July 15, 2000

Crypto-Gram Newsletter

by Bruce Schneier
Founder and CTO
Counterpane Internet Security, Inc.
schneier@schneier.com
http://www.counterpane.com

A free monthly newsletter providing summaries, analyses, insights, and commentaries on computer security and cryptography.

Back issues are available at http://www.schneier.com. To subscribe or unsubscribe, see below.

In this issue:

Full Disclosure and the CIA

On its Web site, the New York Times published a previously classified 1954 CIA document about the overthrow of the Iranian government. I'm not sure how the New York Times got its hands on the document, but the newspaper decided to redact the names of Iranian citizens involved in the plot.

The document was a scanned PDF file. What the Times did was to create an overlay on the pages, and drew black boxes over the names they wanted to obscure. The result was a file that still had the original digital information, but contained extra commands to obscure that information. Predictably, someone reverse-engineered the original names (it was suprisingly easy) and posted the restored document. The Times panicked and removed the original document, but it's too late; copies are available on various sites worldwide.

Reactions to this have been bizarre, and almost uniformly missing the point. Here's one, by William Hugh Murray: "Protecting the identity of intelligence sources is one of the few legitimate reasons for government secrecy. A 'freedom of information activist' does not serve his own cause, much less our national interest, by such reckless behavior." Ignoring Murray's hubris in claiming to know someone's motivations more strongly than he, what does this case have to do with government secrecy? It was the New York Times that decided to redact the document, not the CIA. The people who gave us the Pentagon Papers decided, on their own, not to release information that the CIA released to them. This is very different than the government trying to keep secrets. The last time I checked, the Times is not part of the intelligence community.

John Young, the activist Murray slams, makes classic full-disclosure arguments. The information was been published by the New York Times. Even though it took some work to extract, people are extracting it. Given this truth, it is better to release the information so that everyone can have access to it -- not only those random few who reverse-engineered the

PDF file. Young said "Those folks who are named have a stake in knowing about it."

The opposing side has an equally predictable reaction: The information should be kept secret. Just because the New York Times erred and made the information available to a few people doesn't mean that we should compound the error and make the information available to everyone.

It's a complicated question, made even more muddy by the New York Times's decision to redact the document. If the CIA gave the New York Times the document in unredacted form, why did the Times decide to limit the public's knowledge of this information?

To me, this is a classic example of the power of full disclosure. The information in question -- the names of the Iranian citizens -- has been released to the public. Initially, it was known by only those few who had the means to reverse-engineer the PDF file, and those who learned as the information spread. Left to its own devices, this information would spread slowly. Maybe someone who wants to do these people harm would learn about this information; he certainly already knows that this information exists. Maybe some of the people named would learn that they are named, and be able to take appropriate action. But this process is slow and haphazard.

Better is to release the information to everyone, quickly. This limits the damage that can be done. The people named are more likely to find out they are named, and they are more likely to find out about it quickly. Those who want to take advantage of the relative secrecy of the information cannot. Everyone knows, and this levels the playing field. It is the situation where only a few random people know, and others find out piecemeal, that is unstable and dangerous.

News articles:
<http://www.wired.com/news/politics/0,1283,37205,00.html>

<http://www.securityfocus.com/news/51>

The document:
<http://cryptome.org/cia-iran-all.htm>

Counterpane Internet Security News

Counterpane is pleased to announce a new insurance tie-in with Lloyd's of London. This is an exclusive offering for Counterpane customers: if Counterpane monitors your network, then you can purchase this insurance. For the first time ever, organizations can buy insurance against hacking without a security audit and without regard to the particular security products they use. Organizations can also buy, for the first time ever, warranty coverage that protects their customers against loss of income and data.

Computer security has always been sold as "threat prevention." Encryption, firewalls, anti-virus, PKI...these are all technologies that prevent particular threats. Threat prevention is a cost, and if an organization doesn't understand the threats, then it might not be willing to pay for prevention. Real business security, on the other hand, is about risk management. Risk management is not a cost, it's a way to make money. If one organization can manage its risk better than another, then it will be more profitable. Smart companies embrace risk, look for more of it, and figure out how to do business in the face of it.

Looking at computer security as a risk management tool, there are many more options than threat prevention. There is detection and response: managing risk by finding attacks in process and stopping them. And there is insurance: packaging risk and selling it to someone else. These are the future of computer security, not prophylactic technologies that promise a magical security blanket.

