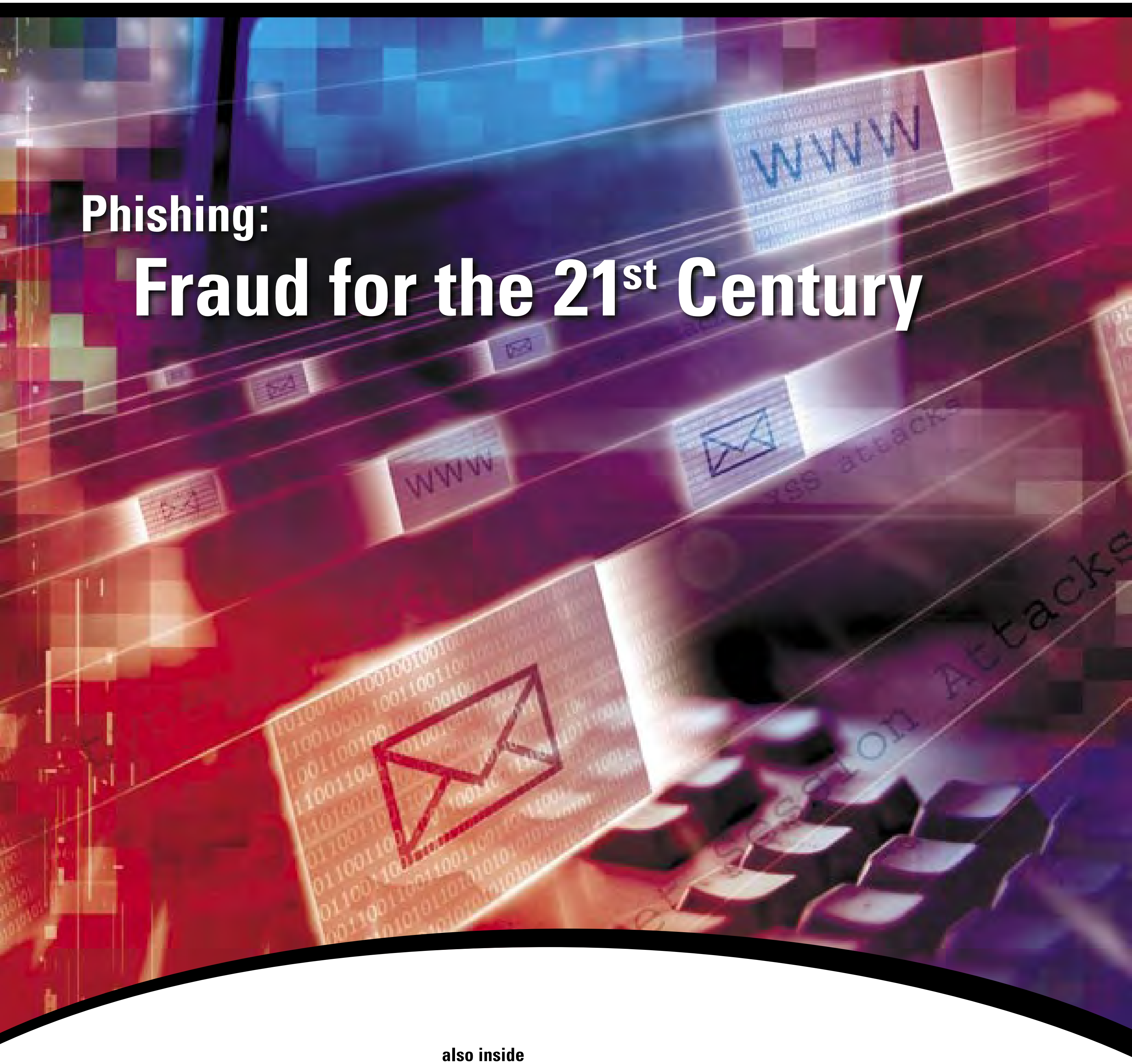
9/4

Volume 9 Number 4 • **Winter 2006**

IAnewsletter

The Newsletter for Information Assurance Technology Professionals

Phishing: Fraud for the 21st Century



also inside

Verifying Network Intrusion Detection Alerts

Data Integrity and Proof of Service in BitTorrent-Like P2P Environments

An Overview of Voice over Internet Protocol (VoIP)

Countering DDoS Attacks with Multi-Path Overlay Networks

Ask the Expert

IATAC Spotlight on Education

IATAC Spotlight on Research

contents

About IATAC and the *IAnewsletter*
The *IAnewsletter* is published quarterly by the Information Assurance Technology Analysis Center (IATAC). IATAC is a Department of Defense (DoD) sponsored Information Analysis Center, administratively managed by the Defense Technical Information Center (DTIC), and Director, Defense Research and Engineering (DDR&E).

Contents of the *IAnewsletter* are not necessarily the official views of or endorsed by the US Government, DoD, or DDR&E. The mention of commercial products and/or does not imply endorsement by DoD or DDR&E.

Inquiries about IATAC capabilities, products, and services may be addressed to—

IATAC Director:	Gene Tyler
Inquiry Services:	Peggy O'Connor

***IAnewsletter* Staff**

Promotional Director:	Christina P. McNemar
Creative Director:	Ahnie Jenkins
Art Directors:	Bryn Farrar Don Rowe
Copy Editor:	Diane Ivone
Designers:	Dustin Hurt Kacy Cummings Ricardo Real Brittany Biggs
Editorial Board:	Ronald Ritchey Tara Shea Gene Tyler Buzz Walsh

***IAnewsletter* Article Submissions**
To submit your articles, notices, programs, or ideas for future issues, please visit http://iac.dtic.mil/iatac/IA_newsletter.html and download an "Article Instructions" packet.

***IAnewsletter* Address Changes/Additions/Deletions**
To change, add, or delete your mailing or e-mail address (soft-copy receipt), please contact us at—

IATAC
Attn: Peggy O'Connor
13200 Woodland Park Road
Suite 6031
Herndon, VA 20171

Phone:	703/984-0775
Fax:	703/984-0773
E-mail:	iatac@dtic.mil
URL:	http://iac.dtic.mil/iatac

Deadlines for future Issues

Spring 2007	Dec. 30th, 2006

Cover design:	Ricardo Real Dustin Hurt
Newsletter design:	Bryn Farrar Donald Rowe

Distribution Statement A:
Approved for public release;
distribution is unlimited.

feature

4 **Phishing: Fraud for the 21st Century**
Phishing refers to a new form of cyber crime that is quickly gaining popularity. During the past several years, there has been a steady increase in the use of online financial services for everything from paying utility bills to conducting banking and brokerage transactions.

10 **Ask the Expert: Sharing the Wealth**
A significant challenge facing Information Assurance (IA) professionals today is the general reluctance to share techniques and ideas that could further the overall IA practice.

12 **Verifying Network Intrusion Detection Alerts**
We developed a simple and effective method to determine the result of signature-based NIDS alerts. It demonstrates that protocol analysis can significantly increase the quality of NIDS alerts.

15 **IATAC Spotlight on Education**
The University of California, Davis (UC Davis) Department of Computer Science (CS) within the College of Engineering was founded in 1993 to advance the research frontiers of computer science while producing outstanding computer science professionals.

16 **Data Integrity and Proof of Service in BitTorrent-Like P2P Environments**
The mSSL protocol offers a security solution in this environment, and in this article we described its approach to data integrity and proof of service.

20 **An Overview of Voice over Internet Protocol (VoIP)**
VoIP is an up-and-coming technology that has a good possibility of replacing current telephone technologies.

25 **IATAC Spotlight on Research**
The SME profiled in this article is Dr. Matt Bishop. Dr. Bishop has been a Professor at the Department of Computer Science at University of California, Davis, since 2004 and is a co-director of the UC Davis Computer Security Laboratory.

26 **Countering DDoS Attacks with Multi-Path Overlay Networks**
Our approach offers an attractive solution against congestion-based DDoS attacks in most environments, as it does not require modifications to clients, servers, protocols, or routers, both in terms of hardware and in existing software.

in every issue

3 IATAC Chat
30 Letter to the Editor
31 Product Order Form
32 Calendar

IATAC Chat

Gene Tyler, IATAC Director

IATAC strives to find, build, and maintain strong relationships with those who provide value and Scientific and Technological Information (STI) to the Information Assurance (IA) community.

As we close out calendar year 2006, I want to focus on some of this year's important IATAC events. Early in the year, we made a big move to our new One Dulles facility in Herndon, VA, where we hosted an Information Assurance Technology Analysis Center (IATAC) Open House for government, industry, and academia. In conjunction with our move, we decided to enhance the face we show to the Information Assurance (IA) community we support. We made changes to our website (*http://iac.dtic.mil/iatac*), created a new face for the *IAnewsletter* and other products, and launched the IATAC Research Update. The update is an IA quarterly email publication that is a targeted voice for the Research and Development (R&D) and academic communities. The Research Update focuses on the Scientific and Technological Information (STI) to which we have access. If you would like to receive this publication or any IATAC product, please visit our website to subscribe. One recent transformation is one that you may have already noticed. We now offer the IA Digest as a Real Simple Syndication (RSS) feed, a file format used to distribute news and news-style content quickly and easily over the Internet. While we still offer the IA Digest in its original email format, the RSS feed allows us to distribute it *via* another media. When you visit our website, click on the RSS icon in the email format and copy the URL to your reader—now you have it, easy! Our appearance may have changed, and we may be adding new tools, but our mission of being the central point of access for IA remains the same.

IATAC strives to find, build, and maintain strong relationships with those who provide value and STI to the IA community. There are many behind-the-scenes activities, and not everything is highlighted in the *IAnewsletter* or other publications. But, as shown by the above changes, some events demand special attention, and our relationship with The Institute for Applied Network Security (IANS) is one such event. Recently, IATAC has been collaborating with IANS, which has a solid IA relationship with the commercial world in knowledge, experience, and STI that the IATAC community may find beneficial. In this edition of the *IAnewsletter*, you will find a new an article, *Sharing the Wealth*, provided by IANS. In future editions of the *IAnewsletter*, IANS will contribute an *Ask the Expert* article highlighting responses from questions that IANS' members ask of the Institute. These questions and responses have an IA perspective, as view by IA professionals servicing their community—analogous to industry best practices. IATAC will work closely with IANS in its future forums to provide a government perspective. The first event will be IANS' Mid-Atlantic Information Security Forum, 5–6 March 2007, at the Sheraton Premier Hotel at Tyson's Corner, VA. I hope to see you there—I will be! You may learn more about this event and IANS on its website, *http://www.ianetsec.com*. You can also learn more about the Mid-Atlantic Forum on the IATAC website and in the Information Assurance/Information Operations (IA/IO) Events Scheduler. IATAC will be responsible for a track in this forum. I encourage all to visit IANS' website. I think you will see they have capabilities that are interesting and valuable to us all.

In this edition of the *IAnewsletter*, you will find, as always, many intelligently written articles. From a Voice over Internet Protocol (VoIP) article written by one of our Core Team members, to an intriguing article on Intrusion Detection, co-authored by this edition's featured Subject Matter Expert (SME), there is something for everyone. I would also like to mention that we are always interested in new articles for inclusion in the *IAnewsletter*. If you have a knack for writing about IA or related topics and have an interest in being published, please let us know, or visit our website for more information on how to submit your work. In closing, I would like to take this opportunity to wish you all a joyous holiday season and a Happy New Year.

Gene Tyler

FEATURE STORY

Phishing: Fraud for the 21st Century

by Lawrence Lauderdale and Ron Ritchey

When you hear the term "phishing," you may think of spending a lazy day by a river, waiting for fish to bite. Unfortunately, phishing refers to a new form of cyber crime that is quickly gaining popularity. During the past several years, there has been a steady increase in the use of online financial services for everything from paying utility bills to conducting banking and brokerage transactions. This has been closely followed by an increase of phishing attacks. Because of increased media attention, many people have heard the term but don't have a clear idea of what it really is. So what *is* phishing and what can be done about it?

What is Phishing?

Phishing is a type of social engineering attack that centers on email. Its goal is to fraudulently acquire sensitive information through a website (or, in some cases, the telephone [1]) by posing as a legitimate entity and requesting the information. The initial emails that are sent as part of the attack can be spoofed to appear to be from a legitimate organization because of well-known weaknesses in the Simple Mail Transport Protocol (SMTP) that permit a sender to set header fields such as FROM: and REPLY-TO: to any arbitrary value. The information gathered by phishers may be credit-card information, personal information such as a Social Security Number (SSN), log-in information to financial sites, or even a corporate user's information for use in corporate espionage. These attacks use social engineering techniques, exploit technological deficiencies in web browsers and websites, and commonly exploit human emotion to harvest their targeted information. Common themes used in phishing attacks include "Your account has been fraudulently accessed;" "Systems have changed, and you will lose access if you do not respond;" "Please verify we have your correct contact information;" *etc.* These attacks play on the human emotions of fear and curiosity to drive people to act without thinking rationally and to cause them to ignore warning signs. Real-world examples of phishing attacks will be shown later in this article.

Phishing—The Problem is Only Getting Worse

According to the Anti-Phishing Working Group (*http://www.antiphishing.org*), an industry association focused on eliminating identity theft and fraud that result from the growing problem of phishing and email spoofing, there were 14,191 [2] phishing sites detected in July 2006 that attempted to hijack 154 unique brands. That figure represents a 211% increase from 4,564 [3] just one year before in July 2005 and a 12,134% increase from 116 [4] in December 2003, the first month recorded by the organization. Obviously, the problem is quickly getting worse . Figure 1 shows the increase in number of new phishing sites reported each month from July 2005 through July 2006.

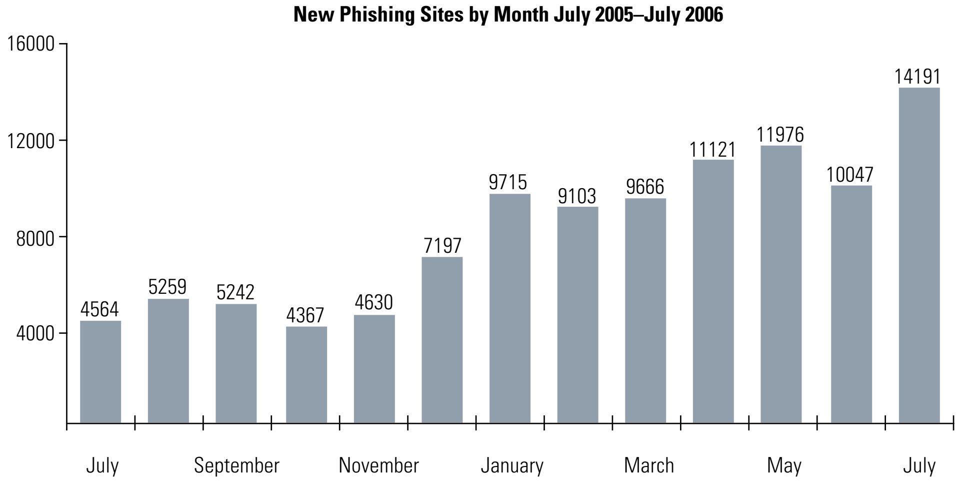


Figure 1 Anti-Phishing Working Group report

Common Targets

Although the first thing that may pop into most people's heads when they think of phishing is emails that request eBay credentials, phishing has moved away from targeting retail brands. The overwhelming majority of phishing attacks are targeted at where the money is. In July, 2006, 93.5% [5] of all reported phishing attacks were targeted at financial services. A distant second, at 3.2%, were Internet Service Providers (ISPs), followed by 1.9% of other market sectors. Bringing up the rear, with just 1.3%, was the retail sector. Although the majority of financial institutions targeted by phishing scams are based in the United States, RSA Security, Inc. reports that, in August 2006, 27% [6] of financial institutions targeted by phishing attacks were non-US brands. This is a 6% increase from the month before and the first time that non-US financial institutions have made up more than 25% of phishing targets. Thus far, the targeted international brands have been mainly European-based financial institutions, with the United Kingdom and Spain leading the way. (See Figure 2).

