in sets S_0 and S_1. Obviously, if the attacker correctly guessed the value of $d[k-i]$ (i.e., if $d[k-i]$ is indeed 0) then we have $\tau_0 > \tau_1$ as all messages in set S_0 give rise to a longer running time resulting from the wrong prediction. Otherwise, we have $\tau_0 \approx \tau_1$ as the partitioning behaves as a random splitting, and thus $d[k-i] = 1$.

Once the value of $d[k-i]$ is recovered, the attacker reiterates the previous attack from the knowledge $d[k-1], d[k-2], \ldots, d[k-i]$ to recover the value of $d[k-i-1]$, and so on until he recovers the whole value of exponent d.

M. JOYE, Ç. KOC

Where will we be?

- Marc JOYE, Invited Speaker,
  2nd International Conference
  on Security of Information
  and Networks (SIN 2007),
  Gazimagusa, North Cyprus,
  May 8-10, 2007

References

[1] ACIICMEZ O., KOÇ Ç.K., SEIFERT J.-P., ***On the power of simple branch prediction analysis***. In *2007 ACM Symposium on InformAtion, Computer and Communications Security* (ACM AsiaCCS 2007), ACM Press. To appear

[2] ACIICMEZ O., KOÇ Ç.K., SEIFERT J.-P., ***Predicting secret keys via branch prediction***. In *Topics in Cryptology – CT-RSA 2007*, vol. 4437 of Lecture Notes in Computer Science, Springer-Verlag, 2007

[3] AMIT Y. et al., ***Overtaking Google Desktop***, www.watchfire.com

[4] BRUMLEY D., BONEH D., ***Remote timing attacks are practical***, *Computer Networks* **48**(5), 2005

[5] CHEN M., FRIDRICH J., GOLJAN M., ***Source digital camcorder identification using CCD photo response nonuniformity***. In *Proc. Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, California, USA, 29 January – 1 February 2007, SPIE Vol. 6505

[6] DHEM J.-F., et al., ***A practical implementation of the timing attack***. In *Smart Card Research and Advanced Applications (CARDIS '98)*, vol. 1820 of Lecture Notes in Computer Science, Springer-Verlag, 2000

[7] GANDOLFI K., MOURTEL C., OLIVIER F., ***Electromagnetic analysis: Concrete results***. In *Cryptographic Hardware and Embedded Systems – CHES 2001*, vol. 2162 of Lecture Notes in Computer Science, Springer-Verlag, 2001

[8] KOCHER P.C., ***Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems***. In *Advances in Cryptology – Crypto'96*, vol. 1109 of Lecture Notes in Computer Science, Springer-Verlag, 1996

[9] KOCHER P.C., JAFFE J., JUN B., ***Differential Power Analysis***. In *Advances in Cryptology – CRYPTO '99*, vol. 1666 of Lecture Notes in Computer Science, Springer-Verlag, 1999

[10] LEFEBVRE F., ***Fingerprinting cameras***. *The Security Newsletter N°3*, Fall 2006

[11] OSVIK D.A., SHAMIR A., TROMER E., ***Cache attacks and countermeasures: the case of AES***. In *Topics in Cryptology – CT-RSA 2006*, vol. 3860 of Lecture Notes in Computer Science, Springer-Verlag, 2006

[12] QUISQUATER J.-J., SAMYDE D., ***Electromagnetic analysis (EMA): Measures and counter-measures for smart cards***. In *Smart Card Programming and Security (E-smart 2001)*, vol. 2140 of Lecture Notes in Computer Science, Springer-Verlag, 2001

[13] RAGER A., ***XSS-Proxy:*** a tool for real-time XSX. http://www.sourceforge.net

[14] http://copycontrol.emi-artistes.com/copie.html

[15] http://www.nytimes.com/2006/10/23/business/23card.html?pagewanted=1&ei=5090&en=76401b1601fc06e3&ex=131925600

[16] http://www.schneier.com/book-beyondfear.html

[17] http://www.schneier.com/blog/archives/2006/08/technological_a_1.html

[18] http://www.theregister.com/2007/03/06/daily_mail_passport_clone/

[19] http://forum.doom9.org/showthread.php?t=119871

[20] http://www.tomtom.com/news/category.php?ID=2&NID=349&Language=1

[[21] http://www.engadget.com/2006/10/16/mcdonalds-mp3-players-ship-with-trojan-horse/

[22] http://www.apple.com/support/windowsvirus/