From the beginning, I have maintained that Counterpane Internet Security will address the real problems in network security. I have never believed that simply installing products will ever protect you, and have focused on the process of security. One part of that process is Managed Security Monitoring, which is what Counterpane's business is. The other part is insurance. Now Counterpane customers, and only Counterpane customers, have access to both.

Summary of the offering:

Counterpane Internet Security Inc., Lloyd's of London, Frank Crystal & Co., and SafeOnline Ltd. have jointly developed a first-of-its-kind, comprehensive risk management insurance solution specifically targeted to meet the needs of today's e-businesses.

Up to $100 million in protection available.

Two products available:

1. Internet Asset and Income Protection Coverage provides insurance for Counterpane's Managed Security Monitoring customers who incur a loss of or damage to information assets resulting from a breach of security or technology failure. Also covers business interruption due to loss of use due to a breach.

2. Internet Asset and Income Protection Warranty Plan for ISPs/ASPs that utilize Counterpane's Managed Security Monitoring services; this is a turnkey, insurance-backed warranty plan to extend the Internet Asset and Income Protection to their customers.

These insurance products are sold through authorized insurance brokers.

Quick Summary:
<http://www.counterpane.com/pr-lloydssl.html>

Press Release:
<http://www.counterpane.com/pr-lloyds.html>

Q&A:
<http://www.counterpane.com/pr-lloydsqa.html>

White Paper describing the insurance offering and its context:
<http://www.counterpane.com/pr-lloydswp.html>

Press Coverage:
<http://news.cnet.com/news//0-1005-200-2232221.html?...>
<http://www.internetwk.com/story/INW20000710S0001>
<http://www.computerworld.com/cwi/story/...>
<http://www.zdnet.com/zdnn/stories/news/...>
<http://www.crn.com/dailies/digest/dailyarchives.asp?...>
<http://slashdot.org/articles/00/07/10/139201.shtml>

More Counterpane Internet Security News

More information on Counterpane's Managed Security Monitoring service can be found at:
<http://www.counterpane.com/oursol.html> [link dead; please see the Counterpane home page] <http://www.counterpane.com/coverage.html> [link dead; please contact Counterpane for information on supported sensors]

Bruce Schneier is speaking at BlackHat (July 26) and DefCon (July 29) in Las Vegas.
<http://www.blackhat.com>
<http://www.defcon.org>

Bruce Schneier will address the Digital Commerce Society of Boston on August 1.

At WebDefense, in Washington DC (Aug 9), Bruce Schneier will give a talk entitled "What Level of Security Can We Reasonably Expect?" <http://www.acius.net/webd_overview.html>

For the full Counterpane conference schedule, see: <http://www.counterpane.com/conf.html>

News

Another interesting article on writing computer viruses: <http://www.hackernews.com/bufferoverflow/99/nitmar/...>

Software (children's software, no less) that automatically spies on you: <http://www.salon.com/tech/col/garf/2000/06/15/brodcast/>

The need for security. Good article. <http://www.acm.org/crossroads/columns/onpatrol/...>

Problems with public-key infrastructures. This article was written well before I started thinking about the problems: <http://world.std.com/~dtd/compliance/compliance.ps>

Countries are conducting military reconnaissance on U.S. computer networks: <http://www.zdnet.com/zdtv/cybercrime/...>

Provocative articles on the risks of the new digital signature law: <http://www.pfir.org/statements/2000-06-17> <http://www.securityfocus.com/templates/article.html?...>

Interesting cryptography story. A political party in Mexico wants an encryption key broken, because it believes that the resulting plaintext will embarrass the ruling party. <http://www.theregister.co.uk/content/1/11508.html>

<http://www.wired.com/news/politics/0,1283,37337,00.html>

Commentary on the hype surrounding computer-security press coverage.
<http://www.computerworld.com/cwi/story/frame/...>
I don't agree: I think that the current media hype will desensitize people to the real threats and the real risks.

NATO releases a virus. (There's enough odd in this story for me to doubt its veracity, but who knows.)
<http://www.the-times.co.uk/news/pages/sti/2000/06/...>

The motives and psychology of the black-hat community. Another good essay by Lance Spitzner.
<http://www.securityfocus.com/focus/ids/articles/kye/...> [link moved to http://www.securityfocus.com/infocus/1448]

Publius is a system for anonymous, censorship-resistant publishing on the Web.
<http://cs.nyu.edu/waldman/publius/>
<http://www.washingtonpost.com/wp-dyn/articles/...>
They're looking for people willing to host a Publius server.