Although the current common targets of phishing attacks have been financial institutions, some recent attacks have been unique. During the last US tax-filing season, phishers targeted US taxpayers by posing as representatives of the IRS. This is a major concern for several reasons—

- The IRS commonly deals with sensitive information that could be used to conduct identity theft.
- The target audience of this type of attack is *all* US taxpayers (approximately 134 million [7], as of the 2005 filing season).
- Most US taxpayers are intimidated by or afraid of the IRS.

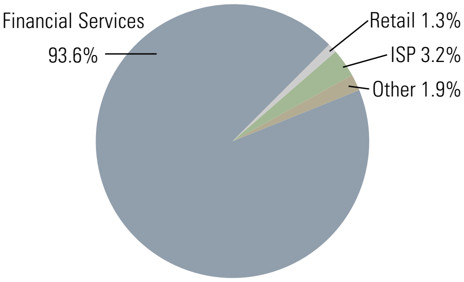


Figure 2 Phishing target sectors

This fear of an audit or other retribution in response to missing information from a return can easily be leveraged by an attacker and used to gather information. Another target of phishing fraud surfaced during the aftermath of Hurricane Katrina. Many phishing sites were uploaded that purported to be gathering relief funds for Hurricane Katrina victims but instead were used to gather credit-card and other information. Earlier this year, research into domain name registrations revealed that fraudsters were gearing up to launch similar attacks if any hurricane named for the 2006 hurricane season caused major damage. These examples indicate that phishers and fraudsters are looking to sociological events to help define and refine their phishing attempts, which shows how far they are willing to go and what lengths must be taken to curb their success.

Why Phishers Phish

Phishing is a growing problem—but *why?* In many ways, the answer is based on the same reasons that SPAM emails are successful. SPAM succeeds because it is very cheap to mail very large numbers of messages. Even very low response rates are sufficient to justify the cost. Because of the nature of phishing attacks, not that many people have to fall for each attack for the attack to be financially successful. According to research conducted by Gartner, Inc. between May 2004 and May 2005, 73 million US adults received phishing attempts, and of these 1.2 million reported losing collectively $929 million [8] as a result of the scams. That computes to approximately $774 per phishing victim. This figure represents an increase from $674 per victim, based on the same study conducted between April 2003 and April 2004 that reported $1.2 billion [9] in losses by 1.78 million consumers. The bottom line? Even though only a small percentage of phishing targets respond, enough people are duped to make phishing very lucrative.

Life Cycle of a Phishing Attack

Although the details of each phishing attack vary widely, the life cycle of the attack follows the same general pattern. At some point, an attacker chooses a target. The target may be chosen because of a vulnerability in the site that permits an attacker to inject code and engineer a more believable attack, but, in most cases, any site can be targeted. An attacker pulls a copy of the pages he requires to get the information he seeks. This is usually just a log-in page, but in some cases phishers copy or create account information pages that include more than just log-in information (address, credit-card number, SSN, *etc.*). The attacker then puts up a web server, most often on a machine that has been compromised by an unpatched computer vulnerability. Then the phisher sends email to a potential victim, using the scheme he has cooked up. He may send email to millions of addresses and hope for the best, or he may gather a target list that he knows have accounts with his targeted institution. Either way, he is likely to get at least some responses. After victims have visited the phishing site and the phisher has extracted the information he wants, victims are very often forwarded to a real site so they don't suspect that something has gone awry. If the attacker is targeting log-in information for an online account, he will often design his site to verify the credentials he has received. Once verified, these credentials can be sold in bulk on the underground market. Depending on the harvested information, the buyer can then use the credentials to electronically transfer funds using "mules" or even perform outright identity theft by opening new credit accounts in the victim's name. The part that makes phishing so successful and hard to stop is that the life cycle of the typical phishing attack is less than five days. [10]

Why Phishing Works

So *why* does phishing work? With so many warning signs in real-world phishing attacks, it is surprising that so many people fall for them. Several groups [11, 12] have done research to better understand why users are fooled by these messages. Their results show that the success of phishing attacks can be attributed to four weaknesses—

1. Lack of user and consumer knowledge
2. Ineffective browser security indicators and warnings
3. Bad practices employed by targeted websites
4. Technological deceptions exploiting software vulnerabilities and weaknesses

Lack of Knowledge

Many studies have found that the leading cause of successful phishing attacks is that users and consumers don't possess the knowledge they need to detect an attack. Although most of the general public is wary of email they receive, many lack the knowledge to detect the other warning signs that are found in a typical attack. A recent study [13] showed that a good phishing site fooled 90% of site visitors. Most users lack the working knowledge of how computer browsers, Secure Socket Layer (SSL) certificates, and Uniform Resource Locators (URLs) work and how to use them to ensure that a session is secure. This lack of knowledge facilitates phishing attacks. And, unfortunately, those who don't know how these features work, don't *know* they don't know.

Most users do not understand the syntax of URLs and how they can be manipulated in a multitude of ways to obfuscate the real site that they point to. For example, the following are all essentially the same URL and will direct you to the same site:

http://www.google.com (Normal)
http://64.233.161.147 (IP Address)
http://www.someothersite.com@www.google.com (With a user)
http://www.google.com@1089053075 (Decimal with a user)
http://0x40E9A193 (Hexadecimal)

You can see that some URLs would be a bit confusing, and most users don't know that the @ symbol in a URL means to log in as the user specified *before* the @ symbol on the site that is specified *after* the @ symbol. Luckily, most recent browsers will throw a warning when a username is used with a website to protect against these types of attacks.

A surprisingly large number of people will not examine any browser error messages and click past them. The most common reason? They don't understand what they mean and are so used to seeing pop-ups that they immediately click *OK* to close them.

This allows phishers to set invalid SSL certificates, knowing that this will fool a certain percentage of the population. The general population just does not understand the trust relationships that Public Key Infrastructure (PKI) authentication is built on. Certificate information can be viewed by double clicking on the "lock" icon in the status bar of most browsers. This information should be checked and verified each time a user submits highly sensitive information. Fortunately, most browsers will give users warnings if inconsistencies are found in SSL certificates.

Most users also do not understand the significance of browser SSL indicators that appear in the "chrome" of the browser *vs.* those that appear in the content of the page and can be easily fooled by "lock" icons imbedded in the webpage. The "chrome" of the browser is the portion of the browser window that displays indicators from the browser, rather than from the webpage being visited. The "chrome" includes the URL bar and status bars. These same users would also be fooled by "tested secure" messages embedded in the page. Finally, some users are not familiar with how simple it is to create a fake FROM: and REPLY-TO: headers to make it appear as if an email came from a legitimate organization.

Ineffective Browser Indicators

Although designers of web browsers are well aware of the phishing problem and have made major improvements in all major browsers during the last few years, it is apparent that the browser indicators in use today are just not effective. Firefox, one of the most popular browsers today, has introduced new design features to aid users in determining if a session is SSL encrypted. Figure 3 shows an example of the real Paypal site, as viewed through Firefox 1.5.0.7, beside a fake *http://www.paypa1.com*. Although the browser shows four separate indicators (yellow background in URL field, "lock" icon in URL field, certificate domain in status bar, and "lock" icon in status bar) that indicate the real site is SSL encrypted and the fake site is not, the majority of users simply do not notice.

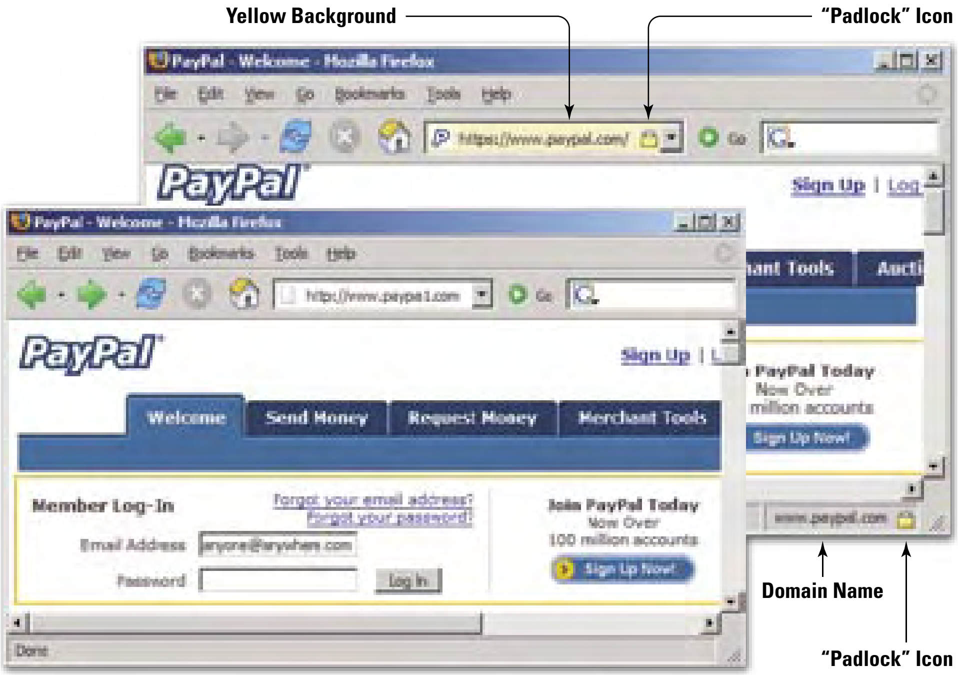


Figure 3 Browser indicators example

Bad Practices

Another reason phishing has been successful is poor website design. As an example, many financial sites have recently moved log-in pages from dedicated pages to their main, unencrypted landing page. To log in, a user enters his or her authentication credentials on a non-encrypted page. The user log-in data is then sent to an SSL encrypted page. While this is effective in protecting user credentials from eavesdropping and does slightly reduce the load on the financial institution's SSL accelerators, it permits users to become used to entering credentials before they have authenticated the site they are sending them to. This effectively lowers the importance of the SSL indicators in the mind of the common user.

Technological Deceptions

Many methods fall into the realm of technologic deceptions. Phishing attacks have used "typejacking" or "cousin domains," which are domains that appear to be the site targeted by the phishing attack. An example might be *http://www.paypa1.com* or *http://www.citibánk.com*. Other technological deceptions used by attackers include the following—

- **Proxies**—Using proxies to execute a Man-In-The-Middle (MITM) attack so that a user sees the real site and the phisher retrieves credentials as they are sent.
- **Cross–Site Scripting (XSS)**—Exploiting sites that are vulnerable to XSS attacks so that the targeted site directs the user to the attacker's site.
- **Pre-set Session Attacks**—Sending a link to the real website with a pre-specified Session ID and then polling the targeted site until the session is authenticated.
- **Deceptive Graphics**—Using graphics to mimic browser security cues such as the URL bar and status bar.

Anatomy of a Phishing Email

Figure 4 shows an example of a phishing email. You can see in this example that the phisher was able to gather information about a user through other means. This enabled the phisher to gain a user's trust by specifying his or her name and partial account number (outlined in red). You can also see that the phisher used fear to initiate a response by indicating unusual activity on the account in question (outlined in blue). Finally, the phisher included a link that is obfuscated to hide the actual IP address of the site to be visited.

Real-World Examples

The following examples are real phishing attempts that were reported to the Anti-Phishing Working Group. These are just a few examples of the more then 200 phishing examples in the phishing archive on the APWG site.

Figure 5 shows a phishing attempt that makes use of a simple "cousin" domain of *http://www.ameritrading.net* to entice victims who may have accounts with Ameritrade. The *real* domain name of Ameritrade is *http://www.tdameritrade.com* or *http://www.ameritrade.com*. This attempt also uses the scare tactic of telling the victim that a new account has been opened in his or her name.

Figure 6 shows a more sophisticated phishing attack using an encoded URL in the email to send users to a non-SSL encrypted site, then uses graphics to display a fake URL bar showing an Hypertext Transfer Protocol (HTTPS) URL. You can see they also used a "lock" icon on the page to help entice the victim, and that there is no "lock" icon in the browser chrome of the status bar. Again, this plays on fear by telling the victim that his or her account has been accessed from another country.

Anti-phishing

You've seen some examples of phishing, but the real question everyone is asking is what can be done to *prevent* damages caused by phishing? Many security vendors are offering services to help organizations curb phishing attacks on their websites, and web browser designers are integrating anti-phishing features. While these technological advancements help,

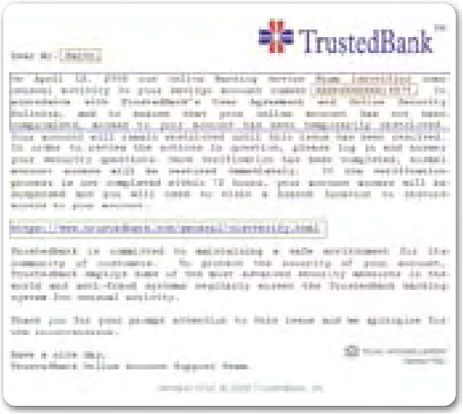


Figure 4 Example of a phishing email

they must be used in conjunction with proper user and consumer education to make a real difference.

Browser Enhancements

The two most popular web browsers with approximately 90% [14] of the web browser market share between them, Microsoft Internet Explorer and Firefox, are leading the way in anti-phishing enhancements. Firefox is planning to integrate the Google Safe Browsing feature of the Google Toolbar into Firefox 2.0 to help web surfers identify fraudulent websites. Microsoft is also implementing a similar feature in the upcoming release of Internet Explorer 7. Both features will work on the same principle. Each request that the user's web browser makes is checked against a list of known phishing sites. If the site is on the list, then the user is blocked from the site. This methodology has merit, but it also does have some drawbacks. It relies on each entity's ability to quickly identify phishing sites and add them to appropriate databases. This method suffers from the same shortcoming as those of anti-virus software and Intrusion Detection Systems (IDSs) in that they will not protect initial victims and will only prevent damages after the signature is added to the database.

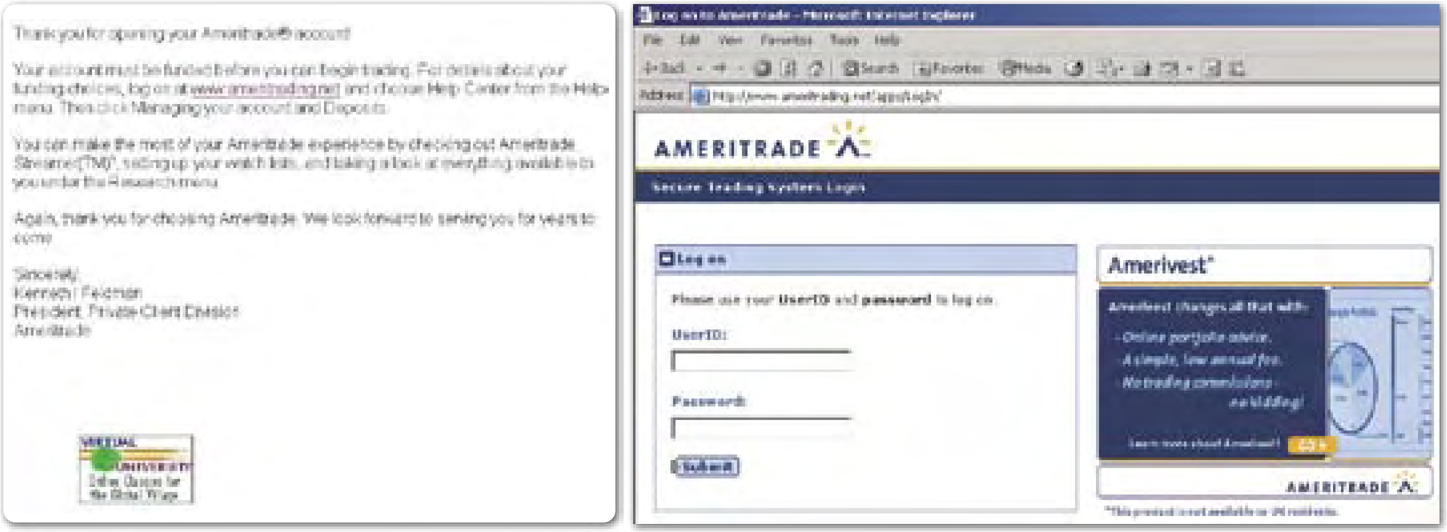
Vendor Services

Many security vendors offer a broad variety of anti-phishing and brand protection services. Some of these include Brandimensions, MarkMonitor, RSA Security (formerly Cyota), Verisign, and many more. (For a complete list, visit *http://www.antiphishing.org.*) Some services include "cousin" and "typo" domain monitoring and registration, site blocking, site takedown, honeypot account tracing, adaptive and strong authentication, and watermarking of site content. Each solution provides a different type of protection from phishing attacks.

One anti-phishing product that has been adopted by some major financial-services sites such as Bank of America, Stanford Federal Credit Union, and ING Direct, is PassMark SiteKey. The idea is simple. When a user signs up for an account they register their computer by having it "fingerprinted." This is done by creating a user-specific token and placing it onto the user's system with a browser cookie. The user also chooses a picture and a security phrase. On subsequent visits to the site, when the user signs into his or her account, they enter only an account number. The site then checks the fingerprint cookie on the computer. If it matches the one on file, then the user's picture and phrase are displayed. This confirms to the user that the site is the real site. If the computer is not one that has been associated with the account, then the site will ask more security questions before displaying the security picture to prevent phishers from harvesting the pictures and phrases to use on a phishing site.

For "cousin" and typo domain monitoring and registration, vendors work with domain name registrars to identify domains that match certain criteria. Then the vendors either monitor the domains to see if they are registered to an entity other than the proper organization or register them for the organization and provide a redirect to the legitimate site. Domain name monitoring can provide a jump-start on a phishing attack by possibly detecting the attack before emails are sent out.

Vendors also provide more reactive services such as site takedown and site blocking. Site blocking is a temporary solution that prevents users from getting to a fraudulent site until the vendor can


Thank you for opening your Ameritrade® account!

Your account must be funded before you can begin trading. For details about your funding choices, log on at www.ameritrading.net and choose Help Center from the Help menu. Then click Managing your account and Deposits.

You can make the most of your Ameritrade experience by checking out Ameritrade Streamer(TM)*, setting up your watch lists, and taking a look at everything available to you under the Research menu.

Again, thank you for choosing Ameritrade. We look forward to serving you for years to come.

Sincerely,
Kenneth J. Feldman
President, Private Client Division
Ameritrade

Virtual University
Online Classes for the Global Village
Log on to Ameritrade - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.ameritrading.net/apps/login/
AMERITRADE
Secure Trading System Login
Log on
Submit
Amerivest*
Learn more about Amerivest! GO
AMERITRADE

Please use your UserID and password to log on.

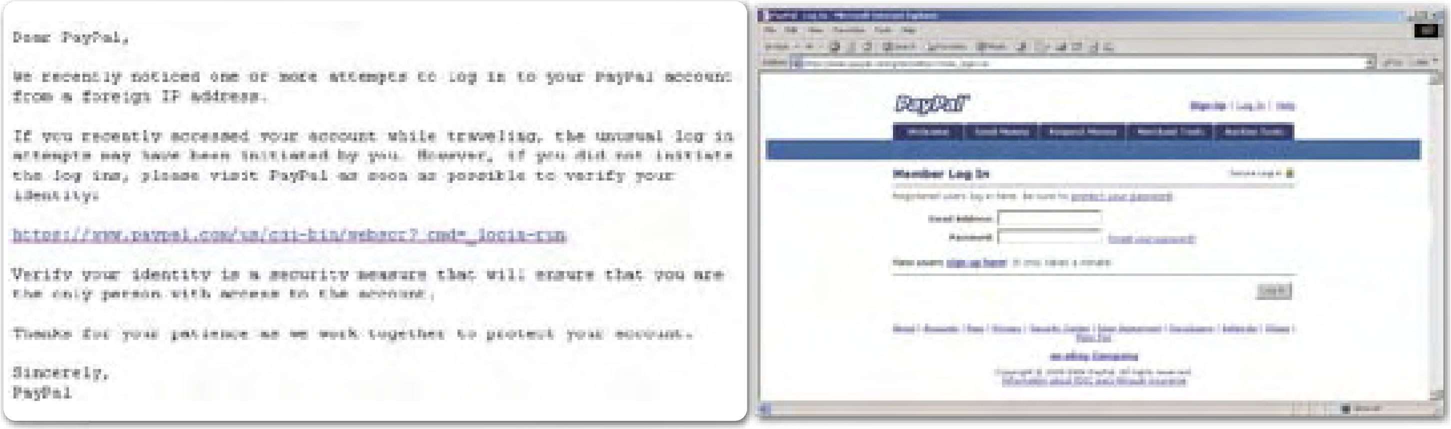
UserID:

Password:

Amerivest changes all that with:
- Online portfolio advice.
- A simple, low annual fee.
- No trading commissions - no kidding!

*This product is not available to UK residents.

Figure 5 Ameritrade phishing example. Images courtesy of the Anti-Phishing Working Group (http://www.antiphishing.org)



Dear PayPal,

We recently noticed one or more attempts to log in to your PayPal account from a foreign IP address.

If you recently accessed your account while traveling, the unusual log in attempts may have been initiated by you. However, if you did not initiate the log ins, please visit PayPal as soon as possible to verify your identity:

https://www.paypal.com/us/cgi-bin/webscr?cmd=_login-run

Verify your identity is a security measure that will ensure that you are the only person with access to the account.

Thanks for your patience as we work together to protect your account.

Sincerely,
PayPal

Figure 6 Paypal phishing example. Images courtesy of the Anti-Phishing Working Group (http://www.antiphishing.org)

find and take down the site. To block the site, the vendor works with domain name registrars and major ISPs to block access to the site shortly after an attack is launched. Once an attack is detected, vendors work with ISPs worldwide to identify and take down the site by removing the machine from the web.

One technique that has been very useful in limiting the impact of phishing attacks is the use of "honeypot" accounts. The vendor sets up a set of bogus site accounts, credit cards, and bank accounts. When a new phishing site is found, these accounts are submitted to the site before it is taken down. The vendor can then track these accounts to see if a phisher attempts to access them. If this happens, all transaction details are recorded, including the source IP addresses of machines that access these accounts. The vendor can then use this information to determine other accounts that were victims of the phishing attack and take action before money begins to disappear. They can also use this information to set up blacklists of IPs that are known to have used the "honeypot" accounts. By cross-referencing between many phishing attacks, a vendor is able to quickly build a blacklist of machines used during phishing attacks.

What You Can Do—As an Organization

The following is a list of some actions an organization can take to fight phishing—

- **Use consistent branding**—This includes email FROM: and REPLY-TO: fields and will force a phisher to use your branding, thereby making attacks easier to detect.
- **Issue notification**—Continuously notify users of communication policies.
- **Use personalization in correspondence**—An email missing a simple "Dear Mr. Smith" will cause users to notice something is wrong.
- **Use digital signatures for email**—Although most web based email does not support digital signatures, all traditional email applications (Thunderbird, Outlook, *etc.*) do.
- **Use simple URLs**—Try to avoid confusing URLs so that users will become accustomed to looking at them and "decoding" them for validity.
- **Monitor email bounce-back**—If a phisher uses a real REPLY-TO header to be consistent with your brand, the email bounce-back may provide an early warning of a phishing attack.
- **Monitor referrer sites**—Many phishers refer users to the real site after information is harvested. Monitoring referrers can help detect a phishing attack and identify the phishing site.
- **Gather information about phishing attempts**—Provide a formal method for users to report phishing attempts against your brand.
- **Use strong authentication**—If your information is important enough, use strong authentication such as SecurID tokens.
- **Watermark content**—This can be used to trace an attack and determine who pulled a copy of the site.

What You Can Do—As a Consumer and User

The following is a list of some actions a consumer or user may take to fight phishing—

- **Keep all patches up to date and use anti-virus software and a personal firewall**—These are three actions you should already be taking to help prevent your computers from being used in a phishing scam.
- **Disable dangerous browser functionality**—Disable browser functionality that can be used to deceive you during a phishing attack. Some of these include Pop-ups, Java, and ActiveX.
- **Use plug-ins**—Install browser anti-phishing plug-ins.
- **Be alert to suspicious emails**—Be wary of unexpected emails, especially those that "warn" you about something.
- **Avoid following links in email**—Contact the sender using a URL you are familiar with, or call a telephone number that you can confirm is legitimate (*e.g.*, from a statement or back of a bank card).

- **Verify that a site is encrypted**—Look for the HTTPs URL and verify the SSL certificate by clicking on the "lock" icon in your browser's status bar.

Conclusion

Phishing is a serious problem that continues to grow worse and cause severe financial loses worldwide. As long as there is money to be made, phishing will continue to plague society. Both anti-phishing vendors and web browser developers have made major strides to help prevent the success of phishing attacks; however, these efforts will never be perfect and must be combined with user and consumer education to effectively stem the flow of phishing attacks. ■

References

1. Skoudis, E. (2006, June 13). *Phone phishing: The role of VoIP in phishing attacks*. Retrieved September 21, 2006, from *http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1193304,00.html*
2. Anti-Phishing Working Group. (2006 July). *Phishing Activity Trends Report*. Retrieved September 21, 2006, from *http://www.antiphishing.org/reports/apwg_report_july_2006.pdf*
3. *Ibid.*
4. Anti-Phishing Working Group. (2004 January). *phishing Activity Trends Report*. Retrieved September 21, 2006, from *http://www.antiphishing.org/reports/APWG.Phishing.Attack.Report.Jan2004.pdf*
5. Anti-Phishing Working Group. (2006, July). *Phishing Activity Trends Report*. Retrieved September 21, 2006, from *http://www.antiphishing.org/reports/apwg_report_july_2006.pdf*
6. RSA Security. (2006 August). *Monthly Online Fraud Intelligence Report*. Retrieved September 21, 2006, from *http://www.rsasecurity.com/solutions/consumer_authentication/intelreport/RSA%20Online%20Fraud%20Intel%20Report%20-%20Aug%202006.pdf*
7. Internal Revenue Service (IRS). *Individual Tax Filing Statistics Filing Year 2006*. Retrieved September 21, 2006, from *http://www.irs.gov/taxstats/index.html*
8. Gartner Press Release. *Gartner Survey Shows Frequent Data Security Lapses and Increased Cyber Attacks Damage Consumer Trust in Online Commerce*. Retrieved September 21, 2006, from *http://www.gartner.com/press_releases/asset_129754_11.html*
9. *Gartner Press Release. Gartner Study Finds Significant Increase in E-Mail phishing Attacks.* Retrieved September 21, 2006, from *http://www.gartner.com/press_releases/asset_71087_11.html*
10. Anti-Phishing Working Group. (2006 July). *Phishing Activity Trends Report*. Retrieved September 21, 2006, from *http://www.antiphishing.org/reports/apwg_report_july_2006.pdf*
11. Dhamija, R., Tygar J. D., & Hearst, M. (2006 April). *Why phishing Works*. Retrieved September 21, 2006, from *http://people.deas.harvard.edu/~rachna/papers/why_phishing_works.pdf*
12. *University Study Why phishing Works*. Retrieved September 21, 2006, from *http://www.securityfocus.com/brief/176*
13. Dhamija, R., Tygar J. D., & Hearst, M. (2006 April). *Why phishing Works*. Retrieved September 21, 2006, from *http://people.deas.harvard.edu/~rachna/papers/why_phishing_works.pdf*
14. W3Schools. *Browser Statistics*. Retrieved September 21, 2006, from *http://www.w3schools.com/browsers/browsers_stats.asp*

About the Authors

Lawrence Lauderdale | is an Information Security Consultant. He has more than five years experience in network penetration testing and specializes in social engineering and wireless network security testing. His interests include assembly-level programming and software reverse engineering. He may be contacted at iatac@dtic.mil.

Ronald Ritchey | is the Chief Scientist of the Information Assurance Technology Analysis Center (IATAC). He has over 20 years experience researching information technologies and is completely fascinated by the challenges we face in securing them. His role in IATAC permits him to be actively engaged across the Information Assurance (IA) community, in which he regularly learns from organizations that are at the leading edge of this exciting field. He may be contacted at iatac@dtic.mil.

ASK THE EXPERT

Sharing the Wealth

by Jack Phillips

A significant challenge facing Information Assurance (IA) professionals today is the general reluctance to share techniques and ideas that could further the overall IA practice. Within both government and industry, the fear of the "CNN moment," coupled with competitive concerns and confidentiality requirements, means that the most promising techniques and practices developed within an organization often cannot be shared with the IA community at large. The goal of this column is to help bridge the information gap between IA professionals in government and those in industry by documenting some of the more useful ideas coming from the commercial sector today.

Over the past five years, the Institute for Applied Network Security (IANS)[1] has assembled a vibrant community of experienced IA professionals from the world's most important commercial, educational, and government organizations. IANS moderates an ongoing series of in-person, telephone-based, and online gatherings among these professionals and captures the most critical ideas in written form for its members. This column will gather emerging trends and challenges from our community to allow us to share with you the most strategic issues that our member organizations are facing.

The topic for this inaugural column is data security. In the US Department of Defense (DoD), data security is much more formalized, but within industry, it is an emerging area of interest. The highest priority for 2007 among the Information Technology (IT) security teams we work with is getting business units to take responsibility for data and application breaches. In most cases, this requires moving spending approval out of the technology infrastructure and assigning it where it should live—within the operating business units. Most teams today agree—

"If you're talking technology, you're behind. It's all about risk, and where responsibility for that risk should reside." [2]

Selfishly, most IA professionals in industry do not want to be caught holding the bag if there is a data breach or loss. Just as importantly, high-performing teams are also building the case that moving ownership to business units also improves the overall IT security posture of the organization. Business-unit owners are the best-qualified executives inside an organization to estimate the cost of, say a lost laptop with sensitive information on it, and what level of spending is appropriate to mitigate that risk.

"It's pretty hard to know how much to spend to secure something you don't truly know the value of. We are probably over-securing much of our data records, because we haven't been told what's valuable and not valuable." [3]

Most organizations today leave the data-security planning to the IT security team. Business-unit staff read headlines and listen to warnings but are generally reluctant to spend the dollars it takes to protect critical business assets—they do not like spending money on things they do not understand. Security teams, therefore, are making many of these decisions for the business units without a complete understanding of the actual risk faced by the business.

"Everyday we deal with the imbalance of risk between IT and the business units. They (the business units) won't answer the hard questions, so we do it for them. Since we sign on the bottom line, the buck probably stops back with us if something goes wrong." [4]

More importantly for the organization, the most-qualified minds are not quantifying the risk and allocating the right amount of resources to mitigate the risk. This is inefficient and dangerous.

IANS has been following a group of high-functioning IT security teams who are using innovative techniques to properly assign data and application risk directly to the business units. The Chief Information Security Officers (CISOs) of these teams are assigning staff to operate business units to help educate the sales, marketing, and operations professionals about the risks to their business and the best techniques (and products) to mitigate those risks. Over time, ownership of the mitigation gradually transfers over to business-unit leaders. CISOs have also begun to explore how the idea of security can be used as a differentiating product feature in the marketplace. Online trading companies now use secure web and ordering applications as a way to attract more customers, which in turn makes IT security a priority to the business unit and puts the risk in the right place. In sum, high-performing security teams are increasingly decentralized groups that set security policy and leave the actual implementation role to the business units.

While the idea of risk pervades the entire IA profession today, we will be sharing insights in the future about a series of other pressing topics, including the following:

- ▶ Securing small form factor devices: a whole new set of issues
- ▶ Making the investment in Security Information and Event Management (SIEM) pay off
- ▶ Organizing an IA team for optimal performance
- ▶ Presenting security in a way that changes your organizational culture

Finally, an open channel of communication with IANS is always available at *http://www.ianetsec.com*. We hope to hear from you. ■

References

1. http://*www.ianetsec.com*
2. Quote from Fortune 50 Financial Institution, Midwest Information Security Forum, November, 2006.
3. Quote from Retail Organization, Midwest Information Security Forum, November, 2006.
4. Major Health Insurer, quoted at the New England Information Security Forum, September 2006.