The Secret Service is developing a Electronic Crimes Special Agent Program. Agents analyze new security products and alert vendors to weaknesses.
<http://www.fcw.com/fcw/articles/2000/0626/...>

Failing dot-coms are selling private information:
<http://news.cnet.com/news/0-1007-200-2176430.html>
More proof that they should never be allowed to collect it in the first place, and that posted privacy policies aren't worth the phosphors they're displayed on.

The Sega Dreamcast copy-protection scheme has been cracked. Is

anyone surprised?
<http://www.mhzgaming.com/articles/dc629.htm>
<http://news.cnet.com/news/0-1005-200-2181596.html>

Special news report on Echelon:
<http://www.zdnet.co.uk/news/specials/2000/06/echelon/>
The government of France is investigating:
<http://dailynews.yahoo.com/h/nm/20000704/ts/...>
<http://news6.thdo.bbc.co.uk/hi/english/world/europe/...>

Interesting article on the lack of security in Australian government Web sites:
<http://www.theregister.co.uk/content/1/11686.html>
And other articles on the same topic:
<http://www.it.fairfax.com.au/breaking/20000630/...>
<http://www.afr.com.au/news/20000701/...>

There were rumors of hackers disrupting the Space Shuttle a few years ago. Here is what seems to be a coherent story:
<http://news.excite.com/news/ap/000702/20/...>
NASA denies it:
<http://news.excite.com/news/ap/000703/19/nasa-hacker>
The question to ask is: why in the world are these critical systems attached to public networks in the first place?

Electronic Commerce: Who Carries the Risk of Fraud?
<http://www.fipr.org/WhoCarriesRiskOfFraud.htm>

NIST has published an index of computer vulnerabilities. Called ICAT, it links users into a variety of publicly available vulnerability databases and patch sites. ICAT is not itself a vulnerability database, but instead a searchable index leading one to vulnerability resources and patch information. ICAT allows one to search at a fine granularity, a feature unavailable with most vulnerability databases, by characterizing each

vulnerability by over 40 attributes (including software name and version number). ICAT indexes the information available in CERT advisories, ISS X-Force, Security Focus, NT Bugtraq, Bugtraq, and a variety of vendor security and patch bulletins. ICAT uses the CVE vulnerability naming standard.
<http://csrc.nist.gov/icat>

The insecurity of wireless computing:
<http://www.msnbc.com/news/429748.asp?cp1=1>

New technology for recovering erased data on electronic media:
<http://www.sciencenews.org/20000708/fob1.asp>

Good editorial. The biggest problem in computer security is the user, not the computer.
<http://www.osopinion.com/Opinions/DanielHiggs/...>

Lots of "pay-to-surf" sites are springing up, paying users to look at web advertisements, along with a cottage industry of programs that help rogue surfers bypass the rules and get paid anyway.
<http://www.wired.com/news/culture/...>

Social Security numbers of the rich and famous:
<http://news.cnet.com/news/0-1005-200-340248.html>

Internet voting is unsafe (no surprise), says a new study.
News article:
<http://www.wired.com/news/politics/0,1283,37504,00.html>
Actual study:
<http://www.voting-integrity.org/text/2000/...>

Spam with huge media attachments. Anyone care to ponder the implications for virus writers?
<http://adage.com/interactive/articles/20000710/...>

"The Carnivore," from the FBI. An expert system that seaches through email.
<http://www.zdnet.com/zdnn/stories/news/...>
The ACLU complains:
<http://www.wired.com/news/politics/0,1283,37470,00.html>

The Cyber Group Network Corp claims to have a technology that allows you to locate a stolen computer, remotely retrieve information from it, and the destroy it. Sounds a bit far fetched. But they take "security by obscurity to new heights: "According to Nish Kapoor, a spokesperson for The Cyber Group Network, the patent pending technology that makes all this possible is being manufactured and developed at a remote, top-secret location identified only as 'Area 74.'" Wow.
<http://www.newsbytes.com/pubNews/00/151921.html>

And even worse, software that allows companies to track down (and presumably prosecute) those who say unkind things about them on the Internet. What happened to free speech on the Internet?
<http://www.businessweek.com/bwdaily/dnflash/...>

Even the President Can't Choose a Good Password

According to a story regarding Clinton signing the e-signature bill with a smart card:

"With a magnetic card and his dog Buddy's name as a password, President Clinton e-signed a bill Friday that will make electronic signatures as real as those on paper."