About the Author

Jack Phillips | began his career in media and information publishing as an investment banker at Morgan Stanley & Co. in New York. After filling senior operating positions at McGraw-Hill beginning in 1994, Mr. Phillips joined the founding team of Internet Securities in 1995 (subsequently purchased by Euromoney in 1998) and later joined CCBN.com in 1999. Mr. Phillips left CCBN.com in mid-2000 to launch the IANS. He is a graduate of Williams College and the Harvard Business School. He may be reached at the Institute for Applied Network Security, 15 Court Square, Suite 110, Boston, MA 02108, by telephone at 617/399-8100, or by email at jphillips@ianetsec.com

Verifying Network Intrusion Detection Alerts

by Jingmin Zhou and Matt Bishop

Today's signature-based Network Intrusion Detection Systems (NIDS) have been widely deployed to protect our network infrastructure. These systems detect network attacks (intrusion attempts) based on the known signatures of these attacks. However, they do not verify whether the attacks are successful.

Detecting unsuccessful attacks can be beneficial to intrusion analysis. It makes more information available to system administrators and helps them understand how attackers probe their systems. However, many occurrences of "false alarms" significantly increase the workload of system administrators, thus decreasing their ability to assess *actual* damage to the systems under attack. System administrators cannot quickly identify successful intrusions from all alerts, and the successful intrusions are those that present the most serious system threats. Therefore, it is difficult for system administrators to appropriately respond to successful intrusions in a timely manner. To mitigate this problem, NIDS should verify the result of the alerts and provide this information to system administrators, who would then prioritize the alerts and focus on the most serious attacks.

A common method of doing this is to profile the systems under attack and compare the profile to the vulnerability an attack is exploiting. [1, 2] For example, if a NIDS observes that an attack for Windows systems is used against a Linux system or an attack targeting a Solaris system before Version 2.8 is used against a Solaris Version 2.9 system, the NIDS can determine that the attack cannot succeed. The profiling process can be done either before or after attacks. Though sometimes effective, this approach has several serious limitations. It can be difficult to profile all systems before attacks occur, especially in a large, dynamic network in which active nodes are constantly changing. Profiling also may not be as accurate as required by the NIDS to determine the attack results. A vulnerability may exist in a server program before a certain version, but profiling may not be able to identify the exact version number of the server program on a system. Moreover, even if the system profile is accurate, an attack can be unsuccessful because the conditions required to exploit the vulnerability are not present. However, a NIDS may still report the attack as successful.

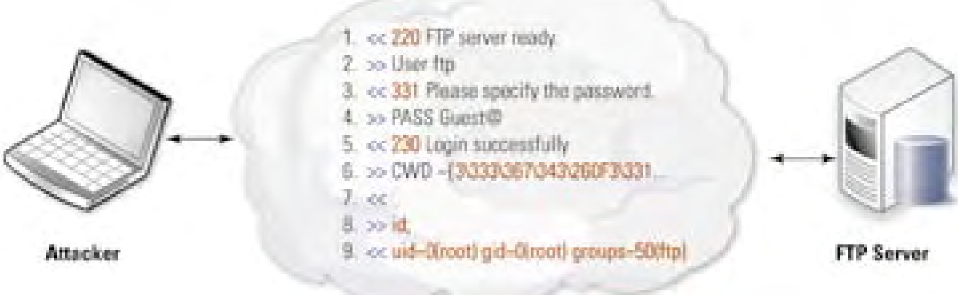
A different approach is to permit the NIDS to connect to systems after observing an attack and to locally verify the attack result. [3] For example, a successful buffer overflow attack against a web server leaves no entry in the server log, while a non-malicious connection would create such an entry. After detecting the attack, a NIDS can log into the server host and check whether there is an entry of the attack in the server log. If the entry is not found, the NIDS determines that the attack is successful. This type of host-level verification is more accurate than profiling-based verification. However, it requires each system to have a "back door" for the NIDS to log into. This poses a new risk to the systems, in that an intruder may be able to obtain the key to this "back door." Also, it is often impossible to implement this method for a dynamic network in which users are able to simply sit in and hook up their laptop computers.

Verifying NIDS Alerts Using Protocol Analysis

Our solution is to verify attack results using lightweight protocol analysis. The nature of network applications is that they all follow certain well-defined application protocols. For example, a web browser must communicate with a web server by Hypertext Transfer Protocol (HTTP). A successful attack often forces the application being attacked to unexpectedly change its behavior. Thus the application's response to the attack either violates the protocol or contains information about the state of the application after the attack. This can be used to determine the attack result. This information is also often contained in the beginning of the application's response, and a NIDS does not need to analyze the entire application response.

Our first technique is to verify an attack result if an application's response violates the relevant protocol. For example, Figure 1 shows the interaction between an attacker and a File Transfer Protocol (FTP) server. In this attack, the intruder logs into the FTP server anonymously (Steps 1–5), exploits a buffer overflow vulnerability in the FTP server (Step 6), and obtains a root shell after the attack (Step 7–9). Before the attack is successful, all data transmitted between the attacker and the FTP server follow the FTP protocol. After the successful invocation of the shell program, the program replaces the FTP server process. Its behavior is completely different from the FTP server. Thus the interactions between the attacker and the shell program no longer follow the FTP protocol. This fact is used in our approach to determine whether a buffer overflow attack is successful. This technique can be used for other popular attacks that force an application to behave abnormally, (*e.g.*, integer overflow, double free, format string).

For attacks that do not trigger abnormal program behavior, we use a second technique. For example, Figure 2 shows the interactions between an attacker and a Microsoft Internet Information Server (IIS). The attacker tries to execute the command interpreter program (`cmd.exe`) on the web server. If the attack is successful, the web server responses often begin with the string, "200 OK." On the other hand, if the attack is unsuccessful because the program does not exist or the attacker is not allowed to access the program, the web server responses often begin with the string "404 Object Not Found" or "403 Access Forbidden." This difference is used to determine the result of the attacks that do not trigger abnormal application behavior.

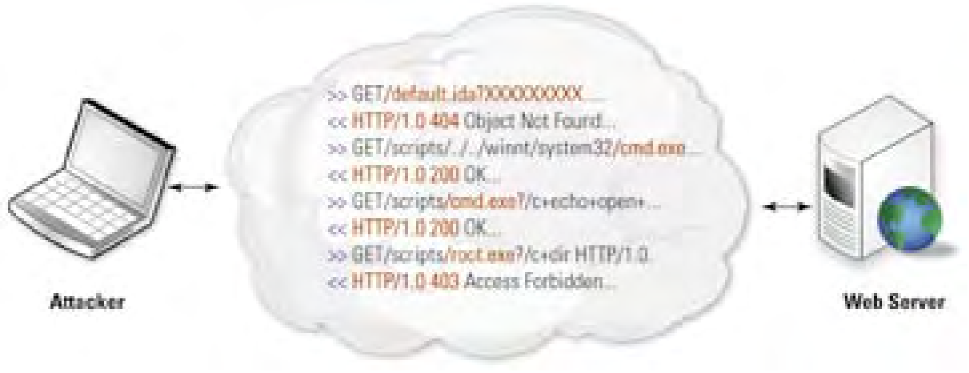


```
1. << 220 FTP server ready
2. >> User ftp
3. << 331 Please specify the password.
4. >> PASS Guest@
5. << 230 Login successfully
6. >> CWD ~{3\333\367\343\260F3\331...
7. <<
8. >> id;
9. << uid=0(root) gid=0(root) groups=50(ftp)
```

Figure 1 FTP buffer overflow attack

Since most application protocols have well-defined formats for valid responses, and a status code appears at the beginning of a response, our approach minimizes the effort to verify the result of an attack. When detecting an attack against an application server, the NIDS can simply capture the header of the response to this attack and use one of the two techniques to determine the attack result, depending on the nature of the attack. Because many NIDS already track the connections between an application server and its clients, the only extra work is to analyze the first packet of a server response, which contains the protocol header and status code.

Our approach is different from that of the simple method to detect the common response of a successful intrusion. For example, the popular NIDS Snort [4] has several signatures to detect strings like `uid=0(root) gid=0(root)` in application server responses, because these strings often indicate a successful buffer overflow attack. However, there are several problems with this type of signature. First, if an attacker does not execute the command `id` after a successful attack, such a string will not be generated. Therefore, it is trivial to inhibit detection. Secondly, this kind of signature is prone to false alerts. For example, one of our security tutorial webpages



```
>> GET/default.ida?XXXXXXXXXX...
<< HTTP/1.0 404 Object Not Found...
>> GET/scripts/../../winnt/system32/cmd.exe...
<< HTTP/1.0 200 OK...
>> GET/scripts/cmd.exe?/c+echo+open+...
<< HTTP/1.0 200 OK...
>> GET/scripts/root.exe?/c+dir HTTP/1.0...
<< HTTP/1.0 403 Access Forbidden...
```

Figure 2 Web Common Gateway Interface (CGI) attacks

contains the string `uid=0(root) gid=0(root)`—and it always triggers an alert when visited! These signatures also increase the overhead of NIDS because they are applied to all traffic. Our approach, on the other hand, combines analysis of the application's response to detection of the attacks and is therefore more accurate and efficient.

Fixing Snort Signatures

We have developed a method to integrate our approach into the signatures of Snort. Our method turns out to be surprisingly simple. For each Snort signature, we first determine the application protocol and the verification technique to use. If multiple signatures share the same application protocol and verification technique, we define a new rule to verify the attack result and modify these signatures to point to this new rule. For 687 Snort signatures detecting web, FTP, and Post Office Protocol 3 (POP3) attacks, the number of new rules is no more than 23. Moreover, our modification was accomplished in just a few minutes, thanks to the good formatting and organization of Snort signatures.

We validated our approach with experiments using real-world intrusion datasets collected on three of our Honeypot systems. [5] The results are shown in Table 1. Our new rules determined that 83.28%, 90.91%, and 96.95% attacks against three Honeypot systems, respectively, were unsuccessful. This demonstrates that the ability to determine the result of detected attacks can significantly reduce the workload of system administrators. Our new rules correctly report the successful buffer overflow attacks—usually the most serious threats to our systems. They also effectively distinguish successful web CGI attacks from those that are unsuccessful, which comprise the vast majority of attacks in the dataset and often in the real world.

Honeypots	Alerts (Old Rule Sets)	Alerts (New Rule Sets)
Windows NT 4.0	16,989	2,481
Windows 2000	13,660	1,242
RedHat Linux 7.2	4,978	1,52

Table 1 Reported alerts

Conclusion and Future Work

We developed a simple and effective method to determine the result of signature-based NIDS alerts. It demonstrates that protocol analysis can significantly increase the quality of NIDS alerts. Our future plan includes improving the performance, evaluating this technique with more sophisticated applications, and combining anomaly detection techniques. ■

References

1. Kruegel, C & Robertson, W. Alert verification: Determining the success of intrusion attempts. *Proceedings of the 1st Workshop on Detection of Intrusions and malware & Vulnerability Assessment (DIMVA),* July 2004.
2. Lippmann, R. P, Webster, S. E. & Stetson, D. The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection. *Proceedings of 5th International Symposium of Recent Advances in Intrusion Detection (RAID),* 2002.
3. Vigna, G., Robertson, W, Kher, V. & Kemmerer, R. A stateful intrusion detection system for world-wide web servers. *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC),* December 2003.
4. Roesch, M. Snort—lightweight intrusion detection for networks. *Proceedings of the USENIX LISA '99 Conference*, November 1999.
5. The Honeypot Project. (2001). *Know your enemy: Revealing the security tools, tactics, and motives of the blackhat community. http://www.honeynet.org*

Acknowledgement

This research was supported by a grant from Promia, Inc. to the University of California, Davis.

About the Authors

Jingmin Zhou | is a PhD student at the Computer Security Laboratory of the University of California, Davis. His research focuses on intrusion detection, vulnerability analysis, static program analysis for software security, and other computer security issues. He may be reached at zhouji@cs.ucdavis.edu.

Matt Bishop | is a professor in the Department of Computer Science of the University of California, Davis. His research focuses on vulnerability analysis, attack analysis, data sanitization, policy modeling, software assurance, and formal models of access control. His textbook, *Computer Security: Art and Science* was published in December 2002 by Addison-Wesley Professional. He may be reached at bishop@cs.ucdavis.edu.

IATAC SPOTLIGHT ON EDUCATION

Center of Excellence University of California, Davis

by Ron Ritchey

The University of California, Davis (UC Davis) Department of Computer Science (CS) within the College of Engineering was founded in 1993 to advance the research frontiers of computer science while producing outstanding computer science professionals. The department offers undergraduate degrees in two curricula: Computer Science and Engineering and Computer Science. The department also offers MS and PhD programs in CS, offering students research specialization in all aspects of computers from algorithms and computational biology to computer security and software engineering. [1]

The department hosts the Computer Security Lab, one of the nation's leading centers for research in network security, vulnerability, information integrity, intrusion detection, and security policy. The Computer Security Lab has four co-directors—

- **Matt Bishop**, who is profiled in this issue's IATAC Spotlight on Research
- **Hao Chen**, who researches malicious code and worms
- **Karl Levitt**, who works with Intrusion Detection Systems (IDSs), malicious code and diversity of systems, formal methods, and models
- **Felix Wu**, who works with IDSs and network security

Apart from the four co-directors, there are usually 10–12 other researchers working on overlapping projects. According to Professor Bishop, the Computer Security Lab's research projects span many areas, including the following [2]—

- **Vulnerabilities**—Analyzing systems for and defining and reacting to vulnerabilities
- **Intrusion Detection and Analysis**—Detecting and responding to attacks and developing ways in which to make systems more difficult to attack
- **Malicious Code Such as Computer Worms**—Establishing defenses such as diversifying systems to make them more difficult to attack
- **Privacy**—Balancing individual privacy against the needs of corporate or government entities

One project lab is working on is property-based testing, which researches new ways of looking for vulnerabilities in computer systems. Properties define the desired behavior of software. The property-based testing tools treat the execution of the software as a state machine. As the machine executes, its state is compared to the specification, and any violations sound an alarm.

Another project is examining new methods for analyzing attacks. It is the flip side of vulnerability analysis. Dealing with the aftermath of a system compromise, this project examines methods for aggregating intrusion alerts to determine what the attacker did. This project develops new techniques in computer forensics and reactive postmortem analysis of systems.

In related work, the laboratory staff is also using deception techniques to better understand attacker behavior. The deception project studies methods to deceive attackers once they have compromised a system and examine how they react. This project focuses on host-based techniques rather than honeynets or honeypots. Attackers who break into systems are flipped into a *deceptive mode* that provides false data to attackers and enables observation of attacker responses and actions.

A past project, the electronic recordation (*e.g.*, real estate) project examined problems associated with recording titles, liens, *etc.* over networks such as the Internet. The project discovered numerous potential vulnerabilities if electronic recordation is not implemented properly. The project then developed a formal model of the recordation process to guide the development and implementation of electronic recordation systems. ■

References

1. More information about the UC Davis CS department can be found at *http://www.cs.ucdavis.edu*
2. A list of current and previous Computer Security Lab Research Projects can be found at *http://seclab.cs.ucdavis.edu/Projects.html*

Data Integrity and Proof of Service in BitTorrent-Like P2P Environments

by Jun Li

The success behind BitTorrent [1] or BitTorrent-like Peer-to-Peer (P2P) applications (such as those discussed in Stavrou, Rubenstein and Sahu [2]; Kong and Ghosal [3]; and Sherwood, Braud, and Bhattacharjee [4]) is their innovative use of collaboration among peer clients. While the scalability of client-server applications is often poor when a large number of clients access a single server, permitting a client to download data from other peer clients in addition to directly from its server has tremendously boosted the availability of data to a client and fundamentally addressed data-accessing scalability.

However, compared to receiving data directly from a server, receiving data from arbitrary, often less trustworthy peer clients is subject to much higher security risks. Two obvious concerns that warrant new study are data integrity and proof of service in this BitTorrent-like P2P environment. The integrity of data received from peer clients is more easily breached if malicious clients sneak into the system to corrupt peer communications. Mechanisms to reward peer clients for sharing data, such as crediting the providers, are also vulnerable, since clients may lie about peer-level service.

Conventional approaches designed for the client-server communication paradigm, such as Secure Sockets Layer (SSL) [5], cannot easily address these concerns. With thousands of clients sharing a single server and communicating with each other, simply setting up an SSL security channel between every client and the server and between every pair of peer clients is not only costly but also requires a great deal of extra work to make an SSL function in this environment.

In this article, we describe a feasible approach to addressing the data integrity and proof of service in a BitTorrent-like P2P environment. Both functionalities are included in a new protocol we designed that is called mSSL.