Note his choice of password. It's five characters, all alphas, and probably all lower case. At least he doesn't keep it on a Post-it in his desk.

<http://www.foxnews.com:80/elections/063000/...>

The Doghouse: Intuit QuickBooks

Intuit's QuickBooks is a finance manager for small companies. It has lots of great features and is one of the most popular financial packages available. Unfortunately, the security sucks.

The earliest versions of QuickBooks stored usernames and passwords in the clear. As the version number increased, so did the amount of obfuscation: first they XORed the data with a constant mask, then they added some byte rotations and more bit twiddling. The latest versions of QuickBooks use DES to encrypt blocks of data containing a username and the corresponding password. This could have made the system much harder to break, if only the decryption keys weren't stored in the file.

To verify that the user has the correct password, QuickBooks looks up the decryption key, decrypts the block, and compares the password the user typed in to the one it just decrypted. Nothing prevents an unauthorized program from performing the same steps and printing out all the usernames and passwords.

Thanks to Mike Stay for pointing this out.

Full Disclosure and Lockmaking

"In respect to lock-making, there can scarcely be such a thing as dishonesty of intention: the inventor produces a lock which he honestly thinks will possess such and such qualities; and he declares his belief to the world. If others differ from him in opinion concerning those qualities, it is open to them to say so; and the discussion, truthfully conducted, must lead to public advantage: the discussion stimulates curiosity, and curiosity stimulates invention. Nothing but a partial and limited view of the question could lead to the opinion that harm can result: if there be harm, it will be much more than counterbalanced by good."

-- Charles Tomlinson's Rudimentary Treatise on the Construction of Locks, published around 1850.

Crypto-Gram Reprints

Those of you who have subscribed recently might have missed these essays from back issues.

Declassifying SKIPJACK:
<http://www.schneier.com/crypto-gram-9807.html#skip>

The Future of Crypto-Hacking:
<http://www.schneier.com/...>

Bungled SSL:
<http://www.schneier.com/...>

Security Risks of Unicode

Unicode is an international character set. Like ASCII, it provides a standard correspondence between the binary numbers that computers understand and the letters, digits, and punctuation that people understand. But unlike ASCII, it seeks to provide a code for every character in every language in the world. To do this requires more than 256 characters, the 8-bit ASCII character set; default Unicode uses 16-bit characters, and there are rules to extend even that.

I don't know if anyone has considered the security implications of this.

Remember all those input validation attacks that were based on replacing characters with alternate representations, or that explored alternative delimiters? For example, there was a hole in the IRIX Web server: if you could replace spaces with tabs you could fool the parser, and you could

use hexadecimal escapes, strange quoting, and nulls to defeat input validation.

The Unicode specification includes all sorts of complicated new escape sequences. They have things called UTF-8 and UTF-16, which allow several possible representations of various character codes, several different places where control-characters pop through, a scheme for placing diacriticals and accents in separated characters (looking very much like an escape), and hundreds of brand new punctuation characters and otherwise nonalphabetic characters.

The philosophy behind the Unicode spec is to provide all possible useful characters for applications that are 8- or 16-bit clean. This is admirable, but it is nearly impossible to filter a Unicode character stream to decide what is "safe" in some application and what is not.

What happens when:

- We start attaching semantics to the new characters as delimiters, white space, etc? With thousands of characters and new characters being added all the time, it will be extremely difficult to categorize all the possible characters consistently, and where there is inconsistency, there tends to be security holes.

- Somebody uses "modifier" characters in an unexpected way?

- Somebody uses UTF-8 or UTF-16 to encode a conventional character in a novel way to bypass validation checks?

With the ASCII character set, we could carefully study a small selection of characters, categorize them clearly, and make relatively straightforward decisions about the nature of each character. And even here, there have been mistakes (forgetting about tabs, multicharacter control-sequence snafus, etc). Still, a careful designer can figure out a safe way to deal with

any possible character that can come off an untrusted wire by elimination if necessary.