Data Integrity

Now that a client can receive its server's data from either the server or from its peer clients, the client must be able to verify the integrity of the data to ensure that the data has indeed originated from the server and has never been manipulated.

The mSSL protocol supports a block-based, on-demand data integrity solution. With a block-based solution that divides a data object into many blocks and verifies the data integrity at block level, a client can verify the integrity of every block once it is received. If a client has to verify the signature of an entire data object to verify its integrity, the penalty can be high; in particular, if signature verification fails, the entire data object—which can be a file with many gigabytes of data—has to be retransmitted. The integrity solution is also on demand, in that a client can determine the *integrity path* of a given block and then request the integrity path information to verify the integrity of the block. We explain the integrity path concept below.

To verify data integrity at block level, one solution is to bind a data signature to every block of a data object. A client must verify the block signature to determine its integrity. This method, however, involves encryption and decryption at block level and can lead to high computational overhead. Another approach is to build a superblock for a data object that contains a strong one-way hash result for every block. A client can first obtain the superblock (the superblock itself can carry a signature to prove its own integrity), then, whenever it receives a new block, it can calculate the hash of the block and compare it with the value contained in the superblock. Calculating the hash result is a much faster process than encryption operations, but, for large files, a superblock itself can be very large. In a 100 gigabyte video file, for example, if every block is 1 kilobyte, and every hash is 16 bytes, a superblock of 1.6 gigabytes has to be retrieved first, leading to a high startup latency in retrieving data blocks.

In fact, every data object can have a binary Merkle hash tree [7], and every block of the data object can have an authentication path obtained from that tree. (Merkle hash trees have been used in various contexts such as P2P media streaming [8] or in third-party distribution of integrity-critical databases [9] and XML documents. [10]) When a client receives a block, it can request the authentication path of that block to verify its integrity. The client does not have to download all hash values

beforehand as in the superblock-based approach. Figure 1 shows an example.

The problem with this approach, which is block based and on demand, as is mSSL, is its high traffic volume in receiving authentication paths. For example, if every block is 1kilobyte and every hash is 16 bytes, a 1 gigabyte data object can incur 320 megabytes of such traffic. (The object has 2^{20} blocks, and every block's authentication path consists of 20 hash values; *i.e.* 320 bytes).

The mSSL protocol greatly optimizes both on-demand requests of integrity verification information and the verification procedure itself. Unlike the authentication path above, in mSSL, every block is associated with an integrity path. An integrity path is only composed of those hash values that are not locally available for calculating the root of the Merkle hash tree of a data object. The question then is, *What hash values are locally available or unavailable?*

Denote a block *b*'s authentication path $A(b)$ as $A(b) = H^m, H^{m-1}, ..., H^1$ (H^i is a hash value at level i of the Merkle hash tree and H^m is at the leaf level), and use *mip(b)* to denote *b*'s integrity path at a client. Our research has shown that, if the client already has H^{l-1}, but not yet H^l, then $mip(b) = <H^m, H^{m-1},\cdots, H^l>$, with a total of $|mip(b)|= m - l + 1$ hash values. Also interestingly, a client only needs to specify the value of $|mip(b)|$ for its peer client or the server to determine what the integrity path is at the requesting client. With this approach, we have also proven that for a client to verify the integrity of all n blocks of a data object, the total number of hash values that the client needs to request is also n, if every block is received correctly. In the above example, in which every block is 1 kilobyte and every hash is 16 bytes, a 1–gigabyte data object will incur 16 megabytes of traffic overhead, only 1/20th of the authentication-path-based solution.

Once a client receives the integrity path of a block, it can also verify its integrity at a faster speed than the authentication-path-based approach. Instead of using the integrity path to determine the authentication path of the block and then calculating the root of the tree using the authentication path, mSSL will only calculate the hash value of an intermediary node on the tree, and compare it with that node's correct value, which has already been obtained earlier. More specifically, if a block *b*'s integrity path is $mip(b) = <H^m,H^{m-1}, \cdots H^l>$, a client only needs to calculate the sibling node of H^{l}*'s* parent.

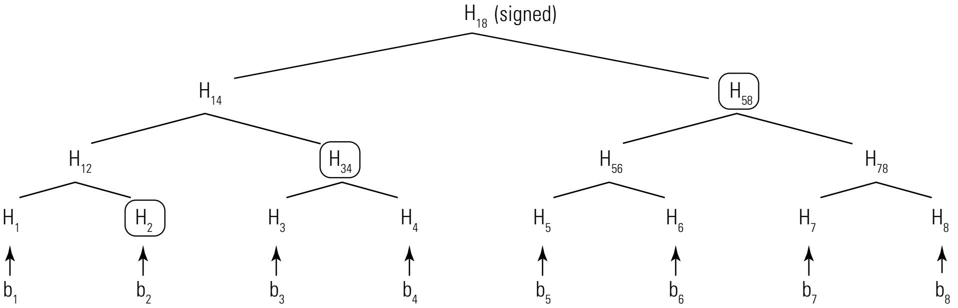


Figure 1 Merkle hash tree of a data object with eight blocks. Every leaf node is a hash value produced from a strong one-way hash function with the data of a corresponding block as input. Every intermediary node is a hash value that is calculated using the hash function with the values of its child nodes as input. Every block has an authentication path that can help calculate the root of the tree when knowing only the block itself. For example, the authentication path of block b_1 is $A(b_1)$ = < *H2, H34, H58* >, where b_1 can lead to H_1, H_1 and H_2 [$H_2 \in A(b_1)$] can lead to H_{12}, *etc.* If a block is corrupted, it can be detected because the root calculated from the block will not be equal to the authentic value of the root of the tree.

Proof of Service

If a client can present to its server an accurate, verifiable, and non-repudiatable proof to describe its service by providing data to some of its peer clients, the server could offer this client certain credit; *e.g.*, assigning a higher priority in providing this client data or charging it a lower price for receiving the server's files. More importantly, the client will also have stronger incentive to continue to provide data to its peers.

A simple approach to obtaining proof of service is to enforce an interlocking block-by-block verification mechanism between every pair of provider and recipient clients. As shown in Figure 2, the recipient must acknowledge its receipt of the current block before it can receive the next block from the provider. The provider can then use the acknowledgment as proof of serving the block being acknowledged.

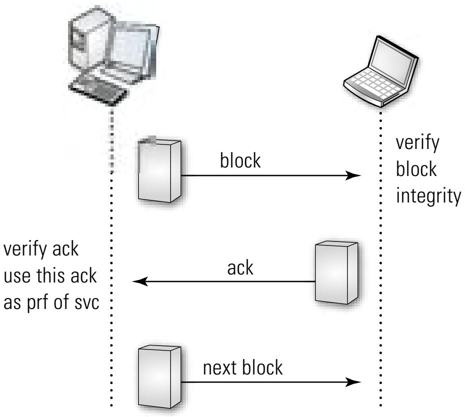


Figure 2 A basic proof of service solution

Unfortunately, this approach faces several serious problems. First, if a provider has to obtain a separate proof for *every* block it offered, it will face the proof-explosion issue when serving large files to its peer clients. Second, if a recipient decides *not* to acknowledge the receipt of a block after receiving it, the provider will not be able to obtain the proof of serving that block. Third, if the acknowledgment from a recipient is corrupted, the provider can be cheated in providing the next block.

The mSSL protocol supports a proof-of-service solution that addresses all these problems in a BitTorrent-like environment. This work is related to the strong, fair exchange of information in other contexts, and its design can be summarized as follows—

- A recipient will receive an encrypted block first and receive the decrypting block key after acknowledging the receipt of the block
- Acknowledgments are cumulative; thus the most recent one can replace previous ones as the proof of service. For example, it can be [1–66,68–128] to acknowledge the receipt of the first 128 blocks of a data object except block 67
- Every acknowledgment is signed and can be verified using the public key of the recipient.

Figure 3 shows the detailed steps for a provider p to obtain a proof, while providing data blocks to a recipient r. More specifically, it contains the following major steps—

1. **Public Key Exchange**—p and r will first exchange public key certificates so that they can know each other's public key. In mSSL, when a client (such as p or r) first contacts its server, it will still set up an SSL channel with the server and obtain both the server's public key certificate and, at the same time, the server's public key. Through this SSL channel, the client can also request the server to sign the client's public key, thus generating a public key certificate signed by the server. This kind of certificate is exactly what p and r exchange, and each can verify the certificate from the other side.
2. **Encrypted Block Request and Transmission**—r then requests a block b from p, which sends an encrypted block back to r. Here, the key used for block encryption, also called **block key**, is generated by p using a strong one-way hash function, f: $k = f(p, r, file, blockid, k_p)$, in which $p, r, file, blockid$ are the identifiers of $p, r,$ the data object that the block belongs to, and the block itself. k_p is the secret key shared between the provider and the server. Note: The server can also apply this formula to calculate the block key.
3. **Cumulative Acknowledgment**—r then sends back an acknowledgment and its signature. Instead of acknowledging only the receipt of the encrypted block, r uses a *sack* field to acknowledge the receipt of all blocks it has received from p. It also includes the digest of the encrypted block. r signs the acknowledgment with its private key, PRV_r. On the receipt of the acknowledgment, p then can verify it. Only if the verification is successful will p use the acknowledgment as its proof of service to r so far—and begin to deliver the block key to r.
4. **Block Key Delivery**—p delivers the block key to r, which is encrypted with r's public key, PUB_r, and signed with p's private key, PRV_p. The delivery must be secure so that only r can obtain the key and r can verify it is from p. After r receives the block key, it then can decrypt the block and verify the integrity of the block. If p refuses or fails to deliver the block key, r can request its server to calculate the block key and deliver the key to r. Note: This operation implies that r probably will stop using p as its data provider and is therefore an infrequent operation that will not overload the server. If the block integrity is found to be corrupted, r will not acknowledge the receipt of the next block. By doing so, when p presents the current acknowledgment to the server as proof of service, the server can calculate the digest of the current block in its encrypted form to detect that r did not correctly receive the current block. This design, however, leads to a slow, stop-and-go data-transmission process, which we revisit below.

The above process handles one block at a time. To improve performance, mSSL further introduces parallelism to concurrently handle multiple blocks. Every acknowledgment can actually include digests of the last m encrypted blocks ($m = 1$ in the above step-by-step description). When an acknowledgment is used as a proof of service and presented to a server, the server will verify the correctness of all m digests to ensure that the last m encrypted blocks are all delivered correctly. This way, instead of first using a block key to decrypt an encrypted block and verify its integrity before acknowledging data reception, a recipient can always first acknowledge the receipt of up to m-1 encrypted blocks.

Two types of attacks may occur against the proof-of-service design:

individual cheating, during which an individual client misreports the amount of service it received or provided, and colluded cheating, during which multiple clients forge proofs of service together. mSSL addresses both attacks.

For individual cheating, a provider cannot overstate its service, since every proof is provided and signed by its peer clients after they receive data from the provider; and a recipient cannot deny the service it receives since, in doing so, the recipient will not be able to receive block keys to decrypt the encrypted data blocks it receives.

Colluded cheating can happen in various ways, especially in the following two scenarios—

- In one scenario, a recipient client sends acknowledgments to its real provider and then may try to copy these acknowledgments to its accomplices, who, in turn, may try to use these acknowledgments to claim that they each also have delivered data to the recipient. This colluding scenario can be detected by the server of these clients when verifying the digest(s) of the acknowledgment, which must be the digest(s) of the block(s) encrypted by the real provider using a block key (or block keys) specific to that provider. This mechanism can also help detect proofs that a client forges by simply providing data to itself.
- In another scenario, even if no data transmission *ever* occurred, clients can collude to forge a proof that one or several of them have provided data to a recipient client. Here, an economical countermeasure may be the most effective: Because the server needs only to make sure that the undeserved credits that fake providers may obtain are always less than the cost that the colluding recipient has to pay, these colluders will lack incentives to proceed.

Summary

While BitTorrent or BitTorrent-like P2P applications have been very successful in sharing among peer clients data that are traditionally only available by directly downloading from a server, security concerns in these applications are severe, especially ensuring data integrity and obtaining trustworthy proof of service that a client has provided to its peers.

The mSSL protocol offers a security solution in this environment, and in this article we described its approach to data integrity and proof of service. In ensuring data integrity, mSSL introduces an integrity path concept to support a prompt, block-based, and on-demand integrity verification mechanism that permits both low traffic and low computational overhead. The proof-of-service design is also advantageous in BitTorrentlike P2P environments. It not only minimizes server overhead and data-transmission slowdown but also ensures that the proofs are accurate and of small size. ■

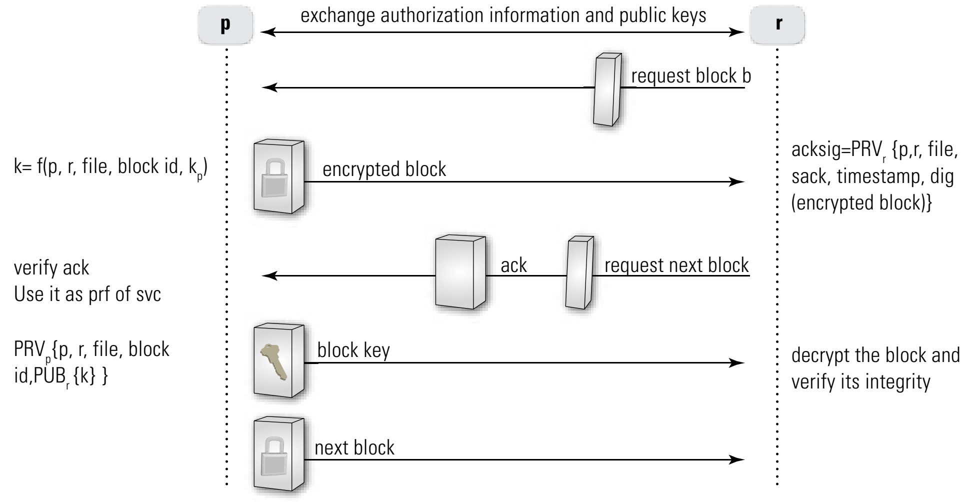

p
exchange authorization information and public keys
r
request block b
k= f(p, r, file, block id, k_p)
encrypted block
acksig=PRV_r {p,r, file, sack, timestamp, dig (encrypted block)}
verify ack
Use it as prf of svc
ack
request next block
PRV_p{p, r, file, block id,PUB_r{k} }
block key
decrypt the block and verify its integrity
next block

Figure 3 mSSL's proof-of-service design

References

1. BitTorrent, *http://bittorrent.com, 2005.*
2. Stavrou, Angelos; Rubenstein, Dan & Sahu, Sambit. A Lightweight, Robust P2P System to Handle Flash Crowds. *10th* ICNP, 2002 (pp. 226–235).
3. Kong, Keith & Ghosal, Dipak. (1999) Mitigating Server-Side Congestion in the Internet through Pseudoserving. *IEEE/ACM Trans. Netw., vol. 7, no. 4* (pp. 530–544).
4. Sherwood, Rob; Braud, Ryan & Bhattacharjee, Bobby. (2004). Slurpie: A Cooperative Bulk Data Transfer Protocol. *IEEE INFOCOM*
5. Rescorla, Eric. *SSL and TLS: Designing and Building Secure Systems. Addison-Wesley*, 2000.
6. Li, Jun & Kang, Xun (December 2005). mSSL: Extending SSL to support data sharing among collaborative clients. *Annual Computer Security Applications Conference, Tucson, Arizona* (pp. 357–368).
7. Merkle, Ralph. (April 1980). Protocols for public key cryptosystems. *IEEE Symposium on Privacy and Security* (pp. 122–134).
8. Habib, Ahsan; Xu, Dongyan; Atallah, Mikhail; Bargava, Bharat & Chuang, John. (2005) Verifying Data Integrity in Peer-to-Peer Media Streaming." *Twelfth Annual Multimedia Computing and Networking (MMCN '05).*
9. Devanbu, Premkumar T.; Gertz, Michael; Martel, Chip & Stubblebine, Stuart G. (2000) Authentic Third-party Data Publication. *IFIP Workshop on Database Security* (pp. 101–112).
10. Bertino, Elisa; Carminati, Barbara; Ferrari, Elena; Thuraisingham, Bhavani M. & Gupta, Amar (2004). Selective and Authentic Third-Party Distribution of XML Documents. *IEEE Trans. Knowl. Data Eng., vol. 16, no. 10.* (pp. 1263–1278).

About the Author

Jun Li | is an assistant professor at the University of Oregon. He received his PhD from the University of California, Los Angeles, in 2002. His current research includes Internet worm defense, Internet routing forensics, Internet Protocol (IP) source address validity enforcement, and P2P security in BitTorrent-like environments. Dr. Li is a member of the Institute of Electrical & Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM). His research is being funded by National Science Foundation and Intel Corporation. He may be reached at lijun@cs.uoregon.edu.

An Overview of Voice over Internet Protocol (VoIP)

by Matthew Warnock

Following the invention of the telephone in 1876, worldwide communication was revolutionized. While computer communication has been very popular in the past 25 years, voice communication maintains a long-standing reign over person-to-person communication, as can be seen in the great amount of cell phone usage. Voice communication has improved significantly since the first successful test by Alexander Graham Bell, famous for the quote, "Mr. Watson, come here. I want you," and voice communication continues to improve as technology warrants.

The telephone started as a closed-loop connection between caller and receiver and could cover reasonably long distances. Long-distance calls from New York to Los Angeles actually completed a one-way circuit over a complete copper wire connection over 3,000 miles. However, that call was on the wire, and, because of this, a pair of wires must be established between every standard analog phone. In the 1960s, the backbone of the long-distance telephone system became digitized, and now the analog connection to the phone in your kitchen only has to connect to the local office. The digital backbone improved the sound quality and maximum possible distance of connections. A new technology, Voice over Internet Protocol (VoIP), improves the connection on the long-distance backbone and the connection to your house by changing the way in which sound travels over the wire. This paper will detail the technology of VoIP, compare it to the standard telephone system, discuss how it is currently implemented, review current standards and security, and discuss government regulation.

Technology

Circuit Switching *vs.* Packet Switching

The Public Switched Telephone Network (PSTN) is the standard telephone network that connects most homes and businesses in the country and the world. The PSTN uses a circuit-switched network, which is a very reliable but inefficient technology first seen in the 1940s. In a circuit-switched network, the connection from caller to receiver is a closed loop occupying an entire wire between the two points (in the case of analog connections) or a specific time slot (in the case of digital connections). The call only takes one route and uses the same bandwidth the entire time, no matter what kind of audio signal is being passed. In contrast, a packet switching network is one in which many signals are sent as packets from multiple senders to multiple receivers and use the specific amount of bandwidth required to send the data. This is how networks and the Internet work, using the Internet Protocol (IP). Many connections can be placed over the same wire, but the source and the destination IP addresses identify the origin and destination of the message. Also, the packets travel through many points and can take a different route each time. When a large amount of data is required, more packets flow over the wires, while smaller amounts of data require smaller amounts of packets. Jongwoo Han of Syracuse University has illustrated the concept of circuit switching and packet switching networks quite clearly on his website, *http://classes.maxwell.syr.edu/psc300_103*. (See Figure 1).

VoIP uses the idea of packet switching networks and changes the way voice data moves across the wire by changing the way the communication is routed. Instead of creating a connection through switches that keep the connection constantly open or uses a fixed bandwidth, it separates the data into packets that move alongside other packets, depending on the required data size and the network routes the data follows. This makes it more efficient, because, like the Internet, VoIP packets can take any path to get to their destination and do not need to constantly send data. However, packet switching networks may not be as reliable as circuit switching networks, as packets can transmit out of order, can be delayed, or may not reach the destination at all. This causes problems with latency, or delay in audio signal, jitter, where the amount of delay changes, and packet loss, where some of the signal does not make it at all.

Since VoIP is simply defined as voice data going over the IP, there are several technologies that fall under VoIP. There

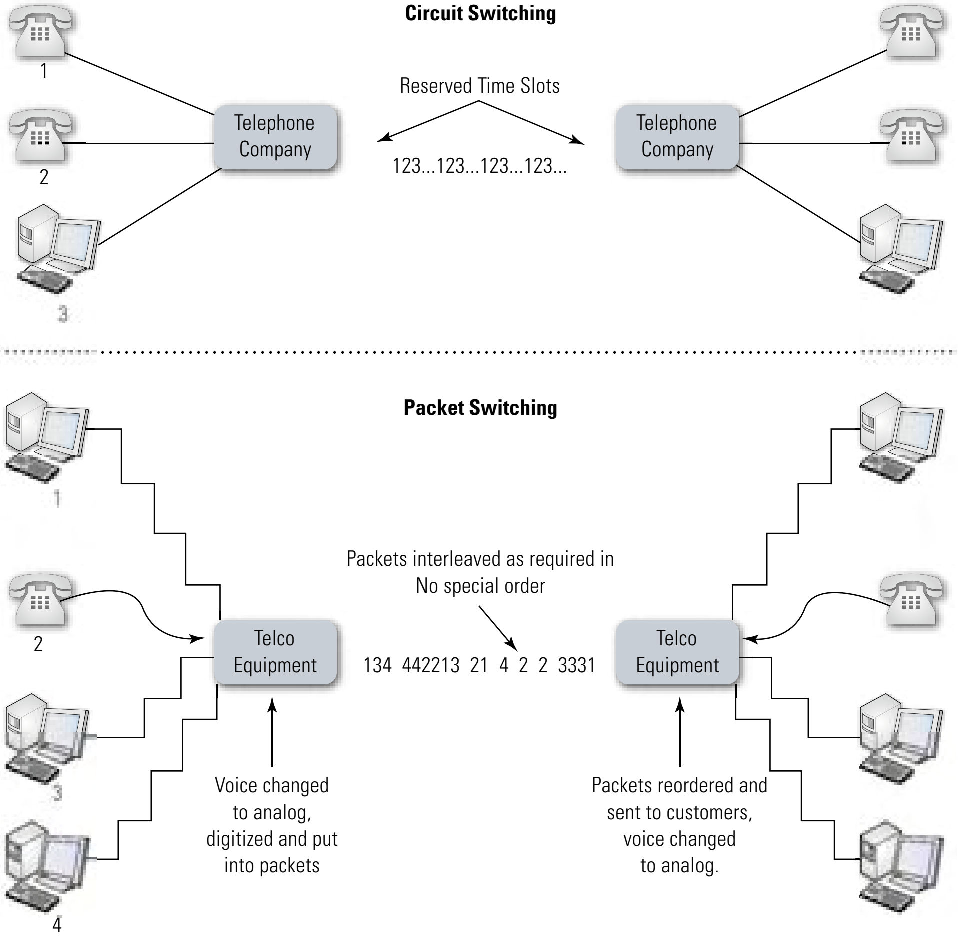


Figure 1 Circuit switching and packet switching

are three types discussed in this paper and include computer-to-computer (shown in Figure 2), Analog Telephone Adaptors (ATAs) (shown in Figure 3), and IP phones (shown in Figure 4). All three technologies are part of VoIP, and the makeup of VoIP, such as standards and protocols, works on all these. All three VoIP technologies do the same thing by converting analog audio signals into digital audio signals, converting the digital audio signals into network packets to send over a distance, and then performing the opposite operation at the receiving end.

Standards

VoIP is still in its infancy, and there is no standard protocol. However, there are several open-source protocols and standards that make up VoIP today. By being open, any company or software creator can create VoIP hardware or software that is interoperable, at least between the same protocols. There are several standards for analog-to-digital audio conversion. A Compression/Decompression Module (CODEC) algorithm is a standard algorithm that converts analog data to digital data. The most common standard of audio CODEC for VoIP is G.729A.

Once the analog voice data is converted to digital, it can use any number of VoIP network protocols to actually send the digital signal over a network. The first of these protocols is the H.323 protocol suite. The H.323 protocols were not designed for VoIP but for video conferencing. They include a very robust but complicated suite of protocols, including features for video, audio, whiteboard, chat, and other controls. The H.323 protocols are regarded as inefficient; however, they were the most popular VoIP protocols until recently. Problems using H.323 over Network Address Translation (NAT) have made them less desirable, but updates to the H.323 suite (as of H.323v6–2006) make them traverse NAT and may help H.323 make a comeback. Also, the H.323 protocol integrates

Figure 2 Screen shot Gizmo

with the PSTN very well, because the robust features include the same features found in the telephone system.

Unlike using H.323 for something for which it was not solely intended, another protocol called Session Initiation Protocol (SIP) was designed for VoIP. It is smaller and more efficient than H.323 and is currently more popular, especially for end-user applications. Because it contains fewer features as compared to H.323, there are some interoperability problems, especially when connecting to the PSTN. Both protocols are considered open source and the standards are publicly available.

Another protocol to mention is the Media Gateway Control Protocol (MGCP), which focuses on end-point control used directly by the caller or recipient, such as call waiting. H.323, SIP, and MGCP are not necessarily compatible and will continue to cause problems until a standard VoIP protocol is established.

Usage

VoIP only details how voice is sent from one point to another and includes computer-to-computer connections, ATAs, and IP phones. Phone calls may be made between two VoIP users or between two analog phone users, during which, somewhere in between the two analog phones, the data is converted to and from VoIP.

Computer-to-Computer

Computer-to-computer connections are used when voice data is received through a microphone connected to a computer. Inside the computer, the data is converted to a digital signal, converted to network packets, routed over IP to another computer, converted back to digital audio, and finally to analog audio to be played over speakers. This requires no special hardware; only software that can perform conversions on either end. Such applications include Microsoft Netmeeting, included with newer version of Windows, and Apple iChat, included with versions of the Mac Operating System.

ATA

The second type of VoIP technology uses an ATA, which is an adaptor used to plug a phone into a computer to complete calls over the network. This lets you connect a standard telephone, used to connect to the PSTN, to instead connect to your computer or Local Area Network (LAN). The telephone call is then placed using software on the computer; however, any kind of telephone device can be plugged into this device and use VoIP. If the ATA does not require a connection to a computer, but connects to the LAN instead, the ATA makes the connection. The end user will not notice a difference compared to an analog phone. Using an ATA is the only way that non-voice devices can use VoIP. These devices can include a security alarm, fax machine, data modem, or Tivo system. Depending on the ATA device, they may or may not be affected by using VoIP.

IP Phone

IP phones are the third category of technology. The hardware plugs directly into a network using a Registered Jack-45 (RJ45) connector. The entire connection is made through the IP phone hardware and not through a computer. In this case, the user notices no difference between conventional phone communication and VoIP communication. Any of these technologies include additional telephone features for free, such as call waiting, three-way calling, caller ID, *etc.*

PSTN Backbone

When long-distance companies began to digitize their long-distance calls in the 1960s, each long-distance connection was given a constant 128 kilobits per second (*i.e.*, 56 kilobits per second in each direction) connection. It did not matter if only one person was talking or even if no one was taking—the full 128 kilobits per second was utilized. Now, some long-distance backbone uses VoIP, which permits the audio data to be compressed and separated into packets and the bandwidth given on an as-needed basis. This improves throughput significantly, and many more connections can be made on the same bandwidth. Still, the headers of the packets use most of the bandwidth, which is why protocols that are even more efficient are being designed.

Currently, H.323 is the main protocol used on the long-distance backbone.

Connections

The PSTN uses a telephone standard of telephone numbers to route phone calls. The North American Numbering Plan (NANP), a standard detailed in standard E.164, consists of an area code, local exchange, and phone number. VoIP connections route the packets differently using IP addresses. Soft switching is the way in which calls are converted from IP addresses to phone numbers, using a lookup database. A Central Call Processor is the hardware that provides the soft-switching capability. Depending on the caller's location, the IP address and phone number mappings can be updated. A business traveler could take a VoIP phone with him or her and use the same phone number, because the software maintains the bind between the phone number and IP address.

Figure 3 Analog Telephone Adaptor

Figure 4 IP phone

It is possible to use VoIP connections while traveling, and it has recently been determined that VoIP connections could travel over Wireless (WiFi) network connections. If a mobile phone could connect to a local network *via* WiFi, the remainder of the phone call could be completed over VoIP. Since more people are using cell phones and, as a result, disconnecting their land-line phones, WiFi VoIP is a viable option for meeting the demands of this new telephone frontier. It is also a new way to market cell phones in a near-saturated market.

Services

There are several VoIP service providers willing to give a free or commercial alternative to the local phone company. Comcast Digital Phone service uses the data infrastructure built for its television network to provide phone connectivity. AT&T provides CallVantage, and Verizon provides VoiceWing, both of which can be used over any broadband connection. Vonage is a popular service using an ATA, just as do AT&T's and Verizon's plans. These are pay services that offer a fixed number of minutes—or even unlimited usage of local and long distance—for a monthly fee. You use a standard phone and phone number and send and receive calls just as you would when using the standard phone system. You can call anyone who has a phone number on the PSTN; there is no need to connect over VoIP.

There are several other kinds of VoIP services that are computer-to-computer based. Gizmo is a service that provides free voice communication to any other Gizmo user and a per-minute charge to call telephone numbers. You can even attach a telephone number to the service for call-in capability, which is available for a monthly fee. Gizmo uses the SIP protocol. Skype is another computer-based program touting similar features that also includes free calls to telephone numbers in the US and Canada until the end of 2006. Skype also includes encryption on all calls and makes connections in a Peer-to-Peer (P2P) fashion using a proprietary protocol.

Security

VoIP security is something that must be improved before becoming adopted. The PSTN system has some built-in security because it is a closed system, and only telephone companies maintain the connections. Also, the switch-circuit portion of the network does not allow more than one connection over the same wire or digital pipe, so eavesdropping is more difficult. By definition, VoIP allows multiple data signals to travel over the shared network, and eavesdropping is likely to occur. VoIP protocols have limited security benefits. Security is often obtained by way of security-through-obscurity, when the analog-to-digital algorithms or the VoIP protocols are proprietary, so that the data cannot be easily converted to a readable format. This cannot guarantee any level of security. Gizmo, Zfone, and Skype include strong, built-in encryption to their programs. These programs are used to secure the voice data traveling over the wire.

It is also possible to secure traffic by securing the connection itself. Because VoIP travels in IP packets, Secure Internet Protocol (IPSec) can be used to protect packets by encrypting and/or signing the packets. Another option is to secure VoIP communications at the transport layer. The Secure Realtime Protocol (SRTP) replaced the Real Time Protocol (RTP) used in insecure VoIP traffic. Encrypted tunnels, such as those used in VPN connections, can be used to secure all traffic going through the tunnel, including VoIP. No matter where you decide to encrypt traffic, in the voice data or in the entire packet, latency is a major concern in encrypted VoIP connections.

While eavesdropping is a large concern because of the unencrypted VoIP traffic, other Transmission Control Protocol/Internet Protocol (TCP/IP) weaknesses are also present in VoIP. Hackers can threaten VoIP systems just as can worms and viruses that travel over network connections. In a packet switching environment, a Denial of Service (DoS) attack on a VoIP network can bring down all voice communications. The circuit-switched network is much more resilient in this aspect. While not a security risk, the requirement for walled power is an availability issue with VoIP. The standard phone company provides power to its phone lines, which permits you to use a standard phone even when power is out. VoIP phones require power at the phone, the computer, and all the network connections in between. When power fails at any of these points,the failure can prohibit incoming or outgoing calls.

Regulation

Because VoIP basically circumvents the standard telephone company, it also circumvents Federal Communication Commission (FCC) regulations and other state or local taxes and tolls. The FCC has not done much to regulate VoIP traffic. In November 2004, it ruled that state regulations governing telephone companies do not govern Vonage. They have required that VoIP providers clearing display warning labels if they cannot provide 911 support. Also, they support the wiretap regulations established by the Communication Assistance for Law Enforcement Act (CALEA), which require VoIP providers to have capabilities that permit law enforcement to tap phone lines.

Other countries do not enjoy the freedom that the US has in terms of VoIP. Some Latin American countries restrict VoIP connections, while countries like Ethiopia ban any use of VoIP to ensure that all connections are made through its government-operated telephone company. They even operate VoIP detection at the firewalls at their borders.