With Unicode, we probably won't be able to get a consistent definition of what to accept, what is a delimiter under what circumstance, or how to handle arbitrary streams safely. It's just a matter of time before simple validators pass data and upper layer software, trying to be helpful, attach magic-character semantics, and we have a brand-new variety of security holes.

Unicode is just too complex to ever be secure.

Unicode:
<http://www.unicode.org/>

My thanks to Jeffrey Streifling, who provided much of the material for this article.

Comments from Readers

> **From: Bernard Peek <Bernard@postar.co.uk> Subject: Remotely Disabling Software**

> In the UK there is legal protection against disabling software by remote commands. In general this is illegal without the users' prior consent. It could be permitted if the user agreed to a software license that contained appropriate clauses. As I understand it, in the absence of a prior agreement any supplier that disabled a program running on a machine in the UK would be committing an offence within UK jurisdiction. Whether they triggered the event from elsewhere is not a defence.

> I believe that one software supplier embedded an undisclosed time-trap in their program which would disable it. The trap was sprung if the

> supplier failed to switch it off. They would only do that if the customer failed to pay the final invoice.
>
> One invoice was disputed and the trap was sprung. The software supplier was successfully prosecuted and had to pay a fine and damages.
>
> I note, with some interest, that some files have security certificates with expiry dates, and that some programs will not work with expired files. I trust that the suppliers of these files have mechanisms in place to replace all of them prior to their expiry date, or have the customer's signature on a software license that details the conditions under which the programs cease to work.

From: Matt Curtin <cmcurtin@interhack.net> Subject: Breaking DES Keys by Brute Force

> In the June 15 issue of CRYPTO-GRAM, you discuss the (in)security of DES because of its relatively short key length.
>
> > Concerns about its short key length have dogged the
> > algorithm since the beginning, and in 1998 a brute- > force machine capable of breaking DES was built.
>
> John Gilmore and company made an important advancement with Deep Crack in 1998, breaking DES keys very quickly. But the first break of a DES key was actually accomplished in 1997 in software by a team led by Rocke Verser. Justin Dolske and I wrote a paper describing the architecture of the system that was published in the May 1998 special issue of ;login.
>
> This seems an important thing to mention not just because we can move the time that it took to break DES keys back another year, but because it has been demonstrated (twice -- another team did the

> same thing the following year) that the key length was so short that it could be broken in software using only random idle cycles from computers talking together over a simple protocol. Before this is dismissed as an impractical attack, consider that it's believed that recent distributed denial-of-service attacks have been perpetrated by attackers who take over others' machines without their knowledge and then coordinate these "zombie machines" to do their work. If attackers can do this for launching DoS attacks, it could be done to break keys. (Though because of the inefficiency of the attack, this won't be used for anyone but attackers with no money, I'd think...)

> The technical article is online at <http://www.interhack.net/pubs/des-key-crack/> and a nontechnical article (originally written for the press) briefly describing cryptography, brute force attacks, and our project's success is at <http://www.interhack.net/projects/deschall/what.html>.

> **From: Paul Bissex <pb@e-scribe.com> Subject: Re: Cryptogram 6/15, Williams/Agre**

> A quick note on J. Christopher Williams' response to Phil Agre. I believe Mr. Williams has a legitimate difference of opinion here, though I do not share his views. However, to say that "[Agre's] grasp of basic grammar is less than firm" is simply laughable, especially after one has struggled through a few dozen lines of Mr. Williams' stilted prose.

> There are few people who write about technology with as much intelligence, erudition, and accuracy as Phil Agre. It is unfortunate that Mr. Williams felt the need to turn his nitpick into a personal attack.

> From: "Jay R. Ashworth" <jra@baylink.com>
> Subject: Comments to Schneier on Agre