Conclusion

VoIP is an up-and-coming technology that has a good possibility of replacing current telephone technologies. Just as the iPod is attempting to replace audio CDs, VoIP provides a means to obtain the same results by using a new technology. VoIP includes any method of taking voice data and sending it over an IP network. This includes computer-to-computer connections, analog phones using an ATA, or IP telephones. The technology is so new that a single standard has not yet been established—but a single standard must be set. At the moment, the most popular protocols and standards are open, which means any commercial or free product can make interoperable devices. Security—a very weak point for VoIP—must be improved. Security-through-obscurity is never a viable option when securing data. The security problem is being addressed, and there are several good solutions emerging; however, these must be incorporated into VoIP standards. Currently, VoIP is not well regulated, and where it is regulated, such as in 911 support, it is a good thing; however, it's almost inevitable that the technology will be taxed at some point. Whether a caller is using a commercial service like Vonage, AT&T CallVantage, a free service like Gizmo or Skype, or placing a long-distance call, most people are using VoIP. This technology will be here in some form for a long time, but only time will tell in what capacity. ■

Bibliography

AT&T CallVantage, *http://www.usa.att.com/callvantage,* Retrieved July 11, 2006.

H.323 vs. SIP: A Comparison, *http://www.packetizer.com/voip/h323_vs_sip*, Retrieved July 10, 2006.

The History of AT&T, *http://www.att.com/history,* Retrieved July 11, 2006.

How VoIP Works, *http://electronics.howstuffworks.com/ip-telephony.htm,* Retrieved July 12, 2006.

Skype, *http://www.skype.com,* Retrieved July 12, 2006.

National Institute of Technology (NIST), *Special Publication (SP) 800-56, Security Considerations for Voice Over IP Systems*.

Federal Communication Commission, Voice over Internet Protocol, *http://www.fcc.gov/voip,* Retrieved July 12, 2006

Vonage, *http://www.vonage.com*, Retrieved 7/10/2006.

About the Author

Matthew Warnock | is an Information Assurance Specialist with the Information Assurance Technology Analysis Center (IATAC). He graduated from Pennsylvania State University with a BS in Electrical Engineering, holds an Information Security Management Certificate from the University of Virginia, and is currently enrolled in a program for the MS degree in Telecommunications at George Mason University. His background includes assignments with the Defense Logistics Agency (DLA) in firewall and border-protection support. He may be reached at iatac@dtic.mil.

Dr. Matt Bishop

by Ron Ritchey and Ted Winograd

This article continues our series profiling members of the Information Assurance Technology Analysis Center (IATAC) Subject Matter Expert (SME) program. The SME profiled in this article is Dr. Matt Bishop. Dr. Bishop has been a Professor at the Department of Computer Science at University of California, Davis, since 2004 and is a co-director of the UC Davis Computer Security Laboratory. His research focuses on computer security, which is reflected in his numerous journal articles, white papers, [1] and his 2002 textbook, *Computer Security: Art and Science.*

Dr. Bishop's conducts his primary research in analyzing computer systems' vulnerabilities, including modeling, detecting, and reacting to them. His research has included detecting and handling malicious logic, network security, Denial of Service (DoS) attacks and defenses, policy modeling, software assurance testing, and formal modeling of access control.

Since earning his PhD in 1984, Dr. Bishop has written numerous papers on security. Some of his early research has been adopted by the computer industry as a whole. Apple Computer's password-quality measuring tool is based on password research Dr. Bishop performed almost 20 years ago. In the 1990s, he researched static analysis methods; today, static analysis tools are widely available and have proven to be important tools in security analysis.

In addition to his research, Dr. Bishop is heavily involved in integrating security into computer-science curricula. In 2002, Addison-Wesley Professional published Dr. Bishop's textbook, *Computer Security: Art and Science*. The textbook was largely written to complement Dr. Bishop's course structure. It provides a strong background in the theory behind computer security while relating it to the practices of computer security.

At the UC Davis Computer Security Lab, Dr. Bishop's vulnerability analysis project focuses on how to analyze systems for vulnerabilities and how to define vulnerabilities. For example, depending on the conditions under which it exists, a buffer overflow may not be a vulnerability. As another example, an electronic voting system needs to be available on Election Day. If the system fails on Election Day, it has violated the security-policy requirement of being available to record a vote. At other times, however, the failure may be acceptable, because the system is not recording a vote. This is an example of how a policy interacts with a system to define vulnerabilities.

Another of Dr. Bishop's projects at the Computer Security Lab is the Secure Programming Clinic. [2] The clinic follows the model successfully used in English courses. Many universities provide external writing clinics to help students improve their basic writing skills—without adding content to existing curricula. Secure programming practices, like writing skills, can be taught in parallel with regular classes. When students write programs for Computer Science courses, they can go to the clinic, where their programs are analyzed for any obvious problems. When found, the problems are identified and explained to the students. Students seem to enjoy the programming clinic process, and experience indicates this enjoyment results in better programs. The next step is to develop metrics to measure how well the Secure Programming Clinic works in comparison with other teaching methods.

If you have a technical question for Dr. Bishop or another IATAC SME, please contact *iatac@dtic.mil*. The IATAC staff will assist you in reaching the SME best suited to helping you solve the challenge at hand. If you have any questions about the SME program or are interested in joining the SME database and providing technical support to others in your domains of expertise, please contact *iatac@dtic.mil*, and the URL for the SME application will be sent to you. ■

References

1. A selected list of Dr. Bishop's publications may be found at *http://nob.cs.ucdavis.edu/bishop/papers*
2. "A Clinic to Teach Good Programming Practices." M. Bishop, B. J. Orvis, 10th Colloquium for Information Systems Security Education.

Countering DDoS Attacks with Multi-Path Overlay Networks

by Angelos Stavrou and Angelos Keromytis

Distributed Denial of Service (DDoS) has emerged as a major threat to the operation of online network services [1, 2, 3]. Current forms of DDoS attacks implicate multiple groups of Internet machines that have been taken over and controlled by an attacker. These machines, called *bots*, are manipulated by the attacker to produce an excessive surge of traffic toward a target server, the victim. The target server is forced to processing and/or to link-capacity starvation, since malicious traffic is blended with normal traffic, making it difficult to weed out. Figure 1 depicts a DDoS attack and its impact on the target server.

Unfortunately, DDoS attacks can only become worse: Despite network and processing speeds that increase with every passing day, real-world botnet sizes and attack capabilities increase at the same rate. Furthermore, attackers devise sophisticated software to infect and subsequently control thousands of infected machines while remaining stealthy. [4]

Addressing the network (DDoS) problem is extremely hard, given the fundamentally open nature of the Internet and the apparent reluctance of router vendors and network operators to deploy and operate new, potentially complex mechanisms. [5] Overlay-based approaches such as Secure Overlay Services (SOS) [6], funded by the Defense Advanced Research Projects Agency (DARPA) and the National Science Foundation (NSF Grant ITR CNS-04-26623); I3 [7]; and MayDay (Distributed Filtering for Internet Services) [8] offer an attractive alternative, as they do not require changes to the existing routing infrastructure. Furthermore, such systems require minimal or no collaboration from Internet Service Providers (ISPs), making their deployment completely transparent and thus practical. Overlay-based protection systems use an Internet-wide network of nodes that act as first-level firewalls, discriminating between legitimate traffic and potentially malicious traffic, enforcing some form of user or end-host authentication. Their distributed nature requires an extremely well-provisioned adversary to suppress their functionality, because, to disrupt protected communications, attack traffic must be split among all nodes. But how do these systems operate in practice?

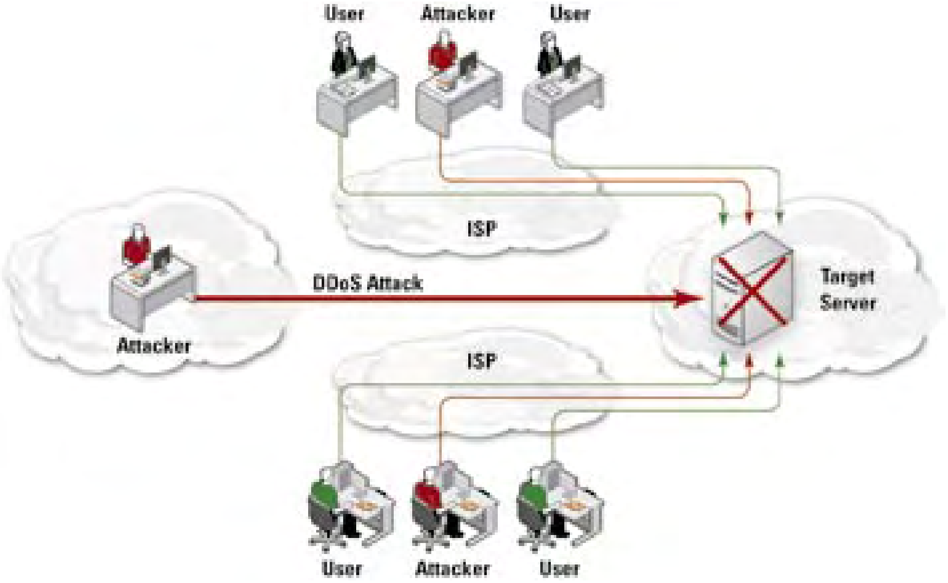


Figure 1 The target server is the victim of a DDoS attack. Legitimate users are denied access to the actual service since attackers generate overwhelming requests toward the target server's network.

Protection *via* Indirection Overlay Networks

In Figure 2, we present the main characteristics of the original SOS architecture, which is representative of indirection *via* overlay-based protection systems. We distinguish the three parts of the system:

the users, the overlay, and the protected server. Users want to establish a connection with the protected server but cannot do so directly—only a few select overlay nodes are allowed to communicate through the router-filtered area. These nodes can change over time in a random manner (but in coordination with the filtering routers). Although all overlay nodes are assumed to be publicly known, the precise identity of those nodes that can forward traffic through the filtered region at any given point in time is kept secret.

Users have to first authenticate themselves to the overlay network by connecting to a publicly advertised overlay node. This authentication can be either *via* cryptographic protocol and/or reverse Graphic Turing Tests (GTTs) [9] to determine valid users. (In some scenarios, this may simply mean "humans," while in other cases, some form of "proper" authentication may be required). Traffic from legitimate users is routed *via* the overlay and through the allowed overlay nodes to the protected service. However, malicious (or simply unknown) traffic is simply dropped by the overlay nodes, keeping the DDoS attack far from the protected service and potentially close to the attacker, using the overlay network as an indirection mechanism. One assumption made by systems such as SOS is that there is enough capacity leading to the filtering router to withstand a direct DDoS attack (*i.e.*, the unprotected links cannot be saturated). In most instances of DDoS attacks to date, the upstream ISP can handle the additional traffic; it is the target's uplink that is typically less well provisioned. By allowing only a few, select overlay nodes to forward traffic through this router, we avoid the need for new (potentially expensive, computationally or otherwise) router features.

Unfortunately, the original approaches of the Indirection-based Overlay Network (ION) depend on the inability of an adversary to discover connectivity information for a given client and the infrastructure (*e.g.*, which overlay node a client is using to route traffic). This makes them susceptible to a variety of easy-to-launch attacks that are not considered in the standard threat model of such systems. For example, adversaries may possess real-time knowledge of the specific overlay node(s) through which a client is routing traffic or may be attacking nodes using a time-based scheme that will try to maximize the impact of the attack on a client's connectivity. Such attacks can be network-oriented such as Transmission

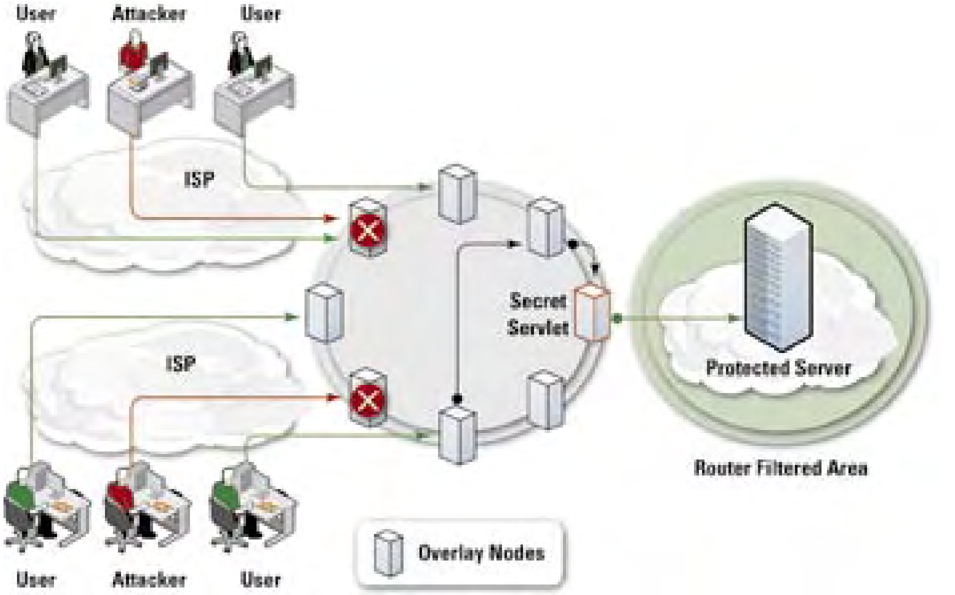


Figure 2 An overlay-based protection system. The users connect through the overlay nodes to the protected server. The overlay nodes act as distributed filters deep inside the network, mitigating the effects of a DDoS attack by dropping all unauthorized and unknown requests.

Control Protocol Synchronize (TCP SYN) attacks, application-related “sweeping” attacks, or “targeted” attacks.

In targeted attacks, an attacker who has knowledge of a client’s communication parameters can “follow” the client’s connections and bring down the nodes that he tries to connect to. As soon as the client realizes (typically, after some time-out period) that the overlay node is unresponsive and switches to a new node, the attacker also switches the attack to this new node. Thus an attacker that can bring down a single node can succeed in a targeted DDoS attack for specific clients. Similar attacks, exploiting information that must only be available to trusted components of the system but which an attacker can feasibly gain access to, are possible against almost all previously proposed anti-DDoS mechanisms.

Furthermore, IO networks are susceptible to an even worse type of attack: the sweeping attack. For this, an attacker uses its power (which is insufficient to bring down an entire ION) to target a small percentage of the overlay nodes at a time. The weak point of the overlay network is the application-level state maintained by the overlay node that is responsible for a client. Destroying this state forces the client to re-establish both network and application-level connectivity, degrading the clients’ connection and leading to DDoS for time-critical or latency-dependent applications. Repeating this attack can force clients to re-establish their credentials multiple times within short periods of time, making IONs completely impractical. Thus, although IONs can counter blind DoS attacks, they remain vulnerable to a range of simple but debilitating attacks.

A Novel, Stateless Architecture

We believe that these inherent limitations of first-generation, overlay-based, traffic-redirection mechanisms can be addressed by adopting a spread-spectrum-like communication paradigm. Note that although we use the term “spread-spectrum” to describe our approach, our work is *not* geared toward wireless networks nor does it touch on physical-layer issues. Our approach, as shown in Figure 3, is straightforward: Spread the packets from the client across all overlay nodes in a random manner, storing no network- or application-level state in the overlay nodes. The path diversity naturally exhibited by a distributed overlay network serves as the “spectrum” over which communications are “spread.” In our system, a token issued by the overlay network to the client is used to verify the authenticity of each packet communicated by the client. The use of a token (akin to a Kerberos ticket) alleviates the necessity to maintain application- or network-level state at any overlay node (unlike previous IONs) at the expense of bandwidth (since the ticket must be included in every packet routed through the ION). In return, our system is impervious to attacks that use this state dependence to attack the overlay.

An attacker will not know which nodes to direct an attack to; randomly attacking a subset of them will only cause a fraction of the client’s traffic to be dropped. By using Forward Error Correction (FEC) or simply duplicating packets (*i.e.*, simultaneously sending the same packet through two or more different overlay nodes), we can guarantee packet delivery with high probability, if we place an upper bound on the number of nodes an attacker can simultaneously attack.

Attack Resilience and Performance

To evaluate our system, we used a testbed consisting of PlanetLab Consortium machines located at various sites in the continental US. These machines were running User Mode Linux (UML) on commodity *x*86 hardware (Intel and compatible computer processors) and were connected using the Abilene Network Internet-2 high-performance backbone. Using these fairly distributed machines, we constructed our overlay network of overlay nodes by running a small forwarding daemon on each of the participating machines. We also used two more machines, acting as client and server, respectively. In our experiments, we measured link characteristics such as end-to-end latency and throughput when we interposed the overlay network of overlay nodes between the client and the protected server. To measure throughput, we used a protected server that was located at Columbia University in the City of New York. For our latency measurements, we used *http://www.cnn.com* as the “target.” In both cases, the goal of the

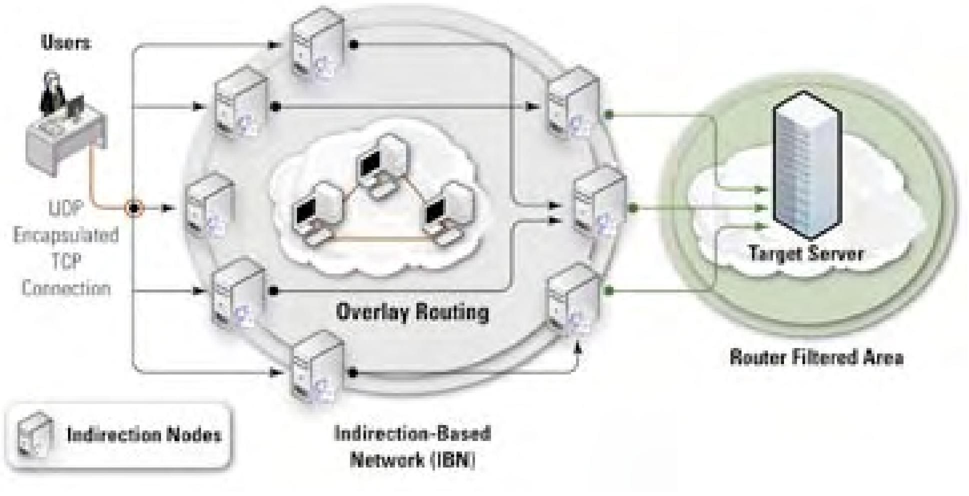


Figure 3 Users spread their packets to the network using a pseudo-random generator to avoid creating state to a single indirection node. An attacker cannot succeed by focusing his attack to some of the indirection nodes. Our system can sustain attacks that bring down up to 40% of indirection nodes, making it suitable for applications that require high levels of resiliency.

client was to establish communication with the protected server. To do so, the client used User Datagram Protocol (UDP) encapsulation on the TCP packets generated by a Secure Copy (SCP) session and then spread the UDP packets to the nodes participating on the overlay network. Those packets were in turn forwarded to a pre-specified overlay node that was permitted to connect to the protected server. Since our throughput connection measurements involve a client and a server that were co-located, we effectively measured the worst-case scenario (since our otherwise-local traffic had to take a tour of the Internet). A non-co-located server would result in a higher latency and lower throughput for a direct client-server connection, leading to comparatively better results when we use the overlay. Surprisingly, in some cases, we can achieve better latency using the overlay rather than by connecting directly to the server.

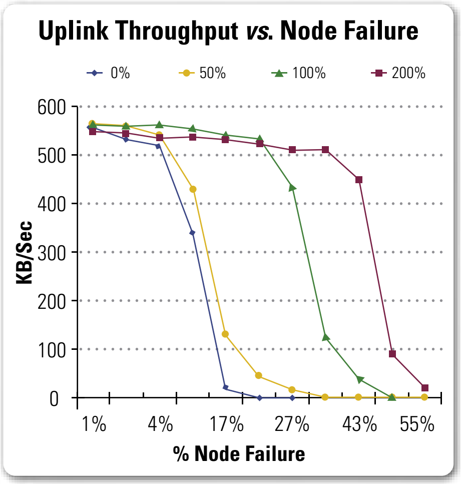


Figure 4 Throughput results in KB/s when we use the uplink of our client under attack. The attack happens on a random fraction of the overlay nodes. Each line represents different packet replication levels: For 100% packet replication, the client sends twice the amount of traffic by replicating each packet. Allowing packet replication helps us achieve higher network resilience.

Through our experiments and theoretical analysis, we show that, for an attacker to successfully attack our system, he will have to subvert or suppress more than 40% of the overlay nodes before the system becomes unusable for all users. Of course, our ability to thwart attacks depends on the packet replication (redundancy) we use. For example, a packet replication of 100% means that the client will replicate all packets once, effectively sending twice the amount of traffic. Figure 4 presents the system uplink performance when we vary both the number of overlay nodes that are under attack and the packet replication factor. For 200% packet replication we can sustain attacks up to 40% of the overlay nodes. Thus, our system has an operational threshold on the order of 40% of the nodes being subverted. Before this 40% threshold is reached, the users will not notice a significant impact to their connectivity. As a comparison, in the original SOS architecture, the user had to find an overlay node that was not under attack, which becomes increasingly difficult as we increase the portion of nodes under attack. We quantify the increase in the system's resistance to attacks using a simple analytical model and provide experimental validation by deploying a prototype over PlanetLab, a wide-area overlay network testbed. PlanetLab nodes are distributed across the Internet, serving as an ideal platform for experimentation.

Our analysis shows that an Akamai-sized ION with 2,500 nodes can withstand attacks that bring down up to 40% of the overlay. This corresponds to attacks that involve several million bots (attacking hosts), which is an order of magnitude larger than the biggest bot network seen to date. One expects that using an ION will impose a performance penalty. In our case, end-to-end latency increases by a factor of 2.0 in the worst case, but, by using packet replication, we maintain latency at the same level as that of the direct-connection case. These results confirm the findings from other research on multi-path routing.

Finally, we evaluated the overhead of our system to the end-to-end latency experienced by the clients. Although latency increase is a big concern whenever we add a network indirection system, our experiments show that, in the worst-case scenario, we have a 2.5 times increase in latency when compared to the direct connection to the protected server. However, this increase drops to just 1.5 times when we introduce a small packet replication of 50%. (For each two packets, we transmit another one.) In Figure 5, we present our latency results: As we increase the replication factor and for larger networks, we get better average latency results. In some cases, the latency observed when the client connects directly to the server can be higher than the one measured through the overlay. [The To (Overlay)/Td (Direct Connection) ratio in Figure 5 is below 1.0] This is true when some overlay nodes happen to have a lower latency route to the protected server when compared to the direct client-to-server route.

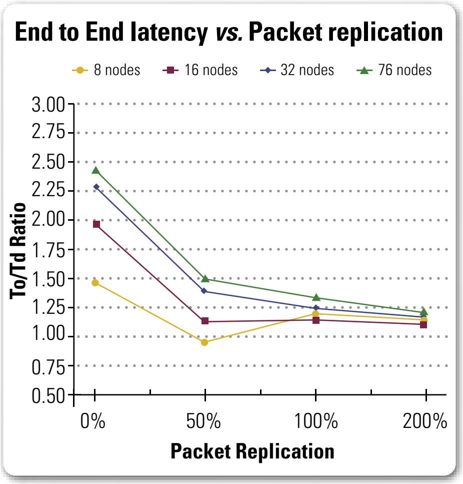


Figure 5 End-to-end average latency results for the index page and a collection of pages for *http://www.cnn.com*. The different points denote the change in the end-to-end latency through the Overlay, To, when compared to the Direct Connection, Td. Different lines represent different-sized overlays. Increasing the replication factor and for larger networks, we get lower average latency results because of the multi-path effect on the transmitted packets.

Conclusion

Our approach offers an attractive solution against congestion-based DDoS attacks in most environments, as it does not require modifications to clients, servers, protocols, or routers, both in terms of hardware and in existing software. Our plans for future work include developing a better characterization of the trade-offs that we have explored so far by introducing a coding scheme for the data transmission that will adapt to the network characteristics of each path used. Furthermore, we are looking into mechanisms to protect our system against attackers that can take over overlay nodes, thereby subverting part of the infrastructure. Finally, we are interested in deploying and using such a protection system on a larger scale than our experimental testbed to acquire operational experience in a real environment. Our article, *Countering DoS Attacks With Stateless Multipath Overlays,* [10] contains additional details about our system and the analysis and experimental evaluation.■

References

1. Hulme, G. (2004, September 13). Extortion online. *Information Week.*
2. Worldwide ISP Security Report. (2005 September). Retrieved from *http://www.arbor.net/downloads/Arbor_Worldwide_ISP_Security_Report.pdf*
3. Lipson, Howard F. (2002 November) Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues. Retrieved from *http://www.cert.org/archive/pdf/02sr009.pdf*
4. Ianelli, Nicholas. & Hackworth, Aaron. (2005 December). Botnets as a Vehicle for Online Crime. Retrieved from *http://www.cert.org/archive/pdf/Botnets.pdf*
5. The Cambridge-MIT Institute. (2005 January) DoS-Resistant Internet Working Group Meetings. Retrieved from *http://www.communicationsresearch.net/dos-resistant*
6. A. D. Keromytis, A. D. Misra, & Rubenstein, D. (2004 January). SOS: An Architecture For Mitigating DDoS Attacks. *IEEE Journal on Selected Areas of Communications (JSAC).*
7. I. Stoica, I., Adkins, D., Zhuang, S. & Surana, S. (2002 August) Internet Indirection Infrastructure. *Proceedings of ACM SIGCOMM.*
8. Anderson, David G. (2003 March). Mayday: Distributed Filtering for Internet Services. *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS).*
9. Luis von Ahn, Luis, Blum Manuel, Hopper, Nicholas J. & Langford, John. (2003 May) CAPTCHA: Using Hard AI Problems For Security. *Proceedings of EUROCRYPT.*
10. Stavrou, A. & Keromytis, A. D. (2005). Countering DoS Attacks With Stateless Multipath Overlays. *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pp. 249–259, Alexandria, VA.

About the Authors

Angelos Stavrou | is currently a Research Assistant at the Network Security Laboratory at Columbia University. His research interests are security using Peer-to-Peer (P2P) and overlay networks. He received a BS in Physics with honors from the University of Patras, Greece, and an MSc in the theory of Algorithms, Logic, and Computation from the University of Athens, Greece. He also holds an MSc in Electrical Engineering from Columbia University and is currently a PhD candidate at Columbia.

Angelos Keromytis | is an Associate Professor of Computer Science at Columbia University. He received Masters and PhD degrees from the University of Pennsylvania and a Bachelors degree (all in Computer Science) from the University of Crete, Greece. His research interests include network and system survivability, authorization and access control, and large-scale systems security. His complete *curriculum vitae* may be found at *http://www.cs.columbia.edu/angelos/cv.html*

Letter to the Editor

Q ***I recently attended the Information Assurance Technical Framework Forum (IATFF) at Johns Hopkins Applied Physics Laboratory in Laurel, MD. While there, I heard a briefing on the protection of data at rest and noted something: the Secure Mobile Environment-Portable Electronic Device. This is the first I've heard of this device. Might you know something more about it?***

A The Secure Mobile Environment-Portable Electronic Device (SME-PED) is the National Security Agency's (NSA) concept for a secure, wireless, handheld product. Currently in development, the SME-PED will be a secure Personal Digital Assistant (PDA) and wireless phone. It will provide users with protected voice and data communications and support security levels up to the Top Secret level and email exchanges up to the Secret level.

The SME-PED will not only permit secure phone usage but will also be the first product to provide remote, wireless access to the Secret IP Router Network (SIPRNet). With NSA's Type 1 and Non-Type 1 encryption implemented, individuals will be able to access the Internet, NIPRNet, and SIPRNet *via* the SME-PED.

Only two companies were awarded the $36M contract to develop this product, with a scheduled delivery date of 2Q 2007. Although the SME-PED's release is scheduled almost a year from now, several government organizations have seen the value of this product and are already integrating the SME-PED in future plans and programs. For more information, please do not hesitate to contact us at iatac@dtic.mil. ■

FREE Products

Order Form

Instructions: All IATAC LIMITED DISTRIBUTION reports are distributed through DTIC. If you are not registered DTIC user, you must do so prior to ordering any IATAC products (unless you are DoD or Government personnel). To register On-line: *http://www.dtic.mil/dtic/registration*. The *IAnewsletter* is UNLIMITED DISTRIBUTION and may be requested directly from IATAC.

Name ____________________

Organization ____________________

Address ____________________

____

____

DTIC User Code ____________________

Ofc. Symbol ____________________

Phone ____________________

E-mail ____________________

Fax ____________________

Please check one:	□ USA	□ USMC	□ USN	□ USAF	□ DoD
	□ Industry	□ Academia	□ Government	□ Other	

Please list the Government Program(s)/Project(s) that the product(s) will be used to support: ____________________

____

LIMITED DISTRIBUTION

IA Tools Reports (softcopy only)

□ Firewalls
□ Intrusion Detection
□ Vulnerability Analysis

Critical Review and Technology Assessment (CR/TA) Reports

□ Biometrics (soft copy only)
□ Configuration Management
□ Defense in Depth (soft copy only)
□ Data Mining (soft copy only)
□ IA Metrics (soft copy only)
□ Network Centric Warfare (soft copy only)
□ Wireless Wide Area Network (WWAN) Security
□ Exploring Biotechnology (soft copy only)
□ Computer Forensics* (soft copy only. DTIC user code MUST be supplied before these reports will be shipped)

State-of-the-Art Reports (SOARs)

□ Data Embedding for IA (soft copy only)
□ IO/IA Visualization Technologies (soft copy only)
□ Modeling & Simulation for IA (soft copy only)
□ Malicious Code (soft copy only)
□ A Comprehensive Review of Common Needs and Capability Gaps

UNLIMITED DISTRIBUTION

IAnewsletters Hardcopies are available to order. The list below represents current stock.
Softcopy back issues are available for download at *http://iac.dtic.mil/iatac/IA_newsletter.html*

Volumes 4		□ No. 2	□ No. 3	□ No. 4
Volumes 5	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 6	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 7	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 8	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 9	□ No. 1	□ No. 2	□ No. 3	□ No. 4

Fax completed form to IATAC at 703/984-0773

Calendar

January

Defending America/SpaceComm 2007
23–25 January 2007
Broadmoor
Colorado Springs, CO
http://www.afceaspacecomm.com/index.php

Total Asset Visibility For Defense
30–31 January 2007
Hilton Arlington
Arlington, VA
http://www.idga.org/cgi-bin/templates/singlecell.html?topic=495&event=11762

WEST 2007
31 January–2 February 2007
San Diego Convention Center
San Diego, CA
http://www.afcea.org/events/West/2007/introduction.asp

February

Camp HM Smith–HQ US Pacific Command
5 February 2007
Pollock Theatre–Building 4 at Camp Smith
Honolulu, HI
http://www.fbcinc.com/event.asp?eventid=Q6UJ9A00C1TM

RSA Conference
5–9 February 2007
Mascone Center
San Francisco, CA
http://www.rsaconference.com/2007/US

Phoenix Challenge 2007 Conference
27 February–1 March 2007
Sandia National Laboratories, Kirtland AFB
Albuquerque, NM
http://www.aia.af.mil/units/afioc/phoenixchallenge.asp

Homeland Security Conference 2007
28 February–1 March 2007
Ronald Reagan International Trade Center
Washington, DC
http://www.afcea.org/events/homeland/landing.asp

March

20th Annual FISSEA Conference
12 March 2007
Bethesda North Marriott
Rockville, MD
http://www.fbcinc.com/event.asp?eventid=Q6UJ9A00CEND

InfoSec World Conference & Expo 2007
19–21 March 2007
Rosen Shingle Creek Resort
Orlando, FL
http://www.misti.com/default.asp?Page=65&Return=70&ProductID=5539&LS=infosecworld2007

Maritime Homeland Security
19–21 March 2007
Sheraton Baltimore North
Baltimore, MD
http://www.iqpc.com/cgi-bin/templates/singlecell.html?topic=221&event=11840

DTIC 2007 Annual Users Meeting & Training Conference
26–28 March 2007
Hilton Alexandria Old Town
Alexandria, VA
http://www.dtic.mil/dtic/annualconf

IATAC

Information Assurance Technology Analysis Center
13200 Woodland Park Road, Suite 6031
Herndon, VA 20171

To change, add, or delete your mailing or e-mail address (soft-copy receipt), please contact us at the address above or call us at: 703/984-0775, fax us at: 703/984-0773 , or send us a message at: iatac@dtic.mil