> In comments published in the June Cryptogram, Mr. Williams, you take

> issue with the comments by Phil Agre pointed to by, I believe, the previous edition of the newsletter...
>
> Microsoft wrote:
> > "... This is by-design behavior, not a security > vulnerability."
>
> Agre commented:
> > "More odd language. It's like saying, 'This is a rock,
> > not something that can fall to the ground'. It's
> > confusing to even think about it.
> > ..."
>
> You observed:
> > The author may be a security or computer expert, but
> > his grasp of basic grammar is less than firm. The more
> > accurate grammar conversion would be "This is an
> > object thrown to the ground, not a free-falling
> > object." The statement in and of itself merely assists
> > in defining what can be called "thrown" and what can
> > be called "free-falling" campaign the author suggests.
> > To me the author is seemingly engaging in the same
> > "blame shifting tactic" that he accuses Microsoft of.
>
> The way *I* see this is this: Agre feels, and I agree with him, that Microsoft, in its phrasing, is suggesting that since [the behavior in question] is by design, that it *cannot be* a security vulnerability; that is, that they are mutually exclusive.
>
> If in fact this is the assertion that Microsoft is trying to cause people to infer, they are wrong. For evidence, we need merely look to one of the most famous security holes of all time: the sendmail DEBUG command.
>
> Obviously this was placed in the software by design. Equally obviously, it was a security vulnerability. Therefore, it must be possible for an

> element of a program to be both of these things.
>
> Therefore, Microsoft's explanation must be disingenuous, as they are trying to assert that these two descriptions of a part of a program cannot both simultaneously be true.
>
> The problem isn't Phil doesn't understand the language, it's that Microsoft is playing fast and loose with it.
>
> **From: kragen@pobox.com (Kragen Sitaker) Subject: Risks of XML**
>
> You write:
> > XML and how to secure it:
> > <http://www.zdnet.co.uk/news/2000/20/ns-15500.html> [link moved to http://news.zdnet.co.uk/story/0,,s2079073,00.html]
>
> This topic is sort of like "ASCII and how to secure it." This paragraph says it best:
>
> > And that is the one big problem. The Internet is as
> > insecure as ever, and ASCII will do nothing to improve
> > it. In fact, the temptation to intercept and alter an
> > ASCII document containing vital data en route from one
> > banking application to another will lure many an
> > Internet vandal.
>
> Well, in the original, it said XML, not ASCII, but it makes just as much sense either way.
>
> From: Darren Cervantes <cervante@roguewave.com>
> Subject: SOAP Security
>
> Your newsletter was forwarded to me recently. I am interested in your opinion on SOAP security issues. Your recent article suggested that any SOAP command may be able to sneak through to your machine

> and do something malicious. You state: Firewalls have good reasons for blocking protocols like DCOM coming from untrusted sources. Protocols that sneak them through are not what's wanted.

> In my understanding of the SOAP protocol, a few things have to happen in order for an RPC to "sneak" through:
> 1) An explicit SOAP server has to be set up
> 2) The SOAP server has to "translate" SOAP to enterprise services or RPCs
> 3) That server has to be configured to execute particular SOAP services (in other words if the service isn't defined, it isn't available through SOAP)
> 4) The server has an opportunity to recognize the SOAP service and refuse or block it if it is invalid

> In addition, in the MS SOAP document, they state that both HTTP encryption (SSL) and HTTP authentication protocols can be used in conjunction with any end-point security (presumably authorization). This would help negate the "untrusted sources" factor.

> Based on my understanding, you would have to set up a SOAP server and an explicit service to "erase harddrive." You would then have to allow your server to accept the request. In a sense, it seems that this same concern could be equally applied to many other technologies. If SOAP were to be implemented irresponsibly, it could certainly be insecure, but isn't it possible to implement it securely?

CRYPTO-GRAM is a free monthly newsletter providing summaries, analyses, insights, and commentaries on computer security and cryptography.

To subscribe, visit http://www.schneier.com/crypto-gram.html or send a blank message to crypto-gram-subscribe@chaparraltree.com. To

unsubscribe, visit http://www.schneier.com/crypto-gram-faq.html. Back issues are available at http://www.schneier.com.

Please feel free to forward CRYPTO-GRAM to colleagues and friends who will find it valuable. Permission is granted to reprint CRYPTO-GRAM, as long as it is reprinted in its entirety.

CRYPTO-GRAM is written by Bruce Schneier. Schneier is founder and CTO of Counterpane Internet Security Inc., the author of "Applied Cryptography," and an inventor of the Blowfish, Twofish, and Yarrow algorithms. He served on the board of the International Association for Cryptologic Research, EPIC, and VTW. He is a frequent writer and lecturer on computer security and cryptography.

Counterpane Internet Security, Inc. is a venture-funded company bringing innovative managed security solutions to the enterprise.

http://www.counterpane.com/

next issue
previous issue
back to Crypto-Gram index

Schneier.com is a personal website. Opinions expressed are not necessarily those of BT.